# Paraconsistent Logics: Preamble

**Jair Minoro Abe**

**Abstract** In this introductory chapter it is introduced some aspects of paraconsistent logics, such as its brief historical developments, some main systems and mention some applications. The chapter obviously does not cover many topics: moreover it is far from to be complete. In fact, the theme is now widespread and occupies a distinguished position in academia. The most part of the chapter focuses on a special class of paraconsistent logic, namely the paraconsistent annotated logics.

**Keywords** Paraconsistent logic · Annotated logic · Non-classical logic · Rival logic · Logic and applications

## 1 Introduction

Nowadays its is accepted by innumerous logicians that the classical first order predicate calculus (and some of its important subsystems like propositional calculus and some of its extensions like ZF-set theory) constitutes the nucleus of the so-called classical logic.

The term non-classical logic is a generic term which has been employed to refer to any logic other than classical logic. Roughly they are divided in two groups: those that complement or extend classical logic and those that rival classical logic.

The first group, complementary to the classical logic, keeps the classical logic on its basis. It is supplemented by extending its language enriching its vocabulary and/or the theorems of these non-classical logics supplement those of classical logic. Thus all valid schemes of classical logic remains valid with the addition of

J.M. Abe (✉)
Graduate Program in Production Engineering, ICET—Paulista University,
São Paulo, Brazil
e-mail: jairabe@uol.com.br

J.M. Abe
Institute for Advanced Studies, University of São Paulo, São Paulo, Brazil



J.M. Abe (ed.), *Paraconsistent Intelligent-Based Systems*,
Intelligent Systems Reference Library 94, DOI 10.1007/978-3-319-19722-7_1

new theorems. As example, modal logics add symbols like L (it is necessary that) in its language to express modalities (also I—it is impossible that, C—it is contingent that, M—it is possible that). Additional axiom schemes are added, e.g., $A \rightarrow \mathrm{M}A$ ($A$ implies that it is possible $A$) and appropriate inference rules are also considered. In the same way, we can also consider deontic modalities O (it is obligatory that), P (it is permitted that), F (it is forbidden that), etc. In the second group, rival logics (or heterodox logics) substitute partially or even totally the classical logic. So, in heterodox systems, some principles valid in classical logic are not valid in the last. For example, let us consider the law of the excluded middle, in the form $A$ or not $A$. This is not valid, v.g. in intuitionistic logics, some many-valued logics, in paraconsistent logics, etc.

It should be emphasized that such division is somewhat vague; in effect, there are complementary logics that can be viewed as rival ones and vice versa. The choice how to consider them should take into account pragmatic issues.

In this chapter we are concerned with a category of rival logics, namely the paraconsistent logics.

## 1.1 Historical Aspects

The paraconsistent logic had as precursors the Russian logician Nicolai Alexandrovich Vasiliev (1880–1940) and the Polish logician Jan Łukasiewicz (1878–1956). Both, in 1910, independently published papers in which they treated the possibility of a logic that does not eliminate ab initio the contradictions. However, the works of these authors, regarding to paraconsistency, were restricted to the traditional Aristotelian logic. Only in 1948 and 1954 that the Polish logician Stanislaw Jaśkowski (1906–1965) and the Brazilian logician Newton C.A. da Costa (1929), respectively, although independently, built the paraconsistent logic.

Jaśkowski has formalized a paraconsistent propositional calculus called discursive (or discussive) propositional calculus while da Costa constructed for the first time hierarchies of paraconsistent propositional calculi $C_{\mathrm{i}}(1 \leq i \leq \omega)$, of paraconsistent first-order predicate calculi $C_{\mathrm{i}}^{*}(1 \leq i \leq \omega)$ (with or without equality), of paraconsistent description calculi $D_{\mathrm{i}}(1 \leq i \leq \omega)$, and paraconsistent higher-order logics (systems $NF_{\mathrm{i}}$, $1 \leq i \leq \omega$).

Also, in parallel (Nels) David Nelson (1918–2003) in 1959 investigated his constructive logics with strong negation closely related with ideas of paraconsistency.

The term 'paraconsistent' was coined by the Peruvian philosopher Francisco Miró Quesada Cantuarias (1918) at the Third Latin America Conference on Mathematical Logic held in State University of Campinas, in 1976. At the time Quesada seems to have in mind the meaning 'quasi' (or 'similar to, modelled on') for the prefix 'para'.

## *1.2 Inconsistent Theories and Trivial Theories*

The most important reason to consider the paraconsistent logic is the construction of theories in which inconsistencies are allowed without trivialization. In logics not properly distinguishable from classical logic, it is valid in general the scheme $A \to (\neg A \to B)$ (where '$A$' and '$B$' are formulas, '$\neg A$' is the negation of '$A$' and '$\to$' is the symbol of implication). Let us admit as premises contradictory formulas $A$ and $\neg A$. As noted above, $A \to (\neg A \to B)$ is a valid scheme. Taking into account the assumptions made for the deduction, the Modus Ponens rule (from $A$ and $A \to B$ we deduce $B$) give us $(\neg A \to B)$. By applying again the Modus Ponens rule to this last formula and premises, we obtain $B$. Thus, from a contradictory formulas we can deduce any statement. This is the trivialization phenomena.

It is worth mentioning that the converse is immediate: in fact, if all formulas of a theory are derivable, in particular, it can be proved a contradiction. Thus in the majority of the logical systems, a contradictory (or inconsistent) theory is trivial and, conversely, if a theory is trivial, it is contradictory. Thus, in most logical systems, the concepts of inconsistent and trivial theories coincide.

In fact, surely, when we think about applications, such property is not at all intuitive and reveals how the classical logic is "fragile" in that scope. Imagine a person reasoning and suppose he reaches a contradiction: it is unusual that in the mind of such a person everything becomes true (unless the person presents a very special insanity).

Still on contradictions, we finish this part, pointing out another implication of the heterodox systems, with the famous paradox by Eubulides (or popularly, the liar paradox), in the following form:

(S) I'm lying

We note that S is true if and only if S is false. It is important to emphasize that the paradox of the liar, for many centuries after its discovery was an aporia.

After the considerations of Tarski based on his correspondence theory of truth, it became a fallacy. Finally, with the advent of paraconsistent logics, there are grounds for regard it as aporia again.

## *1.3 Basic Concepts*

Let T be a theory founded on a logic L, and suppose that the language of T and L contains a symbol for negation—if there are more negation operators, one of them must be chosen for their formal-logical characteristics. T is said to be inconsistent if it has contradictory theorems; i.e., one is the negation of the other; otherwise, T is said to be consistent. T is said to be trivial if all formulas of L–or all closed formulas of L—are theorems of T; otherwise T is said to be non-trivial.

Similarly, the same definition applies to systems of propositions, set of information, etc. (taking into account, of course, the set of its consequences). In classical logic, and in many categories of logic, consistency plays a fundamental role. Indeed, in the most usual logic systems, a theory T is trivial, then T is inconsistent and *vice versa*.

A logic L is called paraconsistent if it can serve as a basis for inconsistent but non-trivial theories.

Its dual concept is the idea of paracomplete logics. A logical system is called *paracomplete* if it can function as the underlying logic of theories in which there are formulas such that these formulas and their negations are simultaneously false. Intuitionistic logic and several systems of many-valued logics are paracomplete in this sense (and the dual of intuitionistic logic, Brouwerian logic, is therefore paraconsistent).

As a consequence, paraconsistent theories do not satisfy the principle of non-contradiction, which can be stated as follows: of two contradictory propositions, i.e., one of which is the negation of the other, one must be false. And, paracomplete theories do not satisfy the principle of the excluded middle, formulated in the following form: of two contradictory propositions, one must be true.

Finally, logics which are simultaneously paraconsistent and paracomplete are called *non-alethic logics.*

## *1.4 Concepts Regarding Actual World*

Almost all concepts regarding actual world encompasses a certain imprecision degree. Let us take colors fulfilling a rainbow. Let us suppose, for instance that the rainbow begins with yellow band and ends with red one. If we consider the statement, “This point of the rainbow is yellow”, surely it is true if the point is on the first band and false if it is on the last band. However, if the point ranges between the extremes, there are points in which the statement is neither true nor false; or it can be both true and false.

