Logic and Logical Philosophy
Volume 8 (2000), 115–152

Walter A. Carnielli[*]
João Marcos[†]
Sandra de Amo[‡]

FORMAL INCONSISTENCY AND EVOLUTIONARY DATABASES

Abstract. This paper introduces new logical systems which axiomatize a formal representation of inconsistency (here taken to be equivalent to contradictoriness) in classical logic. We start from an intuitive semantical account of inconsistent data, fixing some basic requirements, and provide two distinct sound and complete axiomatics for such semantics, **LFI1** and **LFI2**, as well as their first-order extensions, **LFI1*** and **LFI2***, depending on which additional requirements are considered. These formal systems are examples of what we dub Logics of Formal Inconsistency (**LFI**) and form part of a much larger family of similar logics. We also show that there are translations from classical and paraconsistent first-order logics into **LFI1*** and **LFI2***, and back. Hence, despite their status as subsystems of classical logic, **LFI1*** and **LFI2*** can codify any classical or paraconsistent reasoning.

1. Introduction and motivation

Investigations of the formal possibilities of handling contradictions and inconsistencies are usually carried out under the common label of *paraconsistent*

[‡] This research was initiated while the first author visited the Seminar für Logik und Grundlagenforschung, Universität Bonn, granted by the Alexander von Humboldt Foundation, Germany, and by CAPES, Brazil. This work is also supported by an individual research grant from CNPq, Brazil.

[‡] Supported by a CAPES graduate fellowship.

[‡] Supported by an individual research grant from CNPq.

logics. Although in principle it seems clear that logical mechanisms for handling contradictions and inconsistencies can be applied to a wide variety of problems, it is usually difficult to point out real applications, because the extant syntax of the better developed paraconsistent systems is quite often simply inappropriate to deal with them. The reader should be aware that the concepts of contradiction and inconsistency are not necessarily to be identified, as argued in [CM01], and that distinct philosophical positions can be taken according to whether or not one adopts such an identification. We will here, however, identify inconsistency with contradiction throughout this study, as it will be shown that inconsistent information (formalized as $\bullet A$) and contradictory information (in the form $A \wedge \neg A$) are identified in our systems through the equivalence $\bullet A \leftrightarrow (A \wedge \neg A)$.

A general formal system apt to handle contradictions in informatics is clearly of great interest, in particular for the management of databases. A relational database (cf. [Cod70]) is a finite collection of finite relations in which information is stored. In general, this information must verify some conditions, the so-called *integrity constraints*, in order for it to be safely introduced in the database. Integrity constraints are expressed by (fixed) first-order sentences; so, for example, a database storing information about flights may contain the requirement that no flight can have more than one duration, a condition which could be expressed by the following first-order formula:

$$\forall x \forall y \forall z(\mathit{Flight}(x, y) \wedge \mathit{Flight}(x, z) \to y = z,$$

where $\mathit{Flight}(x, y)$ means that flight number x has a duration of y hours. Traditionally, when a database is updated — that is, when some information is added, modified or removed from its relations — the management system verifies if the new upcoming database state satisfies the integrity constraints. The update is only performed in the affirmative case, and, when refused, the database maintains its previous state. So, in a traditional database system, contradictory information is never really allowed into the database, due to this kind of preventive control.

However, with the development of new network technology, information was allowed to be accessed and updated from several sources, making the updating process harder and harder to carry out. Local databases have their inner integrity constraints, and are thus free of contradictions. Nevertheless, two local databases can quite naturally be mutually contradictory, requiring complex and costly procedures to restore or maintain consistency at a global level.

Besides such inherent problems of databases, another very desirable feature of database systems which could produce an inconsistent situation is to let integrity constraints themselves change in time, instead of remaining fixed forever. We will call such evolving databases *evolutionary databases*. Traditionally, the integrity constraints are defined by the database designer and cannot be modified by the users during the lifetime of the database. However, this is a rather restrictive requirement. Now, if one allows databases to be evolutionary, then one is unable to guarantee that new constraints would not be contradictory with previous ones. Thus, we are faced once more with the possibility of inconsistency arising inside our informational environment.

Our logical foundations for evolutionary databases allow us to implement several policy managements concerning storage and representation of information. In case a piece of information "*Inf*" is proposed by a source, it enters the database either with the token ✓ or the token ✗ appended to it. In case "not-*Inf*" is proposed, it enters with the token ✗ or it does not enter at all. In case there is no information about a subject "*Inf*", nothing in this respect is added to the database. As a consequence, in case "*Inf*" and "not-*Inf*" are simultaneously proposed (for instance, by different sources), then "*Inf*" enters the database with the token ✗.

Suppose, as an example, we have a database schema ***DS*** containing three relations:

– $author(Name, Country)$,

– $wrote(Name, Book)$, and

– $has\_been\_translated(Book, Language)$.

Suppose also that two different sources, ***A*** and ***B***, provide information to our database, telling us that:

– Source ***A***:

(1) Joaquim Maria Machado de Assis was born in Brazil.

(2) Gabriel García Marquez was born in Colombia.

(3) Machado de Assis wrote "Dom Casmurro".

(4) "Dom Casmurro" has not been translated into Polish.

– Source ***B***:

(5) García Marquez was not born in South America.

(6) García Marquez wrote "One Hundred Years of Solitude".

(7) "One Hundred Years of Solitude" has been translated into Polish.

Information (4) is negative, hence it is either stored in $\boldsymbol{DS}$ with a ✗ token or not stored at all. On the other hand, the information represented in (2) and (5) is inconsistent, and in this case that fact is added with the token ✗ appended to it. The remaining positive information may be added having either ✓ or ✗ as suffix. What may result is the following:

author		
Name	*Country*	*status*
JMMdA	Brazil	✓
GGM	Colombia	✗

wrote		
Name	*Book*	*status*
JMMdA	DC	✓
GGM	OHYoS	✓

has_been_translated		
Book	*Language*	*status*
OHYoS	Polish	✓

A query language can be used to infer new information from $\boldsymbol{DS}$. By way of an example, one can infer that *has_been_translated*(DC, Hungarian) is *not* the case (expressing the fact that Dom Casmurro has not been translated into Hungarian), since the sources have nothing to say about that. This kind of negated assertion is known as *default negation*, and is connected to the so-called *closed world assumption*, which presupposes complete knowledge about a given subject.

Besides facts, sources can also add sentences, i.e. integrity constraints, to our database. In traditional databases, constraints are fixed, but our model allows constraints to be either added or deleted. For example, suppose that source $\boldsymbol{A}$ adds the new constraint: 'No South American author has ever been translated into Polish'. After $\boldsymbol{DS}$ has been updated taking this new constraint into consideration, the relation *has_been_translated* will also contain inconsistent information. While traditional databases cannot support this situation, our model permits to reason with this other type of inconsistency, even taking advantage of this controversy to get better knowledge about the sources.

The predicament of inconsistency has been already tackled in the database field by several articles, where a series of logical frameworks have been proposed in order to reason in the presence of inconsistent data: many-valued logics in [Bel77], annotated logics in [KL92] and [Fit91], traditional paraconsistent logics in [CLM92], and also semi-formal treatments like the one in [RM70]. In this paper we intend to discuss some of the underlying questions of paraconsistency by means of proposing a database model based on an axiomatic treatment of the basic properties of inconsistent information. We argue that, in this way, it is possible to offer an alternative view of the question of consistency of the updating process, which at the same time permits one to consider also evolutionary databases. Our approach allows not only inconsistent facts to be represented, but also new constraints to be added which could change the state of already stored facts.

A logical system, consisting of a pair $(\boldsymbol{S}, \Vdash_{\boldsymbol{S}})$ formed by a set $\boldsymbol{S}$ of formulas endowed with a consequence relation $\Vdash_{\boldsymbol{S}}$, is called *paraconsistent* when it permits us to distinguish, on the one hand, between *contradictory* theories Γ (in the sense that $\Gamma \Vdash_{\boldsymbol{S}} A$ and $\Gamma \Vdash_{\boldsymbol{S}} \neg A$, for some formula A, where $\neg A$ denotes the negation of A), theories which are at most inconvenient, and, on the other hand, *trivial* theories Δ (in the sense that every $\Delta \Vdash_{\boldsymbol{S}} B$, for any formula B), theories which are evidently uninteresting. Equivalently, if we call *explosive* the logics in which the *Principle of Explosion*: $A, \neg A \Vdash_{\boldsymbol{S}} B$ holds, for any A and B, we can assert then that a logical system is paraconsistent if, and only if, it is non-explosive (the equivalence of the two formulations above holds at least for logics having a reflexive and transitive consequence relation — see [CM01] — , like the ones we study here).

There exist many different systems of paraconsistent logic. The first to be introduced (and perhaps the most influential) were the systems in the hierarchy C_n, $0 < n < \omega$ (see [dC74]). In the C_n systems, trivialization is separated from contradictoriness (and yet contradiction and inconsistency coincide) by means of a finer control of the inferential mechanisms of logic, by assuming that:

(**C1**) not all formulas are necessarily subject to the Principle of Explosion, but only some special formulas (in the case of C_1, for instance, these would be the formulas A which were such that $\neg(A \wedge \neg A)$ was provable);

(**C2**) if both A and B are subject to the Principle of Explosion, so are $A \wedge B$, $A \vee B$, and $A \to B$.

Substituting only (**C2**) for:

(**C3**) if either A or B are subject to the Principle of Explosion, so are $A \wedge B$, $A \vee B$, and $A \to B$,

one can easily formulate stronger versions of these paraconsistent logics, namely those called C_n^+, for $0 < n < \omega$ (cf. [dCBB95]).

Although the logics above are not many-valued, there may be found in the literature several many-valued paraconsistent systems, known for a long time, such as the three-valued paraconsistent systems $\mathbf{P}^1$ (cf. [Set73]) and $\mathbf{J}_3$ (cf. [DOdC70]), both of them constituting but special cases of the 8,192 three-valued paraconsistent logics studied in [Mar??].

Systems like $\mathbf{P}^1$ and $\mathbf{J}_3$ are not appropriate for dealing with information, as they have been designed with different motivations. The logics C_n and C_n^+, on the other hand, are not fully appropriate for handling information, as they are in some sense too weak — they are not maximal (in the sense that they can be properly extended in arbitrarily many ways, yet remaining both paraconsistent and fragments of classical logic). Some further semantical considerations concerning these last systems were raised in [Mar99] (and in [Car00]), where possible-translations semantics for (slightly stronger versions of) them are proposed, in terms of combinations of some appropriate three-valued logics, aiming to provide them with better interpretations.

We introduce here new logical systems to handle inconsistent data, ones which are naturally motivated by our evolutionary database, and are at the same time close enough to usual first-order logic so as to allow good control of inconsistent data without renouncing the existent theory of traditional databases. Moreover, we start from the semantical properties which would seem desirable for a well-founded treatment of inconsistent data, and derive the logics **LFI1** and **LFI2** from such desiderata. We then show that **LFI1** and $\mathbf{J}_3$ are interdefinable, but **LFI2** is independent of any other known three-valued logic, and moreover both **LFI1** and **LFI2** are maximal systems. The approaches towards **LFI1** and $\mathbf{J}_3$, however, are totally distinct, as **LFI1** axiomatizes a form of inconsistency by means of the connective $\bullet$, while $\mathbf{J}_3$ axiomatizes a form of possibility by means of the connective ∇.

Differently from $\mathbf{J}_3$, which is intended to be related to the proposal of Jaśkowski in [Jas68], and also differently from the proposal of da Costa in [dC74], which identified consistency with the formula $\neg(A \wedge \neg A)$ (though, strangely enough, *not* with the similar formula $\neg(\neg A \wedge A)$ — cf. [Mar99]), and from any other proposal currently known to the authors (cf. [PRN89] for discussions about the nature and reach of several approaches to paraconsistency), our approach regards contradictions and inconsistencies as phenomena waiting to be formally treated. Indeed, the deductive proof system introduced in this paper tries to formalize what an inconsistency *should be* from the point of view of classical logic. That is the reason we call the systems described here *Logics of Formal Inconsistency* — **LFI** (which constitute a very large and interesting family of paraconsistent logics — cf. [CM01] for a thorough discussion about such logics). These formal systems are shown to be sound and complete with respect to some very natural semantics, appropriate for monitoring linguistic inconsistencies (or contradictions), particularly within those found in evolutionary databases. So, instead of refusing new contradictory data arising from updates, each of these proof systems will act as a kind of sensor by tracking down inconsistency and pointing out what part of the data is likely to contain inconsistencies.

In other papers we concentrate on further technical details of showing how logics of this kind can be used to handle evolutionary databases and to update procedures. In [CdA99] we raised the main problems of evolutionary databases and outlined the desirable features of an underlying logic suitable for the management of evolutionary databases. Indeed, as we argue in the present paper, several logical systems can be used to such an effect, and in [dACM??] we show how the particular system **LFI1** offers some good perspectives on this question.

