Intelligent Systems Reference Library 94

Jair Minoro Abe *Editor*

Paraconsistent Intelligent-Based Systems

New Trends in the Applications of Paraconsistency

Springer

Intelligent Systems Reference Library

Volume 94

About this Series

The aim of this series is to publish a Reference Library, including novel advances and developments in all aspects of Intelligent Systems in an easily accessible and well structured form. The series includes reference works, handbooks, compendia, textbooks, well-structured monographs, dictionaries, and encyclopedias. It contains well integrated knowledge and current information in the field of Intelligent Systems. The series covers the theory, applications, and design methods of Intelligent Systems. Virtually all disciplines such as engineering, computer science, avionics, business, e-commerce, environment, healthcare, physics and life science are included.

More information about this series at http://www.springer.com/series/8578

Jair Minoro Abe
Editor

Paraconsistent Intelligent-Based Systems

New Trends in the Applications of Paraconsistency

Springer

Editor
Jair Minoro Abe
São Paulo
Brazil

ISSN 1868-4394 ISSN 1868-4408 (electronic)
Intelligent Systems Reference Library
ISBN 978-3-319-19721-0 ISBN 978-3-319-19722-7 (eBook)
DOI 10.1007/978-3-319-19722-7

Library of Congress Control Number: 2015940988

Springer Cham Heidelberg New York Dordrecht London

Printed on acid-free paper

Springer International Publishing AG Switzerland is part of Springer Science+Business Media
(www.springer.com)

To my wife Tiyo and my lovely daughters
Clarissa Sayumi and Letícia Hisae

Preface

The studying of Logic in Brazil has enjoyed a wide influence on the field due in large measure to the teachings, writings, and efforts of Newton C.A. da Costa spanning a period of half century or more.

The many hours I have spent in seminairs, lectures, and private conversations with Prof. Newton Costa were a real privilege.

This present book gives me special satisfaction to be able to record my gratitude and admiration to Prof. Newton Costa on this occasion.

One of da Costa's major virtues is the construction (with other logicians independently) of a new type of non-classical logic, namely the so-called paraconsistent logics (da Costa's C_n systems). Roughly speaking, such systems allow contradictions in their interior without trivialization. This is a rare contribution in pure science made by Brazilian scientists.

In the beginning of my career I have experienced people observing 'paraconsistent systems can be even interesting, but there is no intrinsic value at all, so there are not relevant applications'…

This book is devoted to some applications of annotated logics, which are a kind paraconsistent (and generally also paracomplete) logics.

I hope that this tiny collaboration deserves the attention of more experts to become interested in the subject that is undoubtedly promising.

I would like to express thanks to the collaborators who were willing with high-level works. Without them this book would have not been possible.

Last but not least, I am deeply grateful to Prof. Lakhmi C. Jain for his encouragement and opportunity offered to accomplish this project.

Jair Minoro Abe

Contents

Paraconsistent Logics: Preamble 1
Jair Minoro Abe

Constructive Discursive Logic: Paraconsistency in Constructivism 23
Seiki Akama, Jair Minoro Abe and Kazumi Nakamatsu

Paraconsistent Annotated Logic Program EVALPSN and Its Applications. 39
Kazumi Nakamatsu, Jair Minoro Abe and Seiki Akama

The New Hardware Structure of the Emmy II Robot 87
Cláudio Rodrigo Torres and Régis Reis

A Suggestion for Sample Size Determination Using an Instrument for Collecting Medical Data Based on Discrete Items. 105
Euro de Barros Couto Junior

Paraconsistent Logic Algorithms Applied to Seasonal Comparative Analysis with Biomass Data Extracted by the Fouling Process. 131
João Inácio da Silva Filho, Irapajy da Silva Caetano, Floriana Nascimento Pontes, Maurício Conceição Mario, Jair Minoro Abe and Fábio Giordano

An Algorithmic Method Supported by Paraconsistent Annotated Logic Applied to the Determination of Friction Factors for Turbulent Flow in Smooth Pipes 153
Maurício Conceição Mário, Marcílio Dias Lopes,
Cláudio Luís Magalhães Fernandes, Dorotéa Vilanova Garcia,
João Inácio Da Silva Filho, José Carlos Morilla,
Clóvis Misseno Da Cruz, Jair Minoro Abe,
Claudio Rodrigo Torres and Deovaldo de Moraes Júnior

Paraconsistent Logic Study of Image Focus in Cylindrical Refraction Experiments 175
Paulo Henrique Ferraz Masotti and Roberto Navarro de Mesquita

Paraconsistent Artificial Neural Networks and Aspects of Pattern Recognition 207
Sheila Souza and Jair Minoro Abe

Paraconsistent Logic in Decision Making: Paraconsistent Decision Method (PDM) 233
Fábio Romeu de Carvalho

Paraconsistent Neurocomputing and Biological Signals Analysis 273
Jair Minoro Abe, Helder Frederico da Silva Lopes
and Renato Anghinah

Contributors

Jair Minoro Abe Graduate Program in Production Engineering, ICET—Paulista University, São Paulo, Brazil; Institute for Advanced Studies, University of São Paulo, São Paulo, Brazil

Seiki Akama C-Republic, Asao-Ku, Kawasaki, Japan

Renato Anghinah Reference Center of Behavioral Disturbances and Dementia (CEREDIC) of Faculty of Medicine, University of São Paulo, São Paulo, Brazil

Clóvis Misseno Da Cruz UNISANTA—Santa Cecília University, Santos, SP, Brazil

Irapajy da Silva Caetano UNISANTA—Santa Cecília University, Santos, SP, Brazil

João Inácio da Silva Filho UNISANTA—Santa Cecília University, Santos, SP, Brazil; IEA-USP—Institute of Advanced Studies of the University of São Paulo, São Paulo, Brazil

Helder Frederico da Silva Lopes Faculty of Medicine, University of São Paulo, São Paulo, Brazil

Euro de Barros Couto Junior Municipal Government of São Paulo, São Paulo, Brazil

Fábio Romeu de Carvalho Paulista University, UNIP, São Paulo, Brazil

Cláudio Luís Magalhães Fernandes UNISANTA—Santa Cecília University, Santos, SP, Brazil

Dorotéa Vilanova Garcia UNISANTA—Santa Cecília University, Santos, SP, Brazil

Fábio Giordano UNISANTA—Santa Cecília University, Santos, SP, Brazil

Marcílio Dias Lopes UNISANTA—Santa Cecília University, Santos, SP, Brazil

Maurício Conceição Mario UNISANTA—Santa Cecília University, Santos, SP, Brazil

Paulo Henrique Ferraz Masotti Nuclear Engineering Center, Instituto de Pesquisas Energéticas e Nucleares, IPEN/CNEN-SP, São Paulo, Brazil

Roberto Navarro de Mesquita Nuclear Engineering Center, Instituto de Pesquisas Energéticas e Nucleares, IPEN/CNEN-SP, São Paulo, Brazil

Deovaldo de Moraes Júnior UNISANTA—Santa Cecília University, Santos, SP, Brazil

José Carlos Morilla UNISANTA—Santa Cecília University, Santos, SP, Brazil

Kazumi Nakamatsu School of Human Science and Environment, University of Hyogo, Himeji, Japan

Floriana Nascimento Pontes UNISANTA—Santa Cecília University, Santos, SP, Brazil

Régis Reis Methodist University of São Paulo, São Paulo, Brazil

Sheila Souza PRODESP—Data Processing Company of São Paulo State, Taboão da Serra, Brazil

Cláudio Rodrigo Torres UNISANTA—Santa Cecília University, Santos, SP, Brazil; Methodist University of São Paulo, São Paulo, Brazil

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_{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 $A \rightarrow (\neg A \rightarrow B)$ (where 'A' and 'B' are formulas, '$\neg A$' is the negation of 'A' and '$\rightarrow$' is the symbol of implication). Let us admit as premises contradictory formulas A and $\neg A$. As noted above, $A \rightarrow (\neg A \rightarrow B)$ is a valid scheme. Taking into account the assumptions made for the deduction, the Modus Ponens rule (from A and $A \rightarrow B$ we deduce B) give us $(\neg A \rightarrow B)$. 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_{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_{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^{o} abbreviates $\neg(A \wedge \neg A)$. A^{i} abbreviates $A^{\mathrm{o}\ldots\mathrm{o}}$, where the symbol $^{\mathrm{o}}$ appears i times, $i \geq 1$. (thus, A^1 is A^{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_ω 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_ω 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 $C_0, C_1, \ldots, C_n, \ldots, C_\omega$ is such that C_i is strictly stronger than C_{i+1}, for all $i, 1 \leq i < \omega$. C_ω is the weakest calculus of the hierarchy. Notice that we can extend C_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_n is a generalization of the usual semantics.

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

Therefore the semantics for C_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 τ 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 $\sim : |\tau| \rightarrow |\tau|$ which constitutes the "meaning" of the negation of $Q\tau$. The least element of τ 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 τ) .

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 $\rightarrow$ (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 τ 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

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

The symbol '$\leftrightarrow$' is called the *biconditional* and '$\neg *$' 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 $f_S: |S|^n \rightarrow |S|$. (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 $p_S: |S|^n \rightarrow |\tau|$.

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 $\exists x B$, then $S(A) = 1$ iff $S(B_x[i]) = 1$ for some i in L_S.
6. If A is $\forall x B$, 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 Γ is a set of formulas of L we say that A is a *semantic consequence* of Γ 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 Aτ whose underlying language is L: A, B, C are any formulas whatsoever, F, G are complex formulas, and $p_\lambda t_1 \ldots t_n$ an annotated atom. Aτ 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)
$(\wedge_1)$ $A \wedge B \rightarrow A$

$(\wedge_2)$ $A \wedge B \to B$
$(\wedge_3)$ $A \to (B \to (A \wedge B))$
$(\vee_1)$ $A \to A \vee B$
$(\vee_2)$ $B \to A \vee B$
$(\vee_3)$ $(A \to C) \to ((B \to C) \to ((A \vee B) \to C))$
$(\neg_1)$ $(F \to G) \to ((F \to \neg G) \to \neg F)$
$(\neg_2)$ $F \to (\neg F \to A)$
$(\neg_3)$ $F \vee \neg F$
$(\exists_1)$ $A(t) \to \exists x A(x)$
$(\exists_2)$ $\frac{A(x) \to B}{\exists x A(x) \to B}$
$(\forall_1)$ $\forall x A(x) \to A(t)$
$(\forall_2)$ $\frac{B \to A(x)}{B \to \forall x A(x)}$
(τ_1) $p_\perp t_1 \ldots t_n$
(τ_2) $\neg^k p_\lambda t_1 \ldots t_n \to \neg^{k-1} p_{\sim\lambda} t_1 \ldots t_n, k \geq 1$
(τ_3) $p_\lambda t_1 \ldots t_n \to p_\mu t_1 \ldots t_n, \lambda \geq \mu$
(τ_4) $p_{\lambda 1} t_1 \ldots t_n \wedge p_{\lambda 2} t_1 \ldots t_n \wedge \ldots \wedge p_{\lambda m} t_1 \ldots t_n \to p_\lambda t_1 \ldots t_n$, where $\lambda = \bigvee_{i=1}^{m} \lambda_i$
$(=_1)$ $x = x$
$(=_2)$ $x = y \to (A[x] \leftrightarrow A[y])$

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. $\vdash (A \to B) \to ((A \to \neg * B) \to \neg * A)$
4. $\vdash A \to \neg * \neg * A$
5. $\vdash \neg * A \to (A \to B)$
6. $\vdash (A \to \neg * A) \to B$

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

Corollary 2.12.1 *In $Q\tau$ the connectives $\neg *$, $\wedge$, $\vee$, and $\to$ 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 $p_S(a_1, \ldots, a_n) = \bot$ and $p_S(b_1, \ldots, b_n) = \top$ where $(a_1, \ldots, a_n) \neq (b_1, \ldots, b_n)$.

Then, $S(p_\bot i_1 \ldots i_n) = 1$, where i_j the name of b_j, $j = 1, \ldots, n$, and $S(\neg p_\bot i_1 \ldots i_n) = 1$. 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 τ 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 Γ 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 Γ.

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, $Q\tau$ is, in general, a paraconsistent, paracomplete and non-alethic logic.

We give a Henkin-type proof of the completeness theorem for the logics $Q\tau$.

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 (τ_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 (τ_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 Γ 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 τ of the logics $Q\tau$ is infinite, due the axiom scheme (τ_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 Eτ. The lattice τ is composed by the set [0, 1] × [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 Eτ 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 μ and contrary evidence is λ." 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 (μ, λ) is from Inconsistent or Paracomplete state. Similarly, $G_{ce}(\mu, \lambda)$ show us how close (or far) the annotation constant (μ, λ) 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

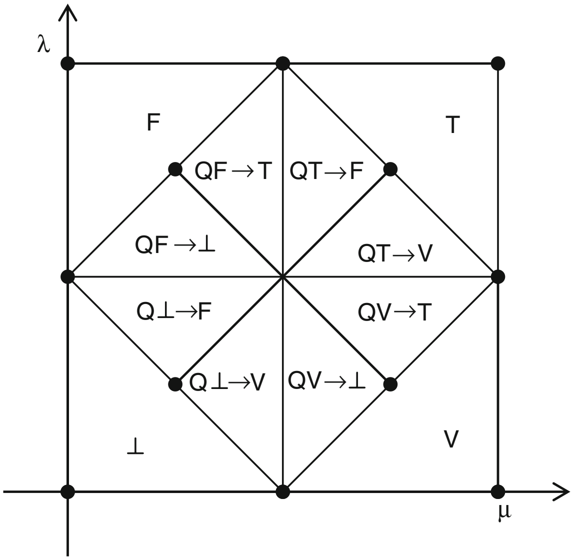


Fig. 1.1 Lattice of extreme and non-extreme states

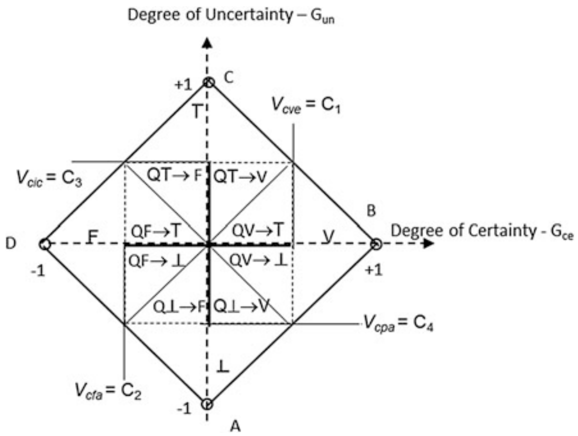


Fig. 1.2 Certainty and uncertainty degrees and decision states

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 Eτ 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 Eτ 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 Eτ, is a logical controller that we call Para-control. It was built an experimental robot based on paraconsistent annotated evidential logic Eτ 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)

Constructive Discursive Logic: Paraconsistency in Constructivism

Seiki Akama, Jair Minoro Abe and Kazumi Nakamatsu

Abstract We propose a constructive discursive logic with strong negation *CDLSN* based on Nelson's constructive logic N^- as a constructive version of Jaśkowski's discursive logic. In *CDLSN*, discursive negation is defined similar to intuitionistic negation and discursive implication is defined as material implication using discursive negation. We give an axiomatic system and Kripke semantics with a completeness proof. We also discuss some possible applications of *CDLSN* for common-sense reasoning.

Keywords Jaśkowski · Constructive discursive logic · Common-sense reasoning

1 Introduction

Jaśkowski proposed *discursive logic* (or *discussive logic*) in 1948. It is the first formal *paraconsistent logic* which is classified as a *non-adjunctive system*; see Jaśkowski [10]. The gist of discursive logic is to consider the nature of our ordinary discourse. In a discourse, there are several *participants* who have some information, beliefs, and such. In this regard, truth is formalized by means of the sum of opinions supplied by participants. Even if each participant has consistent information, some participant could be inconsistent with other participants.