This is not because the particular instruments that we use nor it is lack of our vocabulary. The vagueness of the terms and concepts of real science has no subjective nature, arising from causes inherent to the observer, nor objective, in the sense that the reality is indeed imprecise or vague.

Such a condition is imposed on us by our relationship with reality, how we are constituted psycho-physiologically to grasp it, and also by the nature of the universe.

So when we need to describe portions of our reality, unavoidably we face with imprecise description and inconsistency becomes a natural phenomena.

## *1.5 Some Motivations for Paraconsistent Logics*

Problems of various kinds give rise to paraconsistent logics: for instance, the paradoxes of set theory (v.g. Russell’s paradox), the semantic antinomies, some issues originating in dialectics, in Meinong's theory of objects, in the theory of fuzziness, and in the theory of constructivity.

However, since end of last century paraconsistent logics began to be applied in a variety of themes: AI, automation and robotics, information systems, pattern recognition, computability beyond Turing, physics, quantum mechanics, besides in human sciences, like issues in psychoanalysis, etc.

In the scope of this book, we will not consider the question whether our world, is in fact, inconsistent or not. What is of interest for us is what we can call a kind of 'weak ontology': for instance, suppose that a doctor A says that patient P has a certain disease, while another doctor B says that the same patient A has not such disease. In an automated system, we have to make a decision from these conflicting data. On the other hand, systems based on classical logic cannot deal it at least directly. So we need another type of logic to deal with such contradictions without the need for extra-logical devices.

## *1.6 The Systems* $\mathbf{C}_n$ *of da Costa*

Presently it is known an infinitely many paraconsistent logic systems. One of the first important systems in the literature is the (propositional) system $C_{\mathrm{n}}(1 \leq n < \omega)$ of da Costa which we sketch it briefly below.

A hierarchy of paraconsistent propositional calculi $C_{\mathrm{n}}(1 \leq n < \omega)$ was introduced by da Costa [21]. For each n, $1 \leq n < \omega$, we have different calculi symbolized by $C_{\mathrm{n}}$. Such calculi were formulated with the aim of satisfying the following conditions:

(a) The principle of non-contradiction, in the form $\neg(A \wedge \neg A)$ is not valid in general;
(b) From two contradictory propositions, $A$ and $\neg A$, we can not deduce any formula $B$;
(c) The calculus should contain the most important schemes and inference rules of the classic propositional calculus compatible with the conditions (a) and (b) above.

The language of $C_{\mathrm{n}}$ calculi $(1 \leq n < \omega)$ is the same for all of them; let us denote it by L. The primitive symbols of L are:

1. propositional variables: an denumerable set of symbols.
2. logical connectives: $\neg$ (negation) $\wedge$ (conjunction), $\vee$ (disjunction) and $\rightarrow$ (implication).
3. Auxiliary symbols: parentheses.

It is introduced usual syntactic concepts, for example, the idea of formula.

Let $A$ be a formula. Then $A^{\mathrm{o}}$ abbreviates $\neg(A \wedge \neg A)$. $A^{\mathrm{i}}$ abbreviates $A^{\mathrm{o} \ldots \mathrm{o}}$, where the symbol $^{\mathrm{o}}$ appears $i$ times, $i \geq 1$. (thus, $A^1$ is $A^{\mathrm{o}}$). We write $A^{(\mathrm{i})}$ for $(A \wedge A^1 \wedge A^2 \wedge \ldots \wedge A^{\mathrm{i}})$.

The postulates (axioms schemes and inference rules) of $C_n(1 \leq n < \omega)$ are as follows: $A$, $B$ and $C$ denote any formulas.

(1) $A \to (B \to A)$
(2) $(A \to (B \to C)) \to ((A \to B) \to (A \to C))$
(3) $((A \to B) \to A) \to A$
(4) $\frac{A, A \to B}{B}$
(5) $A \wedge B \to A$
(6) $A \wedge B \to B$
(7) $A \to (B \to (A \wedge B))$
(8) $A \to A \vee B$
(9) $B \to A \vee B$
(10) $(A \to C) \to ((B \to C) \to ((A \vee B) \to C))$
(11) $B^{(n)} \to ((A \to B) \to ((A \to \neg B) \to \neg A))$
(12) $(A^{(n)} \wedge B^{(n)}) \to ((A \wedge B)^{(n)} \wedge (A \vee B)^{(n)} \wedge (A \to B)^{(n)})$
(13) $(A \vee \neg A)$
(14) $\neg\neg A \to A$

The postulates of $C_\omega$ are those $C_n$ with the exception of (3), (11) and (12).

In the calculi $C_n(1 \leq n < \omega)$ the formula $A^{(n)}$ expresses the intuitively that the formula $A$ “behaves” classically; therefore motivation for postulates (11) and (12) is clear. In addition, the postulates show us that the remaining connectives $\wedge, \vee, \to$ have all properties of conjunction, disjunction and the classical implication respectively.

We have the following result correlating to classical positive logic: in $C_n(1 \leq n < \omega)$ is true all valid schemes of classical positive propositional logic. In particular, the deduction theorem is valid in $C_n(1 \leq n < \omega)$.

$C_\omega$ contains the positive intuitionistic logic.

We write $\neg^{(n)}A$ for $\neg A \wedge A^{(n)}$.

In $C_n(1 \leq n < \omega)$ we have the following metatheorem: $\vdash \neg A^{(n)} \to (\neg A)^{(n)}$

Also, the connective defined $\neg^{(n)}$ has all the properties of classical negation. As a result, the classical propositional calculus is contained in $C_n(1 \leq n < \omega)$, although the latter is a strict subcalculus of the first. In this way the previous conditions (a), (b) and (c) are satisfied. Thus, the principle of non-contradiction $\neg(A \wedge \neg A)$ is not valid in general.

A semantical consideration can be made for $C_n$ known as valuation theory. $A$ and $B$ are any formulas. $F$ symbolizes the set of formulas of $C_1$ and 2 indicates the set $\{0, 1\}$. An interpretation (or validation) for $C_1$ is a function $\varphi : F \to 2$ such that:

(1) $\varphi(A) = 0 \Rightarrow \varphi(\neg A) = 1;$
(2) $\varphi(\neg\neg A) = 1 \Rightarrow \varphi(A) = 1;$
(3) $\varphi(B^{o}) = \varphi(A \to B) = \varphi(A \to \neg B) = 1 \Rightarrow \varphi(A) = 0;$
(4) $\varphi(A \to B) = 0 \Rightarrow \varphi(A) = 0$ ou $\varphi(B) = 1;$
(5) $\varphi(A \wedge B) = 1 \Rightarrow \varphi(A) = \varphi(B) = 1;$

(6) $\varphi(A \vee B) = 1 \Rightarrow \varphi(A) = 1$ ou $\varphi(B) = 1$;
(7) $\varphi(A^{o}) = \varphi(B^{o}) = 1 \Rightarrow \varphi((A \rightarrow B)^{o}) = \varphi((A \wedge B)^{o}) = \varphi((A \vee B)^{o}) = 1.$

If we denote by $C_0$ the classical propositional calculus, then the hierarchy $C_0, C_1, \ldots, C_{\mathrm{n}}, \ldots, C_\omega$ is such that $C_{\mathrm{i}}$ is strictly stronger than $C_{\mathrm{i+1}}$, for all $\mathrm{i}, 1 \leq \mathrm{i} < \omega$. $\mathrm{C}_\omega$ is the weakest calculus of the hierarchy. Notice that we can extend $C_{\mathrm{n}}$ to the first order logic and higher order logics; also it can built strong set theories based on these first-order logics. With the bi-valued semantic presented it can be proved several basic metatheorems: soundness, strong and weak completeness, and such calculi also are decidable.