This paper is organized as follows: In Section 2, we introduce some basic concepts and notations concerning our database model interpretation based on **LFI1*** and discuss how its semantics is motivated and defined. In Section 3 we show that the semantics we have introduced for **LFI1**, the propositional fragment of **LFI1***, is equivalent to a three-valued semantics, and present as well some results concerning the expressive capacity of this logic. Section 4 presents an axiomatic system for **LFI1** which formalizes its semantics in precise terms, as shown by the completeness theorem proved in details (via a constructive Kalmár-like procedure). In Section 5 we study **LFI1*** and outline its completeness proof. Section 6 presents some conservative translations from classical and paraconsistent logics into **LFI1***, and comment on translations in the opposite direction. Finally, in Section 7 we present **LFI2***, a competitor of **LFI1***, and exhibit its main properties. Although a large class of logics of formal inconsistency can be defined following our approach, one should observe that **LFI1** and **LFI2**, as well as their first-order extensions, occupy, in a certain sense, diametrically opposite positions in the spectrum of logical systems which formalize inconsistency: Indeed, we shall argue that **LFI1** can be seen as regarding inconsistency as an *external* phenomenon, while for **LFI2** inconsistency is an *internal* phenomenon. We conclude in Section 8 by discussing some further perspectives related to our present approach.

2. A formal semantics for a theory which supports contradictions

We assume the language $\mathbf{L}^{+}$ of our logic to be defined in the usual way, as in the classical first-order logic, with the addition of a new symbol $\bullet$ (read as 'it is inconsistent'). So, all syntactical notions are the familiar ones, with the obvious modifications. A formula B is called *inconsistent* in case it assumes the form $\bullet A$, for some formula A. A formula is called an *extended literal* if it is either an atomic formula, or the negation of an atomic formula, or an inconsistent atomic formula.

As to the semantical ground on which we base our study, we assume that our structures (or models) are the same as in the classical case, with the only difference that we reserve two distinguished *non-standard constants*, ✗ and ✓, present in the universe of every structure. The interpretation of terms and predicates is defined in the expected way, straightforwardly adapted to include those new constants. So, given an n-ary predicate R in $\mathbf{L}^{+}$, the *standard interpretation* $R^{\boldsymbol{I}}$ of R in the structure $\boldsymbol{I}$ (with universe $|\boldsymbol{I}|$) is an n-ary relation $R^{\boldsymbol{I}} \subseteq |\boldsymbol{I}|^{n}$, and the *extended interpretation* of R is a new relation $R^{\boldsymbol{I}+} \subseteq R^{\boldsymbol{I}} \times \{✗, ✓\}$, where $(r, ✗)$ and $(r, ✓)$ do not occur simultaneously, for $r \in R^{\boldsymbol{I}}$. Now, for **LFI1**:

DEFINITION 2.1. The *interpretation* $\vDash$ for the sentences in **LFI1**, written in the propositional fragment of $\mathbf{L}^{+}$, in a given structure $\boldsymbol{I}$, is inductively defined as follows (we use $\nvDash$ to denote the failure of a clause):

(1.1) $\boldsymbol{I} \vDash A \wedge B \iff \boldsymbol{I} \vDash A$ and $\boldsymbol{I} \vDash B$

(1.2) $\boldsymbol{I} \vDash A \vee B \iff \boldsymbol{I} \vDash A$ or $\boldsymbol{I} \vDash B$

(1.3) $\boldsymbol{I} \vDash A \rightarrow B \iff \boldsymbol{I} \nvDash A$ or $\boldsymbol{I} \vDash B$

(2.0) $\boldsymbol{I} \vDash \neg\neg A \iff \boldsymbol{I} \vDash A$

(2.1) $\boldsymbol{I} \nvDash \bullet\bullet A$

(2.2) $\boldsymbol{I} \vDash \bullet A \Longrightarrow \boldsymbol{I} \vDash A$

(2.3) $\boldsymbol{I} \vDash \neg A \iff \boldsymbol{I} \nvDash A$ or $\boldsymbol{I} \vDash \bullet A$

(3.1) $\boldsymbol{I} \vDash \bullet(A \wedge B) \iff \boldsymbol{I} \vDash \bullet A \wedge B$ or $\boldsymbol{I} \vDash \bullet B \wedge A$

(3.2) $\boldsymbol{I} \vDash \bullet(A \vee B) \iff \boldsymbol{I} \vDash \bullet A \wedge \neg B$ or $\boldsymbol{I} \vDash \bullet B \wedge \neg A$

(3.3) $\boldsymbol{I} \vDash \bullet(A \rightarrow B) \iff \boldsymbol{I} \vDash A \wedge \bullet B$

And to interpret the sentences in **LFI1***, i.e. the first-order sentences in $\mathbf{L}^{+}$, one just has to add:
– For extended literals, given c_1, c_2, …, c_n closed terms:

(0.1) $\boldsymbol{I} \vDash R(c_1, c_2, \ldots, c_n) \iff (c_1^{\boldsymbol{I}}, c_2^{\boldsymbol{I}}, \ldots, c_n^{\boldsymbol{I}}, \text{✗}) \in R^{\boldsymbol{I}+}$ or $(c_1^{\boldsymbol{I}}, c_2^{\boldsymbol{I}}, \ldots, c_n^{\boldsymbol{I}}, \checkmark) \in R^{\boldsymbol{I}+}$

(0.2) $\boldsymbol{I} \vDash \neg R(c_1, c_2, \ldots) \iff (c_1^{\boldsymbol{I}}, c_2^{\boldsymbol{I}}, \ldots, c_n^{\boldsymbol{I}}, \text{✗}) \in R^{\boldsymbol{I}+}$, or both $(c_1^{\boldsymbol{I}}, c_2^{\boldsymbol{I}}, \ldots, c_n^{\boldsymbol{I}}, \text{✗}) \notin R^{\boldsymbol{I}+}$ and $(c_1^{\boldsymbol{I}}, c_2^{\boldsymbol{I}}, \ldots, c_n^{\boldsymbol{I}}, \checkmark) \notin R^{\boldsymbol{I}+}$

(0.3) $\boldsymbol{I} \vDash \bullet R(c_1, c_2, \ldots, c_n) \iff (c_1^{\boldsymbol{I}}, c_2^{\boldsymbol{I}}, \ldots, c_n^{\boldsymbol{I}}, \text{✗}) \in R^{\boldsymbol{I}+}$

– Remaining clauses, for quantification:

(FO.1) $\boldsymbol{I} \vDash \forall x A(x) \iff \boldsymbol{I} \vDash A(t)$ for all t

(FO.2) $\boldsymbol{I} \vDash \exists x A(x) \iff \boldsymbol{I} \vDash A(t)$ for some t

(FO.3) $\boldsymbol{I} \vDash \neg(\forall x A(x)) \iff \boldsymbol{I} \vDash \exists x \neg A(x)$

(FO.4) $\boldsymbol{I} \vDash \neg(\exists x A(x)) \iff \boldsymbol{I} \vDash \forall x \neg A(x)$

(FO.5) $\boldsymbol{I} \vDash \bullet(\forall x A(x)) \iff \boldsymbol{I} \vDash \forall x A(x)$ and $\boldsymbol{I} \vDash \exists x \bullet A(x)$

(FO.6) $\boldsymbol{I} \vDash \bullet(\exists x A(x)) \iff \boldsymbol{I} \vDash \forall x \neg A(x)$ and $\boldsymbol{I} \vDash \exists x \bullet A(x)$

The basic idea behind conditions (2.1)–(2.3) is that having an inconsistent formula A is equivalent to having both A and $\neg A$, and an inconsistent information cannot be itself inconsistent, by way of stipulation. As to conditions (3.1)–(3.3), one may regard **LFI1** as modelling inconsistencies as external phenomena: Thus, having an inconsistency in a complex formula $(A \# B)$, where $\#$ is a binary connective, would intuitively be the same as having $(A \# B)$ and $\neg(A \# B)$, assuming furtherly that formulas can be represented in a sort of disjunctive normal form. As an example, in (3.1), the conjunction of $(A \wedge B)$ and $\neg(A \wedge B) = (\neg A \vee \neg B)$, produces $(A \wedge B \wedge \neg A)$ or $(A \wedge B \wedge \neg B)$, what would result, following the intended intuition, in $\bullet A \wedge B$ or $\bullet B \wedge A$. As to conditions (FO.5) and (FO.6), the idea is to regard $\forall$ and $\exists$, respectively, as an unbounded conjunction and an unbounded disjunction.

The interpretation above defines a consequence relation for the logic **LFI1***. All other usual semantical notions, such as validity, tautology, etc. are similar to the classical notions. It is important to insist that our semantics is in some sense an extension of the usual semantical notion, basically differing from the classical one by the treatment given to negation.

The following properties of an **LFI1**-interpretation are then easily verified:

Fact 2.2. *Under the interpretation above, the following property obtains:*

(3.0) $\boldsymbol{I} \vDash \bullet(\neg A) \iff \boldsymbol{I} \vDash \bullet A$ □

Proposition 2.3. (i) *Clauses* (2.1)–(2.3) *may be substituted by:*

(2.4) $\boldsymbol{I} \vDash \bullet A \iff \boldsymbol{I} \vDash A \wedge \neg A$

(2.5) $\boldsymbol{I} \vDash \neg(\bullet A) \iff \boldsymbol{I} \nvDash \bullet A$

(2.6) $\boldsymbol{I} \nvDash \neg A \Longrightarrow \boldsymbol{I} \vDash A$

(ii) *And clauses* (3.1)–(3.3) *may be substituted by:*

(1.4) $\boldsymbol{I} \vDash \neg(A \wedge B) \iff \boldsymbol{I} \vDash \neg A$ *or* $\boldsymbol{I} \vDash \neg B$

(1.5) $\boldsymbol{I} \vDash \neg(A \vee B) \iff \boldsymbol{I} \vDash \neg A$ *and* $\boldsymbol{I} \vDash \neg B$

(1.6) $\boldsymbol{I} \vDash \neg(A \to B) \iff \boldsymbol{I} \vDash A$ *and* $\boldsymbol{I} \vDash \neg B$

Remember from last section that a logic with a consequence relation $\Vdash$ is called paraconsistent if it is non-explosive, that is, if there are formulas A and B such that $A, \neg A \nVdash B$. Following [CM99], we say that a logic has a *strong negation* if it has an operator $\sim$ such that $A, \sim A \Vdash B$, for any formulas A and B. A paraconsistent logic in which all positive inferences hold and a strong negation is present is said to constitute a **C**-*system* (see [CM01] for sharper definitions of that). According to the requisite (**C1**) of da Costa, in the last section, formulas such as $\neg(A \wedge \neg A)$ should not be theorems of some sort of paraconsistent logics, for these logics would not only identify $\bullet A$ with $A \wedge \neg A$, but also their negations, through the equivalence $\neg \bullet A \leftrightarrow \neg(A \wedge \neg A)$. Paraconsistent logics with a strong negation which make this identification, or, in a wider sense, which identify both $\bullet A$ and $\neg \bullet A$ with some other formulas in which the operator $\bullet$ does not occur, are called **dC**-*systems*. Evidently:

Remark 2.4. **LFI1** is a **C**-system, but not a **dC**-system.

To see that **LFI1** is a **C**-system, just notice that, in general, $\boldsymbol{I} \vDash (A \wedge \neg A) \to B$ does *not* hold (though if A assumes the form $\bullet C$ or $\sim C$, for some C, it does), but that, on the other side, $\boldsymbol{I} \vDash (A \wedge \neg A \wedge \neg \bullet A) \to B$ *do* always hold, for every structure $\boldsymbol{I}$. To see that it is not a **dC**-system, in a strict sense, hare a look at the matrics to see that $\bullet$ is not definable fromany combination of the other connectives. Furthermore, notice that, even though $\boldsymbol{I} \vDash \neg \bullet A$ does *not* hold, in general, $\boldsymbol{I} \vDash \neg(A \wedge \neg A)$ *do* hold. □

As we will see in the next section, the operator $\sim$ defined as $\sim A \overset{\text{def}}{=} \neg A \wedge \neg \bullet A$, for every formula A, is *indeed* a strong negation — and, moreover, has all the properties of classical negation. We will also see there, in Fact 3.1, why $\neg \bullet A$ cannot be identified with any other formula in which $\bullet$ does not occur. In Section 5 we show that this strong negation precisely formalizes the notion of default negation, mentioned in the ***DS*** example of the introduction.

An underlying assumption of the interpretation above, reflected in (2.4), was that an inconsistent formula $\bullet A$ is true iff A and $\neg A$ are simultaneously true, or, in other words, iff A is contradictory. The inconsistency (or contradictory) operator $\bullet$ can be understood as a kind of linguistic modality, but not coincident with usual alethic or deontic modalities. In fact, as we will show in the next few sections, **LFI1** is a *three-valued logic*, and **LFI1*** its immediate first-order extension.

3. A three-valued logic in disguise

This section will show that the interpretation proposed in the last section defines a three-valued logic, in the canonical sense, and some further semantical properties of this logic will be exhibited. We assume the reader to be familiar with the usual notions of matrix semantics for many-valued logics, and recall only that a *many-valued valuation* is a homomorphism between sentences and truth-values induced by the matrices. In the case of **LFI1**, the truth-values are 1 and $1/2$ (for “true” and “partially true”) and 0 (for “false”) and the matrices are the following:

$\wedge$	1	$1/2$	0
1	1	$1/2$	0
$1/2$	$1/2$	$1/2$	0
0	0	0	0

$\vee$	1	$1/2$	0
1	1	1	1
$1/2$	1	$1/2$	$1/2$
0	1	$1/2$	0

$\to$	1	$1/2$	0
1	1	$1/2$	0
$1/2$	1	$1/2$	0
0	1	1	1

	$\neg$	$\bullet$
1	0	0
$1/2$	$1/2$	1
0	1	0

where 1 and $1/2$ are the designated values.