S. Akama (✉)
1-20-1 Higashi-Yurigaoka, Asao-Ku, Kawasaki 215-0012, Japan
e-mail: akama@jcom.home.ne.jp

J.M. Abe
Paulista University, São Paulo, Brazil
e-mail: jairabe@uol.com.br

K. Nakamatsu
University of Hyogo, Himeji, Japan
e-mail: nakamatsu@shse.u-hyogo.ac.jp

J.M. Abe (ed.), *Paraconsistent Intelligent-Based Systems*,
Intelligent Systems Reference Library 94, DOI 10.1007/978-3-319-19722-7_2

It is reasonable to suppose that $A \wedge \sim A$ (A and $not(A)$) does not hold while both A and $\sim A$ hold to describe such situations. This means that the so-called *adjunction*, i.e., from $\vdash A, \vdash B$ to $\vdash A \wedge B$, is invalid. Here, $\vdash A$ reads "A is provable". Jaśkowski modeled the idea founded on modal logic S5 and reached the discursive logic in which adjunction and *modus ponens* cannot hold. In addition, Jaśkowski introduced discursive implication $A \rightarrow_d B$ as $\diamondsuit A \rightarrow B$ satisfying *modus ponens*, where $\diamondsuit$ denotes the possibility operator.

Akama, Abe and Nakamatsu [5] proposed a constructive discursive logic based on constructivism. It can be viewed as a constructive version of Jaśkowski's original system; also see Akama [4]. Its base is Nelson's constructive logic [17], although Jaśkowski developed his discursive logic based on classical modal logic. Our approach is seen as a new way of formalizing discursive logic.

The rest of this paper is as follows. Section 2 reviews Jaśkowski's discursive logic. In Sect. 3, we introduce constructive discursive logic with strong negation *CDLSN* with an axiomatic system. Section 4 outlines a Kripke semantics. We establish the completeness theorem. In Sect. 5, we suggest possible applications of *CDLSN* for common-sense reasoning. Section 6 concludes the paper with a discussion on future work. This paper is based on the materials in Akama, Abe and Nakamatsu [5] and Akama, Nakamatsu and Abe [6].

2 Jaśkowski's Discursive Logic

Discursive logic, due to the Polish logician S.Jaśkowski [10], is a formal system J satisfying the conditions: (a) from two contradictory propositions, it should not be possible to deduce any proposition; (b) most of the classical theses compatible with (a) should be valid; (c) J should have an intuitive interpretation.

Such a calculus has, among others, the following intuitive properties remarked by Jaśkowski himself: suppose that one desires to systematize in only one deductive system all theses defended in a discussion. In general, the participants do not confer the same meaning to some of the symbols. One would have then as theses of a deductive system that formalize such a discussion, an assertion and its negation, so both are "true" since it has a variation in the sense given to the symbols. It is thus possible to regard discursive logic as one of the so-called *paraconsistent logics*.

Jaśkowski's D_2 contains propositional formulas built from the logical symbols of classical logic. In addition, possibility operator $\diamondsuit$ in S5 is added. $\diamondsuit A$ reads "A is possible". Based on the possibility operator, three discursive logical symbols can be defined as follows:

$$\begin{array}{ll} \text{discursive implication}: & A \rightarrow_d B =_{def} \diamondsuit A \rightarrow B \\ \text{discursive conjunction}: & A \wedge_d B =_{def} \diamondsuit A \wedge B \\ \text{discursive equivalence}: & A \leftrightarrow_d B =_{def} (A \rightarrow_d B) \wedge_d (B \rightarrow_d A) \end{array}$$

Additionally, we can define discursive negation $\neg_d A$ as $A \rightarrow_d false$. Jaśkowski's original formulation of D_2 in [10] used the logical symbols: $\rightarrow_d, \leftrightarrow_d, \vee, \wedge, \neg$, and he later defined $\wedge_d$ in [11].

The axiomatization due to Kotas [12] has the following axioms and the rules of inference. Here, $\Box$ is the necessity operator, and is definable by $\neg\Diamond\neg$. $\Box A$ reads "A is necessary".

Axioms

(A1) $\Box(A \rightarrow (\neg A \rightarrow B))$
(A2) $\Box((A \rightarrow B) \rightarrow ((B \rightarrow C) \rightarrow (A \rightarrow C)))$
(A3) $\Box((\neg A \rightarrow A) \rightarrow A)$
(A4) $\Box(\Box A \rightarrow A)$
(A5) $\Box(\Box(A \rightarrow B) \rightarrow (\Box A \rightarrow \Box B))$
(A6) $\Box(\neg\Box A \rightarrow \Box\neg\Box A)$

Rules of Inference

(R1) substitution rule
(R2) $\Box A, \Box(A \rightarrow B)/\Box B$
(R3) $\Box A/\Box\Box A$
(R4) $\Box A/A$
(R5) $\neg\Box\neg\Box A/A$

Note that discursive implication $\rightarrow_d$ satisfies *modus ponens* in S5, but $\rightarrow$ does not. There are other axiomatizations of D_2. For example, da Costa and Dubikajtis gave an axiomatization based on the connectives $\rightarrow_d, \wedge_d, \neg$; see [8]. Semantics for discursive logic can be obtained by a Kripke semantics for modal logic S5. Jaśkowski's three conditions for J mentioned above are solved by many workers in different ways. For a comprehensive survey on discursive logic, see da Costa and Doria [9].

3 Constructive Discursive Logic with Strong Negation

The gist of discursive logic is to use the modal logic S5 to define discursive logical connectives which can formalize a non-adjunctive system. It follows that discursive logic can be seen as a paraconsistent logic, which does not satisfy *explosion* of the form: $\{A, \neg A\} \models B$ for any A and B, where $\models$ is a consequence relation. We say that a system is *trivial* iff all the formulas are provable. Therefore, paraconsistent logic is useful to formalize inconsistent but *non-trivial* theories.

Most works on discursive logic utilize classical logic and S5 as a basis. However, we do not think that these are essential. For instance, different modal logics yield the corresponding discursive logics. We can use non-classical logics as the base. An intuitionist hopes to have a discursive system in a constructive setting. It is the starting point of Akama, Abe and Nakamatsu [5].

To make the idea formal, it is worth considering Nelson's constructive logic with strong negation N^- of Almukdad and Nelson [7]. In N^-, $\sim$ denotes *strong negation* satisfying the following axioms:

(N1) $\sim\sim A \leftrightarrow A$
(N2) $\sim(A \wedge B) \leftrightarrow (\sim A \vee \sim B)$
(N3) $\sim(A \vee B) \leftrightarrow (\sim A \wedge \sim B)$
(N4) $\sim(A \rightarrow B) \leftrightarrow (A \wedge \sim B)$

and the axiomatization of the intuitionistic positive logic Int^+ with *modus ponens* (MP), i.e. $A, A \rightarrow B/B$ as the rule of inference.

Strong negation can express explicit negative information which cannot be described by intuitionistic negation. In this sense, strong negation is constructive, but intuitionistic negation is not. As the name shows, strong negation is stronger than intuitionistic negation in that $\sim A \rightarrow \neg A$ holds but the converse does not. Note here that N^- is paraconsistent in the sense that $\sim(A \wedge \sim A)$ and $(A \wedge \sim A) \rightarrow B$ do not hold.

If we add (N0) to N^-, we have N of Nelson [17].

$$\text{(N0)}\ (A \wedge \sim A) \rightarrow B$$

In N, *intuitionistic negation* $\neg$ can be defined as follows:

$$\neg A =_{def} A \rightarrow \ \sim A$$

If we add the law of *excluded middle*: $A \vee \sim A$ to N, the resulting system is classical logic.

Indeed, N^- is itself a paraconsistent logic; see Akama [3]. But it can also be accommodated as a version of discursive logic.

Now, we introduce the *constructive discursive logic with strong negation CDLSN*. It diverges in two ways from D_2: (1) it does not take classical logic as its starting point; and (2) it does not use the possibility operator $\Diamond$ as a modality, but use two negation operators.

CDLSN can be defined in two ways. One is to extend N^- with discursive negation $\neg_d$. The other is to weaken intuitionistic negation in N^-. We adopt the first approach.

Here, we fix the language of the logics which we use in this paper. The language of Int^+ is defined as the set of propositional variables and logical symbols: $\wedge$ (conjunction), $\vee$ (disjunction) and $\rightarrow$ (implication). The language of *Int* is the extension of that of Int^+ with $\neg$ (intuitionistic negation). The language of N^- is the extension of that of Int^+ with $\sim$ (strong negation). The language of *CDLSN* is the extension of N^- with $\neg_d$ (discursive negation). Additionally, we use the logical constant *false* as the abbreviation of $\sim(A \rightarrow A)$.

We believe that *CDLSN* is (constructive) improvement of D_2. First, *CDLSN* uses Int^+ rather than classical logic as the base. Second, *CDLSN* simulates modality in D_2 by negations, although D_2 needs the possibility operator.

$\neg_d$ is similar to $\neg$, but these are not equivalent. The motivation of introducing $\neg_d$ is to interpret discursive negation as the negation used by an intuitionist in the discursive context. Unfortunately, intuitionistic negation is not a discursive negation. And we need to re-interpret it as $\neg_d$. Based on $\neg_d$, we can define $\rightarrow_d$ and $\wedge_d$.

Discursive implication $\rightarrow_d$ and discursive conjunction $\wedge_d$ can be respectively introduced by definition as follows.

$$A \rightarrow_d B =_{def} \neg_d A \vee B$$
$$A \wedge_d B =_{def} \sim\neg_d A \wedge B$$

Observe that $A \rightarrow (\sim A \rightarrow B)$ is not a theorem in *CDLSN* while $A \rightarrow (\neg_d A \rightarrow B)$ is a theorem in *CDLSN*. The axiomatization of *CDLSN* is that of N^- with the following three axioms.

$$\text{(CDLSN1)}\ \neg_d A \rightarrow (A \rightarrow B)$$
$$\text{(CDLSN2)}\ (A \rightarrow B) \rightarrow ((A \rightarrow \neg_d B) \rightarrow \neg_d A)$$
$$\text{(CDLSN3)}\ A \rightarrow\ \sim\neg_d A$$

Here, an explanation of these axioms may be in order. (CDLSN1) and (CDLSN2) describe basic properties of intuitionistic negation. By (CDLSN3), we show the connection of $\sim$ and $\neg_d$. The intuitive interpretation of $\sim\neg_d$ is like possibility under our semantics developed below.

$\neg_d$ is weaker than $\neg$. Vorob'ev [20] proposed a constructive logic having both strong and intuitionistic negation. It extends N with the following two axioms:

$$\sim\neg A \leftrightarrow A$$
$$\sim A \rightarrow \neg A,$$

where A is atomic

If we replace (CDLSN3) by the axiom of the form $\sim\neg_d A \leftrightarrow A$ and add the axiom $\sim A \rightarrow \neg_d A$, then $\neg_d$ agrees with $\neg$. Thus, it is not possible to identify $\neg$ and $\neg_d$ in our axiomatization.

We use $\vdash A$ to mean that A is a theorem in *CDLSN*. Here, the notion of a proof is defined as usual. Let $\Gamma = \{B_1, \ldots, B_n\}$ be a set of formulas and A be a formula. Then, $\Gamma \vdash A$ iff $\vdash \Gamma \rightarrow A$.

Notice that $\neg_d$ has some similarities with $\neg$, as the following lemma indicates.

Lemma 1 *The following formulas are provable in CDLSN.*

(1) $\vdash A \rightarrow \neg_d\neg_d A$
(2) $\vdash (A \rightarrow B) \rightarrow (\neg_d B \rightarrow \neg_d A)$
(3) $\vdash (A \wedge \neg_d A) \rightarrow B$
(4) $\vdash \neg_d(A \wedge \neg_d A)$
(5) $\vdash (A \rightarrow \neg_d A) \rightarrow \neg_d A$

Proof Ad(1): From (CDSLN1) and Int^{+} i.e. $\vdash (A \to (B \to C)) \to (B \to (A \to C)))$, we have (i).

(i) $\vdash A \to (\neg_d A \to A)$
(ii) is an instance of (CDLSN2).
$\vdash (\neg_d A \to A) \to ((\neg_d A \to \neg_d A) \to \neg_d \neg_d A)$
(iii) is a theorem of Int^{+}, i.e., $\vdash (A \to B) \to ((B \to C) \to (A \to C)))$
$\vdash (A \to (\neg_d A \to A)) \to (((\neg_d A \to A) \to ((\neg_d A \to \neg_d A)$
$\to \neg_d \neg_d A)) \to (A \to ((\neg_d A \to \neg_d A) \to \neg_d \neg_d A)))$
From (i) and (iii) by (MP), we have (iv).
(iv) $\vdash (((\neg_d A \to A) \to ((\neg_d A \to \neg_d A) \to \neg_d \neg_d A))$
$\to (A \to ((\neg_d A \to \neg_d A) \to \neg_d \neg_d A)))$
From (ii) and (iv) by (MP), we have (v).
(v) $\vdash A \to ((\neg_d A \to \neg_d A) \to \neg_d \neg_d A))$
by $\vdash (A \to (B \to C)) \to (B \to (A \to C))$ we can derive (vi)
(vi) $\vdash (\neg_d A \to \neg_d A) \to (A \to \neg_d \neg_d A)$
since $\vdash A \to A$ we have (vii)
(vii) $\vdash \neg_d A \to \neg_d A$From (vi) and (vii) by (MP), we can finally obtain (viii).
(viii) $\vdash A \to \neg_d \neg_d A$
Ad(2): By (CDLSN2), we have (i).
(i) $\vdash (A \to B) \to ((A \to \neg_d B) \to \neg_d A)$
(ii) is a theorem of Int^{+}.
$\vdash (\neg_d B \to (A \to \neg_d B)) \to (((A \to \neg_d B) \to \neg_d A)$
$\to (\neg_d B \to \neg_d A))$
(iii) is an instance of $A \to (B \to A)$ which is the axiom of Int^{+}
$\vdash \neg_d B \to (A \to \neg_d B)$
From (ii) and (iii) by (MP), (iv) is obtained.
(iv) $\vdash ((A \to \neg_d B) \to \neg_d A) \to (\neg_d B \to \neg_d A)$
(v) is a theorem of Int^{+}.
$\vdash ((A \to B) \to ((A \to \neg_d B) \to \neg_d A)$
$\to ((((A \to \neg_d B) \to \neg_d A) \to (\neg_d B \to \neg_d A))$
$\to ((A \to B) \to (\neg_d B \to \neg_d A)))$
From (i) and (v) by (MP), (vi) can be proved.
(vi) $\vdash ((A \to \neg_d B) \to \neg_d A) \to (\neg_d B \to \neg_d A))$
$\to ((A \to B) \to (\neg_d B \to \neg_d A))$
From (iv) and (vi) by (MP), we can reach (vii).
(vii) $\vdash (A \to B) \to (\neg_d B \to \neg_d A)$
Ad(3): By (CDLSN1), we have (i).
(i) $\vdash \neg_d A \to (A \to B)$
From $\vdash (A \to (B \to C)) \to (B \to (A \to C))$, we can derive (ii).
(ii) $\vdash A \to (\neg_d A \to B)$
since $\vdash (A \to (B \to C)) \to ((A \wedge B) \to C)$, we have (iii).