Let us turn our attention to the calculus $C_1$. It can be proved that it is a paraconsistent calculus and therefore we can use it to manipulate inconsistent sets of formulas without immediate trivialization (this means that all formulas of language can not be deduced from this inconsistent set of formulas, as has been noted before). In $C_1$ there are inconsistent theories that have models and, as a consequence, they are not trivial. In other words, $C_1$ may serve as underlying logic of paraconsistents theories. However, it should be noted that when we are working with formulas that satisfy the principle of non-contradiction, then $C_1$ reduce to $C_0$.

It should observed that the previous semantics is such that the criterion (T) by Tarski remains valid. Indeed, if $A$ is a formula and $[A]$ its name, we have:

$[A]$ is true (in a validation) if, and only if, $A$.

In a certain sense, the semantics proposed for $C_{\mathrm{n}}$ is a generalization of the usual semantics.

The foregoing observations can be extended easily to other calculi $C_{\mathrm{n}}(1 \leq \mathrm{n} < \omega)$ and to first order calculi $C_{\mathrm{n}}*$ and $C_{\mathrm{n}}^{=}(1 \leq \mathrm{n} < \omega)$. A similar semantics can be constructed to $C_\omega$, as well as for $C_\omega^{*}$ and $C_\omega^{=}$.

Therefore the semantics for $C_{\mathrm{n}}$ extends the classical propositional calculus semantics. In general, ‘paraconsistent’ semantics generalize the classical semantic. So there are “Tarskian alternatives” of Tarski’s truth’s theory, and paraconsistent logic again becomes a starting point for a dialectic of classical doctrine of logicism.

Moreover, the abstract and idealized pure semantic character, traditional or not, shows that the logical systems are theoretical reconstructions of aspects of our boundary; and, bearing in mind all the above observation, it becomes clear the existence of large distance between logical systems and real logical structures.

## *1.7 Other Issues*

Here if a modality which expresses knowledge is added to classical logic, we face another undesirable aspect. In effect, one peculiarity of the deductive structure of the usual modal logic is that an agent knows all logical consequences of their body of knowledge, in particular, all tautologies. This property is known as the question of logical omniscience. This context is not natural in general. Take the example of human reasoning. When we think of agents as humans, surely they are not logically omniscient: in fact, a person can learn a set of facts without knowing all logical consequences of this set of facts. For example, a person can know the chess game rules without knowing whether the white pieces have a strategy to win or not. In practice, the lack of logic omniscience can have several reasons. An obvious example is the computational limitations; for example, an agent may simply not have computing resources to see if the white pieces have a strategy to win at the game of chess.

## 2 Paraconsistent Annotated Logic

Annotated logics are class of 2-sorted logics. They were introduced in logic programming by Subrahmanian [39] and subsequently by Blair and Subrahmanian [14]. Simultaneously, some other applications were made: declarative semantics for inheritance networks [28], object-oriented databases [27], among other issues.

In view of the applicability of annotated logics to these differing formalisms in computer science, it has become essential to study these logics more carefully, mainly from the foundational point of view. In [22] the authors studied the propositional level of annotated systems. In sequence, Abe [1] studied the first order predicate calculi $Q\tau$ in details, obtaining completeness and soundness theorems for the case when the associated lattice $\tau$ is finite. Also this author has established some main theorems concerning the theory of models (Łos theorem, Chang theorem, interpolation theorem, Beth definability theorem, among others). Also an annotated set theory was proposed [1, 18] 'inside' usual ZF set theory which encompasses the fuzzy set theory in totum.

In general, annotated logics are a kind of paraconsistent, paracomplete, and non-alethic logic.

### *2.1 The Annotated Logics $Q\tau$*

$Q\tau$ is a family of two-sorted first-order logics, called annotated two-sorted first-order predicate calculi. They are defined as follows: throughout this paragraph, $\tau = <|\tau|, \leq, \sim>$ will be some arbitrary, but finite fixed lattice of truth values with the operator $\sim : |\tau| \to |\tau|$ which constitutes the "meaning" of the negation of $Q\tau$. The least element of $\tau$ is denoted by $\perp$, while its greatest element by $\top$; $\vee$ and $\wedge$ denote, respectively, the least upper bound and the greatest lower bound operators (of $\tau$) .

The primitive symbols of the language $L$ of $Q\tau$ are the following:

1. Individual variables: a denumerable infinite set of variable symbols.
2. Logical connectives: $\neg$, (negation), $\wedge$ (conjunction), $\vee$ (disjunction), and $\to$ (conditional).

3. For each *n,* zero or more *n*-ary function symbols (*n* is a natural number).
4. For each $n \neq 0$, zero or more *n*-ary predicate symbols.
5. Quantifiers: $\forall$ (for all) and $\exists$ (there exists).
6. The equality symbol: = .
7. Annotated constants: each member of $\tau$ is called an annotational constant.
8. Auxiliary symbols: parentheses and commas.

For each *n,* the number of *n*-ary function symbols may be zero or non-zero, finite or infinite. A 0-ary function symbol is called a *constant.* Also, for each *n*, the number of *n*-ary predicate symbol may be finite or infinite.

In the sequel, we suppose that $Q\tau$ possesses at least one predicate symbol.

We define the notion of *term* as usual. Given a predicate symbol *p* of arity *n,* an annotational constant λ and *n* terms $t_1, \ldots, t_n$, an *annotated atom* is an expression of the form $p_\lambda t_1 \ldots t_n$. In addition, if $t_1$ and $t_2$ are terms whatsoever, $t_1 = t_2$ is an *atomic formula.* We introduce the general concept of *formula* in the standard way. Among several intuitive readings, an annotated atom $p_\lambda t_1 \ldots t_n$ can be read is *it is believed that*

$p_\lambda t_1 \ldots t_n$*'s truth value is at least* λ.

Syntactical notions, as well as terminology, notations, etc. are those of current literature with obvious adaptations. We will employ them without extensive comments.

**Definition 2.1** Let *A* and *B* formulas of *L*. We put

$(A \leftrightarrow B) =_{\text{Def.}} ((A \to B) \wedge (B \to A))$ and $(\neg{*}A) =_{\text{Def.}} (A \to ((A \to A) \wedge \neg (A \to A))$.

The symbol '↔' is called the *biconditional* and '¬*' is called *strong negation.*

Let *A* be a formula. $\neg^0 A$ indicates *A,* $\neg^1 A$ indicates $\neg A$, and $\neg^n A$ indicates $(\neg(\neg^{n-1}A))$, $(n \geq 1)$. Also, if $\mu \in \tau$, $\sim{}^0\mu$ indicates μ, $\sim{}^1\mu$ indicates $\sim \mu$, and $\sim{}^n\mu$ indicates $(\sim (\sim^{n-1}\mu))$, $(n \geq 1)$.

**Definition 2.2** Let $p_\lambda t_1 \ldots t_n$ be an annotated atom. A formula of the form $\neg^k p_\lambda t_1 \ldots t_n$ $(k \geq 0)$ is called a *hyper-literal.* A formula other than hyper-literal is called a *complex formula.*

We now introduce the concept of structure for *L*.

**Definition 2.3** A *structure S* for *L* consists of the following objects:

1. A non-empty set $|S|$, called the *universe* of *S.* The elements of $|S|$ are called *individuals* of *S.*
2. For each *n*-ary function symbol *f* of *L* an *n*-ary function $f_S$: $|S|^n \to |S|$. (In particular, for each constant *e* of *L*, $e_S$ is an individual of *A.*)
3. For each *n*-ary predicate symbol *p* of *L* an *n*-ary function $p_S$: $|S|^n \to |\tau|$.

Let $A$ be a structure for $L$. The *diagram language* $L_S$ is obtained as usual.

If $a$ is a free-variable term, we define the individual $S(a)$ of $S$. We use $i$ and $j$ as syntactical variables which vary over names.

We define a truth value $S(A)$ for each closed formula $A$ in $L_S$.