Fact 3.1. *The connectives* $\vee$, $\neg$ *and* $\bullet$ *may be taken as primitive in* **LFI1**. Just notice that $A \wedge B$ and $A \to B$ may be defined, respectively, as $\neg(\neg A \vee \neg B)$ and $B \vee \neg(A \vee \bullet A)$. The reader may also observe that one cannot define the connective $\bullet$, or its negation, $\neg \bullet$, from the other connectives of **LFI1**, once their output value is $1/2$ whenever all the input values are $1/2$. □

Observe also that the matrices above can be written in a functional way as follows:

$$v(A \vee B) \overset{\text{def}}{=} \max(v(A), v(B))$$
$$v(\neg A) \overset{\text{def}}{=} 1 - v(A)$$
$$v(\bullet A) \overset{\text{def}}{=} 1 - |2v(A) - 1|$$

Consider now the logic $\mathbf{J}_3$, which also has 1 and $^1\!/_2$ as designated values, and whose matrices of $\wedge$, $\vee$, $\rightarrow$ and $\neg$ coincide with those of **LFI1**, but instead of the (inconsistency) connective $\bullet$ it has the (possibility) connective ∇, whose table is:

	∇
1	1
$^1\!/_2$	1
0	0

Then we may prove:

Remark 3.2. **LFI1** and $\mathbf{J}_3$ are inter-definable.
Notice that ∇A may be defined in **LFI1** as $A \vee \bullet A$, and, conversely, $\bullet A$ may be defined in $\mathbf{J}_3$ as $\nabla A \wedge \nabla\neg A$. □

As an obvious consequence, one obtains:

Corollary 3.3. ***LFI1*** *and* $\mathbf{J}_3$ *have the same theorems in the language* **L** *of the Classical Propositional Logic.* □

Let us now prove in details that the three-valued matrices presented above and the **LFI1**-interpretation defined in the last section really define the same logic:

Theorem 3.4 (Convenience). *Given an interpretation* $\boldsymbol{I}$ *of* **LFI1***, there is a three-valuation* v *obeying the matrices above and such that:*

$$v(A) \in \{1, {}^1\!/_2\} \iff \boldsymbol{I} \vDash A, \text{ for every formula } A.$$

Proof. Define v as:

$$v(A) = \begin{cases} 1 & \text{if } \boldsymbol{I} \nvDash \neg A \qquad \text{(i)}, \\ {}^1\!/_2 & \text{if } \boldsymbol{I} \vDash \bullet A \qquad \text{(ii)}, \\ 0 & \text{if } \boldsymbol{I} \nvDash A \qquad \text{(iii)}. \end{cases}$$

The reader should observe that, in case (i), it follows from (2.6) that $\boldsymbol{I} \vDash A$; in case (ii), it follows from (2.4) and (1.1) that $\boldsymbol{I} \vDash A$ and $\boldsymbol{I} \vDash \neg A$; in case (iii), it follows from (2.6) that $\boldsymbol{I} \vDash \neg A$. We proceed to show that v is a three-valuation.

If A is an atomic formula p, then $v(p) \in \{1, {}^1\!/\!_2\} \iff \boldsymbol{I} \vDash p$ follows immediately from definitions (i), (ii) and (iii). Let us now test the cases where A is a complex formula, to check if the function v here defined really obeys the three-valued matrices introduced above:

Case 1. A assumes the form $C \wedge D$.

Subcase 1.1. Suppose $v(C) = 1$ and $v(D) = 1$. By definition (i), $v(C) = 1$ and $v(D) = 1 \iff \boldsymbol{I} \nvDash \neg C$ and $\boldsymbol{I} \nvDash \neg D$. By clause (1.4) of the definition of $\boldsymbol{I}$, we have that $\boldsymbol{I} \nvDash \neg C$ and $\boldsymbol{I} \nvDash \neg D \iff \boldsymbol{I} \nvDash \neg(C \wedge D)$ By definition (i) again, $\boldsymbol{I} \nvDash \neg(C \wedge D) \iff v(C \wedge D) = 1$.

Subcase 1.2. Suppose $v(C) = {}^1\!/\!_2$ and $v(D) \neq 0$. By (i) and (ii), $v(C) = {}^1\!/\!_2$ and $v(D) \neq 0 \Longrightarrow \boldsymbol{I} \vDash \bullet C$ and $\boldsymbol{I} \vDash D$. From clause (1.1), $\boldsymbol{I} \vDash \bullet C$ and $\boldsymbol{I} \vDash D \iff \boldsymbol{I} \vDash \bullet C \wedge D$, and, from (3.1), $\boldsymbol{I} \vDash \bullet C \wedge D \Longrightarrow \boldsymbol{I} \vDash \bullet(C \wedge D)$. By (ii), $\boldsymbol{I} \vDash \bullet(C \wedge D) \iff v(C \wedge D) = {}^1\!/\!_2$.

Subcase 1.3. Suppose $v(D) = {}^1\!/\!_2$ and $v(C) \neq 0$. As in Subcase 1.2., *mutatis mutandis.*

Subcase 1.4. Suppose $v(C) = 0$ or $v(D) = 0$. By (iii), $v(C) = 0$ or $v(D) = 0 \iff \boldsymbol{I} \nvDash C$ or $\boldsymbol{I} \nvDash D$. From (1.1), $\boldsymbol{I} \nvDash C$ or $\boldsymbol{I} \nvDash D \iff \boldsymbol{I} \nvDash C \wedge D$. By (iii), $\boldsymbol{I} \nvDash C \wedge D \iff v(C \wedge D) = 0$.

Case 2. A assumes the form $C \vee D$.

Subcase 2.1. Suppose $v(C) = 1$ or $v(D) = 1$. By (i), $v(C) = 1$ or $v(D) = 1 \iff \boldsymbol{I} \nvDash \neg C$ or $\boldsymbol{I} \nvDash \neg D$. From clause (1.5), $\boldsymbol{I} \nvDash \neg C$ or $\boldsymbol{I} \nvDash \neg D \iff \boldsymbol{I} \nvDash \neg(C \vee D)$. By (i), $\boldsymbol{I} \nvDash \neg(C \vee D) \iff v(C \vee D) = 1$.

Subcase 2.2. Suppose $v(C) = {}^1\!/\!_2$ and $v(D) \neq 1$. By (ii) and (iii), $v(C) = {}^1\!/\!_2$ and $v(D) \neq 1 \Longrightarrow \boldsymbol{I} \vDash \bullet C$ and $\boldsymbol{I} \vDash \neg D$. From (1.1), $\boldsymbol{I} \vDash \bullet C$ and $\boldsymbol{I} \vDash \neg D \iff \boldsymbol{I} \vDash \bullet C \wedge \neg D$, and, from (3.2), $\boldsymbol{I} \vDash \bullet C \wedge \neg D \Longrightarrow \boldsymbol{I} \vDash \bullet(C \vee D)$. By (ii), $\boldsymbol{I} \vDash \bullet(C \vee D) \iff v(C \vee D) = {}^1\!/\!_2$.

Subcase 2.3. Suppose $v(D) = {}^1\!/\!_2$ and $v(C) \neq 1$. As in Subcase 2.2.

Subcase 2.4. Suppose $v(C) = 0$ and $v(D) = 0$. By (iii), $v(C) = 0$ and $v(D) = 0 \iff \boldsymbol{I} \nvDash C$ and $\boldsymbol{I} \nvDash D$. From (1.2), $\boldsymbol{I} \nvDash C$ and $\boldsymbol{I} \nvDash D \iff \boldsymbol{I} \nvDash C \vee D$. By (iii), $\boldsymbol{I} \nvDash C \vee D \iff v(C \vee D) = 0$.

Case 3. A assumes the form $C \to D$.

Subcase 3.1. Suppose $v(C) = 0$ or $v(D) = 1$. By (iii) and (i), $v(C) = 0$ or $v(D) = 1 \iff \boldsymbol{I} \nvDash C$ or $\boldsymbol{I} \nvDash \neg D$. From (1.6), $\boldsymbol{I} \nvDash C$ or $\boldsymbol{I} \nvDash \neg D \iff \boldsymbol{I} \nvDash \neg(C \to D)$. By (i), $\boldsymbol{I} \nvDash \neg(C \to D) \iff v(C \to D) = 1$.

Subcase 3.2. Suppose $v(C) \neq 0$ and $v(D) = {}^1\!/\!_2$. By (i) and (ii), $v(C) \neq 0$ and $v(D) = {}^1\!/\!_2 \Longrightarrow \boldsymbol{I} \vDash C$ and $\boldsymbol{I} \vDash \bullet D$. From (1.1), $\boldsymbol{I} \vDash C$ and $\boldsymbol{I} \vDash \bullet D \iff$

$\boldsymbol{I} \vDash C \wedge \bullet D$, and, from (3.3), $\boldsymbol{I} \vDash C \wedge \bullet D \iff \boldsymbol{I} \vDash \bullet(C \to D)$. By (ii), $\boldsymbol{I} \vDash \bullet(C \to D) \iff v(C \to D) = {}^1\!/_2$.

Subcase 3.3. Suppose $v(C) \neq 0$ and $v(D) = 0$. By (i), (ii) and (iii), $v(C) \neq 0$ and $v(D) = 0 \Longrightarrow \boldsymbol{I} \vDash C$ and $\boldsymbol{I} \nvDash D$. From (1.3), $\boldsymbol{I} \vDash C$ and $\boldsymbol{I} \nvDash D \iff \boldsymbol{I} \nvDash C \to D$. By (iii), $\boldsymbol{I} \nvDash C \to D \iff v(C \to D) = 0$.

Case 4. A assumes the form $\neg C$.

Subcase 4.1. Suppose $v(C) = 1$. By (i), $v(C) = 1 \iff \boldsymbol{I} \nvDash \neg C$, and, by (iii), $\boldsymbol{I} \nvDash \neg C \iff v(\neg C) = 0$.

Subcase 4.2. Suppose $v(C) = {}^1\!/_2$. By (ii), $v(C) = {}^1\!/_2 \iff \boldsymbol{I} \vDash \bullet C$. From (3.0), $\boldsymbol{I} \vDash \bullet C \iff \boldsymbol{I} \vDash \bullet(\neg C)$ and, by (ii) again, $\boldsymbol{I} \vDash \bullet(\neg C) \iff v(\neg C) = {}^1\!/_2$.

Subcase 4.3. Suppose $v(C) = 0$. By (iii), $v(C) = 0 \iff \boldsymbol{I} \nvDash C$. From (2.0), $\boldsymbol{I} \nvDash C \iff \boldsymbol{I} \nvDash \neg\neg C$, and, by (i), $\boldsymbol{I} \nvDash \neg\neg C \iff v(\neg C) = 1$.

Case 5. A assumes the form $\bullet C$.

Subcase 5.1. Suppose $v(C) \neq {}^1\!/_2$. By (i) and (iii), $v(C) \neq {}^1\!/_2 \iff \boldsymbol{I} \nvDash C$ or $\boldsymbol{I} \nvDash \neg C$. From (2.4) and (1.1), $\boldsymbol{I} \nvDash C$ or $\boldsymbol{I} \nvDash \neg C \iff \boldsymbol{I} \nvDash \bullet C$. By (iii), $\boldsymbol{I} \nvDash \bullet C \iff v(\bullet C) = 0$.

Subcase 5.2. Suppose $v(C) = {}^1\!/_2$. By (ii), $v(C) = {}^1\!/_2 \iff \boldsymbol{I} \vDash \bullet C$. From (2.5), $\boldsymbol{I} \vDash \bullet C \iff \boldsymbol{I} \nvDash \neg(\bullet C)$. By (i), $\boldsymbol{I} \nvDash \neg(\bullet C) \iff v(\bullet C) = 1$. □

Theorem 3.5 (Representability). *Given a three-valuation v obeying the matrices above, the relation "$\boldsymbol{I} \vDash$" defined as:*

$$\boldsymbol{I} \vDash A \iff v(A) \in \{1, {}^1\!/_2\}, \text{ for every formula } A,$$

provides an interpretation $\boldsymbol{I}$ of **LFI1**.

Proof. Here one should show either that clauses (1.1)–(1.3), (2.0)–(2.3), (3.1)–(3.3), or, alternatively, that clauses (1.1)–(1.6), (2.0), (2.4)–(2.6) are respected by this definition. We leave this as an exercise to the reader. □

It is now easy to see now that the *strong negation* $\sim$, defined for a formula A as:

$$\sim A \overset{\text{def}}{=} \neg A \wedge \neg \bullet A\,,$$

has all the properties of classical negation. Its matrix can be seen at below:

	$\sim$
1	0
${}^1\!/_2$	0
0	1

The corresponding clause, in terms of the interpretation $\boldsymbol{I}$ of **LFI1**, is:

(4.0) $\quad \boldsymbol{I} \vDash A \iff \boldsymbol{I} \nvDash {\sim} A.$

Expressibility. One way of measuring the *expressibility* of a many-valued logic is by tracking its capacity of defining, or 'expressing', different matrices, or functions. Thus, for instance, the two-valued Classical Propositional Logic, or the many-valued logics of Post are said to be *functionally complete*, for one can define all n-ary matrices using their connectives.

Let's say that a three-valued matrix is *hyper-classical* if the restriction of its associated function to the classical domain (values 1 and 0) will have its image in the classical codomain (values 1 and 0). Now we can state (cf. [Mar99]):

Theorem 3.6 (Expressibility I). *All hyper-classical three-valued matrices, and only them, are definable in* **LFI1**.