(iii) $\vdash (A \rightarrow (\neg_d A \rightarrow B)) \rightarrow ((A \wedge \neg_d A) \rightarrow B)$
From (ii) and (iii) by (MP), we can obtain (iv).
(iv) $\vdash (A \wedge \neg_d A) \rightarrow B$
Ad(4): By (3), we have (i) and (ii).
(i) $\vdash (A \wedge \neg_d A) \rightarrow B$
(ii) $\vdash (A \wedge \neg_d A) \rightarrow \neg_d B$
From (CDLSN2), (iii) holds.
(iii) $((A \wedge \neg_d A) \rightarrow B) \rightarrow (((A \wedge \neg_d A) \rightarrow \neg_d B) \rightarrow \neg_d (A \wedge \neg_d A))$
From (i) and (iii) by (MP), we have (iv).
(iv) $((A \wedge \neg_d A) \rightarrow \neg_d B) \rightarrow \neg_d (A \wedge \neg_A)$
From (ii) and (iv) by (MP), we can derive (v).
(v) $\vdash \neg_d (A \wedge \neg_d A)$
Ad(5): By (CDLSN2), we have (i).
(i) $\vdash (A \rightarrow A) \rightarrow ((A \rightarrow \neg_d A) \rightarrow \neg_d A)$
(ii) is a theorem of Int^+.
$\vdash A \rightarrow A$
From (i) and (ii) by (MP), we can obtain (iii).
(iii) $(A \rightarrow \neg_d A) \rightarrow \neg_d A$ □

It should be, however, pointed out that the following formulas are not provable in *CDLSN*.

$\nvdash \sim (A \wedge \sim A)$

$\nvdash A \vee \sim A$

$\nvdash (A \rightarrow B) \rightarrow (\sim B \rightarrow \sim A)$

$\nvdash \neg_d \neg_d A \rightarrow A$

$\nvdash A \vee \neg_d A$

$\nvdash (\neg_d A \rightarrow A) \rightarrow A$

$\nvdash \sim \neg_d A \rightarrow A$

$\nvdash A \rightarrow_d A$

4 Kripke Semantics

It is possible to give a Kripke semantics for *CDLSN* which is a discursive modification of that for *N*. A Kripke semantics for *N* can be formalized as an extension of that for intuitionistic logic. It first provided by Thomason [19]; also see Akama [1, 2]. Akama [3]studied a Kripke semantics for N^-.

Now, we define a Kripke model for *CDLSN*. Let *PV* be a set of propositional variables and *p* be a propositional variable, and *For* be a set of formulas. A *CDLSN-model* is a tuple $\langle W, w_0, R, V \rangle$, where $W \neq \emptyset$ is a set of *worlds*, $w_0 \in W$ satisfying $\forall w(w_0 R w)$, $R \subseteq W \times W$ is a reflexive and transitive relation, and $V : PV \times W \rightarrow \{0, 1\}$ is a partial *valuation* satisfying:

$$V(p, w) = 1 \text{ and } wRv \;\Rightarrow\; V(p, v) = 1$$
$$V(p, w) = 0 \text{ and } wRv \;\Rightarrow\; V(p, v) = 0$$

for any formula $p \in PV$ and $w, v \in W$. Here, $V(p, w) = 1$ is read "p is true at w" and $V(p, w) = 0$ is read "p is false at w", respectively. Both truth and falsity are independently given by a constructive setting.

We can now extend V for any formula A, B in a tandem way as follows.

$$\begin{array}{ll}
V(\sim A, w) = 1 & \text{iff } V(A, w) = 0 \\
V(A \wedge B, w) = 1 & \text{iff } V(A, w) = 1 \text{ and } V(B, w) = 1 \\
V(A \vee B, w) = 1 & \text{iff } V(A, w) = 1 \text{ or } V(B, w) = 1 \\
V(A \rightarrow B, w) = 1 & \text{iff } \forall v(wRv \text{ and } V(A, v) = 1 \Rightarrow V(B, v) = 1) \\
V(\neg_d A, w) = 1 & \text{iff } \forall v(wRv \Rightarrow V(A, v) = 0) \\
V(\sim A, w) = 0 & \text{iff } V(A, w) = 1 \\
V(A \wedge B, w) = 0 & \text{iff } V(A, w) = 0 \text{ or } V(B, w) = 0 \\
V(A \vee B, w) = 0 & \text{iff } V(A, w) = 0 \text{ and } V(B, w) = 0 \\
V(A \rightarrow B, w) = 0 & \text{iff } V(A, w) = 1 \text{ and } V(B, w) = 0 \\
V(\neg_d A, w) = 0 & \text{iff } \exists v(wRv \text{ and } V(A, v) = 1)
\end{array}$$

Additionally, we need the following condition:

$V(A \wedge \sim A, w) = 1$ for some A and some w.

This condition is used to invalidate $(A \wedge \sim A) \rightarrow B$, and guarantees the paraconsistency of $\sim$ in *CDSLN*.

Here, observe that truth and falsity conditions for $\sim\neg_d A$ are implicit in the above clauses from the equivalences such that $V(\sim\neg_d A, w) = 1$ iff $V(\neg_d A, w) = 0$, and $V(\sim\neg_d A, w) = 0$ iff $V(\neg_d A, w) = 1$. One can claim that $\sim\neg_d$ behaves as a modality. In this regard, we do not need to introduce a possibility operator into *CDLSN* as a primitive.

We say that A is *valid*, written $\vDash A$, iff $V(A, w_0) = 1$ in all *CDLSN*-models. Let $\Gamma = \{B_1, \ldots, B_n\}$ be a set of formulas. Then, we say that Γ *entails* A, written $\Gamma \vDash A$, iff $\Gamma \rightarrow A$ is valid.

Lemma 2 states the monotonicity of valuation in a Kripke model.

Lemma 2 *The following hold for any formula A which is not of the form* $\sim\neg_d B$, *and any worlds* $w, v \in W$.

$$V(A, w) = 1 \text{ and } wRv \;\Rightarrow\; V(A, v) = 1,$$
$$V(A, w) = 0 \text{ and } wRv \;\Rightarrow\; V(A, v) = 0.$$

Proof By induction on A.

Ad($\sim$): Suppose $V(\sim A, w) = 1$and wRv. Then, we have that $V(A, w) = 0$ and wRv. By induction hypothesis (IH), we have that $V(A, v) = 0$, i.e $V(\sim A, v) = 1$.

Suppose $V(\sim A, w) = 0$and wRv. Then, we have that $V(A, w) = 1$ and wRv. By (IH), we have that $V(A, v) = 1$, i.e. $V(\sim A, v) = 0$.

Ad(∧): Suppose $V(A \wedge B, w) = 1$ and wRv. Then, we have $V(A, w) = 1$ and $V(B, w) = 1$. By (IH), $V(A, v) = 1$ and $V(B, v) = 1$, i.e. $V(A \wedge B, v) = 1$.

Suppose $V(A \wedge B, w) = 0$ and wRv. Then, we have $V(A, w) = 0$ or $V(B, w) = 0$. By (IH), $V(A, v) = 0$ or $V(B, v) = 0$, i.e. $V(A \wedge B, v) = 0$.

Ad(∨): Suppose $V(A \vee B, w) = 1$ and $wRv) = 1$. Then, we have $V(A, w) = 1$ or $V(B, w) = 1$. By (IH), $V(A, v) = 1$ or $V(B, v) = 1$, i.e. $V(A \vee B, v) = 1$.

Suppose $V(A \vee B, w) = 0$ and wRv. Then, we have $V(A, w) = 0$ and $V(B, w) = 0$. By (IH), $V(A, v) = 0$ and $V(B, v) = 0$, i.e. $V(A \vee B, v) = 0$.

Ad(→): Suppose $V(A \to B) = 1$ and wRv. Then, we have $\forall v(wRv \text{ and } V(A, v) = 1 \Rightarrow V(B, v) = 1)$. By (IH) and the transitivity of R $\forall z(vRz \text{ and } V(A, z) = 1 \Rightarrow V(B, z) = 1)$, i.e. $V(A \to B, v) = 1$.

Suppose $V(A \to B, w) = 0$ and wRv. Then, we have $V(A, w) = 1$ and $V(B, w) = 0$. By (IH), $V(A, v) = 1$ and $V(B, v) = 0$, i.e. $V(A \to B, v) = 0$. □

Lemma 2 does not hold for the formula of the form $\sim \neg_d A$. We can easily construct a counter model. We only treat the case of $V(\sim \neg_d A, w) = 1$. The case of $V(\sim \neg_d A, w) = 0$ is similar. Assume that $V(\sim \neg_d A, w) = 1$ and wRv. Then, $V(\neg_d A, w) = 0$ iff $\exists u(wRu \text{ and } V(A, u) = 1)$. Now, suppose that there exists a world t distinct from u such that vRt and a valuation such that $V(A, t) = 0$. This means that $V(\sim \neg_d A, v) = 0$. Thus, $V(\sim \neg_d A, w) = 1$ and wRv, but $V(\sim \neg_d A, v) = 0$.

We think that the fact is intuitive because $\sim \neg_d A$ behaves as possibility. There are no reasons for possibility in discourse to satisfy the monotonicity.

Next, we present a soundness theorem.

Theorem 1 (soundness)$\vdash A \Rightarrow \models A$

Proof It suffices to check that (CDLSN1), (CDLSN2) and (CDLSN3) are valid and (MP) preserves validity. The proof of preservation of validity under (MP) is well-known in constructive and intuitionistic logic. Thus, we here prove the validity of three axioms.

Ad(CDLSN1): Suppose it is not valid. Then, $V(\neg_d A, w_0) = 1$ and $V(A \to B, w_0) \neq 1$. From the first conjunct, $\forall v(w_0 Rv \Rightarrow V(A, v) \neq 1)$holds. From the second conjunct, $\exists v(w_0 Rv \text{ and } V(A, v) = 1 \text{ and } V(B, v) \neq 1)$. However, $V(A, v) = 1$ and $V(A, v) \neq 1$are contradictory.

Ad(CDLSN2): Suppose it is not valid. Then, $V(A \to B, w_0) = 1$and $V(A \to \neg_d B, w_0) = 1$ and $V(\neg_d A, w_0) \neq 1$. From the first conjunct, $\forall v(w_0 Rv \text{ and } V(A, v) = 1 \Rightarrow V(B, v) = 1)$ holds. From the second conjunct, $\forall v(w_0 Rv \text{ and } V(A,v) = 1 \Rightarrow V(\neg_d B, v) = 1)$iff $\forall v(wRv \text{ and } V(A,v) = 1 \Rightarrow \forall z(vRz \Rightarrow V(A,z) \neq 1)$. From the third conjunct, $\exists v(w_0 Rv \text{ and } V(A,v) = 1$holds. However, $V(A,v) = 1$ and $V(A,z) \neq 1$ for any z such that vRz are contradictory.

Ad(CDLSN3): Suppose it is not valid. Then, $V(A, w_0) = 1$ and $V(\sim \neg_d A, w_0) \neq 1$. From the second conjunct, we have $V(\neg_d A, w_0) \neq 0$ iff $\forall v(w_0 Rv \Rightarrow V(A,v) \neq 1)$. However, $V(A, w_0) = 1$ and $V(A,v) \neq 1$or any v such that $w_0 Rv$ are contradictory. □

Theorem 3 Theorem 3 can be generalized as a strong form, i.e. $\Gamma \vdash A \Rightarrow \Gamma \models A$.

Now, we give a completeness proof. We say that a set of formulas Γ^* is a *maximal non-trivial discursive theory* (mntdt) iff (1) Γ^* is a theory, (2) Γ^* is *non-trivial*, i.e. $\Gamma^* \nvdash B$ for some B, (3) Γ^* is *maximal*, i.e. $A \in \Gamma^*$ or $A \notin \Gamma^*$, (4) Γ^* is *discursive*, i.e. $\neg_d A \notin \Gamma^*$ iff $\sim \neg_d A \in \Gamma^*$. Here, discursiveness is needed to capture the property of discursive negation.

Lemma 3 For any mntdt Γ and any formula A, B the following hold:

(1) $A \wedge B \in \Gamma$ iff $A \in \Gamma$and $B \in \Gamma$
(2) $A \vee B \in \Gamma$ iff $A \in \Gamma$ or $B \in \Gamma$
(3) $A \rightarrow B \in \Gamma$ iff $\forall \Delta(\Gamma \subseteq \Delta \text{ and } A \in \Delta \Rightarrow B \in \Delta)$
(4) $\neg_d A \in \Gamma$ iff $\forall \Delta(\Gamma \subseteq \Delta \Rightarrow A \notin \Delta)$
(5) $\sim (A \wedge B) \in \Gamma$ iff $\sim A \in \Gamma$ or $\sim B \in \Gamma$
(6) $\sim (A \vee B) \in \Gamma$ iff $\sim A \in \Gamma$ and $\sim B \in \Gamma$
(7) $\sim (A \rightarrow B) \in \Gamma$ iff $A \in \Gamma$ and $\sim B \in \Gamma$
(8) $\sim \sim A \in \Gamma$ iff $A \in \Gamma$
(9) $\sim \neg_d A \in \Gamma$ iff $\exists \Delta(\Gamma \subseteq \Delta \text{ and } A \in \Delta)$.

Proof We only prove (4) and (9). Other cases are similarly justified from the literature on constructive logic (cf. Thomason [19]).

Ad(4): $\neg_d A \in \Gamma$ iff (by axiom (CDLSN1)) iff (by axiom (CDLSN1)) $A \rightarrow B \in \Gamma$ iff (by Lemma 3 (3)) $\forall \Delta(\Gamma \subseteq \Delta \text{ and } A \in \Delta \Rightarrow B \in \Delta)$. Since Γ is non-trivial, $B \notin \Gamma$ for some B. Thus, $B \in \Delta$ does not always hold, i.e.$\forall \Delta(\Gamma \subseteq \Delta$ and $A \in \Delta \Rightarrow$ *false*$)$iff $\forall \Delta(\Gamma \subseteq \Delta \Rightarrow A \notin \Delta)$.

Ad(9): We prove it by contraposition from (4). Contraposition can derive $\exists \Delta(\Gamma \subseteq \Delta$ and $A \in \Delta)$ by negating the left and right sides of (4). Then, it is shown to be equivalent to $\neg_d A \notin \Gamma$. By (discursiveness), $\neg_d A \notin \Gamma$ iff $\sim \neg_d A \in \Gamma$.

Based on the maximal non-trivial discursive theory, we can define a canonical model $(\Gamma, \subseteq, V)$ such that Γ is a mntdt, $\subseteq$ is the subset relation, and V is a valuation satisfying the conditions that $V(p, \Gamma) = 1$ iff $p \in \Gamma$ and that $V(p, \Gamma) = 0$ iff $\sim p \in \Gamma$.□

The next lemma is a truth lemma.

Lemma 4 (truth lemma) *For any mntdt* Γ *and any* A, *we have the following*:
$V(A, \Gamma) = 1$ *iff* $A \in \Gamma$
$V(A, \Gamma) = 0$ *iff* $\sim A \in \Gamma$

Proof It suffices to check the case $A = \neg_d B$.

$$
\begin{array}{ll}
V(\neg_d B, \Gamma) = 1 & \textit{iff}\ \forall \Delta \in \Gamma^* (\Gamma \subseteq \Delta \Rightarrow V(B, \Delta) \neq 1) \\
(IH) & \textit{iff}\ \forall \Delta \in \Gamma^* (\Gamma \subseteq \Delta \Rightarrow B \notin \Delta) \\
(Lemma\ 4\ (4)) & \textit{iff}\ \neg_d B \in \Gamma
\end{array}
$$

□

$$
\begin{array}{ll}
V(\neg_d B, \Gamma) = 0 & \textit{iff}\ \exists \Delta \in \Gamma^* (\Gamma \subseteq \Delta \text{ and } V(B, \Delta) = 1) \\
(IH) & \textit{iff}\ \exists \Delta \in \Gamma^* (\Gamma \subseteq \Delta \text{ and } B \in \Delta) \\
(Lemma\ 4\ (9)) & \textit{iff}\ \sim \neg_d B \in \Gamma
\end{array}
$$

Then, we can state the (strong) completeness of *CDLSN* as follows:

Theorem 2 (completeness). $\Gamma \vDash A \ \Rightarrow \ \Gamma \vdash A$

Proof Assume $\Gamma \nvdash A$. Then, by the Lindenbaum lemma, there is a mntdt Γ such that $A \notin \Gamma$. By using a canonical model defined above, we have $V(A, \Gamma) \neq 1$ by Lemma 4. Consequently, completeness follows. □

Finally, we justify the formal properties of *CDLSN* as a discursive logic. It is extremely important because we can understand the differences of *CDLSN* and standard discursive logics like D_2. As mentioned in Sect. 1, Jaśkowski suggested three conditions of discursive logics. We check them here.