1. If $A$ is $a = b$
   $S(A) = 1$ iff $S(a) = S(b)$; otherwise $S(A) = 0$.
2. If $A$ is $p_\lambda t_1 \ldots t_n$
   $S(A) = 1$ iff $p_S(S(t_1) \ldots S(t_n) \geq \lambda$
   $S(A) = 0$ iff it is not the case that $p_S(S(t_1) \ldots S(t_n) \geq \lambda$
3. If $A$ is $B \wedge C$, or $B \vee C$, or $B \rightarrow C$, we let
   $S(B \wedge C) = 1$ iff $S(B) = S(C) = 1$.
   $S(B \vee C) = 1$ iff $S(B) = 1$ or $S(C) = 1$.
   $S(B \rightarrow C) = 0$ iff $S(B) = 1$ and $S(C) = 0$.
   If $A$ is $\neg^k p_\lambda t_1 \ldots t_n$ $(k \geq 1)$, then $S(A) = S(\neg^{k-1} p_{\sim\lambda} t_1 \ldots t_n)$.
4. If $A$ is a complex formula, then, $S(\neg A) = 1$—$S(A)$.
5. If $A$ is $\exists xB$, then $S(A) = 1$ iff $S(B_x[i]) = 1$ for some $i$ in $L_S$.
6. If $A$ is $\forall xB$, then $S(A) = 1$ iff $S(B_x[i]) = 1$ for all $i$ in $L_S$.

A formula $A$ of $L$ is said to be *valid in S* if $S(A') = 1$ for every $S$-instance $A'$ of $A$. A formula $A$ is called *logically valid* if it is valid in every structure for $L$. In this case, we symbolize it by $\vDash A$. If $\Gamma$ is a set of formulas of $L$ we say that $A$ is a *semantic consequence* of $\Gamma$ if for any structure $S$ in what $S(B) = 1$ for all $B \in \Gamma$, it is the case that $S(A) = 1$. We symbolize this fact by $\Gamma \vDash A$. Note that when $\Gamma = \varnothing$, $\Gamma \vDash A$ iff $\vDash A$.

**Lemma 2.1** *We have:*

1. $\vDash p_\perp t_1 \ldots t_n$
2. $\vDash \neg^k p_\lambda t_1 \ldots t_n \leftrightarrow \neg^{k-1} p_{\sim\lambda} t_1 \ldots t_n)\ (k \geq 1)$
3. $\vDash p_{\mu 1} t_1 \ldots t_n \rightarrow p_{\mu 2} t_1 \ldots t_n, 1 \geq \mu$
4. $\vDash p_{\mu 1} t_1 \ldots t_n \wedge p_{\mu 2} t_1 \ldots t_n \wedge \ldots \wedge p_{\mu m} t_1 \ldots t_n \rightarrow p_{\mu \bigvee_{i=1}^{m}} t_1 \ldots t_n$

*Now, we shall describe an axiomatic system which we call $A\tau$ whose underlying language is L: A, B, C are any formulas whatsoever, F, G are complex formulas, and $p_\lambda t_1$ ... $t_n$ an annotated atom. $A\tau$ consists of the following postulates (axiom schemes and primitive rules of inference), with the usual restrictions:*

$(\rightarrow_1)$ $A \rightarrow (B \rightarrow A)$
$(\rightarrow_2)$ $(A \rightarrow (B \rightarrow C) \rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow C))$
$(\rightarrow_3)$ $((A \rightarrow B) \rightarrow A \rightarrow A)$
$(\rightarrow_4)$ $\frac{A, A \rightarrow B}{B}$ (Modus Ponens)
$(\wedge_1)$ $A \wedge B \rightarrow A$

$(\wedge_2)$ $A \wedge B \rightarrow B$
$(\wedge_3)$ $A \rightarrow (B \rightarrow (A \wedge B))$
$(\vee_1)$ $A \rightarrow A \vee B$
$(\vee_2)$ $B \rightarrow A \vee B$
$(\vee_3)$ $(A \rightarrow C) \rightarrow ((B \rightarrow C) \rightarrow ((A \vee B) \rightarrow C))$
$(\neg_1)$ $(F \rightarrow G) \rightarrow ((F \rightarrow \neg G) \rightarrow \neg F)$
$(\neg_2)$ $F \rightarrow (\neg F \rightarrow A)$
$(\neg_3)$ $F \vee \neg F$
$(\exists_1)$ $A(t) \rightarrow \exists x A(x)$
$(\exists_2)$ $\frac{A(x) \rightarrow B}{\exists x A(x) \rightarrow B}$
$(\forall_1)$ $\forall x A(x) \rightarrow A(t)$
$(\forall_2)$ $\frac{B \rightarrow A(x)}{B \rightarrow \forall x A(x)}$
$(\tau_1)$ $p_{\perp} t_1 \ldots t_n$
$(\tau_2)$ $\neg^k p_{\lambda} t_1 \ldots t_n \rightarrow \neg^{k-1} p_{\sim\lambda} t_1 \ldots t_n, k \geq 1$
$(\tau_3)$ $p_{\lambda} t_1 \ldots t_n \rightarrow p_{\mu} t_1 \ldots t_n, \lambda \geq \mu$
$(\tau_4)$ $p_{\lambda 1} t_1 \ldots t_n \wedge p_{\lambda 2} t_1 \ldots t_n \wedge \ldots \wedge p_{\lambda m} t_1 \ldots t_n \rightarrow p_{\lambda} t_1 \ldots t_n$, where $\lambda = \bigvee_{i=1}^{m} \lambda_i$
$(=_1)$ $x = x$
$(=_2)$ $x = y \rightarrow (A[x] \leftrightarrow A[y])$

**Theorem 2.12** *In $Q\tau$, the operator $\neg *$ has all properties of the classical negation. For instance, we have:*

1. $\vdash A \vee \neg * A$
2. $\vdash \neg * (A \wedge \neg * A)$
3. $\vdash (A \rightarrow B) \rightarrow ((A \rightarrow \neg * B) \rightarrow \neg * A)$
4. $\vdash A \rightarrow \neg * \neg * A$
5. $\vdash \neg * A \rightarrow (A \rightarrow B)$
6. $\vdash (A \rightarrow \neg * A) \rightarrow B$

*among others, where A, B are any formulas whatsoever.*

**Corollary 2.12.1** *In $Q\tau$ the connectives $\neg *$, $\wedge$, $\vee$, and $\rightarrow$ together with the quantifiers $\forall$ and $\exists$ have all properties of the classical negation, conjunction, disjunction, conditional and the universal and existential quantifiers, respectively. If A, B, C are formulas whatsoever, we have, for instance:*

1. $(A \wedge B) \leftrightarrow \neg * (\neg * A \vee \neg * B)$
2. $\neg * \forall A \leftrightarrow \exists x \neg * A$
3. $\exists x B \vee C \leftrightarrow \exists x (B \vee C)$
4. $B \vee \exists x C \leftrightarrow \exists x (B \vee C)$

**Theorem 2.13** *If A is a complex formula, then $\vdash \neg A \leftrightarrow \neg * A$*

**Definition 2.10** We say that a *structure* $S$ is *non-trivial* if there is a closed annotated atom $p_\lambda t_1 \ldots t_n$ such that $S(p_\lambda t_1 \ldots t_n) = 0$.

Hence a structure $S$ is non-trivial iff there is some closed annotated atom that is not valid in $S$.

**Definition 2.11** We say that a *structure A* is *inconsistent* if there is a closed annotated atom $p_\lambda t_1 \ldots t_n$ such that

$$S(p_l t_1 \ldots t_n) = 1 = S(p_l t_1 \ldots t_n).$$

So, a structure $S$ is inconsistent iff there is some closed annotated atom such that it and its negation are both valid in $S$.

**Definition 2.12** A *structure S* is called *paraconsistent* if $S$ is both inconsistent and non-trivial. The *system* $Q\tau$ is said to be *paraconsistent* if there is a structure $S$ for $Q\tau$ such that $S$ is paraconsistent.

**Definition 2.13** A *structure S* is called *paracomplete* if there is a closed annotated atom $p_\lambda t_1 \ldots t_n$ such that $S(p_\lambda t_1 \ldots t_n) = 0 = S(p_\lambda t_1 \ldots t_n)$.