Proof. First of all, let s define the connective $\supset$ as: $A \supset B \stackrel{\text{def}}{=} (\neg A \vee B) \vee (\bullet A \wedge \bullet B)$. It is straightforward to check that this connective is such that: $v(A \supset B) = \min(1, 1 - v(A) + v(B))$ (this is just the implication from Łukasiewicz's logic $\mathbf{Ł}_3$). We will now define a series of unary and binary connectives, respectively $\otimes_i^k$ and $\oplus_{ij}^k$, such that:

$$\begin{aligned} v(\otimes_i^k A) &= \begin{cases} k & \text{if } v(A) = i \\ 0 & \text{otherwise} \end{cases} \\ v(A \oplus_{ij}^k B) &= \begin{cases} k & \text{if } v(A) = i \text{ and } v(B) = j \\ 0 & \text{otherwise} \end{cases} \end{aligned}$$

These connectives may be defined as follows:

$$\begin{aligned} \otimes_1^1 A &\stackrel{\text{def}}{=} \neg(A \supset \neg A) \\ \otimes_{1/2}^1 A &\stackrel{\text{def}}{=} (A \supset \neg A) \wedge (\neg A \supset A) \\ \otimes_{1/2}^{1/2} A &\stackrel{\text{def}}{=} A \wedge \neg A \\ \otimes_{1/2}^0 A &\stackrel{\text{def}}{=} \otimes_1^1 A \wedge \otimes_{1/2}^1 A \\ \otimes_0^1 A &\stackrel{\text{def}}{=} \otimes_1^1(\neg A) \\ A \oplus_{11}^1 B &\stackrel{\text{def}}{=} \neg(A \supset (A \supset \neg \otimes_1^1 B)) \\ A \oplus_{1\,1/2}^1 B &\stackrel{\text{def}}{=} \otimes_{1/2}^1 (A \oplus_{1\,1/2}^{1/2} B) \end{aligned}$$

$$\begin{aligned}
A \oplus^{1/2}_{1\,1/2} B &\overset{\text{def}}{=} \neg(A \supset (B \vee \neg B))\\
A \oplus^{1}_{10} B &\overset{\text{def}}{=} A \oplus^{1}_{11} \neg B\\
A \oplus^{1}_{1/2\,1} B &\overset{\text{def}}{=} \otimes^{1}_{1/2}(A \oplus^{1/2}_{1/2\,1} B)\\
A \oplus^{1/2}_{1/2\,1} B &\overset{\text{def}}{=} B \oplus^{1/2}_{1\,1/2} A\\
A \oplus^{1}_{1/2\,1/2} B &\overset{\text{def}}{=} \otimes^{1}_{1/2}(A \oplus^{1/2}_{1/2\,1/2} B)\\
A \oplus^{1/2}_{1/2\,1/2} B &\overset{\text{def}}{=} \neg((A \supset B) \supset ((\neg A \supset B) \supset B))\\
A \oplus^{0}_{1/2\,1/2} B &\overset{\text{def}}{=} (A \oplus^{1}_{11} B) \wedge (A \oplus^{1/2}_{1\,1/2} B)\\
A \oplus^{1}_{1/2\,0} B &\overset{\text{def}}{=} \otimes^{1}_{1/2}(A \oplus^{1/2}_{1/2\,0} B)\\
A \oplus^{1/2}_{1/2\,0} B &\overset{\text{def}}{=} \neg B \oplus^{1/2}_{1\,1/2} \neg A\\
A \oplus^{1}_{01} B &\overset{\text{def}}{=} \neg A \oplus^{1}_{11} B\\
A \oplus^{1}_{0\,1/2} B &\overset{\text{def}}{=} \otimes^{1}_{1/2}(A \oplus^{1/2}_{0\,1/2} B)\\
A \oplus^{1/2}_{0\,1/2} B &\overset{\text{def}}{=} \neg A \oplus^{1/2}_{1\,1/2} \neg B\\
A \oplus^{1}_{00} B &\overset{\text{def}}{=} \neg A \oplus^{1}_{11} \neg B
\end{aligned}$$

Using now the disjunction to ‘sum up’ the connectives above, we can easily build any hyper-classical unary or binary matrix. For example, if, for any reason, we needed to build the following matrices:

%	1	1/2	0
1	1	1/2	1
1/2	1	0	1
0	0	1/2	1

	$
1	1
1/2	0
0	1

then we should only write:

$$\begin{aligned}
A \,\%\, B &\overset{\text{def}}{=} (A \oplus^{1}_{11} B) \vee (A \oplus^{1/2}_{1\,1/2} B) \vee (A \oplus^{1}_{10} B) \vee\\
&\quad (A \oplus^{1}_{1/2\,1} B) \vee (A \oplus^{1/2}_{0\,1/2} B) \vee (A \oplus^{1}_{00} B)\\
\$\, A &\overset{\text{def}}{=} \otimes^{1}_{1} A \vee \otimes^{1}_{0} A
\end{aligned}$$

This shows that all unary and binary hyper-classical three-valued matrices are definable in **LFI1**. To see that no other unary or binary matrix is thereby defined, one should only observe that all connectives of **LFI1** have hyper-classical matrices.

This result can be easily adapted for any n-ary connective. For suppose one needed to build a hyperclassical 3-ary matrix §. Let A, B and C be its components. Fixing $v(C) = 1$, A and B will form a hyperclassical binary matrix, that we may represent by $\S_1(A, B)$, and the same will happen when we fix $v(C) = {}^1\!/_2$ and $v(C) = 0$, forming the hyperclassical matrices $\S_{1/2}(A, B)$ and $\S_0(A, B)$. Now it is immediate to check that $\S(A, B, C)$ may be written as $\otimes^1_1(\S_1(A, B)) \vee \otimes^1_{1/2}(\S_1(A, B)) \vee \otimes^1_0(\S_0(A, B))$. And so on, for any n-ary connective. □

Evidently, the result in Theorem 3.6 holds equally good for $\mathbf{J}_3$ (by Remark 3.2) and also for $\mathbf{Ł}_3$, the three-valued logic of Łukasiewicz, from which we borrowed the connective $\supset$ that we have defined and used above. In fact, $\mathbf{Ł}_3$ is usually defined using but the matrix of $\supset$, plus the matrix of $\neg$, the same negation of **LFI1** or $\mathbf{J}_3$. The (fundamental) difference between these logics lies in the fact that $\mathbf{Ł}_3$ has only 1 as designated value.

Constrasting with the concept of functional completeness, we may say now that a given many-valued logic is *functionally pre-complete* if, when we add to the logic any n-ary function which is not definable in it, then this logic turns to be functional complete. As pointed out in [Kar00], one may easily prove:

COROLLARY 3.7 (Expressibility II). **LFI1**, *as well as* $\mathbf{J}_3$ *and* $\mathbf{Ł}_3$, *are functionally pre-complete.*

PROOF. The reader just has to notice that the addition of a new n-ary function to **LFI1** will immediately enable us to define the unary connectives $\otimes^{1/2}_1$ and $\otimes^{1/2}_0$, as well as the binary connectives $\oplus^{1/2}_{11}$, $\oplus^{1/2}_{10}$, $\oplus^{1/2}_{01}$ and $\oplus^{1/2}_{00}$, which, together with the connectives in Theorem 3.6, will do the job. □

4. LFI1: Axiomatization and maximality

Hereunder we shall use $A \leftrightarrow B$ as an abbreviation for $(A \to B) \wedge (B \to A)$, and use also $\circ A$ as an abbreviation for $\neg \bullet A$. We will now show that **LFI1** may be axiomatized by the following axioms:

(Ax1) $A \to (B \to A)$

(Ax2) $(A \to B) \to ((A \to (B \to C)) \to (A \to C))$

(Ax3) $A \to (B \to (A \wedge B))$

(Ax4) $(A \wedge B) \to A$

(Ax5) $(A \wedge B) \to B$

(Ax6) $A \to (A \vee B)$

(Ax7) $B \to (A \vee B)$

(Ax8) $(A \to C) \to ((B \to C) \to ((A \vee B) \to C))$

(Ax9) $A \vee \neg A$

(Ax10) $\neg\neg A \leftrightarrow A$

(Ax11) $\circ A \to (A \to (\neg A \to B))$

(Ax12) $\bullet A \to (A \wedge \neg A)$

(Ax13) $\bullet(A \wedge B) \leftrightarrow ((\bullet A \wedge B) \vee (\bullet B \wedge A))$

(Ax14) $\bullet(A \vee B) \leftrightarrow ((\bullet A \wedge \neg B) \vee (\bullet B \wedge \neg A))$

(Ax15) $\bullet(A \to B) \leftrightarrow (A \wedge \bullet B)$

Having as rule: (MP) $A, A \to B/B$

Remark 4.1. The Deduction Theorem: $\Delta, A \vdash B \iff \Delta \vdash A \to B$, holds for **LFI1**.
This is clear from the presence of (Ax1) and (Ax2), and the fact that (MP) is the sole rule of **LFI1**.

Theorem 4.2 (Soundness). *All axioms above are validated by the matrices of* **LFI1**, *and the sole rule above,* (MP), *preserves validity.* □

The completeness of the above axiomatization of **LFI1** with respect to the matrix valuation proposed in the last section is but a special case of the general completeness proof of a class of 2^{13} three-valued paraconsistent logics, presented in [Mar??]. We now proceed to give an alternative proof of completeness of **LFI1**, adapting the well-known constructive method of Kalmár for the Classical Propositional Logic. To this end, we first state two auxiliary lemmas:

Lemma 4.3. *The following schemas are provable in* **LFI1***:*

(T1) $\circ A \to \circ(\neg A)$

(T2) $\bullet A \to \bullet(\neg A)$

(T3) $\circ(\bullet A)$

(T4) $((\circ A \wedge A) \wedge (\circ B \wedge B)) \to (\circ(A \wedge B) \wedge (A \wedge B))$

(T5) $((\circ A \wedge \neg A) \vee (\circ B \wedge \neg B)) \to (\circ(A \wedge B) \wedge \neg(A \wedge B))$

(T6) $((\circ A \wedge \neg A) \wedge (\circ B \wedge \neg B)) \to (\circ(A \vee B) \wedge \neg(A \vee B))$

(T7) $((\circ A \wedge A) \wedge (\circ B \wedge B)) \to (\circ(A \vee B) \wedge (A \vee B))$

(T8) $((\circ A \wedge \neg A) \vee (\circ B \wedge B)) \to (\circ(A \to B) \wedge (A \to B))$

(T9) $((\bullet A \vee (\circ A \wedge A)) \wedge (\circ B \wedge \neg B)) \to (\circ(A \to B) \wedge \neg(A \to B))$

(T10) $(\bullet A \wedge \bullet B) \to (\bullet(A \wedge B) \wedge \bullet(A \vee B) \wedge \bullet(A \to B))$ □

Lemma 4.4 (Kalmár-like). *Given a three-valuation v of* **LFI1***, in what follows we define for each formula A an associated formula A^v:*

$$A^v = \begin{cases} \circ A \wedge A & \text{if } v(A) = 1 \qquad \text{(i);} \\ \bullet A & \text{if } v(A) = 1 \qquad \text{(ii);} \\ \circ A \wedge \neg A & \text{if } v(A) = 0 \qquad \text{(iii).} \end{cases}$$

Given a formula G whose set of atomic variables is $\{p_1, p_2, \dots, p_n\}$, let's denote by Δ^v the set $\{p_1^v, p_2^v, \dots, p_n^v\}$. We state that the following holds: $\Delta^v \vdash G^v$.

Proof. The proof is by induction on the complexity of G. If G is an atomic variable itself, it is straight-forward, for $p^v \vdash p^v$. If G is complex, we suppose by induction hypothesis, (IH), that the lemma holds for any C and D subformulas of G, and must prove that the result for G follows from this supposition.

Case 1. G assumes the form $C \wedge D$.

Subcase 1.1. Suppose $v(C) = 1$ and $v(D) = 1$. By the matrices of **LFI1**, $v(G) = v(C \wedge D) = 1$. From definition (i), we have that $C^v = \circ C \wedge C$, $D^v = \circ D \wedge D$ and $G^v = \circ G \wedge G$, and, by (IH), we have $\Delta^v \vdash C^v$ and $\Delta^v \vdash D^v$. So, from (Ax3) and (T4), we conclude by (MP) that $\Delta^v \vdash G^v$.

Subcase 1.2. Suppose $v(C) = 1$ and $v(D) = 1/2$. So, $v(G) = 1$, and, from (i) and (ii), $C^v = \circ C \wedge C$, $D^v = \bullet D$ and $G^v = \bullet G$. Now, from (Ax13), we have in particular that $(\bullet D \wedge C) \to \bullet(C \wedge D)$. Therefore, sucessively applying (Ax5), (Ax3) and (IH), with (MP), we conclude once more that $\Delta^v \vdash G^v$.

Subcase 1.3. Suppose $v(C) = 1$ and $v(D) = 0$. So, $v(G) = 0$, and, from (i) and (iii), $C^v = \circ C \wedge C$, $D^v = \circ D \wedge \neg D$ and $G^v = \circ G \wedge \neg G$. We now use (Ax7), (T5), (IH) and (MP) to obtain the result.

Subcase 1.4. Suppose $v(C) = {}^1\!/_2$ and $v(D) = 1$. As in Subcase 1.2, *mutatis mutandis.*

Subcase 1.5. Suppose $v(C) = {}^1\!/_2$ and $v(D) = {}^1\!/_2$. Use now (T10).