CDLSN is *discursive*. First, $\sim (A \wedge \sim A)$ does not hold. The explosion also fails, i.e. $A, \sim A \nvdash B$. But, these hold for $\neg_d$ (cf. Lemma 1), and are not a problem because explosion should be valid for plausible discourses.

Note that the adjunction of the form $\vdash A, \vdash B \ \Rightarrow \ \vdash A \wedge_d B$ does not hold in *CDLSN*. But, it holds for ∧ .

Second, in *CDLSN*, most of the theses of constructive logic are valid. Since *CDLSN* has a constructive base, it is different from D_2 whose base is classical logic.

Third, we can give an intuitive interpretation for *CDLSN* by means of Kripke models as discussed below.

CDLSN is *constructive* because the law of excluded middle, which is a non-constructive principle, does not hold. As discussed above, N^- is a constructive logic, and the fact is not surprising.

From our Kripke semantics given above, we can give an intuitive interpretation of *CDLSN*. The interpretations of the logical symbols of N^- are obvious, and we concentrate on discursive logical symbols.

Here, it may be helpful to explain the interpretation by a brief example. Consider a *discourse* which consists of several persons who are interested in some subjects. Each person has knowledge about subjects, and a discourse is plausibly expanded by adding other persons.

In this setting, a world in our semantics could be identified with the discourse just given. So, the logical symbols can be interpreted with reference to a discourse.

Since the interpretations of $\neg_d$ are crucial, we begin with this, namely

$\neg_d A$ is true iff A is false in all plausible growing discourses,
$\neg_d A$ is false iff A is true in some plausible growing discourse.

Here, the second clause corresponds to the possibility used in discursive logic. Note here that the plausible growth of discourse implies the increase of information (or knowledge) in view of constructive setting.

Other discursive logical symbols can be read as follows:

$A \wedge_d B$ is true iff A is true in one discourse and B is true in another plausible discourse.
$A \rightarrow_d B$ is true iff if A is true in certain plausible discourse then B is true in a discourse.

The interpretations of $\vee_d$ and $\leftrightarrow_d$ can be obtained by definition. The important point here is that the primitive discursive connective is $\neg_d$.

In our approach, two kinds of negations are used and it is necessary to compare them. $\sim$ is a constructive negation which can express constructive falsity of the proposition, whereas $\neg_d$ is a discursive negation of the proposition with modal flavor, which is similar to intuitionistic negation.

They can express the possibility operator needed in discursive logic as $\sim\neg_d$. Here, $\sim$ behaves as classical-like negation and $\neg_d$ as modal-like negation. We know that in classical modal logic the following holds.

$$\Diamond A \equiv -\Box - A$$

Here, $-$ is classical negation and $\equiv$ is classical equivalence. It is therefore natural to consider two negations in classical-like and modal-like ways.

From the above discussion, *CDLSN* is shown to be a constructive discursive logic which is compatible with Jaśkowski's original ideas. It means that a constructivist can formally perform discursive reasoning.

5 Applications

Constructive discursive logic seems to have many applications for several fields. Although discursive logic was originally motivated in a philosophical tradition, it has the potential to be used for other areas. Here, we take up the so-called *common-sense reasoning* like paraconsistent and non-monotonic reasoning, which are of special importance to knowledge representation in *Artificial Intelligence* (AI).

First, we discuss paraconsistent reasoning which can appear in many real situations. That is, it can extract some conclusions in the presence of contradiction. As is well known, it is obliged to have any arbitrary conclusion from contradiction, if we use the underlying logical basis as classical (or intuitionistic) logic.

However, it is not compatible with our common-sense intuition. Human beings can usually do reasoning in a natural manner, even if they are faced with contradiction. Because human beings have obviously limited memory and reasoning capacity, their knowledge is not always consistent. And contradiction naturally arises in common-sense reasoning.

Paraconsistent logics are useful in such contexts. It is also to be noticed that paraconsistent logic can serve as a foundation for inconsistent (or paraconsistent) mathematics. *CDLSN* can describe paraconsistent reasoning in a logical setting, since it is a paraconsistent logic. Consider the following knowledge base KB_1.

$$KB_1 = \{A, \neg B, A \rightarrow B, B \rightarrow C\}$$

Here, we assume that the base logic is classical logic. From KB_1, we should conclude C using *modus ponens*. But, it is impossible in classical logic, since KB_1 produces inconsistency. In fact, both $\vdash_{KB_1} B$ and $\vdash_{KB_1} \neg B$.

However, in classical logic C, $B \wedge \neg B \vdash_C D$, where D denotes an arbitrary formula. In other words, the knowledge base KB_1 is trivial and it is not of use as a knowledge base in that no useful information is derivable. This fact reveals that classical logic is not suited for reasoning under contradiction.

But, we can derive C in *CDLSN*, as required. The reason is that $B \wedge \sim B \nvdash_{CDLSN} D$. This is a desired feature of common-sense reasoning. Normally, a knowledge base is built from incomplete knowledge due to several reasons. Thus, such a knowledge base may contain some contradictions which need to be tolerated.

Second, we show that *CDLSN* can model non-monotonic reasoning, in which old conclusions can be invalidated by new knowledge. Non-monotonic reasoning is regarded as fundamental in common-sense reasoning. But, standard logics like classical logic are monotonic. Minsky addressed the inadequacy of classical logic as the formalism for describing common-sense reasoning by pointing out that classical logic cannot express non-monotonic reasoning; see Minsky [15].

Based on the observation, Minsky considered that a logic-based approach to AI is not adequate and impossible. If we rely on classical logic as the logic, his consideration may be true. But, we can overcome the difficulty by developing a logic which is not monotonic.

In AI, there is a rich literature on *non-monotonic logics*, formalizing non-monotonic reasoning in a logical setting. For instance, McDermott and Doyle proposed a version of non-monotonic logic by extending classical logic with the consistent operator M; see McDermott and Doyle [14].

Their non-monotonic logic is very similar to modal logic. A formula of the form MA in their non-monotonic logic denotes that A is consistent. Unfortunately, their logic lacks formal semantics as discussed below. Later, McDermott [13] worked out non-monotonic logics based on modal logics, but his attempt was not successful. For example, non-monotonic S5 is shown to be monotonic S5.

We also know other interesting non-monotonic logics like the *default logic* of Reiter [18] and the *autoepistemic logic* of Moore [16]. Default logic is an extension of classical logic with default rules, describing default reasoning, i.e., reasoning by default.

Autoepistemic logic extends classical logic with the belief operator, which models beliefs of a rational agent. Since a rational agent can believe his beliefs and lack of beliefs, reasoning based on his beliefs are non-monotonic according to the increase of new beliefs.

Unfortunately, many non-monotonic logics in AI have been criticized due to the lack of theoretical foundations. This is because most non-monotonic logics rely on meta-rules whose interpretation is outside the scope of object-language. For example, the consistency expressed by McDermott and Doyle's non-monotonic logic needs meta-level reasoning. Namely, M cannot be regarded as a modal operator in modal logic in that it has a reasonable semantics in the standard sense.

Now, we see real examples. Consider the knowledge base KB_2.

$$KB_2 = \{A, A \rightarrow B, C\}$$

We can deduce B from KB_2, written $KB_2 \vdash_C B$, in the framework of classical logic. However, a knowledge base grows with new knowledge. Suppose that new knowledge base $\neg B$ is obtained from KB_2 by adding the new knowledge denoted $\neg B$.

$$KB_3 = \{A, A \rightarrow B, C, \neg B\}$$

Here, the desired reasoning is that $\neg B$ is provable, i.e., $KB_3 \vdash \neg B$. This implies that $KB_3 \nvdash B$, where the old conclusion B is withdrawn in KB_3. Classical logic concludes that $B \wedge \neg B$, i.e., contradiction is provable, however.

A typical example is as follows. Normally birds fly, which can be seen as common-sense. Tweety is a bird. Since all birds can fly, we can conclude that Tweety can fly at this stage. Later, we learn that Tweety is a penguin and all penguins cannot fly. At the stage in which new information is supplied, we naturally infer that Tweety cannot fly. The old conclusion that Tweety can fly is invalidated by the new information concerning Tweety, namely, that he is a penguin.

Non-monotonic reasoning can be formally expressed in *CDLSN*. "Normally if A then B" is described as $A \wedge \sim\neg_d B \rightarrow B$. Note here that $\sim\neg_d$ behaves like M in non-monotonic logic. Although non-monotonic logic requires the interpretation of M in the meta-level in that $\Gamma \vdash \mathrm{M}A$ iff $\Gamma \nvdash \sim A$, where Γ denotes a set of formulas, *CDLSN* dispenses with meta-level features in that $\sim\neg_d$ has the formal interpretation in Kripke semantics.

Paraconsistent and non-monotonic reasoning are closely related. Usually, in common-sense reasoning new knowledge seems to be of importance, yielding non-monotonic reasoning. However, it is not always the case. There appear to be situations in which we cannot give a priority of the old conclusion A to the new conclusion $\neg A$.

For example, assume that both A and $\neg A$ are added to a knowledge base at the same time. The case has no reason to give a priority of A and $\neg A$. We may resolve inconsistency for our purposes. But, we cannot decide whether A or $\neg A$ is appropriate in the situation. There are several ways for decision.

One possible solution is to assume that both A and $\neg A$ hold, which is not a problem in paraconsistent logic. This is because we cannot deduce arbitrary B from $A \wedge \neg A$. This aspect is, however, neglected in non-monotonic reasoning based on classical logic.

6 Concluding Remarks

We proposed a constructive discursive logic *CDLSN* with Hilbert-style axiomatization and Kripke semantics. It can be viewed as a constructive version of Jaśkowski's original system. We established some formal results of *CDLSN* including completeness. We also discussed applications to common-sense reasoning. We believe that *CDLSN* can serve as a logical foundation for paraconsistent intelligent systems.

Finally, we mention topics which remain to be worked out. First, we should extend *CDLSN* with quantifiers for dealing with many interesting problems. There seem to be no difficulties with axiomatization and Kripke semantics.

Second, for practical applications, we need efficient proof methods since a Hilbert system is not suitable. Tableau and sequent calculi are desirable as a proof method. Tableau calculi for N^{-} and N have been worked out in Akama [3], and they can be modified for *CDLSN*.

Third, we should elaborate on the formalization of common-sense reasoning in *CDLSN*. It is interesting to study the connections of *CDLSN* and several non-monotonic logics. Non-monotonic formalisms are also related to logic programming, and we should explore relationships in this context.

Acknowledgments We are grateful to the referees for their useful comments. Thanks also to Prof. John Fulcher for his suggestions in the final version of this chapter.

References

1. Akama, S.: Constructive predicate logic with strong negation and model theory. Notre Dame J. Form. Logic **29**, 18–27 (1988)
2. Akama, S.: Subformula semantics for strong negation systems. J. Philos. Logic **19**, 217–226 (1990)
3. Akama, S.: Nelson's paraconsistent logic. Logic and Logical Philos. **7**, 101–115 (1999)
4. Akama, S.: Discursive reasoning in a constructive setting. Int. J. Reason.-Based Intell. Syst. **3**, 88–93 (2011)
5. Akama, S., Abe, J.M., Nakamatsu, K.: Constructive discursive logic with strong negation. Logique et Analyse **215**, 395–408 (2011)
6. Akama, S., Nakamatsu, K., Abe, J.M.: Common-sense reasoning in constructive discursive logic. In: Proceedings of ICCM2012, pp. 596–601, Seoul, South Korea (2012)

7. Almukdad, A., Nelson, D.: Constructible falsity and inexact predicates. J. Symb. Logic **49**, 231–233 (1984)
8. da Costa, N.C.A., Dubikajtis, L.: On Jaśkowski's discursive logic. In: Arruda, A.I., da Costa, N.C.A., Chuaqui, R. (eds.) Non-Classical Logics, Model Theory and Computability, pp. 37–56. North-Holland, Amsterdam (1977)
9. da Costa, N.C.A., Doria, F.A.: On Jaśkowski's discursive logics. Stud. Logica. **54**, 33–60 (1995)
10. Jaśkowski, S.: Propositional calculus for contradictory deductive systems (in Polish). Studia Societatis Scientiarun Torunesis, Sectio A **1**, 55–77 (1948)
11. Jaśkowski, S.: On the discursive conjunction in the propositional calculus for inconsistent deductive systems (in Polish). Studia Societatis Scientiarun Torunesis, Sectio A **8**, 171–172 (1949)
12. Kotas, J.: The axiomatization of S. Jaśkowski's discursive logic, Studia Logica **33**, 195–200 (1974)
13. McDermott, D.: Nonmonotonic logic II. J. ACM **29**, 33–57 (1982)
14. McDermott, D., Doyle, J.: Non-monotonic logic I. Artif. Intell. **13**, 41–72 (1980)
15. Minsky, M.: A framework for representing knowledge. Haugeland, J. (ed.) Mind-Design, pp. 95–128, MIT Press, Cambridge (1975)
16. Moore, R.: Semantical considerations on nonmonotonic logic. Artif. Intell. **25**, 75–94 (1985)
17. Nelson, D.: Constructible falsity. J. Symb. Logic **14**, 16–26 (1949)
18. Reiter, R.: A logic for default reasoning. Artif. Intell. **13**, 81–132 (1980)
19. Thomason, R.H.: Semantical study of constructible falsity. Zeitschrift für mathematische Logik und Grundlagen der Mathematik **15**, 247–257 (1969)
20. Vorob'ev, N.N.: A constructive propositional calculus with strong negation (in Russian). Dokl. Akad. Nauk SSSR **85**, 465–468 (1952)

Paraconsistent Annotated Logic Program EVALPSN and Its Applications

Kazumi Nakamatsu, Jair Minoro Abe and Seiki Akama

Abstract We have already proposed a paraconsistent annotated logic program called EVALPSN. In EVALPSN, an annotation called an extended vector annotation is attached to each literal. In order to deal with before-after relation between two time intervals, we also have introduced a new interpretation for extended vector annotations in EVALPSN, which is named before-after(bf)-EVALPSN. In this chapter, we review paraconsistent annotated logic programs EVALPSN/bf-EVALPSN from the view point of application to safety verification and process order control with simple examples. First, the background and overview of EVALPSN are introduced, and paraconsistent annotated logics $\mathcal{PT}$ and the basic annotated logic program are recapitulated as the formal background of EVALPSN with some simple examples. Then, bf-EVALPSN is formally defined, how to implement and apply bf-EVALPSN to real-time intelligent process order control and its safety verification with simple practical examples. Last, unique and useful features of bf-EVALPSN called transitive bf-inference rules are introduced, and this chapter is concluded with some remarks.