The *system* $Q\tau$ is said to be *paracomplete* if there is a structure $S$ such that $S$ is paracomplete.

**Theorem 2.14** *$Q\tau$ is paraconsistent iff $\#|\tau| \geq 2$.*

*Proof* Suppose that $\#|\tau| \geq 2$. There is at least one predicate symbol $p$. Let $|\tau|$ be a non-empty set which $\#|\tau| \geq 2$. Let us define $p_S{:}|S|^n \rightarrow |\tau|$ setting $p_S(a_1, \ldots, a_n) = \perp$ and $p_S(b_1, \ldots, b_n) = \top$ where $(a_1, \ldots, a_n) \neq (b_1, \ldots, b_n)$.

Then, $S(p_\perp i_1 \ldots i_n) = 1$, where $i_j$ the name of $b_j$, $j = 1, \ldots, n$, and $S(\neg p_\perp i_1 \ldots i_n) = 1$. Likewise, $S(p_\top j_1 \ldots j_n) = 0$, where $j_i$ is the name of $a_i$, $i = 1, \ldots, n$. So, $Q\tau$ is paraconsistent. The converse is immediate.

**Theorem 2.15** *For all $\tau$ there are systems $Q\tau$ that are paracomplete; and also systems that are not paracomplete. If $Q\tau$ is paracomplete, then $\#|\tau| \geq 2$.*

*Proof* Similar to the proof of the preceding theorem.

**Definition 2.15** A *structure S* is called *non-alethic* if $S$ is both paraconsistent and paracomplete. The *system* $Q\tau$ is said to be *non-alethic* if there is a structure $S$ for $Q\tau$ such that $S$ is non-alethic.

**Theorem 2.16** *For all $\#|\tau| \geq 2$ there are systems $Q\tau$ that are non-alethic; and also systems that are not non-alethic. If $Q\tau$ is non-alethic, then $\#|\tau| \geq 2$.*

*Given a structure S, we can define the theory Th(S) associated with S to be the set Th(S) = $C_n(\Gamma)$, where $\Gamma$ is the set of all annotated atoms which are valid in S $C_n(\Gamma)$ indicates the set of all semantic consequences of elements of $\Gamma$.*

**Theorem 2.17** *Given a structure S for $Q\tau$, we have:*

*1. Th(S) is a paraconsistent theory iff S is a paraconsistent structure. 2. Th(A) is a paracomplete theory iff S is a paracomplete structure. 3. Th(S) is a non-alethic theory iff S is a non-alethic structure.*

*In view of the preceding theorem, Qτ is, in general, a paraconsistent, paracomplete and non-alethic logic.*

*We give a Henkin-type proof of the completeness theorem for the logics Qτ.*

**Definition 2.16** A theory $T$ based on $Q\tau$ is said to be *complete* if for each closed formula $A$ we have $\vdash_T A$ or $\vdash_T \neg^* A$.

**Lemma 2.2** *Let* $\lambda_0 = \vee\{\lambda \in |\tau| : \vdash_T p_\lambda t_1 \ldots t_n\}$. *Then* $\vdash_T p_{\lambda 0} t_1 \ldots t_n$.

*Now let T be a non-trivial theory containing at least one constant. We shall define a structure S that we call the canonical structure for T. If a and b are variable-free terms of T, then we define aRb to mean* $\vdash_T a = b$. *It is easy to check that R is an equivalence relation. We let |S| be the quotient set F/R, where F indicates the set of all formulas of L. The equivalence class determined by a is designed by* $a^o$. *We complete the definition of S by setting*

$$f_S(a_1^0, \ldots, a_m^0) = (f_S(a_1, \ldots, a_m))^0 \text{and}$$
$$p_S(a_1^0, \ldots, a_n^0) = \vee\{\lambda \in |t| : \vdash_\tau p_\lambda a_1 \ldots a_n\}$$

It is straightforward to check the formal correctness of the above definitions.

**Theorem 2.18** *If* $p_\lambda a_1 \ldots a_n$ *is a variable-free annotated atom, then*

$$S(p_\lambda a_1 \ldots a_n) = 1 \quad \text{iff} \vdash_\tau p_\lambda a_1 \ldots a_n.$$

*Proof* Let us suppose that $S(p_\lambda a_1 \ldots a_n) = 1$. Then $p_S(a_1^0, \ldots, a_n^0) \geq \lambda$.

But $p_S(a_1^0, \ldots, a_n^0) = \vee\{\lambda \in |\tau| : \vdash_T p_\lambda a_1 \ldots a_n\}$; so, $\vdash_T p_{\lambda 0} a_1 \ldots a_n$ by the preceding lemma. As $\lambda_0 \geq \lambda$, it follows that $\vdash_T p_\lambda a_1 \ldots a_n$ by axiom ($\tau_3$).

Conversely, let us suppose that $\vdash_T p_\lambda a_1 \ldots a_n$. Then $\vee\{\mu \in |\tau| : \vdash_T p_\mu a_1 \ldots a_n\}$. Let $\lambda_0 = \vee\{\mu \in |\tau| : \vdash_T p_\mu a_1 \ldots a_n\}$;). Then it follows that $\lambda_0 \geq \lambda$. But $\lambda_0 = p_S(a_1^0, \ldots, a_n^0)$, and so $p_S(a_1^0, \ldots, a_n^0) \geq \lambda$; hence $S(p_\lambda a_1 \ldots a_n) = 1$.

A formula $A$ is called *variable-free* if $A$ does not contain free variables.

**Theorem 2.19** *Let a = b be a variable-free formula. Then S(a = b) = 1 iff* $\vdash_T a = b$.

*We define the Henkin theory as in the classical case. Now, suppose that T is a Henkin theory and S the canonical structure for T.*

**Theorem 2.20** *Let* $\neg^k p_\lambda a_1 \ldots a_n$ *be a variable-free hyper-literal. Then*

$$S(\neg^k p_\lambda a_1 \ldots a_n) = 1 \quad \text{iff} \vdash_\tau \neg^k p_\lambda a_1 \ldots a_n$$

*Proof* By induction on $k$ taking into account the axiom ($\tau_2$) and theorem 2.7.

**Theorem 2.21** *Let T be a complete Henkin theory, S the canonical structure for T, and A a closed formula. Then, S(A) = 1 iff A.*

*Proof* By induction on the length of $A$.

**Corollary 2.21.1** *Under the conditions of the theorem the canonical structure for T is a model of T.*

*We construct Henkin theories as in the classical case.*

**Theorem 2.22** *(Lindenbaum's theorem). If T is a non-trivial theory, then T has a complete simple extension.*

**Theorem 2.23** *(Completeness theorem). A theory (consistent or not) is non-trivial iff it has a model.*

**Theorem 2.24** Let $\Gamma$ be a set of formulas. Then $\Gamma \vdash A$ iff $\Gamma \vDash A$.

**Theorem 2.25** *A formula A of a theory T is a theorem of T iff it is valid in T.*

Hence usual metatheorems of soundness and completeness are valid for the logics $Q\tau$, as well as the usual theory of models can be adapted for these logics. When the associated lattice $\tau$ of the logics $Q\tau$ is infinite, due the axiom scheme ($\tau_4$), the logic $Q\tau$ is an infinitary logic.

## *2.2 The Paraconsistent Annotated Evidential Logic Eτ*

One logic of particular importance among the family of logics $Q\tau$ is the paraconsistent annotated evidential logic E$\tau$. The lattice $\tau$ is composed by the set $[0, 1] \times [0, 1]$ together with the order relation defined as: $(\mu_1, \lambda_1) \leq (\mu_2, \lambda_2) \Leftrightarrow \mu_1 \leq \mu_2$ and $\lambda_2 \leq \lambda_1$ where [0, 1] is the real unitary interval with the real ordinary order relation.