Subcase 1.6. Suppose $v(C) = {}^1\!/_2$ and $v(D) = 0$. Exactly as in Subcase 1.3.

Subcase 1.7. Suppose $v(C) = 0$ and $v(D) = 1$. As in Subcase 1.3, but now using (Ax6) instead of (Ax7).

Subcase 1.8. Suppose $v(C) = 0$ and $v(D) = {}^1\!/_2$. As in Subcase 1.7.

Subcase 1.9. Suppose $v(C) = 0$ and $v(D) = 0$. As in Subcase 1.3 or Subcase 1.7.

Case 2. G assumes the form $C \vee D$. Here again there are nine subcases but we may make do with them using the axioms of **LFI1** and schemas (T6), (T7) and (T10).

Case 3. G assumes the form $C \to D$. Here we can get some help from (T8), (T9) and (T10).

Case 4. G assumes the form $\neg C$.

Subcase 4.1. Suppose $v(C) = 1$. So, $v(G) = 0$, and hence $C^v = \circ C \wedge C$, $G^v = \circ G \wedge \neg G = \circ(\neg C) \wedge \neg(\neg C)$. We now use (T1) and (Ax10).

Subcase 4.2. Suppose $v(C) = {}^1\!/_2$. Now $v(G) = {}^1\!/_2$, $C^v = \bullet C$, $G^v = \bullet G = \bullet(\neg C)$. Use (T2).

Subcase 4.3. Suppose $v(C) = 0$. Now $v(G) = 1$, $C^v = \circ C \wedge \neg C$, $G^v = \circ G \wedge G = \circ(\neg C) \wedge \neg C$. Use (T1) once more.

Case 5. G assumes the form $\bullet C$.

Subcase 5.1. Suppose $v(C) = 1$. Here, $v(G) = 0$, and then $C^v = \circ C \wedge C$, $G^v = \circ G \wedge \neg G = \circ(\bullet C) \wedge \neg(\bullet C)$. But, by (T3), $\circ(\bullet C)$ is a theorem of **LFI1**, and, by the definition of $\circ$, $\vdash \circ C \to \neg(\bullet C)$, and so it follows once more, by (Ax4), (IH) and (MP), that $\Delta^v \vdash G^v$.

Subcase 5.2. Suppose $v(C) = {}^1\!/_2$. Now $v(G) = 1$, $C^v = \bullet C$, $G^v = \circ G \wedge G = \circ(\bullet C) \wedge \bullet C$. Use (T3) again, and the result follows trivially.

Subcase 5.3. Suppose $v(C) = 0$. As in Subcase 5.1, *mutatis mutandis.* □

Theorem 4.5 (Completeness). *Every tautology of* **LFI1** *is a theorem according to the axiomatic presented above.*

Proof. Let G be a tautology whose atomic variables are, say, p_1, p_2, ..., p_n. So, by Lemma 1.12, $\Delta^v \vdash G$ for any valuation v. Denote by Δ_1^v the set $\Delta \setminus \{p_1^v\}$. We now choose three distinct valuations, differing exactly in p_1, i.e. such that $\Delta_1^{v_1} = \Delta_1^{v_2} = \Delta_1^{v_3}$ ($\overset{\text{def}}{=} \Delta_1$), but: (a) $v_1(p_1) = 1$; (b) $v_2(p_1) = {}^1\!/_2$; and (c) $v_3(p_1) = 0$.

In case (a), $p_1^v = \circ p_1 \wedge p_1$, so we have $\varDelta_1, \circ p_1 \wedge p_1 \vdash G$, or, what amounts to the same, by the Deduction Theorem:

(i) $$\varDelta_1, \circ p_1 \vdash p_1 \to G\,.$$

But, in case (c), $p_1^v = \circ p_1 \wedge \neg p_1$, so we have $\varDelta_1, \circ p_1 \wedge \neg p_1 \vdash G$, or, what amounts to the same:

(ii) $$\varDelta_1, \circ p_1 \vdash \neg p_1 \to G\,.$$

From (i) and (ii), by (Ax8) and (Ax9) — the proof by cases — we conclude:

(iii) $$\varDelta_1, \circ p_1 \vdash G\,.$$

Now, in case (b), $p_1^v = \bullet p_1 = \neg(\circ p_1)$, and so we have:

(iv) $$\varDelta_1, \neg(\circ p_1) \vdash G\,.$$

At last, from (iii) and (iv), using once more the proof by cases, we conclude:

$$\varDelta_1 \vdash G\,.$$

And so we got rid of the variable p_1. We may now recursively define the set $\varDelta_i^v$ as the set $\varDelta_{i-1}^v \setminus \{p_i^v\}$, $1 < i \leq n$, and then repeat the procedure above $n-1$ times. In the end of this process we obtain the empty set $\varDelta_n$, which is such that $\varDelta_n \vdash G$, and the proof is over. □

Maximality. A logic **L1** is said to be *maximal relative* to the logic **L2**, when both logics are written in the same language, **L2** proves all the theorems of **L1**, and, given a theorem G of **L2** which is not a theorem of **L1**, if we add G to **L1** as new axiom schema, then all theorems of **L2** turn to be provable.

Here we understand the Classical Propositional Logic, **CPL**, as the logic written from the connectives $\wedge$, $\vee$, $\to$ and $\neg$, with their usual classical interpretations. If we enrich **CPL** with a new connective, $\bullet$, whose interpretation is the one below:

	$\bullet$
1	0
0	0

we get what we shall call the Extended Classical Propositional Logic, **ECPL**. Axiomatically, this corresponds to adding $\neg \bullet A$ to **CPL** as a new axiom schema. It is easy to see that **ECPL** is a conservative extension of **CPL**, that is, if one restricts **ECPL** to the language of **CPL**, deleting the tautologies of **ECPL** in which the connective $\bullet$ intervenes, one gets no more (and certainly no less) tautologies than one already had in **CPL**.

Theorem 4.6 (Maximality). **LFI1** *is maximal relative to the* **ECPL**.

Proof. Let $g(p_1, p_2, ..., p_n)$, where $p_1, p_2, \ldots, p_n$ are its atomic variables, be a theorem of **ECPL** which is not a theorem of **LFI1**. We may assume, with no loss of generality, that $v(g(p_1, p_2, ..., p_n)) = 0$ iff $v(p_i) = 1$, for all p_i, $1 \leq i \leq n$. For if a given formula $h(p_1, p_2, ..., p_n, p_{n+1})$ assumes the value 0 for a valuation v such that $v(p_i) = {}^1\!/_2$, for $1 \leq i \leq n$, and $v(p_{n+1}) \neq {}^1\!/_2$, we may fix the value of the variable p_{n+1} substituting it for $p_1 \wedge \neg p_1 \wedge \neg \bullet p_1$ or for $p_1 \vee \neg p_1 \vee \bullet p_1$, respectively if $v(p_{n+1}) = 0$ or $v(p_{n+1}) = 1$, obtaining this way a formula of the form $g(p_1, p_2, ..., p_n)$.

Let us now consider another theorem of **ECPL**, say $t(q_1, q_2, ..., q_m)$, equally non-provable in **LFI1**. We must show that this formula is a theorem of **LFI1** plus $g(p_1, p_2, ..., p_n)$ (this last formula added to **LFI1** as a new axiom schema). For all q_j, $1 \leq j \leq m$, we build the formula $g(q_j)$, by substituting each variable in $g(p_1, p_2, ..., p_n)$ for q_j. Hence, for all three-valuation w in **LFI1**, we have $w(g(q_j)) = 0$ iff $w(q_j) = {}^1\!/_2$. Besides, for all such w, two situations may occur:

- $w(t(q_1, q_2, ..., q_m)) \neq 0$;
- $w(t(q_1, q_2, ..., q_m)) = 0$. This situation must occur, for some w, once $t(q_1, q_2, ..., q_m)$ is *not* a theorem of **LFI1**. But, in this case, there must be a q_j, $1 \leq j \leq m$, such that $w(q_j) = {}^1\!/_2$, once $t(q_1, q_2, ..., q_m)$ is a theorem of **ECPL**. And so we conclude that the conjunction $g(q_1) \wedge g(q_2) \wedge \cdots \wedge g(q_m)$ must assume the value 0 under this valuation.

In both cases above, the formula

$$(g(q_1) \wedge g(q_2) \wedge \cdots \wedge g(q_m)) \rightarrow t(q_1, q_2, ..., q_m)$$

will assume a designated value in **LFI1**. By the completeness of **LFI1**, this formula is provable in **LFI1**. But in **LFI1** plus $g(p_1, p_2, ..., p_n)$ the conjunction at the antecedent of the formula above is equally provable and so, by (MP), we prove $t(q_1, q_2, ..., q_m)$ in **LFI1** plus $g(p_1, p_2, ..., p_n)$. □

Being maximal means that there is no 'non-trivial' way of strengthening a given system. So far have we proved **LFI1** (and thus, $\mathbf{J}_3$ as well) to be maximal relative to the **ECPL**, and we come to know this way that **LFI1** has as much rules and schemas of **ECPL** as it could, without becoming **ECPL** itself. Now, what about the fragment of **LFI1** written in the language of the **CPL**, that is, without the contradictory connective $\bullet$? This propositional system built on the connectives $\wedge$, $\vee$, $\rightarrow$ and $\neg$, but no $\bullet$, with the three-valued interpretation above, was studied in [Bat80] and in [Avr86], where it received respectively the names PI^s and $\mathbf{RM}_3^{\tilde{\supset}}$. In those papers, Batens and Avron have shown how to axiomatize their system, and proved it to be maximal relative to the **CPL**.

The next result helps characterizing a certain aspect of the expressive power of **LFI1**, showing which instances of interdefinability of its connectives are valid.

PROPOSITION 4.7. *The following formulas are theorems of* **LFI1***:*

(DM1) $\neg(A \wedge B) \to (\neg A \vee \neg B)$

(DM2) $\neg(A \vee B) \to (\neg A \wedge \neg B)$

(DM3) $(\neg A \vee \neg B) \to \neg(A \wedge B)$

(DM4) $(\neg A \wedge \neg B) \to \neg(A \vee B)$

(ID1) $(A \to B) \to (\neg A \vee B)$

(ID2) $\neg(A \vee B) \to \neg(\neg A \to B)$

(ID3) $\neg(A \to B) \to \neg(\neg A \vee B)$

(IC1) $(A \to B) \to \neg(A \wedge \neg B)$

(IC2) $(A \wedge B) \to \neg(A \to \neg B)$

(IC3) $\neg(A \to B) \to (A \wedge \neg B)$ □

The fact that schemas (DM1)–(DM4), plus (Ax10) $\neg\neg A \leftrightarrow A$, are provable in **LFI1** makes its negation qualify as a *De Morgan negation.* It is easy to see why schemas $(A \vee B) \to (\neg A \to B)$ and $\neg(A \wedge B) \to (A \to \neg B)$ cannot be provable in **LFI1**. First, note that $\neg(C \wedge \neg C)$ is provable in **LFI1** (it is, in fact, equivalent to (Ax9) $C \vee \neg C$, by (DM1), (Ax10) and (MP)). Hence, by (Ax6), $\neg(C \wedge \neg C) \vee D$ is also provable. Now, if $(A \vee B) \to (\neg A \to B)$ were provable, taking A as $\neg(C \wedge \neg C)$ and B as D, one would obtain $\neg\neg(C \wedge \neg C) \to D$. By (Ax10), $(C \wedge \neg C) \to \neg\neg(C \wedge \neg C)$, and so one would prove $(C \wedge \neg C) \to D$ in **LFI1**, and it would not be paraconsistent. The result for $\neg(A \wedge B) \to (A \to \neg B)$ follows from (DM3) and from the previous result.

The reader should notice as well that (ID1) is the only one of the schemas in Proposition 4.7 that holds for logics as weak as the one built from (Ax1)–(Ax9) plus $\neg\neg A \to A$ and $A \vee (A \to B)$ (this last formula being deduced only when one adds (Ax11) to the logic). This pretty weak paraconsistent logic was studied in [CM99], where it was called C_{min}.