K. Nakamatsu (✉)
School of Human Science and Environment, University of Hyogo, 1-1-12 Shinzaike, Himeji 670-0092, Japan
e-mail: nakamatu@shse.u-hyogo.ac.jp

J.M. Abe
Graduate Program in PE, ICET, Paulista University, R. Dr. Bacelar, 1212, São Paulo 04026-002, Brazil
e-mail: jairabe@uol.com.br

J.M. Abe
Inst. Advanced Studies, University of São Paulo, Travessa J, 374 Terreo, Cidade Universitaria, São Paulo 05508-900, Brazil

S. Akama
C-Republic, 1-20-1, Higashi-Yurigaoka, Asao-Ku, Kawasaki 215-0012, Japan
e-mail: akama@jcom.home.ne.jp

J.M. Abe (ed.), *Paraconsistent Intelligent-Based Systems*,
Intelligent Systems Reference Library 94, DOI 10.1007/978-3-319-19722-7_3

1 Introduction and Background

One of the main purposes of paraconsistent logic is to deal with inconsistency in a framework of consistent logical systems. It has been almost six decades since the first paraconsistent logical system was proposed by Jaskowski [12]. It was four decades later that a family of paraconsistent logic called “annotated logic” was proposed by da Costa et al. [8, 47], which can deal with inconsistency by introducing many truth values called “annotations” into their syntax as attached information to formulas.

The paraconsistent annotated logic by da Costa et al.was developed from the viewpoint of logic programming by Subrahmanian et al. [7, 13, 46]. Furthermore, in order to deal with inconsistency and non-monotonic reasoning in a framework of annotated logic programming, ALPSN (Annotated Logic Program with Strong Negation) and its stable model semantics was developed by Nakamatsu and Suzuki [16]. It has been shown that ALPSN can deal with some non-monotonic reasonings such as default logic [44], autoepistemic logic [15] and a non-monotonic Assumption Based Truth Maintenance System(ATMS) [9] in a framework of annotated logic programming [17, 35, 36]. Even though ALPSN can deal with non-monotonic reasoning such as default reasoning and conflicts can be represented as paraconsistent knowledge in it, it is difficult and complicated to deal with reasoning to resolve conflicts in ALPSN. On the other hands, it is known that defeasible logic can deal with conflict resolving in a logical way [5, 39, 40], although defeasible logic cannot deal with inconsistency in its syntax and its inference rules are too complicated to be implemented easily. In order to deal with conflict resolving and inconsistency in a framework of annotated logic programming, a new version of ALPSN, VALPSN (Vector Annotated Logic Program with Strong Negation) that can deal with defeasible reasoning and inconsistency was also developed by Nakamatsu et al. [21]. Moreover, it has been shown that VALPSN can be applied to conflict resolving in various systems [18–20]. It also has been shown that VALPSN provides a computational model of defeasible logic [5, 6]. Later, VALPSN was extended to EVALPSN (Extended VALPSN) by Nakamatsu et al. [22, 23] to deal with deontic notions (obligation, permission, forbiddance, etc.) and defeasible deontic reasoning [41, 42]. Recently, EVALPSN has been applied to various kinds of safety verification and intelligent control, for example, railway interlocking safety verification [26], robot action control [24, 27, 28, 37], safety verification for air traffic control [25], traffic signal control [29], discrete event control [30–32] and pipeline valve control [33, 34].

Considering the safety verification for process control, there is an occasion in which the safety verification for process order control is significant. For example, suppose a pipeline network in which two kinds of liquids, nitric acid and caustic soda are used for cleaning the pipelines. If those liquids are processed continuously and mixed in the same pipeline by accident, explosion by neutralization would be caused. In order to avoid such a dangerous accident, the safety for process order control should be strictly verified in a formal way such as EVALPSN. However, it seems to be a little difficult to utilize EVALPSN for verifying process order control as well as the safety verification for each process in process control. We have already proposed a new EVALPSN called bf(before-after)-EVALPSN that can deal with before-after relations between two time intervals [38].

This chapter mainly focuses on introducing bf-EVALPSN and its application to real-time process order control and its safety verification with simple process order control examples. As far as we know there seems to be no other efficient computational tool that can deal with the real-time safety verification for process order control than bf-EVALPSN.

This chapter is organized as follows: firstly, in Sect. 1, the background and overview of the paraconsistent annotated logic program EVALPSN are reviewed; in Sect. 2, paraconsistent annotated logics and their logic programming as the background knowledge of EVALPSN/bf-EVALPSN and EVALPSN itself are formally recapitulated with simple examples; in Sect. 3, the traffic signal control based on EVALPSN deontic defeasible reasoning and simple simulation results by the cellular automaton method are provided as an application of EVALPSN to intelligent control. in Sect. 4, the basic concepts of bf-EVALPSN are introduced and bf-EVALPSN is formally defined, furthermore, an application of bf-EVALPSN to real-time safety verification for process order control is described with simple practical examples; in Sect. 5, reasoning of before-after relations in bf-EVALPSN is reviewed and a unique and useful inference method of before-after relations in bf-EVALPSN, which can be implemented as a bf-EVALPSN called "transitive bf-inference rules", is introduced with a simple example; lastly, conclusions and remarks are provided.

2 Paraconsistent Annotated Logic Program

This section is devoted to clarify the formal background of the paraconsistent annotated logic program EVALPSN. The more details of EVALPSN has been introduced in [38]. We assume that the reader is familiar with the basic knowledge of classical logic and logic programming [14]. In order to understand EVLPSN and its reasoning we introduce Paraconsistent Annotated Logics $P\mathcal{T}$ [8] in the following subsection.

2.1 Paraconsistent Annotated Logic $P\mathcal{T}$

Here we briefly recapitulate the syntax and semantics for propositional paraconsistent annotated logics $P\mathcal{T}$ proposed by da Costa et al. [8].

Generally, a truth value called an *annotation* is attached to each atomic formula explicitly in paraconsistent annotated logic, and the set of annotations constitutes a complete lattice. We introduce a paraconsistent annotated logic $P\mathcal{T}$ with the four valued complete lattice $\mathcal{T}$.

Definition 2.1 The primitive symbols of $P\mathcal{T}$ are:

1. propositional symbols $p, q, \ldots, p_i, q_i, \ldots$;
2. each member of $\mathcal{T}$ is an *annotation constant* (we may call it simply an annotation);
3. the connectives and parentheses $\wedge$, $\vee$, $\rightarrow$, $\neg$, (,).

Formulas are defined recursively as follows:

1. if p is a propositional symbol and $\mu \in \mathcal{T}$ is an annotation constant, then $p : \mu$ is an *annotated atomic formula (atom)*;
2. if F, F_1, F_2 are formulas, then $\neg F, F_1 \wedge F_2, F_1 \vee F_2, F_1 \rightarrow F_2$ are formulas.

We suppose that the four-valued lattice in Fig. 1 is the complete lattice $\mathcal{T}$, where annotations $\mathtt{t}$ and $\mathtt{f}$ may be intuitively regarded as truth values *true* and *false*, respectively. It may be comprehensible that annotations $\perp, \mathtt{t}, \mathtt{f}$ and $\top$ correspond to the truth values *, *T*, *F* and *TF* in Visser [48] and **None, T, F**, and **Both** in Belnap [4], respectively. Moreover, the complete lattice $\mathcal{T}$ can be viewed as a bi-lattice in which the vertical direction $\overleftarrow{\perp\top}$ indicates *knowledge amount* ordering and the horizontal direction $\overleftarrow{\mathtt{ft}}$ does *truth* ordering [10]. We use the symbol $\leq$ to denote the ordering in terms of knowledge amount (the vertical direction $\overleftarrow{\perp\top}$) over the complete lattice $\mathcal{T}$, and the symbols $\perp$ and $\top$ are used to denote the bottom and top elements, respectively. In the paraconsistent annotated logic $P\mathcal{T}$, each annotated atomic formula can be interpreted epistemically, for example, $p : \mathtt{t}$ may be interpreted epistemically as "the proposition p is known to be true".

There are two kinds of negation in the paraconsistent annotated logic $P\mathcal{T}$, one of them is called *epistemic negation* and represented by the symbol $\neg$ (see Definition 2.1). The epistemic negation in $P\mathcal{T}$ followed by an annotated atomic formula is defined as a mapping between elements of the complete lattice $\mathcal{T}$ as follows:

$$\neg(\perp) = \perp, \quad \neg(\mathtt{t}) = \mathtt{f}, \quad \neg(\mathtt{f}) = \mathtt{t}, \quad \neg\top) = \top.$$

As shown in the above mapping the epistemic negation maps annotations to themselves without changing the knowledge amounts of annotations. Furthermore, the epistemic negation followed by an annotated atomic formula can be eliminated by the mapping. For example, the knowledge amount of annotation $\mathtt{t}$ is the same as that of annotation $\mathtt{f}$ as shown in the complete lattice $\mathcal{T}$, and we have the epistemic negation,[1]

$$\neg(p : \mathtt{t}) = p : \neg(\mathtt{t}) = p : \mathtt{f},$$

which shows that the knowledge amount in terms of the proposition p cannot be changed by the epistemic negation mapping. There is another negation called *ontological*(*strong*) *negation* that is defined by using the epistemic negation.

[1]An expression $\neg p : \mu$ is conveniently used for expressing a negative annotated literal instead of $\neg(p : \mu)$ or $p : \neg(\mu)$.

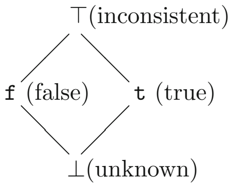


Fig. 1 The 4-valued complete lattice $\mathcal{T}$

Definition 2.2 (*Strong Negation*) Let F be any formula,

$$\sim F =_{def} F \to ((F \to F) \wedge \neg(F \to F)).$$

The epistemic negation in Definition 2.2 is not interpreted as a mapping between annotations since it is not followed by an annotated atomic formula. Therefore, the strongly negated formula $\sim F$ can be interpreted so that if the formula F exists, the contradiction $((F \to F) \wedge \neg(F \to F))$ is implied. Usually, the strong negation is used for denying the existence of a formula following it.

The semantics for the paraconsistent annotated logics $P\mathcal{T}$ is defined.

Definition 2.3 Let v be the set of all propositional symbols and $\mathcal{F}$ be the set of all formulas. An interpretation I is a function,

$$I : v \to \mathcal{T}.$$

To each interpretation I, we can associate the valuation function such that

$$v_I : \mathcal{F} \to \{0, 1\},$$

which is defined as :

1. let p be a propositional symbol and μ an annotation,

$$v_I(p : \mu) = 1 \textbf{ iff } \mu \leq I(p),$$
$$v_I(p : \mu) = 0 \textbf{ iff } \mu \not\leq I(p);$$

2. let A and B be any formulas, and A not an annotated atom,

$$v_I(\neg A) = 1 \textbf{ iff } v_I(A) = 0,$$
$$v_I(\sim B) = 1 \textbf{ iff } v_I(B) = 0;$$

other formulas $A \to B$, $A \wedge B$, $A \vee B$ are valuated as usual.

We provide an intuitive interpretation for strongly negated annotated atoms with the complete lattice $\mathcal{T}$. For example, the strongly negated literal $\sim (p : \texttt{t})$ implies the knowledge "p is false($\texttt{f}$) or unknown($\perp$)" since it denies the existence of the knowledge that "p is true(t)". This intuitive interpretation is proofed by Definition 2.3 as follows: if $v_I(\sim(p:\mathtt{t}))=1$, we have $v_I(p:\mathtt{t})=0$ and for any annotation $\mu\in\{\bot,\mathtt{f},\mathtt{t},\top\}\leq\mathtt{t}$, we have $v_I(p:\mu)=1$, therefore, we obtain that $\mu=\mathtt{f}$ or $\mu=\bot$.

2.2 EVALPSN (Extended Vector Annotated Logic Program with Strong Negation)

Generally, an annotation is explicitly attached to each literal in paraconsistent annotated logic programs as well as the paraconsistent annotated logic $\mathcal{PT}$. For example, let p be a literal, μ an annotation, then $p:\mu$ is called an *annotated literal*. The set of annotations constitutes a complete lattice.

An annotation in EVALPSN has a form of $[(i,j),\mu]$ called an *extended vector annotation*. The first component (i,j) is called a *vector annotation* and the set of vector annotations, which constitutes a complete lattice,

$$\mathcal{T}_v(n)=\{(x,y)|0\leq x\leq n, 0\leq y\leq n, x, y \text{ and } n \text{ are integers}\}$$

shown by the Hasse's diagram as $n=2$ in Fig. 2. The ordering $(\preccurlyeq_v)$ of the complete lattice $\mathcal{T}_v(n)$ is defined as follows: let (x_1,y_1), $(x_2,y_2)\in\mathcal{T}_v(n)$,

$$(x_1,y_1)\preccurlyeq_v(x_2,y_2) \text{ iff } x_1\leq x_2 \text{ and } y_1\leq y_2.$$

For each extended vector annotated literal $p:[(i,j),\mu]$, the integer i denotes the amount of positive information to support the literal p and the integer j denotes that of negative one. The second component μ is an index of fact and deontic notions such as obligation, and the set of the second components constitutes the following complete lattice,

$$\mathcal{T}_d=\{\bot,\alpha,\beta,\gamma,*_1,*_2,*_3,\top\}.$$

The ordering $(\preccurlyeq_d)$ of the complete lattice $\mathcal{T}_d$ is described by the Hasse's diagram in Fig. 2. The intuitive meaning of each member in $\mathcal{T}_d$ is

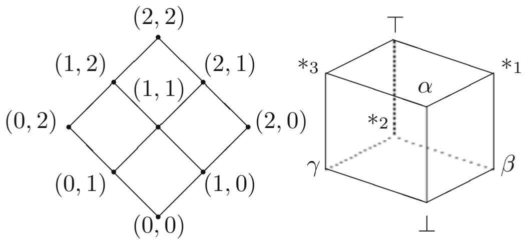


Fig. 2 Lattice $\mathcal{T}_v(2)$ and Lattice $\mathcal{T}_d$

$$\begin{aligned}
&\bot(\text{unknown}),\\
&\alpha(\text{fact}),\quad \beta(\text{obligation}),\quad \gamma(\text{non-obligation}),\\
&*_1\ (\text{fact and obligation}),\\
&*_2\ (\text{obligation and non-obligation}),\\
&*_3\ (\text{fact and non-obligation}),\quad \text{and}\\
&\top(\text{inconsistency}).
\end{aligned}$$

The complete lattice $\mathcal{T}_d$ is a quatro-lattice in which the direction $\overrightarrow{\bot\top}$ measures *knowledge amount*, the direction $\overrightarrow{\gamma\beta}$ does *deontic truth*, the direction $\overrightarrow{\bot *_2}$ does *deontic knowledge amount* and the direction $\overrightarrow{\bot\alpha}$ does *factuality*. For example, annotation β (obligation) can be intuitively interpreted to be more obligatory than annotation γ (non-obligation), and annotations $\bot$ (no knowledge) and $*_2$(obligation and non-obligation) are deontically neutral, that is to say, it cannot be said whether they represent obligation or non-obligation.

The complete lattice $\mathcal{T}_e(n)$ of extended vector annotations is defined as the product,

$$\mathcal{T}_v(n) \times \mathcal{T}_d.$$

The ordering ($\preccurlyeq_e$) of the complete lattice $\mathcal{T}_e(n)$ is also defined as follows: let $[(i_1,j_1),\mu_1], [(i_2,j_2),\mu_2] \in \mathcal{T}_e$,

$$[(i_1,j_1),\mu_1] \preccurlyeq_e [(i_2,j_2),\mu_2] \quad \text{iff} \quad (i_1,j_1) \preccurlyeq_v (i_2,j_2) \quad \text{and} \quad \mu_1 \preccurlyeq_d \mu_2.$$

There are two kinds of epistemic negation ($\neg_1$ and $\neg_2$) in EVALPSN, which are defined as mappings over the complete lattices $\mathcal{T}_v(n)$ and $\mathcal{T}_d$, respectively.