The atomic formulas of the logic E$\tau$ are of the type $p_{(\mu,\, \lambda)}$, where $(\mu, \lambda) \in [0, 1]^2$ ($p$ denotes a propositional variable). $p_{(\mu,\, \lambda)}$ can be intuitively read: "It is assumed that $p$'s favorable evidence is $\mu$ and contrary evidence is $\lambda$." Thus:

- $p_{(1.0,\, 0.0)}$ can be read intuitively as a true proposition.
- $p_{(0.0,\, 1.0)}$ can be read intuitively as a false proposition.
- $p_{(1.0,\, 1.0)}$ can be read intuitively as an inconsistent proposition.
- $p_{(0.0,\, 0.0)}$ can be read intuitively as a paracomplete proposition.
- $p_{(0.5,\, 0.5)}$ can be read intuitively as an indefinite proposition.

We introduce the following concepts (all considerations are taken with $0 \leq \mu, \lambda \leq 1$):

- Uncertainty degree: $G_{un}(\mu, \lambda) = \mu + \lambda - 1$
- Certainty degree: $G_{ce}(\mu, \lambda) = \mu - \lambda$

Intuitively, $G_{un}(\mu, \lambda)$ show us how close (or far) the annotation constant $(\mu, \lambda)$ is from Inconsistent or Paracomplete state. Similarly, $G_{ce}(\mu, \lambda)$ show us how close (or far) the annotation constant $(\mu, \lambda)$ is from True or False state. In this way we can manipulate the information given by the annotation constant (μ, λ). Note that such degrees are not metrical distance.

With the uncertainty and certainty degrees we can get the following 12 output states (Table 1.1): *extreme states*, and *non-extreme states*.

Some additional control values are:

- $V_{scct}$ = maximum value of uncertainty control = $Ft_{un}$
- $V_{scc}$ = maximum value of certainty control = $Ft_{ce}$
- $V_{icct}$ = minimum value of uncertainty control = -$Ft_{un}$
- $V_{icc}$ = minimum value of certainty control = $-Ft_{ce}$

Such values are determined by the knowledge engineer, depending on each application, finding the appropriate control values for each of them.

**Table 1.1** Extreme and Non-extreme states

| Extreme states | Symbol | Non-extreme states | Symbol |
|---|---|---|---|
| True | V | Quasi-true tending to Inconsistent | QV → T |
| False | F | Quasi-true tending to Paracomplete | QV → ⊥ |
| Inconsistent | T | Quasi-false tending to Inconsistent | QF → T |
| Paracomplete | ⊥ | Quasi-false tending to Paracomplete | Qf → ⊥ |
| | | Quasi-inconsistent tending to True | QT → V |
| | | Quasi-inconsistent tending to False | QT → F |
| | | Quasi-paracomplete tending to True | Q⊥ → V |
| | | Quasi-paracomplete tending to False | Q⊥ → F |

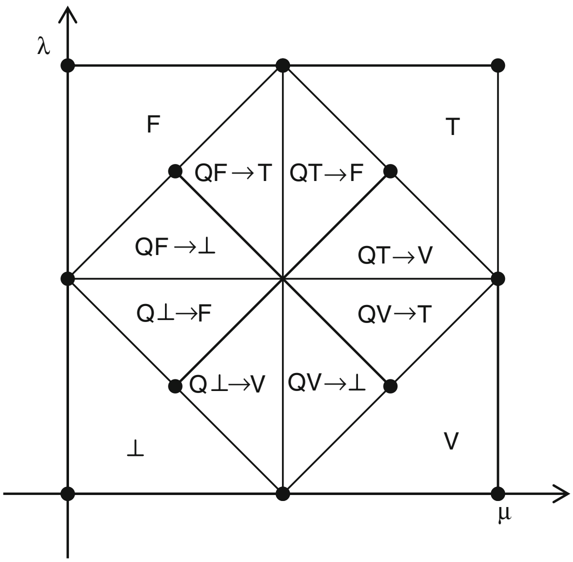


**Fig. 1.1** Lattice of extreme and non-extreme states

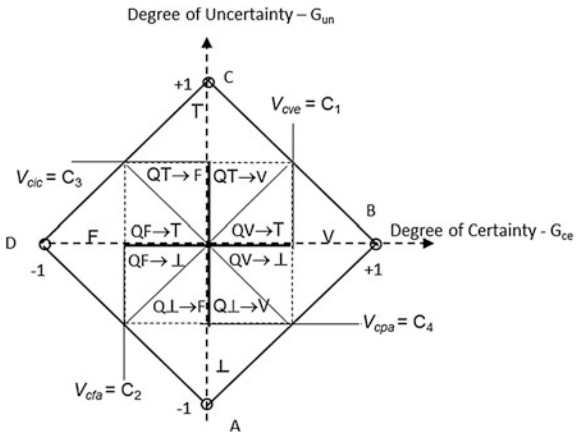


**Fig. 1.2** Certainty and uncertainty degrees and decision states

All states are represented in the next Figs. 1.1 and 1.2 includes their relationship with certainty and uncertainty degrees.

Given the inputs μ-favorable evidence and λ—contrary evidence, there is the Para-analyzer algorithm (below) which figure out a convenient output [4, 25].

## 3 Algorithm "Para-Analyzer"

```
*/ Definitions of the values */
  Max_vcc = C_1 */ maximum value of certainty Control*/
  Max_vctc = C_2 */ maximum value of uncertainty Control
*/
  Min_vcc = C_3 */ minimum value of certainty Control */
  Min_vctc = C_4 */ minimum value of uncertainty Con-
trol*/
*/ Input Variables */
  μ
  λ
*/ Output Variables */
  digital output = S1
  Analog output = S2a
  Analog output = S2b
```

```
* / Mathematical expressions * /
  begin:
  0≤ μ ≤ 1 e 0≤ λ≤ 1
  Gct = μ + λ - 1
  Gc = μ - λ
* / determination of the extreme states * /
  if Gce ≥ C1 then S1 = V
  if Gce ≥ C2 then S1 = F
 if Gun ≥ C3 then S1 = T
 if Gun ≤ C4 then S1 = ⊥
*/ determination of the non-extreme states * /
 for 0 ≤ Gce < C1 and 0 ≤ Gun < C3
 if Gce ≥ Gun then S1 = QV→T
   else S1 = QT→V
 for 0 ≤ Gce < C1 and C4 < Gun ≤ 0
 if Gce ≥ | Gun | then S1 = QV→⊥
 else S1 = Q⊥→V
 for C2 < Gce ≤ 0 and C4 < Gun ≤ 0
 if |Gce | ≥ | Gun | then S1 = QF→⊥
 else S1 = Q⊥→F
 for C2 < Gce ≤ 0 and 0 ≤ Gun < C3
 If |Gce| ≥ Gun then S1 = QF→T
 else S1 = QT→F
 Gun = S2a
 Gce = S2
*/ END */
```

## 4 Applications

Nowadays it is known innumerous applications of paraconsistent logics in various fields of human knowledge. Here we restrict to a fragment of them, mainly having as basis the annotated logics. A number of them were initiated by Abe around 1993 and together with some students have implemented an annotated logic program dubbed Paralog (Paraconsistent Logic Programming) [12] independently of Subrahmanian [39]. These ideas were applied in a construction of a specification and prototype of an annotated paraconsistent logic-based architecture, which integrates various computing systems—planners, databases, vision systems, etc.—of a manufacture cell [36] and knowledge representation by Frames, allowing representing inconsistencies and exceptions [13].

Also, in [23, 25] it was introduced digital circuits (logical gates Complement, And, Or) inspired in a class of paraconsistent annotated logics $P\tau$. These circuits allow "inconsistent" signals in a nontrivial manner in their structure. Such circuits consist of six states; due the existence of literal operators to each of them, the underlying logic is functionally complete.

There are various intelligent systems including nonmonotonic reasoning in the field of AI. Each system has different semantics. More than two nonmonotonic reasoning maybe required in complex intelligent systems. It is more desirable to have a common semantics for such nonmonotonic reasoning. It was proposed a common semantics for the nonmonotonic reasoning by annotated logics and annotated logic programs [32–34].

Also, annotated systems encompass deontic notions in this fashion. Such ideas were implemented successfully in safety control systems: railway signals, traffic intelligent signals, pipeline cleaning system, and many other applications [34].