Remark 4.8. The connective $\leftrightarrow$ does not define a congruence in the algebra of **LFI1**. In particular, one has, for instance, that $\bullet A \leftrightarrow (A \wedge \neg A)$ holds in **LFI1**, while $\neg\bullet A \leftrightarrow \neg(A \wedge \neg A)$ does not (for $\neg(A \wedge \neg A) \rightarrow \neg\bullet A$ is not valid in **LFI1**). □

5. The quantified case: LFI1*

LFI1*, the first-order extension of **LFI1**, can be axiomatized by adding to the axioms of **LFI1** the following schemas:

(Ax16) $A(t) \rightarrow \exists x A(x)$, where t is term

(Ax17) $\forall x A(x) \rightarrow A(t)$, where t is term

(Ax18) $\neg\forall x A(x) \leftrightarrow \exists x \neg A(x)$

(Ax19) $\neg\exists x A(x) \leftrightarrow \forall x \neg A(x)$

(Ax20) $\bullet\forall x A(x) \leftrightarrow (\exists x \bullet A(x) \wedge \forall x A(x))$

(Ax21) $\bullet\exists x A(x) \leftrightarrow (\exists x \bullet A(x) \wedge \forall x \neg A(x))$

And the rules:

($\exists$-introduction) $A(x) \rightarrow B / \exists x A(x) \rightarrow B$, where x is not free in B

($\forall$-introduction) $B \rightarrow A(x) / B \rightarrow \forall x A(x)$

As mentioned in Definition 2.1, the interpretation for a sentence of $\mathbf{L}^{+}$ in a structure $\boldsymbol{I}$ is defined by the following clauses on the case of extended literals (and inductively extended to all first-order sentences through the other clauses, including (FO.1)–(FO.6)):

(0.1) $\boldsymbol{I} \vDash R(c_1, c_2, ..., c_n) \iff (c_1^{\boldsymbol{I}}, c_2^{\boldsymbol{I}}, ..., c_n^{\boldsymbol{I}}, \text{✗}) \in R^{\boldsymbol{I}+}$
or $(c_1^{\boldsymbol{I}}, c_2^{\boldsymbol{I}}, ..., c_n^{\boldsymbol{I}}, \checkmark) \in R^{\boldsymbol{I}+}$

(0.2) $\boldsymbol{I} \vDash \neg R(c_1, c_2, ..., c_n) \iff (c_1^{\boldsymbol{I}}, c_2^{\boldsymbol{I}}, ..., c_n^{\boldsymbol{I}}, \text{✗}) \in R^{\boldsymbol{I}+}$,
or both $(c_1^{\boldsymbol{I}}, c_2^{\boldsymbol{I}}, ..., c_n^{\boldsymbol{I}}, \text{✗}) \notin R^{\boldsymbol{I}+}$ and $(c_1^{\boldsymbol{I}}, c_2^{\boldsymbol{I}}, ..., c_n^{\boldsymbol{I}}, \checkmark) \notin R^{\boldsymbol{I}+}$

(0.3) $\boldsymbol{I} \vDash \bullet R(c_1, c_2, ..., c_n) \iff (c_1^{\boldsymbol{I}}, c_2^{\boldsymbol{I}}, ..., c_n^{\boldsymbol{I}}, \text{✗}) \in R^{\boldsymbol{I}+}$

Recalling that $\circ A$ is an abbreviation for $\neg\bullet A$, and using clause (2.5) of Proposition 2.3, we obtain:

(0.4) $\boldsymbol{I} \vDash \circ R(c_1, c_2, ..., c_n) \iff (c_1^{\boldsymbol{I}}, c_2^{\boldsymbol{I}}, ..., c_n^{\boldsymbol{I}}, \checkmark) \in R^{\boldsymbol{I}+}$,
or both $(c_1^{\boldsymbol{I}}, c_2^{\boldsymbol{I}}, ..., c_n^{\boldsymbol{I}}, \text{✗}) \notin R^{\boldsymbol{I}+}$ and $(c_1^{\boldsymbol{I}}, c_2^{\boldsymbol{I}}, ..., c_n^{\boldsymbol{I}}, \checkmark) \notin R^{\boldsymbol{I}+}$

as a consequence of (0.1)–(0.3). Now, the strong negation of A, denoted as $\sim A$ and defined by $\sim A \overset{\text{def}}{=} \neg A \wedge \neg\bullet A$, gives immediately:

(0.5) $\boldsymbol{I} \vDash\ \sim R(c_1, c_2, ..., c_n) \iff (c_1^{\boldsymbol{I}}, c_2^{\boldsymbol{I}}, ..., c_n^{\boldsymbol{I}}, \text{✗}) \notin R^{\boldsymbol{I}+}$
and $(c_1^{\boldsymbol{I}}, c_2^{\boldsymbol{I}}, ..., c_n^{\boldsymbol{I}}, \checkmark) \notin R^{\boldsymbol{I}+}$

The reader is invited to compare this last clause with the notion of default negation, discussed in the introduction, just after the $\boldsymbol{DS}$ example, so as to convince themselves that the strong and the default negations coincide.

We will now show that clauses (0.1)–(0.3) are equivalent to defining the interpretation $R^{\boldsymbol{I}}$ of a relation R as a function:

$$\chi\colon |\boldsymbol{I}|^n \to \{1, {}^1\!/\!_2, 0\},$$

such that the relation "$\boldsymbol{I} \vDash$" obeys the matrices of **LFI1** for extended literals, that is, given an atomic sentence $R(c_1, c_2, ..., c_n)$, then: (i) $\boldsymbol{I} \vDash R(c_1, c_2, ..., c_n)$ iff $\chi(c_1^{\boldsymbol{I}}, c_2^{\boldsymbol{I}}, ..., c_n^{\boldsymbol{I}})$ is designated; (ii) $\boldsymbol{I} \vDash \neg R(c_1, c_2, ..., c_n)$ iff $\neg\chi(c_1^{\boldsymbol{I}}, c_2^{\boldsymbol{I}}, ..., c_n^{\boldsymbol{I}})$ is designated; (iii) $\boldsymbol{I} \vDash \bullet R(c_1, c_2, ..., c_n)$ iff $\bullet\chi(c_1^{\boldsymbol{I}}, c_2^{\boldsymbol{I}}, ..., c_n^{\boldsymbol{I}})$ is designated.

Indeed, given $R^{\boldsymbol{I}+}$, the extended interpretation of R, define:

$$\chi(c_1^{\boldsymbol{I}}, c_2^{\boldsymbol{I}}, ..., c_n^{\boldsymbol{I}}) = \begin{cases} 1 & \text{if } (c_1^{\boldsymbol{I}}, c_2^{\boldsymbol{I}}, ..., c_n^{\boldsymbol{I}}, \text{✗}) \notin R^{\boldsymbol{I}+} \text{ and } (c_1^{\boldsymbol{I}}, c_2^{\boldsymbol{I}}, ..., c_n^{\boldsymbol{I}}, \checkmark) \in R^{\boldsymbol{I}+} \\ {}^1\!/\!_2 & \text{if } (c_1^{\boldsymbol{I}}, c_2^{\boldsymbol{I}}, ..., c_n^{\boldsymbol{I}}, \text{✗}) \in R^{\boldsymbol{I}+} \\ 0 & \text{if } (c_1^{\boldsymbol{I}}, c_2^{\boldsymbol{I}}, ..., c_n^{\boldsymbol{I}}, \text{✗}) \notin R^{\boldsymbol{I}+} \text{ and } (c_1^{\boldsymbol{I}}, c_2^{\boldsymbol{I}}, ..., c_n^{\boldsymbol{I}}, \checkmark) \notin R^{\boldsymbol{I}+} \end{cases}$$

Then, clearly the function χ does what it was proposed to do, and thus our structure $\boldsymbol{I}$ provides us with a way of talking about a relation being true (value 1), partially true (value ${}^1\!/\!_2$) or false (value 0) in a strictly set-theoretical way. Hence, the following result is immediate from the definition above and the truth-tables of **LFI1**:

Theorem 5.1 (Representability for relations). *Given a structure $\boldsymbol{I}$, then for any atomic sentence $R(c_1, c_2, ..., c_n)$ it holds that:*

(i) $\boldsymbol{I} \vDash R(c_1, c_2, ..., c_n) \iff \chi(c_1^{\boldsymbol{I}}, c_2^{\boldsymbol{I}}, ..., c_n^{\boldsymbol{I}}) \in \{1, {}^1\!/\!_2\}$,

(ii) $\boldsymbol{I} \vDash \neg R(c_1, c_2, ..., c_n) \iff \neg\chi(c_1^{\boldsymbol{I}}, c_2^{\boldsymbol{I}}, ..., c_n^{\boldsymbol{I}}) \in \{1, {}^1\!/\!_2\}$,

(iii) $\boldsymbol{I} \vDash \bullet R(c_1, c_2, ..., c_n) \iff \bullet\chi(c_1^{\boldsymbol{I}}, c_2^{\boldsymbol{I}}, ..., c_n^{\boldsymbol{I}}) \in \{1, {}^1\!/\!_2\}$. □

A many-valued interpretation for the quantifiers may be obtained, for instance, by means of the concept of *distribution quantifiers*, introduced in [Car87]. Using our basic intuition that a universal quantifier should work as a kind of unbounded conjunction, and an existential quantifier as an unbounded disjunction, the matrices of the conjunction and the disjunction of **LFI1** will give Table 1 on p. 140).

Now completeness for **LFI1*** can be proved extending the traditional Henkin's proof for classical first-order logic (on what would constitute an easy adaptation of the proof found in [DOt85] for $\mathbf{J}_3$), as follows:

Theorem 5.2 (Soundness and completeness of **LFI1***). *Given any set $\Delta \cup \{A\}$ of sentences in* **LFI1***, *we have $\Delta \vdash A \iff \Delta \vDash A$.*

distribution of instances x	$\forall x A(x)$	$\exists x A(x)$	– So we have, for example:
0, $^1\!/_2$, 1	0	1	this line means that there are instances of $A(x)$ assuming the values 0, $^1\!/_2$ and 1
0, $^1\!/_2$,	0	$^1\!/_2$	
0, 1	0	1	
$^1\!/_2$, 1	$^1\!/_2$	1	
0	0	0	this line means that all instances of $A(x)$ assume the value 0
$^1\!/_2$	$^1\!/_2$	$^1\!/_2$	
1	1	1	

Table 1.

Proof. As usual, the non-trivial part of the proof is to prove that $\Delta \vDash A \Longrightarrow \Delta \vdash A$. This can be done using the contrapositive form: $\Delta \nvdash A \Longrightarrow D \nvDash A$, via the Henkin construction of maximal consistent theories, by inductively extending the valuations starting from extended literals, whose interpretation will be defined by way of the function χ presented above, and defining then the interpretation of more complex propositions according to the clauses found on Definition 2.1. This is quite a standard procedure (though not free of encumbrances) and we will omit the details here. □

6. Comparing LFI1* with classical logic and with other paraconsistent logics

Given two logical systems $\mathbf{LS}_1 = (\boldsymbol{S}_1, \Vdash_1)$ and $\mathbf{LS}_2 = (\boldsymbol{S}_2, \Vdash_2)$, where $\boldsymbol{S}_1$ and $\boldsymbol{S}_2$ are sets of formulas, and $\Vdash_1$ and $\Vdash_2$ are consequence relations defined on them, what we call a translation from $\mathbf{LS}_1$ into $\mathbf{LS}_2$ is simply a morphism between these two structures, i.e. a function between their universes preserving their consequence relations. Formally, a translation Tr from $\mathbf{LS}_1$ into $\mathbf{LS}_2$ is a mapping $Tr\colon \boldsymbol{S}_1 \to \boldsymbol{S}_2$ such that, given $\Gamma \cup \{A\} \subseteq \boldsymbol{S}_1$, we have $\Gamma \Vdash_1 A \Longrightarrow Tr(\Gamma) \Vdash_2 Tr(A)$. If the converse also holds, i.e. if $\Gamma \Vdash_1 A \Longleftrightarrow Tr(\Gamma) \Vdash_2 Tr(A)$, we call the translation a *conservative translation*. (Conservative) Translations have proved to be a powerful tool on the study of some meta-theoretical properties of logics, for their mere existence allows us to talk about one logic inside some other logic, reproducing the inferential mechanisms of one inside the other. A well-known translation, the 'quantifier-forgetful functor', is used, for instance, to prove the consistency of the first-order classical logic from the consistency of propositional classical logic. General motivation and results about translations can be found in [CD98] and [dSDS98].

This section shows that the First-Order Classical Logic, **FOCL**, and the paraconsistent first-order logic C_1^* can be (conservatively) translated into **LFI1***. Thus, all first-order classical proofs and all proofs in C_1^* can be reproduced (up to a translation) inside **LFI1***.

We will first present two distinct translations from **FOCL** into **LFI1***. Note first that **FOCL** can be axiomatized by (Ax1)–(Ax10) (Section 4), (Ax16)–(Ax19) (Section 5), plus $A \to (\neg A \to B)$ and (MP). For the first translation we define the mapping Tr_1 from **FOCL** into **LFI1*** as follows (recalling that $\circ A$ is an abbreviation for $\neg \bullet A$):

(Tr1.1) $Tr_1(p) = \circ\, p$, if p is an atomic sentence or literal

(Tr1.2) $Tr_1(\neg A) = \neg\, Tr_1(A)$

(Tr1.3) $Tr_1(A \# B) = Tr_1(A) \# Tr_1(B)$, for # a binary connective

(Tr1.4) $Tr_1(@\, x A(x)) = @\, x\, Tr_1(A(x))$, for @ a quantifier

Now one may prove:

Theorem 6.1. *Tr_1 is a conservative translation from* **FOCL** *into* **LFI1***.

Proof. For most cases it is obvious that, if A is an axiom of **FOCL**, then Tr_1(A) is provable in **LFI1***. The only crucial case is the translation of $A \to (\neg A \to B)$, since this formula is not valid in **LFI1*** (compare it with (Ax11) $\circ A \to (A \to (\neg A \to B))$). However, $Tr_1(p \to (\neg p \to q)) = \circ\, p \to (\neg \circ p \to \circ\, q)$ is provable in **LFI1***. The converse is straightforward. □

For the second translation, Tr_2, define it as follows (recall that $\sim A$ is the strong negation of A):

(Tr2.1) $Tr_2(p) = p$, if p is an atomic sentence

(Tr2.2) $Tr_2(\neg A) = \sim Tr_2(A)$

(Tr2.3) $Tr_2(A \# B) = Tr_2(A) \# Tr_2(B)$, for # a binary connective

(Tr2.4) $Tr_2(@\, x A(x)) = @ x\, Tr_2(A(x))$, for @ a quantifier

From which follows:

Theorem 6.2. *Tr_2 is a conservative translation from* **FOCL** *into* **LFI1***.