Definition 2.4 (*epistemic negations $\neg_1$ and $\neg_2$ in EVALPSN*)

$$\begin{aligned}
&\neg_1([(i,j),\mu]) = [(j,i),\mu], \quad \forall \mu \in \mathcal{T}_d\\
&\neg_2([(i,j),\bot]) = [(i,j),\bot], \quad \neg_2([(i,j),\alpha]) = [(i,j),\alpha],\\
&\neg_2([(i,j),\beta]) = [(i,j),\gamma], \quad \neg_2([(i,j),\gamma]) = [(i,j),\beta],\\
&\neg_2([(i,j),*_1]) = [(i,j),*_3], \quad \neg_2([(i,j),*_2]) = [(i,j),*_2],\\
&\neg_2([(i,j),*_3]) = [(i,j),*_1], \quad \neg_2([(i,j),\top]) = [(i,j),\top].
\end{aligned}$$

If we regard the epistemic negations in Definition 2.4 as syntactical operations, an epistemic negation followed by a literal can be eliminated by the syntactical operation. For example, $\neg_1 p : [(2,0),\alpha] = p : [(0,2),\alpha]$ and $\neg_2 q : [(1,0),\beta] = p : [(1,0),\gamma]$. The strong negation ($\sim$) in EVALPSN is defined as well as the paraconsistent annotated logic $P\mathcal{T}$.

Definition 2.5 (*well extended vector annotated literal*) Let p be a literal. $p : [(i,0),\mu]$ and $p : [(0,j),\mu]$ are called *weva*(*well extended vector annotated*)- *literals*, where $i, j \in \{1, 2, \ldots, n\}$, and $\mu \in \{\ \alpha,\ \beta,\ \gamma\ \}$.

Defintion 2.6 (*EVALPSN*) If $L_0, \ldots, L_n$ are weva-literals,

$$L_1 \wedge \cdots \wedge L_i \wedge \sim L_{i+1} \wedge \cdots \wedge \sim L_n \rightarrow L_0$$

is called an *EVALPSN clause*. An *EVALPSN* is a finite set of EVALPSN clauses.

Fact and deontic notions, "obligation", "forbiddance" and "permission" are represented by extended vector annotations,

$$[(m,0),\alpha],\ [(m,0),\beta],\ [(0,m),\beta],\ \text{and}\ [(0,m),\gamma],$$

respectively, where m is a positive integer. For example,

$p : [(2,0),\alpha]$ is intuitively interpreted as "it is known to be true of strength 2 that p is a fact";

$p : [(1,0),\beta]$ is as "it is known to be true of strength 1 that p is obligatory";

$p : [(0,2),\beta]$ is as "it is known to be false of strength 2 that p is obligatory", that is to say, "it is known to be true of strength 2 that p is forbidden";

$p : [(0,1),\gamma]$ is as "it is known to be false of strength 1 that p is not obligatory", that is to say, "it is known to be true of strength 1 that p is permitted".

Generally, if an EVALPSN contains the strong negation $\sim$, it has stable model semantics [38] as well as other ordinary logic programs with strong negation. However, the stable model semantics may have a problem that some programs may have more than two stable models and others have no stable model. Moreover, computation of stable models takes a long time compared to usual logic programming such as PROLOG programming. Therefore, it does not seem to be so appropriate for practical application such as real time processing in general. However, we fortunately have cases to implement EVALPSN practically, if an EVALPSN is a *stratified* program, it has a tractable model called a *perfect model* [43] and the strong negation in the EVALPSN can be treated as the *Negation as Failure* in logic programming with no strong negation. The details of stratified program and some tractable models for normal logic programs can be found in [3, 11, 43, 45], furthermore the details of the stratified EVALPSN are described in [38]. Therefore, inefficient EVALPSN stable model computation does not have to be taken into account in this chapter since all EVALPSNs that will appear in the subsequent sections are stratified.

3 Traffic Signal Control in EVALPSN

3.1 Deontic Defeasible Traffic Signal Control

Traffic jam caused by inappropriate traffic signal control is a serious issue that should be resolved. In this section, we introduce an intelligent traffic signal control system based on EVALPSN defeasible deontic reasoning, which may provide one solution for traffic jam reduction. We show how the traffic signal control is implemented in EVALPSN with taking a simple intersection example in Japan.

We suppose an intersection in which two roads are crossing described in Fig. 3 as an example for implementing the traffic signal control method based on EVALPSN.[2] The intersection has four traffic lights $T_{1,2,3,4}$, which indicate four kinds of signals, green, yellow, red and right-turn arrow. Each lane connected to the intersection has a sensor to detect traffic amount. Each sensor is described by symbols $S_i(1 \le i \le 8)$ in Fig. 3. For example, the sensor S_6 detects the right-turn traffic amount confronting traffic light T_1. Basically, the traffic signal control is performed based on the traffic amount detected by the sensors. The chain of signaling is supposed as follows:

$$\rightarrow \text{red} \rightarrow \text{green} \rightarrow \text{yellow} \rightarrow \text{right arrow} \rightarrow \text{red} \rightarrow .$$

For simplicity, we assume that the durations of yellow and right arrow signals are constant, and if traffic lights $T_{1,2}(T_{3,4})$ are green or right arrow, traffic lights $T_{3,4}(T_{1,2})$ are red as follows:

$$\begin{aligned}
&\text{Signal cycle of traffic lights } T_{1,2}\\
&\qquad\qquad \rightarrow \text{red} \rightarrow \text{red} \rightarrow \text{green} \rightarrow \text{right arrow} \rightarrow \text{red} \rightarrow,\\
&\text{Signal cycle of traffic lights } T_{3,4}\\
&\qquad\qquad \rightarrow \text{green} \rightarrow \text{right arrow} \rightarrow \text{red} \rightarrow \text{red} \rightarrow \text{green} \rightarrow .
\end{aligned}$$

Only the turns green to right arrow and right arrow to red are controlled. The turn red to green of the front traffic signal follows the turn right arrow to red of the neighbor one. Moreover, the signaling is controlled at each unit time $t \in \{0, 1, 2, \ldots, n\}$. The traffic amount of each lane can be regarded as permission or forbiddance from turning such as green to right arrow. For example, if there are many cars waiting for traffic lights $T_{1,2}$ turning red to green, it can be regarded as permission for turning the crossing traffic lights $T_{3,4}$ green to right arrow, yellow and red. On the other hand, if there are many cars passing through the intersection with traffic lights $T_{3,4}$ signaling green, it can be regarded as forbiddance from turning traffic lights $T_{3,4}$ green to right arrow. Then, there is a conflict between those permission and forbiddance in terms of the same traffic lights $T_{3,4}$.

[2]The intersection is supposed to be in Japan where we need to keep left if driving a car.

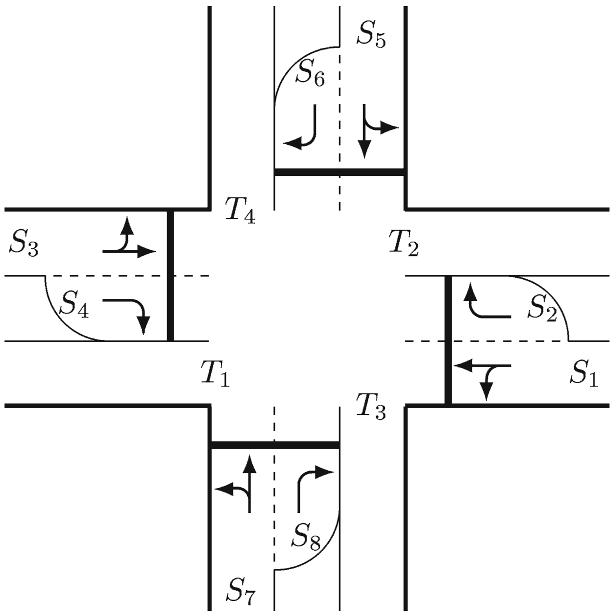


Fig. 3 Intersection

We formalize such a conflict resolving in EVALPSN. We assume that the minimum and maximum durations of green signal are previously given for all traffic lights, and the duration of green signal must be controlled between the minimum and maximum durations. We consider the following four states of traffic lights $T_{1,2,3,4}$,

state 1: traffic lights $T_{1,2}$ are red and traffic lights $T_{3,4}$ are green,
state 2: traffic lights $T_{1,2}$ are red and traffic lights $T_{3,4}$ are right arrow,
state 3: traffic lights $T_{1,2}$ are green and traffic lights $T_{3,4}$ are red,
state 4: traffic lights $T_{1,2}$ are right arrow and traffic lights $T_{3,4}$ are red.

Here we take the transit from the state 1 to the state 2 into account to introduce the traffic signal control properties in the state 1 and its translation into EVALPSN. The traffic signal control consists of the traffic signal control properties for the state transit the state 1 to the state 2, green light length rules, and deontic defeasible reasoning rules for traffic signal control.

We use the following EVALP literals:

- $S_i(t) : [(2,0), \alpha]$ can be informally interpreted as the traffic sensor $S_i (i = 1, 2, \ldots, 8)$ has detected traffic at time t.
- $T_{m,n}(c,t) : [(2,0), \alpha]$ can be informally interpreted as the traffic light $T_{m,n}$ indicates a signal color C at time t, where $m, n = 1, 2, 3, 4$ and c is one of signal colors green(g), red(r), or right arrow(a).

- $MIN_{m,n}(g,t) : [(2,0),\alpha]$ can be informally interpreted as the green duration of traffic lights $T_{m,n}(m,n = 1,2,3,4)$ is shorter than its minimum green duration at time t.
- $MAX_{m,n}(g,t) : [(2,0),\alpha]$ can be informally interpreted as the green duration of traffic lights $T_{m,n}$ is longer than its maximum green duration at time t.
- $T_{m,n}(c,t) : [(0,k),\gamma]$ which can be informally interpreted as it is permitted for traffic lights $T_{m,n}$ to indicate signal color C at time t, where $m,n = 1,2,3,4$ and c is one of the signal colors green(g), red(r), or right arrow(a); if $k = 1$, the permission is weak, and if $k = 2$, the permission is strong.
- $T_{m,n}(c,t) : [(0,k),\beta]$ can be informally interpreted as it is forbidden for traffic lights $T_{m,n}$ from indicating the signal color C at time t, where $m,n = 1,2,3,4$ and c is one of the signal colors green(g), red(r), or right arrow(a); if $k = 1$, the forbiddance is weak, and if $k = 2$, the forbiddance is strong.

[Traffic Signal Control Properties in State 1]

1 If traffic sensor S_1 detects traffic amount, it has already passed the minimum green duration of traffic lights $T_{3,4}$, and neither traffic sensors S_5 nor S_7 detect traffic amount at time t, then it is weakly permitted for traffic lights $T_{3,4}$ to turn green to right arrow at time t; which is translated into the EVALPSN,

$$\begin{aligned}
&S_1(t) : [(2,0),\alpha] \wedge \\
&T_{1,2}(r,t) : [(2,0),\alpha] \wedge T_{3,4}(g,t) : [(2,0),\alpha] \wedge \\
&\sim MIN_{3,4}(g,t) : [(2,0),\alpha] \wedge \\
&\sim S_5(t) : [(2,0),\alpha] \wedge \sim S_7(t) : [(2,0),\alpha] \\
&\rightarrow T_{3,4}(a,t) : [(0,1),\gamma].
\end{aligned} \tag{1}$$

2 If traffic sensor S_3 detects traffic amount, it has already passed the minimum green duration of traffic lights $T_{3,4}$, and neither traffic sensors S_5 nor S_7 detect traffic amount at time t, then it is weakly permitted for traffic lights $T_{3,4}$ to turn green to right arrow at time t; which is translated into the EVALPSN,

$$\begin{aligned}
&S_3(t) : [(2,0),\alpha] \wedge \\
&T_{1,2}(r,t) : [(2,0),\alpha] \wedge T_{3,4}(g,t) : [(2,0),\alpha] \wedge \\
&\sim MIN_{3,4}(g,t) : [(2,0),\alpha] \wedge \\
&\sim S_5(t) : [(2,0),\alpha] \wedge \sim S_7(t) : [(2,0),\alpha] \\
&\rightarrow T_{3,4}(a,t) : [(0,1),\gamma].
\end{aligned} \tag{2}$$

3 If traffic sensor S_2 detects traffic amount, it has already passed the minimum green duration of traffic lights $T_{3,4}$, and neither traffic sensors S_5 nor S_7 detect traffic amount at time t, then it is weakly permitted for traffic lights $T_{3,4}$ to turn green to right arrow at time t, which is translated into the EVALPSN,

$$\begin{aligned}
&S_2(t) : [(2,0),\alpha] \wedge \\
&T_{1,2}(r,t) : [(2,0),\alpha] \wedge T_{3,4}(g,t) : [(2,0),\alpha] \wedge \\
&\sim MIN_{3,4}(g,t) : [(2,0),\alpha] \wedge \\
&\sim S_5(t) : [(2,0),\alpha] \wedge \sim S_7(t) : [(2,0),\alpha] \\
&\rightarrow T_{3,4}(a,t) : [(0,1),\gamma],
\end{aligned} \tag{3}$$

4 If traffic sensor S_4 detects traffic amount, it has already passed the minimum green duration of traffic lights $T_{3,4}$, and neither traffic sensors S_5 nor S_7 detect traffic amount at time t, then it is weakly permitted for traffic lights $T_{3,4}$ to turn green to right arrow at time t; which is translated into the EVALPSN,

$$\begin{aligned}
&S_4(t) : [(2,0),\alpha] \wedge \\
&T_{1,2}(r,t) : [(2,0),\alpha] \wedge T_{3,4}(g,t) : [(2,0),\alpha] \wedge \\
&\sim MIN_{3,4}(g,t) : [(2,0),\alpha] \wedge \\
&\sim S_5(t) : [(2,0),\alpha] \wedge \sim S_7(t) : [(2,0),\alpha] \\
&\rightarrow T_{3,4}(a,t) : [(0,1),\gamma],
\end{aligned} \tag{4}$$

5 If traffic sensor S_6 detects traffic amount, it has already passed the minimum green duration of traffic lights $T_{3,4}$, and neither traffic sensors S_5 nor S_7 detect traffic amount at time t, then it is weakly permitted for traffic lights $T_{3,4}$ to turn green to right arrow at time t; which is translated into the EVALPSN,

$$\begin{aligned}
&S_6(t) : [(2,0),\alpha] \wedge \\
&T_{1,2}(r,t) : [(2,0),\alpha] \wedge T_{3,4}(g,t) : [(2,0),\alpha] \wedge \\
&\sim MIN_{3,4}(g,t) : [(2,0),\alpha] \wedge \\
&\sim S_5(t) : [(2,0),\alpha] \wedge \sim S_7(t) : [(2,0),\alpha] \\
&\rightarrow T_{3,4}(a,t) : [(0,1),\gamma],
\end{aligned} \tag{5}$$

6 If traffic sensor S_8 detects traffic amount, it has already passed the minimum green duration of traffic lights $T_{3,4}$, and neither traffic sensors S_5 nor S_7 detect traffic amount at time t, then it is weakly permitted for traffic lights $T_{3,4}$ to turn green to right arrow at time t; which is translated into the EVALPSN,

$$\begin{aligned}
&S_6(t) : [(2,0),\alpha] \wedge \\
&T_{1,2}(r,t) : [(2,0),\alpha] \wedge T_{3,4}(g,t) : [(2,0),\alpha] \wedge \\
&\sim MIN_{3,4}(g,t) : [(2,0),\alpha] \wedge \\
&\sim S_5(t) : [(2,0),\alpha] \wedge \sim S_7(t) : [(2,0),\alpha] \\
&\rightarrow T_{3,4}(a,t) : [(0,1),\gamma],
\end{aligned} \tag{6}$$