Annotated systems were extended to involve usual modalities [1] and this framework was utilized to obtain annotated knowledge logics [5], temporal annotated logics, deontic logics and versions of Jaśkowski systems. Also multimodal systems were obtained as a formal system to serve as underling logic for distributed systems in order to manage inconsistencies and/or paracompleteness [5].

Also a particular annotated version, namely, paraconsistent annotated evidential logic $E\tau$ was employed in decision-making theory in many questions in production engineering [15] with the aid of Para-analyzer algorithm. An expert system can be considered with the novelty that the database is constituted by date from experts expressing their favorable evidence and unfavorable evidence regarding to the problem. Some positive aspects: the formal language for the experts to express their opinions is richer than ordinary ones. The study employing the logic $E\tau$ was compared with statistics [15–17] and with fuzzy set theory [15] (see also [8]).

Logical circuits and programs can be designed based on the Para-analyzer. A hardware or a software built by using the Para-analyzer, in order to treat logical signals according the structure of the logic $E\tau$, is a logical controller that we call Para-control. It was built an experimental robot based on paraconsistent annotated evidential logic $E\tau$ which basically it has two ultra-sonic sensors (one of them capturing a favorable evidence and another, the contrary evidence regarding to the existing obstacles ahead) and such signals are treated according to Paracontrol. The first robot built on the basis of Pacontrol was dubbed Emmy [24]. Also it was built a robot based on a software using Paralog before mentioned, which was dubbed Sofya [36]. Then several other prototypes were made with many improvements [40].

Also a suitable combination of such logical analyzer, it allowed to built a new artificial neural network dubbed paraconsistent artificial neural network [24] and innumerous applications were succeeded: in the aid of Alzheimer diagnosis [29, 30], Cephalometric analysis [31], speech disfluence [6], numerical recognition [38], sample in Statistics, Robotics [40], decision-making theory, many disease auxiliary diagnosis [2, 11], etc.

Also annotated logics encompass fuzzy set theory and we have presented even an axiomatization of versions of Fuzzy theory [8].

## 5 Conclusions

The paraconsistent annotated logic, still very young, discovered in the eventide of the last century, is one of the great achievements in non-classical logics. Its composition as two-sorted logic, in which the set of one of the variables has a mathematical structure produced useful results regarding to computability and electronic implementations. Its main feature is the capability of handling concepts such as uncertainty, inconsistency, and paracompleteness.

We believe that the annotated logics has very wide horizons, with enormous potential for application and also as a foundation for elucidating the common denominator of many non-classical logics.

The appearance of the paraconsistent logics can be considered one of the most original and imaginative logical systems in the beginning of last century. It constitutes in a paradigm of the human thinking. As rationality and logicality were identical until the advent of non-classical systems, nowadays it can be considered in this way yet? Are there really alternatives to the classical logic? In consequence are there distinct rationalities? All these questions occupy logicians, philosophers, and scientists in general.

## Notes

1. According to da Costa, in a personal letter to the author, Feb. 2015: "The term "Curry Algebras", when I've employed it for the first time had no relationship directly with paraconsistent logics. In formulating the theory, based on Curry's work around l965, they were conceived as "algebras" containing non-monotonic (i.e., non-compatible) operators regarding to their basic equivalence relations. When I've made the algebraization of the paraconsistent $C_n$ systems, I've noted that the negations of my systems were not monotonic regarding to equivalence relations of these algebras; therefore, they were Curry algebras. (Interestingly enough, I personally spoke with Curry, asking him permission to use his name for these algebras and the Curry's response was: "You can do it, if the theory is worthy of being named". Note that what I call Curry algebra is not algebra in the usual sense, as they have always equality as the basic relation. Curry was the first to replace equality by an equivalence relation.) The first presentation of the concept of Curry algebra appeared in C.R. Acad. Sc. Paris[1], as you know. Before it had appeared in my monograph "Algebras de Curry"[2], which you also know and it contains all the literature at the time."
   [1]N.C.A. da Costa, Opérations non monotones dans les treillis. Comptes Rendus de l'Académie des Sciences. Série 1, Mathématique, Paris, v. 263, p. 429–432, 1966.

[1]N.C.A. da Costa, Filtres et idéaux d`une algèbre $C_n$, Comptes Rendus de l'Académie des Sciences. Série 1, Mathématique, Paris, v. 264, p. 549–552, 1967.
[2]N.C.A. da Costa, Álgebras de Curry, University of São Paulo, 1967.
So, "Curry algebras" is in homage to the American logician Haskell Brooks Curry (1900–1982).

2. According to da Costa, in a personal letter to the author, Feb. 2015: "The valuation theory appeared by first time, as I conceive it, in my seminars in Campinas State University in the 60s or 70s. It applies to any usual logic, such as intuitionistic and positive classical logic. Thus, it can be applied even in paraconsistent logic. I did so, in the 70s, regarding to paraconsistent logics $C_n$, which led me, in particular, to the decision processes for these calculi and several others, such as intuitionistic. Such result was due not only to me but also to the Campinas' group. In fact, I showed the soundness and completeness (Gödel) theorems for any logic system, via valuation theory (not only propositional calculi, but also quantification calculi and even for set theories, …)."
3. The name 'Emmy' for the first paraconsistent autonomous robot built with hardware-based on paraconsistent annotated logic was suggested by Newton C. A. da Costa and communicated personally to Jair M. Abe in 1999 at University of São Paulo.
4. The name 'Sofya' for the first paraconsistent autonomous robot built with the software based on paraconsistent annotated logic was also suggested by Newton C.A. da Costa and communicated personally to Jair M. Abe in 1999 at University of São Paulo.
5. The name 'Emmy' is a tribute to the mathematician Amalie Emmy Noether (1882–1935). The name 'Sofya' is in honor to the mathematician Sofya Kovalevskaya Vasil'evna (= Kowalewskaja) (1850–1891). Among other reasons for the choice of names, da Costa commented that as such robots have distinctive feature to deal with paraconsistency in a 'natural way' should receive female names.

## References

1. Abe, J.M.: Fundamentos da Lógica Anotada (Foundations of Annotated Logics) (in Portuguese), Ph.D. Thesis, University of São Paulo, São Paulo, Brazil (1992)
2. Abe, J.M., Akama, S., Nakamatsu, K.: Introduction to annotated logics—foundations of reasoning about incomplete and inconsistent information—Springer, series "Intelligent Systems Reference Library", to appear (2015)
3. Abe, J.M., Da Silva Filho, J.I.,: Inconsistency and electronic circuits. In: Alpaydin, E. (ed.) Proceedings of the international ICSC Symposium on Engineering of Intelligent Systems (EIS'98), vol. 3, Artificial Intelligence, ICSC Academic Press International Computer Science Conventions Canada/Switzerland, pp. 191–197 (1998)
4. Abe, J.M., da Silva Filho, J.I.: Manipulating conflicts and uncertainties in robotics. Multiple-Valued Logic Soft Comput **9**, 147–169 (2003)