Proof. Just note that $Tr_2(p \to (\neg p \to q)) = p \to ({\sim} p \to q)$ is provable in **LFI1***, and $\sim$ has all the properties of classical negation. Again, the converse is straightforward. □

Theorem 6.1 and Theorem 6.2 tell us that all forms of classical reasoning can, in a certain sense, be embedded and reproduced inside (a fragment of) **LFI1***, even though **LFI1*** (restricted to the language **L** of classical logic, that is, avoiding all formulas where the operator $\bullet$ occurs) is, strictly speaking, only a subsystem of classical logic. Furthermore, note that the above translations have a 'grammatical' aspect — they copy also the structure of the formulas being translated — which should not be overlooked. Now, if our task had been only that of finding translations, not necessarily conservative nor grammatical, between those logics, then it would have been quite trivial: indeed, any logic can be translated inside any other logic having at least one theorem G — just pick the constant function with output G.

Now, another question is: What is the relationship of **LFI1*** to the (first-order) paraconsistent logic C_1^* (cf. [dC74])? This question is not trivial, and we treat first only the propositional part of C_1^*, namely C_1, which we mentioned in Section 1 as part of the hierarchy C_n, $0 < n < \omega$. Axiomatically, each C_n can be axiomatized by (Ax1)–(Ax9) (Section 4), plus $\neg\neg A \to A$ (part of (Ax10)), plus (Ax11), (MP) and the following further axiom: $(\circ A \wedge \circ B) \to (\circ(A \wedge B) \wedge \circ(A \vee B) \wedge \circ(A \to B))$, which is based on clause (C2), presented in Section 1. The reader should only notice that in C_1, in particular, the formula $\circ A$ is in fact an abbreviation for the more complex formula $\neg(A \wedge \neg A)$, which clause (C1) takes as distinguished — whereas the remaining systems of the hierarchy will take different formulas as being abbreviated by $\circ A$ (cf. [CM99]).

In [Mar99] (and [Car00]), it has been proved that a possible-translations semantics for the C_n systems (and slightly stronger versions of them) can be semantically characterized by way of a combination of an infinite number of translations of C_n into a three-valued logic called $\mathbf{W}_3$ (and **LCD**), having three conjunctions, three disjunctions, three implications and two negations. This logic $\mathbf{W}_3$ turned to be in fact equivalent to $\mathbf{J}_3$ and so, transitively, to **LFI1**. Therefore, we have already at our disposal an infinite number of (grammatical and meaningful) translations from C_1 into **LFI1**. Now, C_1^* is obtained by adding to C_1 the axioms (Ax16)–(Ax19) (Section 5), plus $\forall x \circ A(x) \to \circ\forall x A(x) \wedge \circ\exists x A(x)$. To extend the translations mentioned above from C_1 into **LFI1** to translations from C_1^* into **LFI1*** one could use the 'quantifier-forgetful functor', but this would perhaps not produce very meaningful translations. So, we shall leave at this point as an open problem the task of producing more interesting such translations from C_1^* into **LFI1***.

Are there any ‘meaningful’ converse conservative translations from **LFI1*** into **FOCL** and into C_1^*? We also cannot fully answer this question at this time, but we should note that a partial answer has already been presented in [BDCK99]. This paper shows how a conservative translation from **LFI1** (called **CLuNs** in that paper) into **CPL** may be produced by an ingenious procedure which could be quickly summarized like this: first, the formulas of **CPL** are injectively mapped into an enhanced version of **CPL**, obtained by the addition of a denumerable set of new propositional variables; second, a bijection between the models of **LFI1** (suitably preprocessing its formulas) and the models of this enhanced version of **CPL** is presented (considering the negations of propositional variables as primitive elements, and mapping them into the new variables which have been added to **CPL**). A rearrangement of the propositional variables of enhanced **CPL** will of course transfer this translation into the original version of **CPL**. The task of extending this translation into the first-order version of those logics rests open, but the whole procedure seems feasibly generalizable for a much wider class of non-classical logics.

7. A competitor: the logic LFI2*

As remarked in the introduction, there is not just one single solution to the problem of providing formal systems to treat inconsistencies (or contradictions) with applications to databases. Indeed, with slight modifications on the semantical clauses of the interpretation in Section 2 we may obtain a second logic **LFI2** and its quantified extension **LFI2***. The modifications in order are shown below:

Definition 7.1. The interpretation $\vDash$ for the sentences in **LFI2**, written in the propositional fragment of $\mathbf{L}^+$, in a given structure $\boldsymbol{I}$, is inductively defined by clauses (1.1)–(1.3), (2.0)–(2.3), as in Definition 2.1 for **LFI1**, plus:

(3.4) $\boldsymbol{I} \vDash \bullet(A \wedge B) \iff \boldsymbol{I} \vDash \bullet A \wedge \bullet B$

(3.5) $\boldsymbol{I} \vDash \bullet(A \vee B) \iff \boldsymbol{I} \vDash \bullet A \wedge \bullet B$

(3.6) $\boldsymbol{I} \vDash \bullet(A \to B) \iff \boldsymbol{I} \vDash \bullet A \wedge \bullet B$

instead of (3.1)–(3.3). And for **LFI2***, the first-order version of **LFI2**, conditions (0.1)–(0.3) are once more used to interpret the extended literals, and the clauses for quantification (FO.1)–(FO.4) remain, but instead of (FO.5) and (FO.6) the following clauses are used:

$$\text{(FO.7)} \qquad \boldsymbol{I} \vDash \bullet\forall x A(x) \iff \boldsymbol{I} \vDash \forall x \bullet A(x)$$

$$\text{(FO.8)} \qquad \boldsymbol{I} \vDash \bullet(\exists x A(x)) \iff \boldsymbol{I} \vDash \forall x \bullet A(x)$$

The main intuition here is that a complex formula should be inconsistent exactly when all of its subformulas are inconsistent, independently of the connectives being considered. As a consequence, the intended semantical account for **LFI2** is totally uniform, in the sense that it does not come from the particular structure of a given formula — and that's why we say that the logic **LFI2** models inconsistencies as internal phenomena.

The property in Fact 2.2 still holds good for **LFI2**, as well as part (i) of the succeeding Proposition 2.3, but part (ii) of that same proposition should here be slightly modified, as follows:

Proposition 7.2. (ii) *Clauses* (3.4)–(3.6) *may be substituted by:*

$$\text{(1.7)} \qquad \boldsymbol{I} \vDash \neg(A \wedge B) \iff (\boldsymbol{I} \nvDash A \text{ or } \boldsymbol{I} \nvDash B) \text{ or } \boldsymbol{I} \vDash \bullet(A \wedge B)$$

$$\text{(1.8)} \qquad \boldsymbol{I} \vDash \neg(A \vee B) \iff (\boldsymbol{I} \nvDash A \text{ and } \boldsymbol{I} \nvDash B) \text{ or } \boldsymbol{I} \vDash \bullet(A \vee B)$$

$$\text{(1.9)} \qquad \boldsymbol{I} \vDash \neg(A \to B) \iff (\boldsymbol{I} \vDash A \text{ and } \boldsymbol{I} \nvDash \neg B) \text{ or } \boldsymbol{I} \vDash \bullet(A \to B) \qquad \square$$

The reader might observe that if the rightmost disjunct in (1.7)–(1.9) is omitted, one obtains exactly the Extended Classical Propositional Logic, i.e. the Classical Propositional Logic written in a language enriched with an operator •, interpreted as always false (see Section 4, Maximality).

The analogue of Remark 2.4 still holds good for **LFI2**.

Three-valued matrices for LFI2. We will now show that the interpretation above for **LFI2** also defines a three-valued logic, one which is represented by the matrices below:

$\wedge$	1	½	0
1	1	1	0
½	1	½	0
0	0	0	0

$\vee$	1	½	0
1	1	1	1
½	1	½	1
0	1	1	0

$\to$	1	½	0
1	1	1	0
½	1	½	0
0	1	1	1

	$\neg$	$\bullet$
1	0	0
½	½	1
0	1	0

where 1 and ½ are the designated values.

Fact 7.3. *The connectives* $\vee$, $\neg$ *and* $\bullet$ *may be taken as primitive in* **LFI2**.

Proof. Notice that $A \wedge B$ and $A \to B$ may be defined, respectively, as $\neg(\neg(A \vee \neg(A \vee B)) \vee \neg(B \vee \neg(A \vee \neg B)))$ and $B \vee \neg(A \vee B)$. □

The analogues of Remark 3.2 and its Corollary 3.3 do not apply to **LFI2**: The matrices above define a brand new three-valued logic, different from $\mathbf{J}_3$ and from any other logic we have heard of (but define, once more, only a special case of the 2^{13} logics studied in [Mar??]).

One can now easily adapt the proofs of convenience and representability (Theorem 3.4 and Theorem 3.5) to **LFI2**, in terms of the three-valued matrices presented above.

The strong negation for **LFI2** is defined as in **LFI1**, and evidently has the same properties.

It is also to be noted that the results on expressibility of Theorem 3.6 do not obtain for **LFI2**. Indeed, it is easy to see that all matrices definable in **LFI2** are hyper-classical, but not all hyper-classical matrices are expressible by **LFI2**.

An axiomatization for **LFI2** is obtainable as in **LFI1**, from (Ax1)–(Ax12) and the rule of (MP) (Section 1), plus:

(Ax13′) $\bullet(A \wedge B) \leftrightarrow (\bullet A \wedge \bullet B)$

(Ax14′) $\bullet(A \vee B) \leftrightarrow (\bullet A \wedge \bullet B)$

(Ax15′) $\bullet(A \to B) \leftrightarrow (\bullet A \wedge \bullet B)$

instead of (Ax13)–(Ax15).

The Deduction Theorem and the soundness of the axioms of **LFI2** relative to the matrices above are immediately verifiable, as in the case of **LFI1** (Remark 4.1 and Theorem 4.2). One of the main distinguishing features of **LFI2** is that its metaproofs are in general much easier to produce, due to the homogeneous semantical treatment given to all its connectives. To constructively show its completeness, for instance, one can make use of the following simpler lemmas (analogues of Lemma 4.3 and Lemma 4.4):

Lemma 7.4. *The following schemas are provable in* **LFI2**:

(T1) $\circ A \to \circ(\neg A)$

(T2) $\bullet A \to \bullet(\neg A)$

(T3) $\circ(\bullet A)$

(T10) $(\bullet A \wedge \bullet B) \to (\bullet(A \wedge B) \wedge \bullet(A \vee B) \wedge \bullet(A \to B))$

(T11) $(\circ A \vee \circ B) \to (\circ(A \wedge B) \wedge \circ(A \vee B) \wedge \circ(A \to B))$

Proof. Schemas (T1)–(T3) and (T10) come from Lemma 4.3, and to prove schema (T11) one should use the new axioms, (Ax13′)-(Ax15′). □

Notice that from (T2), and its converse (which is also provable in both **LFI1** and **LFI2**), and from (Ax13′)–(Ax15′), one can easily conclude that $\bullet A$ is provable in **LFI2** iff $\bullet p$ is provable, for every atomic variable p which intervenes in A. Now to the constructive lemma (where definitions (i) and (iii) are simplified, compared to Lemma 4.4):

Lemma 7.5 (Kalmár-like). *Given a three-valuation v of* **LFI2**, *in what follows we define for each formula A an associated formula A^v:*

$$A^v = \begin{cases} \circ A & \text{if } v(A) = 1; \\ \bullet A & \text{if } v(A) = {}^1\!/_2; \\ \circ A, & \text{if } v(A) = 0. \end{cases}$$

Given a formula G whose set of atomic variables is $\{p_1, p_2, \ldots, p_n\}$, let's denote by Δ^v the set $\{p_1^v, p_2^v, \ldots, p_n^v\}$. We state that the following holds: $\Delta^v \vdash G^v$.

Proof. The proof for the atomic case and the case where G assumes the form $\neg C$ or the form $\bullet C$ is identical to the one given before, in Lemma 4.4. Now, for the cases where G assumes the form $C \wedge D$, $C \vee D$ or $C \to D$, one should use (T10) and (T11). □

Completeness itself is also easier to prove for the propositional case:

Theorem 7.6 (Completeness). *Every tautology of* **LFI2** *is a theorem according to its axiomatic.*

Proof. We proceed as in Theorem 4.5, but now using Lemma 7.5. Now, in both the cases where $v_1(p_1) = 1$ and where $v_3(p_1) = 0$, we have $\Delta_1, \circ p_1 \vdash G$. But in the case where $v_2(p_1) = 1$, we have $\Delta_1, \bullet p_1 \vdash G$, which is the same as $\Delta_1, \neg(\circ p_1) \vdash G$. Using the proof by cases, we get $\Delta_1 \vdash G$. □

Finally, one may also prove:

Theorem 7.7 (Maximality). **LFI2** *is maximal relative to the* **ECPL**.

Proof. The proof follows exactly the same path as the proof of maximality of **LFI1** (Theorem 4.6). □

What about the maximality of the fragment of **LFI2** in the language of the **CPL** relative to the same **CPL** — does it hold? We cannot answer this question at this time.