7 If traffic sensor S_5 detects traffic amount and it has not passed the maximum green duration of traffic lights $T_{3,4}$ yet, then it is weakly forbidden for traffic lights $T_{3,4}$ to turn green to right arrow at time t; which is translated into the EVALPSN,

$$\begin{aligned}&S_5(t):[(2,0),\alpha]\wedge\\&T_{1,2}(r,t):[(2,0),\alpha]\wedge T_{3,4}(g,t):[(2,0),\alpha]\wedge\\&\sim MAX_{3,4}(g,t):[(2,0),\alpha]\\&\ \to T_{3,4}(a,t):[(0,1),\beta],\end{aligned}\tag{7}$$

8 If traffic sensor S_7 detects traffic amount and it has not passed the maximum green duration of traffic lights $T_{3,4}$, then it is weakly forbidden for traffic lights $T_{3,4}$ to turn green to right arrow at time t; which is translated into the EVALPSN,

$$\begin{aligned}&S_7(t):[(2,0),\alpha]\wedge\\&T_{1,2}(r,t):[(2,0),\alpha]\wedge T_{3,4}(g,t):[(2,0),\alpha]\wedge\\&\sim MAX_{3,4}(g,t):[(2,0),\alpha]\\&\ \to T_{3,4}(a,t):[(0,1),\beta],\end{aligned}\tag{8}$$

[Green light length rules for the traffic lights $T_{3,4}$]

9 If traffic lights $T_{3,4}$ are green and it has not passed the minimum duration of them yet, then it is strongly forbidden for traffic lights $T_{3,4}$ to turn green to right arrow at time t; which is translated into the EVALPSN,

$$\begin{aligned}&T_{3,4}(g,t):[(2,0),\alpha]\wedge MIN_{3,4}(g,t):[(2,0),\alpha]\\&\ \to T_{3,4}(a,t):[(0,2),\beta],\end{aligned}\tag{9}$$

10 If traffic lights $T_{3,4}$ are green and it has already passed the maximum duration of them, then it is strongly permitted for traffic lights $T_{3,4}$ to turn green to right arrow at time t; which is translated into the EVALPSN,

$$\begin{aligned}&T_{3,4}(g,t):[(2,0),\alpha]\wedge MAX_{3,4}(g,t):[(2,0),\alpha]\\&\ \to T_{3,4}(a,t):[(0,2),\gamma],\end{aligned}\tag{10}$$

[Deontic deasible reasoning rules]

11 If traffic lights $T_{3,4}$ are green, it is weakly permitted at least for traffic lights $T_{3,4}$ to turn green to right arrow at time t, then it is strongly obligatory for traffic lights $T_{3,4}$ to turn green to right arrow at time $t+1$ (at the next step); which is translated into the EVALPSN,

$$T_{3,4}(g,t):[(2,0),\alpha] \wedge T_{3,4}(a,t):[(0,1),\gamma] \\ \rightarrow T_{3,4}(a,t+1):[(2,0),\beta], \tag{11}$$

12 If traffic lights $T_{3,4}$ are green, it is weakly forbidden at least for traffic lights $T_{3,4}$ to turn green to right arrow at time t, then it is strongly obligatory for traffic lights $T_{3,4}$ not to turn green to right arrow at time $t+1$ (at the next step); which is translated into the EVALPSN,

$$T_{3,4}(g,t):[(2,0),\alpha] \wedge T_{3,4}(a,t):[(0,1),\beta] \\ \rightarrow T_{3,4}(g,t+1):[(2,0),\beta]. \tag{12}$$

3.2 Example and Simulation

Let us introduce a simple example of the EVALPSN based traffic signal control. We assume the same intersection in the previous section.

Example 1 Suppose that traffic lights $T_{1,2}$ are red and traffic lights $T_{3,4}$ are green. We also suppose that the minimum duration of green signal has already passed but the maximum one has not passed yet. Then, we obtain the EVALPSN,

$$T_{1,2}(r,t):[(2,0),\alpha] \wedge T_{3,4}(g,t):[(2,0),\alpha], \tag{13}$$

$$\sim MIN_{3,4}(g,t):[(2,0),\alpha], \tag{14}$$

$$\sim MAX_{3,4}(g,t):[(2,0),\alpha], \tag{15}$$

If traffic sensors $S_{1,3,5}$ detect traffic amount and traffic sensors $S_{2,4,6,7,8}$ do not detect traffic amount at time t, we obtain the EVALPSN,

$$S_1(t):[(2,0),\alpha], \tag{16}$$

$$S_3(t):[(2,0),\alpha], \tag{17}$$

$$S_5(t):[(2,0),\alpha], \tag{18}$$

$$\sim S_2(t):[(2,0),\alpha], \tag{19}$$

$$\sim S_4(t):[(2,0),\alpha], \tag{20}$$

$$\sim S_6(t):[(2,0),\alpha], \tag{21}$$

$$\sim S_7(t) : [(2,0), \alpha], \tag{22}$$

$$\sim S_8(t) : [(2,0), \alpha]. \tag{23}$$

Then, by EVALPSN clauses (13), (18), (15) and (7), the forbiddance from traffic lights $T_{3,4}$ turning to right arrow,

$$T_{3,4}(a,t) : [(0,1), \beta] \tag{24}$$

is derived, furthermore, by EVALPSN clauses (13), (24) and (12), the obligation for traffic lights $T_{3,4}$ keeping green at time $t+1$,

$$T_{3,4}(g,t+1) : [(2,0), \beta]$$

is obtained.

On the other hand, if traffic sensors $S_{1,3}$ detect traffic amount and traffic sensors $S_{2,4,5,6,7,8}$ do not detect traffic amount at time t, we obtain the EVALPSN,

$$S_1(t) : [(2,0), \alpha], \tag{25}$$

$$S_3(t) : [(2,0), \alpha], \tag{26}$$

$$\sim S_2(t) : [(2,0), \alpha], \tag{27}$$

$$\sim S_4(t) : [(2,0), \alpha], \tag{28}$$

$$\sim S_5(t) : [(2,0), \alpha], \tag{29}$$

$$\sim S_6(t) : [(2,0), \alpha], \tag{30}$$

$$\sim S_7(t) : [(2,0), \alpha], \tag{31}$$

$$\sim S_8(t) : [(2,0), \alpha]. \tag{32}$$

Then, by EVALPSN clauses (13), (25), (29), (31), (14), and (1), the permission for traffic lights $T_{3,4}$ turning to right arrow,

$$T_{3,4}(a,t) : [(0,1), \gamma] \tag{33}$$

is derived, furthermore, by EVALPSN clauses (13), (33) and (11), the obligation for traffic lights $T_{3,4}$ turning to right arrow at time $t+1$,

$$T_{3,4}(a,t+1) : [(2,0), \beta]$$

is finally obtained.

Here we introduce an EVALPSN traffic control simulation system based on the cellular automaton method and its simulation results comparing to ordinary fixed-time traffic signal control. In order to evaluate the simulation results we define the concepts "step", "move times", and "stop times" as follows:

step: a time unit in the simulation system, which is a transit time that one car moves from its current cell to the next cell.
move times: shows the times that one car moves from its current cell to the next cell without stop.
stop times: shows the times that one car stops during transition from one cell to another cell.

We introduce the simulation results under the following two traffic flow conditions.
[Condition 1]
Cars are supposed to flow into the intersection from each road with the same probabilities, right-turn 5 %, left-turn 5 % and straight 20 %. It is supposed that green signal duration is 30 steps, yellow one is 3 steps, right-arrow one is 4 steps and red one is 40 steps in the fixed-time traffic signal control. It is also supposed that green signal duration is between 14 and 30 steps in the EVALPSN traffic signal control.
[Condition 2]
Cars are supposed to flow into the intersection with the following probabilities,

from South: right-turn 5 %, left-turn 15 % and straight 10 %;
from North: right-turn 15 %, left-turn 5 % and straight 10 %;
from West: right-turn, left-turn and straight 5 % each;
from East: right-turn and left-turn 5 % each, and straight 15 %.

Other conditions are the same as the Condition 1.

We measured the numbers of car stop and move times during 1000 steps, and repeated it 10 times under the same conditions. The average numbers of car stop and move times are listed in Table 1. The simulation results show that the number of car move times in the EVALPSN traffic signal control is larger than that in the fixed-time traffic signal control, and the number of car stop times in the EVALPSN traffic signal control is smaller than that in the fixed time one. Taking the simulation results into account, it could be concluded that the EVALPSN traffic signal control is more efficient for relieving traffic congestion than the fixed-time traffic signal control.

Table 1 Simulation results

	Fixed-time control		EVALPSN control	
	Stop times	Move times	Stop times	Move times
Condition 1	17,690	19,641	16,285	23,151
Condition 2	16,764	18,664	12,738	20,121

4 Before-After EVALPSN

In this section, we define bf(before-after)-EVALPSN formally and introduce how to implement it aiming at the real-time safety verification for process order control.

4.1 Before-After Relation in EVALPSN

First of all, we introduce a special literal $R(pi, pj, t)$ whose vector annotation represents the before-after relation between processes $Pr_i(pi)$ and $Pr_j(pj)$ at time t, where processes can be regarded as time intervals in general, and the literal $R(pi, pj, t)$ is called a *bf-literal*.[3]

Definition 4.1 (*bf-EVALPSN*) An extended vector annotated literal $R(p_i, p_j, t) : [\mu_1, \mu_2]$ is called a *bf-EVALP* literal, where μ_1 is a vector annotation and $\mu_2 \in \{\alpha, \beta, \gamma\}$. If an EVALPSN clause contains a bf-EVALP literal, it is called a *bf-EVALPSN clause* or just a *bf-EVALP clause* if it contains no strong negation. A bf-EVALPSN is a finite set of bf-EVALPSN clauses.

We provide some paraconsistent interpretations of vector annotations for representing bf-relations, which are called *bf-annotations*. Strictly speaking, bf-relations between time intervals are classified into 15 kinds according to bf-relations between start/finish times of two time intervals. We define the 15 kinds of bf-relations in bf-EVALPSN with regarding processes as time intervals.

Suppose that there are two processes, Pr_i with its start/finish times x_s and x_f, and Pr_j with its start/finish times y_s and y_f.

Before (`be`)/After (`af`)

Firstly, we define the most basic bf-relations *before/after* according to the bf-relation between each start time of two processes, which are represented by bf-annotations `be`/`af`, respectively. If one process has started before/after another one started, then the bf-relations between those processes are defined as "before (`be`)/after(`af`)", respectively. The bf-relations also are described in Fig. 4 with the condition that process Pr_i has started before process Pr_j starts. The bf-relation between their start/finish times is denoted by the inequality $\{x_s < y_s\}$.[4] For example, a fact at time t "process Pr_i has started before process Pr_j started" can be represented by the bf-EVALP clause,

$$R(pi, pj, t) : [\mathtt{be}, \alpha].$$

[3]Hereafter, the expression "**b**efore-a**f**ter" is abbreviated as just "bf" in this chapter.

[4]If time t_1 is earlier than time t_2, we conveniently denote the before-after relation by the inequality $t_1 < t_2$.

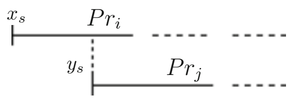


Fig. 4 Bf-relations, before/after

Disjoint Before (db)/After (da)
Bf-relations *disjoint before/after* between processes Pr_i and Pr_j are represented by bf-annotations db/da, respectively. The expressions “disjoint before/after” imply that there is a timelag between the earlier process finish and the later one start. They are also described in Fig. 5 with the condition that process Pr_i has finished before process Pr_j starts. The bf-relation between their start/finish times is denoted by the inequality $\{x_f < y_s\}$. For example, an obligation at time t “process Pr_i must start after process Pr_j finishes” can be represented by the bf-EVALP clause,

$$R(pi, pj, t) : [\mathtt{da}, \beta].$$

Immediate Before (mb)/After (ma)
Bf-relations *immediate before/ after* between the processes Pr_i and Pr_j are represented by bf-annotations mb/ma, respectively. The expressions “immediate before/after” imply that there is no timelag between the finish time of the earlier process and the start time of the later one. The bf-relations are also described in Fig. 6 with the condition that process Pr_i has finished immediately before process Pr_j starts. The bf-relation between their start/finish times is denoted by the equality $\{x_f = y_s\}$. For example, a fact at time t “process Pr_i has finished immediately before process Pr_j starts” can be represented by the bf-EVALP clause,

$$R(pi, pj, t) : [\mathtt{mb}, \alpha].$$

Joint Before (jb)/After (ja)
Bf-relations *joint before/after* between processes Pr_i and Pr_j are represented by bf-annotations jb/ja, respectively. The expressions “joint before/after” imply that the two processes overlap and the earlier process has finished before the later one finishes. The bf-relations are also described in Fig. 7 with the condition that process

Fig. 5 Bf-relations, disjoint before/after

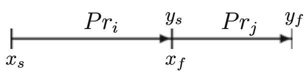


Fig. 6 Bf-relations, immediate before/after

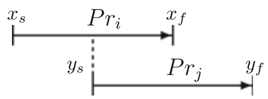


Fig. 7 Bf-relations, joint before/after

Pr_i has started before process Pr_j starts and process Pr_i has finished before process Pr_j finishes. The bf-relation between their start/finish times is denoted by the inequalities $\{x_s < y_s < x_f < y_f\}$. For example, a fact at time t "process Pr_i has started before process Pr_j starts and finished before process Pr_j finishes" can be represented by the bf-EVALP clause,

$$R(pi, pj, t) : [\mathtt{jb}, \alpha].$$

S-included Before (sb), S-included After (sa)

Bf-relations *s-included before/after* between processes Pr_i and Pr_j are represented by bf-annotations sb/sa, respectively. The expressions "s-included before/after" imply that one process has started before another one starts and they have finished at the same time. The bf-relations are also described in Fig. 8 with the condition that process Pr_i has started before process Pr_j starts and they have finished at the same time. The bf-relation between their start/finish times is denoted by the equality and inequalities $\{x_s < y_s < x_f = y_f\}$. For example, a fact at time t "process Pr_i has started before process Pr_j starts and they have finished at the same time" can be represented by the bf-EVALP clause,

$$R(pi, pj, t) : [\mathtt{sb}, \alpha].$$

Included Before (ib)/After (ia)

Bf-relations *included before/after* between processes Pr_i and Pr_j are represented by bf-annotations ib/ia, respectively. The expressions "included before/after" imply that one process has started/finished before/after another one starts/finished, respectively. The bf-relations are also described in Fig. 9 with the condition that process Pr_i has started before process Pr_j starts and finished after process Pr_j finished. The bf-relation between their start/finish times is denoted by the inequalities $\{x_s < y_s,\ y_f < x_f\}$. For example, an obligation at the time t "process Pr_i

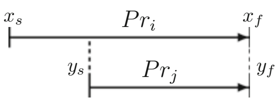


Fig. 8 Bf-relations, S-included before/after

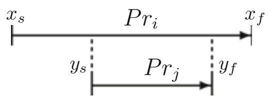


Fig. 9 Bf-relations, included before/after

must start before process Pr_j starts and finish after process Pr_j finishes" can be represented by the bf-EVALP clause,

$$R(pi, pj, t) : [\texttt{ib}, \beta].$$

F-included Before (`fb`)/After (`fa`)

The bf-relations *f-include before/after* between processes Pr_i and Pr_j are represented by bf-annotations `fb`/`fa`, respectively. The expressions "f-included before/after" imply that the two processes have started at the same time and one process has finished before another one finishes. The bf-relations are also described in Fig. 10 with the condition that processes Pr_i and Pr_j have started at the same time and process Pr_i has finished after process Pr_j finished. The bf-relation between their start/finish times is denoted by the equality and inequality $\{x_s = y_s,\ y_f < x_f\}$. For example, a fact at time t "processes Pr_i and Pr_j have started at the same time and process Pr_i has finished after process Pr_j finished" can be represented by the bf-EVALP clause,

$$R(pi, pj, t) : [\texttt{fa}, \alpha].$$

Paraconsistent Before-after (`pba`)

Bf-relation *paraconsistent before-after* between processes Pr_i and Pr_j is represented by bf-annotation `pba`. The expression "paraconsistent before-after" implies that the two processes have started at the same time and also finished at the same time. The bf-relation is also described in Fig. 11 with the condition that processes Pr_i and Pr_j have not only started but also finished at the same time. The bf-relation between

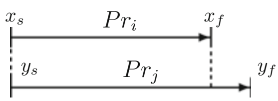


Fig. 10 Bf-relations, F-included before/after

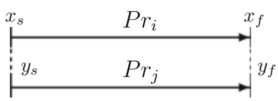


Fig. 11 Bf-relation, paraconsistent before-after

their start/finish times is denoted by the equalities $\{x_s = y_s,\ y_f = x_f\}$. For example, an obligation at time t "processes Pr_i and Pr_j must not only start but also finish at the same time" can be represented by the bf-EVALP clause,

$$R(pi, pj, t) : [\texttt{pba}, \beta].$$

Here we define the epistemic negation $\neg_1$ that maps bf-annotations to themselves in bf-EVALPSN.