5. Abe, J.M., Nakamatsu, K.: Multi-agent systems and paraconsistent knowledge. In: Nguyen, N. T., Lakhmi, C.J. (eds.) Knowledge processing and decision making in agent-based systems, Book series Studies in Computational Intelligence, vol. 167, VIII, 400 p. 92 illus, pp. 101–121, Springer (2009)
6. Abe, J.M., Prado, J.C.A., Nakamatsu, K.: Paraconsistent artificial neural network: applicability in computer analysis of speech productions, Lecture Notes in Computer Science 4252, pp. 844–850, Springer (2006)
7. Abe, J.M.: Some aspects of paraconsistent systems and applications. Logique et Anal. **15**, 83–96 (1997)
8. Akama, S., Abe, J.M.: Fuzzy annotated logics. In: Proceedings of 8th International Conference on Information Processing and Management of Uncertainty in Knowledge Based Systems, IPMU'2000, Organized by: Universidad Politécnica de Madrid, Spain, vol. 1, pp. 504–508 (2000)
9. Akama, S., Abe, J.M., Nakamatsu, K.: Contingent information: a four-valued approach. In: Advances in Intelligent Systems and Computing, vol. 326, ISSN 2194-5357, Springer International Publishing, pp 209–217 ( 2015) doi10.1007/978-3-319-11680-8_17
10. Akama, S., Nakamatsu, K., Abe, J.M.: Common-sense reasoning in constructive discursive logic. In: ICNIT2012, International Conference on Next Generation Information Technology, Seoul, Korea, April 24–26, *Sponsor/Organizer*: IEEE, IEEE Korea Council, ETRI, KISTI, Asia ITNT Association, AICIT, ISBN: 9781467308939, pp. 596–601 (2012)
11. Amaral, F.V.: Classificador Paraconsistente de atributos de imagens de mamográficas aplicado no processo de diagnóstico do câncer de mama assistido por computador, Ph.D. thesis, in Portuguese, Paulista University (2013)
12. Ávila, B.C., Abe, J.M., Prado, J.P.A.: ParaLog-e: A Paraconsistent Evidential Logic Programming Language. In: XVII International Conference of the Chilean Computer Science Society, IEEE Computer Society Press, pp 2–8, Valparaíso, Chile (1997)
13. Ávila, B.C.: Uma Abordagem Paraconsistente Baseada em Lógica Evidencial para Tratar Exceções em Sistemas de Frames com Múltipla Herança, Ph.D. thesis, University of São Paulo, São Paulo (1996)
14. Blair, H.A, Subrahmanian, V.S.: Paraconsistent logic programming. In: Proceedings of 7th Conference on Foundations of Software Technology and Theoretical Computer Science, Lecture Notes in Computer Science, vol. 287, pp. 340–360, Springer (1987)
15. Carvalho, F.R., Abe, J.M.: A simplified version of the fuzzy decision method and its comparison with the paraconsistent decision method. In: AIP Conference Proceedings of November 24, 2010, vol 1303, pp. 216–235, ISBN 978-0-7354-0858-6, One Volume, Print; 458 (2010) doi:10.1063/1.3527158
16. Carvalho, F.R., Brunstein I., Abe, J.M.: Decision making method based on paraconsistent annotated logic and statistical method: a comparison, computing anticipatory systems. In: Dubois, D. (ed.) American Institute of Physics, Melville ISBN 9780735405790, vol 1051, pp. 195–208 (2008) doi:10.1063/1.3020659
17. Carvalho, F.R., Brunstein, I., Abe, J.M.: Paraconsistent annotated logic in analysis of viability: an approach to product launching. In: Dubois, D. (ed.) American Institute of Physics, AIP Conference Proceedings, Springer, Physics and Astronomy, vol. 718, pp. 282–291, ISBN 0-7354-0198-5, ISSN: 0094-243X (2004)
18. da Costa, N.C.A., Abe, J.M., Subrahmanian, V.S.: Remarks on annotated logic. Zeitschrift f. math. Logik und Grundlagen Math. vol. 37, pp. 561–570 (1991)
19. da Costa, N.C.A., Prado, J.P.A., Abe, J.M., Ávila, B.C., Rillo, M., Paralog: Um Prolog Paraconsistente baseado em Lógica Anotada, Coleção Documentos, Série Lógica e Teoria da Ciência, IEA-USP, vol. 18, p. 21 (1995)
20. da Costa, N.C.A.: Logiques Classiques et non classiques. Masson, Paris (1997)
21. da Costa, N.C.A.: On the theory of inconsistent formal systems. Notre Dame J. of Form. Logic **15**, 497–510 (1974)
22. da Costa, N.C.A., Subrahmanian, V.S., Vago, C.: The paraconsistent logics $P\tau$, *Zeitschrift fur Math*. Logik und Grund. der Math. **37**, 137–148 (1991)

23. Da Silva Filho, J.I., Abe, J.M.: Paraconsistent electronic circuits. Int. J. Comput. Anticipatory Syst. **9**, 337–345 (2001)
24. Da Silva Filho, J.I., Torres, G.L, Abe, J.M.: Uncertainty treatment using paraconsistent logic—Introducing Paraconsistent Artificial Neural Networks, IOS Press, Holanda, Vol. 211, pp. 328 (2010)
25. Da Silva Filho, J.I.: Métodos de interpretação da Lógica Paraconsistente Anotada com anotação com dois valores LPA2v com construção de Algoritmo e implementação de Circuitos Eletrônicos, EPUSP, Ph.D. Thesis (in Portuguese), São Paulo (1999)
26. Fagin, R., Halpern, J.Y., Moses, Y., Vardi, M.Y.: Reasoning About Knowledge. The MIT Press, London (1995)
27. Kifer, M., Wu, J.: *A* logic for object oriented logic programming. In: Proceedings of *8*th ACM Symposium on Principles of Database Systems, pp. 379–393 (1989)
28. Kifer, M., Krishnaprasad, T.: *An* evidence based for a theory of inheritance. In: Proceedings 11th International Joint Conference on Artificial Intelligence, pp. 1093–1098, Morgan-Kaufmann (1989)
29. Lopes, H.F.S., Abe, J.M., Kanda, P.A.M., Machado, S., Velasques, B., Ribeiro, P., Basile, L. F.H., Nitrini, R., Anghinah, R.: Improved application of paraconsistent artificial neural networks in diagnosis of alzheimer's disease, Am. J. Neurosci. 2(1): Science Publications, pp. 54–64 (2011)
30. Lopes, H.F.S., Abe, J.M., Anghinah, R.: Application of paraconsistent artificial neural networks as a method of aid in the diagnosis of alzheimer disease, J. Med. Syst., Springer-Netherlands, pp. 1–9 (2009)
31. Mario M.C., Abe, J.M., Ortega, N., Del Santo Jr. M.: Paraconsistent artificial neural network as auxiliary in cephalometric diagnosis, artificial organs, Wiley Interscience, vol. 34, pp. 215–221 (2010)
32. Nakamatsu, K., Abe, J.M., Suzuki, A.: Annotated semantics for defeasible deontic reasoning, rough sets and current trends in computing, Lecture Notes in Artificial Intelligence series 2005, pp. 470–478, Springer (2000)
33. Nakamatsu, K., Abe, J.M., Kountchev, R.: Introduction to intelligent elevator control based on EVALPSN. In: Lecture Notes in Computer Science, ISSN 0302-9743, Springer, Heidelberg, Alemanha, LNAI 627813314210.1007/978-3-642-15393-8_16 (2010)
34. Nakamatsu, K., Abe, J.M., Akama, S.: Intelligent safety verification for pipeline process order control based on bf-EVALPSN, ICONS 2012, The Seventh International Conference on Systems, IARIA, pp. 175–182, ISBN: 9781612081847 (2012)
35. Nogueira, M.: Processo Para Gestão de Riscos em Projetos de Software Apoiado em Lógica Paraconsistente Anotada Evidencial Eτ, Ph.D. thesis, in Portuguese, Paulista University (2013)
36. Prado, J.P.A.: Uma Arquitetura em IA Baseada em Lógica Paraconsistente, Ph.D. Thesis (in Portuguese), University of São Paulo (2006)
37. Reis, N.F.: Método Paraconsistente de Cenários Prospectivos, Ph.D. thesis, in Portuguese, Paulista University (2014)
38. Souza, S., Abe, J.M., Nakamatsu, K.: MICR automated recognition based on paraconsistent artificial neural networks. In: Procedia Computer Science, vol. 22, pp. 1083–1091. Published by Elsevier, Amsterdam, ISSN: 1877-0509 (2013)
39. Subrahmanian, V.S.: On the semantics of quantitative logic programs. In: Proceedings of 4th IEEE Symposium on Logic Programming, Computer Society Press, pp. 173–182. Washington (1987)
40. Torres, C.R.: Sistema Inteligente Baseado na Lógica Paraconsistente Anotada Evidencial Eτ para Controle e Navegação de Robôs Móveis Autônomos em um Ambiente Não-estruturado, Ph.D. Thesis (in Portuguese), Federal University of Itajuba, Brazil (2010)