As to the interdefinability of the connectives of **LFI2**:

Proposition 7.8. *The following formulas are theorems of* **LFI2***:* (DM1), (DM2), (ID1), (ID2) *and* (IC3)*, as in Proposition 4.7, plus the schema:*

(IC4) $\neg(A \wedge B) \to (A \to \neg B)$ □

The (counter-)argument offered just after Proposition 4.7 is still valid for **LFI2**, but one should only notice that the proof of $\neg(C \wedge \neg C)$ in **LFI2** is not as immediate as the proof in **LFI1**, where this formula was equivalent to $C \vee \neg C$, an axiom of both these logics. In fact, in **LFI2**, this formula is a little bit harder to prove, and it is instructive to show how this can be done. First note that (Ax1)–(Ax11) are sufficient to prove $\neg \bullet C \to \neg(C \wedge \neg C)$. Now, from (Ax10), more specifically from $C \to \neg\neg C$, one may prove $\bullet C \to \bullet \neg C$, and from this it immediately follows that $\bullet C \to (\bullet C \wedge \bullet \neg C)$. From (Ax13′), it follows that $(\bullet C \wedge \bullet \neg C) \leftrightarrow \bullet(C \wedge \neg C)$, and from (Ax12) follows $\bullet(C \wedge \neg C) \to ((C \wedge \neg C) \wedge \neg(C \wedge \neg C))$, and so $\bullet(C \wedge \neg C) \to \neg(C \wedge \neg C)$. Therefore, by transitivity, $\bullet C \to \neg(C \wedge \neg C)$. Using the proof by cases, one may at last prove $\neg(C \wedge \neg C)$.

In some sense, one could say that the ‘cost’ of accepting (IC4), for **LFI2**, which was not accepted by **LFI1**, was that of losing half of de Morgan’s laws — (DM3) and (DM4) — and also schemas (ID3), (IC1) and (IC2), once provable in **LFI1**. Notwithstanding, the maximality of both **LFI1** and **LFI2** prevents us from asserting that **LFI1** is in any sense ‘stronger’ than **LFI2**: They are simply different logics, though much similar in many aspects — for example, the analogue of Remark 4.8 also holds good for **LFI2**, which means that both systems are equally hard to algebrize. However, both these logics are certainly solutions to the same problem.

In [Bez?a], Béziau introduces a propositional paraconsistent logic he called **Z**, based on the possible-worlds semantics for the modal logic **S5**, in which the paraconsistent negation $\neg A$ of a formula A is equated with $\Diamond \sim A$, the possibility of the (classical) strong negation of A (cf. also [Bez?b]). The maximal three-valued logic **LFI2** and the modal logic **Z** plus the axiom schema $A \to \neg\neg A$ seem extremely close to each other, and we shall leave as an interesting open problem that of characterizing the exact relation between these two logics.

First-order. **LFI2***, the first-order version of **LFI2**, can be axiomatized by adding (Ax16)–(Ax19) to the axioms of **LFI2**, as in the case of **LFI1*** (Section 5), plus:

(Ax20′) $\bullet \forall x A(x) \leftrightarrow \forall x \bullet A(x)$

(Ax21′) $\bullet \exists x A(x) \leftrightarrow \forall x \bullet A(x)$

instead of (Ax20)–(Ax21), and the rules of (∃-introduction) and (∀-introduction).

The proofs of soundness and completeness of **LFI2*** follow that of **LFI1***, found in Section 5, suitably adapting the previous arguments. Here one should just notice that, following the new matrices of conjunction and disjunction, the adequate (distribution) quantifiers for **LFI2*** would have distinct behaviors from those of **LFI1***, and these new behaviors are pictured below:

distribution of instances x	$\forall x A(x)$	$\exists x A(x)$
0, ½, 1	0	1
0, ½,	0	1
0, 1	0	1
½, 1	1	1
0	0	0
½	½	½
1	1	1

Now the same translations Tr_1 and Tr_2 (Section 6) used to embed **FOCL** into **LFI1*** will work as well to effectively embed **FOCL** into **LFI2***. In fact, these translations dispense (Ax13)–(Ax15) and (Ax20)–(Ax21), in both versions. Stronger versions of the C_n^* logics, called C_n^{+*}, are obtained if one only changes the axioms $(\circ A \wedge \circ B) \to (\circ(A \wedge B) \wedge \circ(A \vee B) \wedge \circ(A \to B))$ and $\forall x \circ A(x) \to \circ \forall x A(x) \wedge \circ \exists x A(x)$ of C_1^* for their counterparts $(\circ A \vee 1B) \to (\circ(A \wedge B) \wedge \circ(A \vee B) \wedge \circ(A \to B))$ and $\exists x \circ A(x) \to \circ \forall x A(x) \wedge \circ \exists x A(x)$, based on clause (C3), Section 1. A possible-translations semantics for the calculus C_1^+ based on a combination of an infinite number of translations of C_1^+ into **LFI2** can be found either in [Mar99] or in [CM??]. The method devised in [BDCK99] for conservatively translating **CPL** inside of **LFI1** seems easily adaptable for the case of **LFI2**.

8. Concluding remarks

We have presented the logics of formal inconsistency **LFI1** and **LFI2** and their first-order extensions, **LFI1*** and **LFI2***. These systems are but two representatives of the possibilities of axiomatizing the notions of consistency (and inconsistency), which help to clarify philosophical positions with respect to contradictions and inconsistencies in logic. **LFI1*** and **LFI2*** have been introduced for the special purpose of founding an approach to the treatment of information in evolutionary databases, and have several interesting features: (i) they have a Hilbertian-style proof theory; (ii) they have been proved to be sound and complete with respect to three-valued semantics; (iii) the propositional fragment of **LFI1** is maximal; (iv) **LFI1*** and **LFI2*** are new paraconsistent first-order logics (although **LFI1*** is in fact equivalent to the first-order extension of the three-valued logic $\mathbf{J}_3$); and, finally, (v) it has been proved, via translations, that all classical and traditional paraconsistent reasoning can be reproduced inside **LFI1***.

As we observed, **LFI1*** and **LFI2*** represent two radically different approaches concerning inconsistent formulas: for **LFI1***, inconsistency is an external phenomenon, while for **LFI2*** it is internal. The present approach to the logics of formal inconsistency and their uses in evolutionary databases is a solution to the problem introduced in [CdA99] concerning the search for a useful, conceptually clear and feasible logic for treating inconsistent or contradictory data. Although these are not the only solutions, we believe them to be fully adequate and specially meaningful ones. We would like to notice that, in the present case, we are dealing with general relations (not necessarily finite), contrary to what is adopted in the database field. We can rather safely conjecture, however, that Trakhtenbrot's Theorem, which proves that the set of sentences of first-order classical logic which are valid in all finite structures is not recursively enumerable (cf. [Tra50]), is also provable for the **LFI** logics. In such cases, what is usual is to consider certain special classes of integrity constraints, for which completeness can be attained. We believe this can be also done for **LFI** logics. A more detailed treatment of databases using **LFI1*** will be given in [dACM??].

We wish to thank the referees of WoLLIC'99 and the audience of the Jaśkowski Memorial Symposium, in Toruń, who encouraged us to pursue this research.

References

[Avr86] A. Avron, “On an implication connective of **RM**,” Notre Dame Journal of Formal Logic 27(2), 201–209, 1986.

[Bat80] D. Batens, “Paraconsistent extensional propositional logics,” Logique et Analyse 90–91, 195–234, 1980.

[BDCK99] D. Batens, K. De Clercq and N. Kurtonina, “Embedding and interpolation for some paralogics. The propositional case,” Reports on Mathematical Logic 33, 29–44, 1999.

[Bel77] N. D. Belnap, “A useful four-valued logic,” pp. 7–37 in *Modern Uses of Multiple-Valued Logic*, J. M. Dunn and G. Epstein (Eds.). D. Reidel Publishing Co, Boston, 1977.

[Bez?a] J.-Y. Béziau, “The paraconsistent logic **Z**. A possible solution to Jaśkowski’s problem,” submitted to publication.

[Bez?b] J.-Y. Béziau, “**S5** is a paraconsistent logic and so is classical first-order logic,” submitted to publication.

[Car87] W. A. Carnielli, “Systematization of finite many-valued logics through the method of tableaux,” The Journal of Symbolic Logic 52(2), 473–493, 1987.

[Car00] W. A. Carnielli, “Possible-translations semantics for paraconsistent logics,” pp. 149–63 in *Frontiers in Paraconsistent Logic*: Proc. of the I World Congress on Paraconsistency, Ghent, Belgium, D. Batens et alia (Eds.), King’s College Publications, 2000.

[CD98] W. A. Carnielli, W. A. and I. M. L. D’Ottaviano, “Translations between logical systems: a manifesto,” Logique et Analyse 157, 67–81, 1997.

[CdA99] W. A. Carnielli and S. de Amo, “A logic-based system for controlling inconsistencies in evolutionary databases,” pp. 89–101 in *Proc. of the VI Workshop on Logic, Language, Information and Computation* (WoLLIC’99), Itatiaia, Brazil, 1999.

[CLM92] W. A. Carnielli and M. Lima-Marques, “Reasoning under inconsistent knowledge,” Journal of Applied Non-Classical Logics 2(1), 49–79, 1992.

[CM99] W. A. Carnielli and J. Marcos, “Limits for paraconsistent calculi,” Notre Dame Journal of Formal Logic, 40(3), 375–390, 1999.

[CM01] W. A. Carnielli and J. Marcos, “A taxonomy of **C**-systems,” in W. A. Carnielli, M. E. Coniglio, and I. M. L. D’Ottaviano, editors, *Paraconsistency: The logical way to the inconsistent.* Proceedings of the II World Congress on Paraconsistency (WCP’2000), pp. 1–94. Marcel Dekker, 2001.

[CM??] W. A. Carnielli and J. Marcos, “Semantics for **C**-systems,” forthcoming.

[Cod70] E. F Codd, "A relational model of data for large shared data banks," Communications of the ACM 13(6), 377–387, 1970.

[dACM??] S. de Amo, W. A. Carnielli, and J. Marcos, "A logical framework for integrating inconsistent information in multiple databases," to appear in Proceedings of the Second International Symposium on Fundations of Information and Knowledge Systems (FoIKS 2002), Schloß Sulzan, Germany, February 2002.

[dC74] N. C. A. da Costa, "On the theory of inconsistent formal systems," Notre Dame Journal of Formal Logic 11, 497–510, 1974.

[dCBB95] N. C. A. da Costa, J.-Y. Béziau and O. Bueno, "Aspects of paraconsistent logic," Bulletin of the IGPL 3(4), 597–614, 1995.

[dSDS98] J. J. da Silva, I. M. L. D'Ottaviano, and A. M. Sette, "Translations between logics," pp. 435–48 in *Models, Algebras and Proofs: Proceedings of the X Latin American Symposium on Mathematical Logic*, X. Caicedo, C. H. Montenegro (Ed.), Lecture Notes in Pure and Applied Mathematics, Marcel Dekker, New York, 1998.

[DOt85] I. M. L. D'Ottaviano, "The completeness and compactness of a three-valued first-order logic", in *Proceedings of the V Latin American Symposium on Mathematical Logic*, Bogotá, 1981. Revista Colombiana de Matemáticas 19(1–2), 77–94, 1985.

[DOdC70] I. M. L. D'Ottaviano and N. C. A. da Costa, "Sur un probleme de Jaśkowski," Comptes Rendus de l'Academie de Sciences de Paris sér. A–B 270, 1349–1353, 1970.

[Fit91] M. Fitting, "Bilattices and the semantics of logic programming," Journal of Logic Programming 11, 91–116, 1991.

[Jas68] S. Jaśkowski, "Rachunek zdan dla systemów dedukcyjnych sprzecznych," Studia Soc. Scient. Torunensis sec. A1(5), 55–77, 1948 (English translation in Studia Logica 24, 143–157, 1969.)

[Kar00] A. S. Karpenko, "A maximal paraconsistent logic: the combination of two three-valued isomorphs of classical propositional logic," pp. 181–187 in *Frontiers in Paraconsistent Logic*: Proc. of the I World Congress on Paraconsistency, Ghent, Belgium, D. Batens et alia (Eds.), King's College Publications, 2000.

[KL92] M. Kifer and E. L. Lozinskii, "A logic for reasoning with inconsistency," Journal of Automated Reasoning 9(2), 179–215, 1992.

[Mar99] J. Marcos, *Possible-Translations Semantics* (in Portuguese), Thesis, State University of Campinas, IFCH, 1999. (ftp://www.cle.unicamp.br/pub/thesis/J.Marcos/)

[Mar??] J. Marcos, "8K solutions and semi-solutions to a problem of da Costa," forthcoming.

[PRN89] G. Priest, R. Routley, and J. Norman (Eds.), *Paraconsistent Logic: essays on the inconsistent*, Philosophia Verlag, München, 1989.

[RM70] N. Rescher and R. Manor, "On inference from inconsistent premises," Theory and Decision 1, 179–219, 1970.

[Set73] A. M. Sette, "On the propositional calculus $\mathbf{P}^1$," Mathematica Japonicae 18, 173–80, 1973.

[Tra50] B. A. Trakhtenbrot, "The impossibility of an algorithm for the decision problem for finite domains" (in Russian), Doklady Akademii Nauk SSSR (N.S.) 70, 569–572, 1950.

Walter A. Carnielli
Group for Theoretical and Applied Logic CLE/IFCH
State University of Campinas, Brazil
carniell@cle.unicamp.br

João Marcos
IFCH
State University of Campinas, Brazil
vegetal@cle.unicamp.br

Sandra de Amo
Department of Computer Sciences
Federal University of Uberlândia, Brazil
deamo@ufu.br