Definition 4.2 (*Epistemic Negation* $\neg_1$ *for Bf-annotations*) The epistemic negation $\neg_1$ over the set of bf-annotations,

$$\{\texttt{be}, \texttt{af}, \texttt{da}, \texttt{db}, \texttt{ma}, \texttt{mb}, \texttt{ja}, \texttt{jb}, \texttt{sa}, \texttt{sb}, \texttt{ia}, \texttt{ib}, \texttt{fa}, \texttt{fb}, \texttt{pba}\}$$

is obviously defined as the following mapping:

$$\begin{array}{ll}
\neg_1(\texttt{af}) = \texttt{be}, & \neg_1(\texttt{be}) = \texttt{af}, \\
\neg_1(\texttt{da}) = \texttt{db}, & \neg_1(\texttt{db}) = \texttt{da}, \\
\neg_1(\texttt{ma}) = \texttt{mb}, & \neg_1(\texttt{mb}) = \texttt{ma}, \\
\neg_1(\texttt{ja}) = \texttt{jb}, & \neg_1(\texttt{jb}) = \texttt{ja}, \\
\neg_1(\texttt{sa}) = \texttt{sb}, & \neg_1(\texttt{sb}) = \texttt{sa}, \\
\neg_1(\texttt{ia}) = \texttt{ib}, & \neg_1(\texttt{ib}) = \texttt{ia}, \\
\neg_1(\texttt{fa}) = \texttt{fb}, & \neg_1(\texttt{fb}) = \texttt{fa}, \\
\neg_1(\texttt{pba}) = \texttt{pba}. &
\end{array}$$

If we consider the before-after measure over the 15 bf-annotations, obviously there exists a partial order $(<_h)$ based on the before-after measure, where $\mu_1 <_h \mu_2$ is intuitively interpreted that the bf-annotation μ_1 denotes a more "before" degree than the bf-annotation μ_2, and $\mu_1, \mu_2 \in \{\texttt{be}, \texttt{af}, \texttt{db}, \texttt{da}, \texttt{mb}, \texttt{ma}, \texttt{jb}, \texttt{ja}, \texttt{ib}, \texttt{ia}, \texttt{sb}, \texttt{sa}, \texttt{fb}, \texttt{fa}, \texttt{pba}\}$. If $\mu_1 <_h \mu_2$ and $\mu_2 <_h \mu_1$, we denote it $\mu_1 \equiv_h \mu_2$. Then we have the following ordering:

$$\texttt{db} <_h \texttt{mb} <_h \texttt{jb} <_h \texttt{sb} <_h \texttt{ib} <_h \texttt{fb} <_h \texttt{pba} <_h \texttt{ia} <_h \texttt{ja} <_h \texttt{ma} <_h \texttt{da}$$
and
$$\texttt{sb} \equiv_h \texttt{be} <_h \texttt{af} \equiv_h \texttt{sa}.$$

On the other hand, if we take the before-after knowledge (information) amount of each bf-relation into account as another measure, obviously there also exists another partial order $(<_v)$ in terms of the before-after knowledge amount, where $\mu_1 <_v \mu_2$ is intuitively interpreted that the bf-annotation μ_1 has less knowledge amount in terms of bf-relation than the bf-annotation μ_2. If $\mu_1 <_v \mu_2$ and $\mu_2 <_v \mu_1$, we denote it $\mu_1 \equiv_v \mu_2$. Then we have the following ordering:

$$\texttt{be} <_v \mu_1, \quad \mu_1 \in \{\texttt{db}, \texttt{mb}, \texttt{jb}, \texttt{sb}, \texttt{ib}\},$$
$$\texttt{af} <_v \mu_2, \quad \mu_1 \in \{\texttt{da}, \texttt{ma}, \texttt{ja}, \texttt{sa}, \texttt{ia}\},$$
$$\texttt{db} \equiv_v \texttt{mb} \equiv_v \texttt{jb} \equiv_v \texttt{sb} \equiv_v \texttt{ib} \equiv_v \texttt{fb} \equiv_v \texttt{pba} \equiv_v$$
$$\texttt{fa} \equiv_v \texttt{ia} \equiv_v \texttt{sa} \equiv_v \texttt{ja} \equiv_v \texttt{ma} \equiv_v \texttt{da}$$
$$\text{and}$$
$$\texttt{be} \equiv_v \texttt{af}.$$

If we take the before-after measure as the horizontal one and the before-after knowledge amount as the vertical one, we obtain the complete bi-lattice $\mathcal{T}_v(12)_{bf}$ of vector annotations including the 15 bf-annotations.

$$\begin{aligned}\mathcal{T}_v(12)_{bf} = \{&\perp_{12}(0,0),\ldots,\texttt{be}(0,8),\ldots,\texttt{db}(0,12),\ldots,\texttt{mb}(1,11),\ldots,\\&\texttt{jb}(2,10),\ldots,\texttt{sb}(3,9),\ldots,\texttt{ib}(4,8),\ldots,\texttt{fb}(5,7),\ldots,\\&\texttt{pba}(6,6),\ldots,\texttt{fa}(7,5),\ldots,\texttt{af}(8,0),\ldots,\texttt{ia}(8,4),\ldots,\\&\texttt{sa}(9,3),\ldots,\texttt{ja}(10,2),\ldots,\texttt{ma}(11,1),\ldots,\texttt{da}(12,0),\ldots,\\&\top_{12}(12,12)\},\end{aligned}$$

which is described as the Hasse's diagram in Fig. 12. We note that a bf-EVALP literal

$$\begin{aligned}&R(pi,pj,t):[\mu_1(m,n),\mu_2],\\&\text{where}\quad \mu_2 \in \{\alpha,\beta,\gamma\}\quad\text{and}\\&\mu_1 \in \{\texttt{be},\texttt{db},\texttt{mb},\texttt{jb},\texttt{sb},\texttt{ib},\texttt{fb},\texttt{pba},\texttt{fa},\texttt{ia},\texttt{sa},\texttt{jb},\texttt{ma},\texttt{da},\texttt{af}\},\end{aligned}$$

is not well annotated if $m \neq 0$ and $n \neq 0$, however, since the bf-EVALP literal is equivalent to the following two well annotated bf-EVALP literals:

$$R(pi,pj):[(m,0),\mu] \quad\text{and}\quad R(pi,pj):[(0,n),\mu].$$

Therefore, such non-well annotated bf-EVALP literals can be regarded as the conjunction of two well annotated EVALP literals. For example, suppose that there is a non-well annotated bf-EVALP clause,

$$R(pi,pj,t_1):[(k,l),\mu_1] \to R(pi,pj,t_2):[(m,n),\mu_2],$$

where $k \neq 0$, $l \neq 0$, $m \neq 0$ and $n \neq 0$. It can be equivalently transformed into the following two well annotated bf-EVALP clauses,

$$R(pi,pj,t_1):[(k,0),\mu_1] \wedge R(pi,pj,t_1):[(0,l),\mu_1] \to R(pi,pj,t_2):[(m,0),\mu_2],$$
$$R(pi,pj,t_1):[(k,0),\mu_1] \wedge R(pi,pj,t_2):[(0,l),\mu_1] \to R(pi,pj,t_2):[(0,n),\mu_2].$$

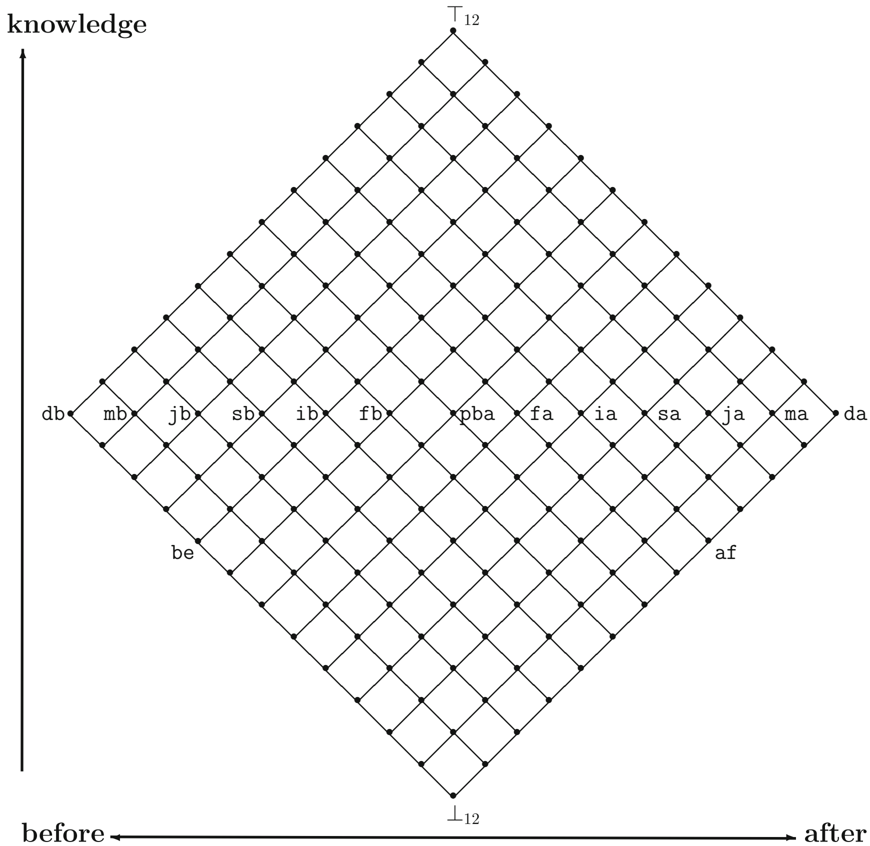


Fig. 12 The complete bi-lattice $\mathcal{T}_v(12)_{bf}$ of bf-annotations

4.2 Implementation of bf-EVALPSN Verification System

We now introduce how to implement bf-EVALPSN process order safety verification systems with a simple example. For simplicity, we do not consider cases in which one process starts/finishes with another one starts/finishes at the same time, however, the process order control system can deal with immediately before/after relations, which means that we consider a case in which two processes are processed in sequence. Then, we do not have to take bf-annotations, sb/sa, fb/fa and pba into account, and we take the following ten bf-annotations with new vector annotations into account:

before(**be**)/after(af),	$(0,4)/(4,0)$,
discrete before(**db**)/after(da),	$(0,7)/(7,0)$,
immediate before(**mb**)/after(ma),	$(1,6)/(6,1)$,
joint before(**jb**)/after(ja),	$(2,5)/(5,2)$,
included before(**ib**)/after(ia),	$(3,4)/(4,3)$.

The complete bi-lattice $\mathcal{T}_v(7)_{bf}$ including the ten bf-annotations is described as the Hasse's diagram in Fig. 13.

Now we show an example of implementing a real-time process order safety verification system in bf-EVALPSN.

Example 2 Suppose three processes Pr_0(id $p0$), Pr_1(id $p1$) and Pr_2(id $p2$) appearing, and the next process Pr_3(id $p3$) not appearing in Fig. 14. Those processes are supposed to be processed according to the processing schedule in Fig. 14, Then, we consider three bf-relations represented by the following bf-EVALP clauses, (34), (35) and (36) :

$$R(p0, p1, t_i) : [(i_1, j_1), \alpha], \tag{34}$$

$$R(p1, p2, t_i) : [(i_2, j_2), \alpha], \tag{35}$$

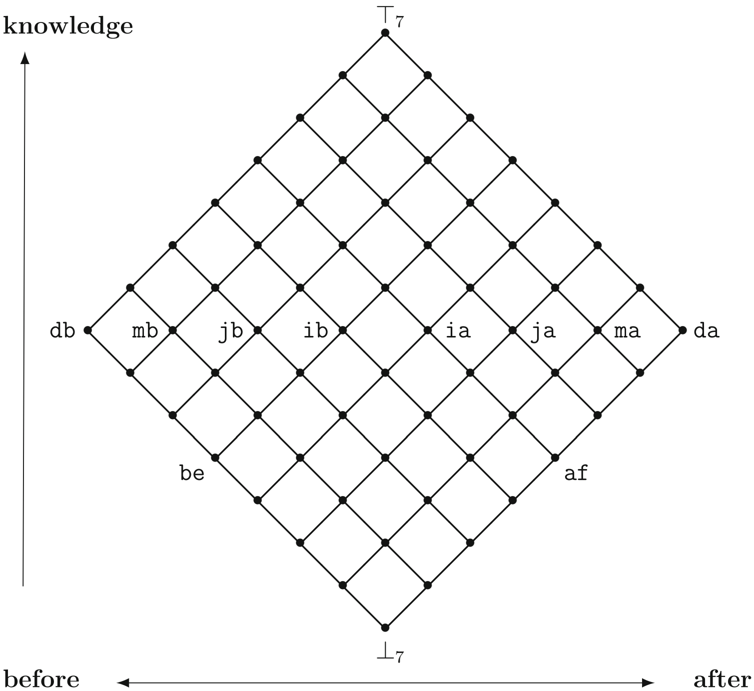


Fig. 13 The complete bi-lattice $\mathcal{T}_v(7)_{bf}$ of bf-annotations

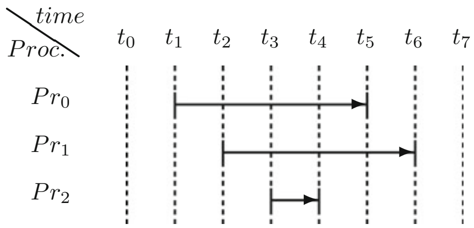


Fig. 14 Process timing chart

$$R(p2, p3, t_i) : [(i_3, j_3), \alpha], \tag{36}$$

which will be inferred based on each process start/finish information at time t_i $(i = 0, 1, 2, \ldots, 7)$.

At time t_0 no process has started yet. Thus, we have no knowledge in terms of any bf-relations. Therefore, we have the bf-EVALP clauses,

$$R(p0, p1, t_0) : [(0, 0), \alpha],$$
$$R(p1, p2, t_0) : [(0, 0), \alpha],$$
$$R(p2, p3, t_0) : [(0, 0), \alpha].$$

At time t_1 only process Pr_0 has started before process Pr_1 starts, Then, bf-annotations, `db` (0, 7), `mb` (1, 6), `jb` (2, 5) or `ib` (3, 4) could be the final bf-annotation to represent the bf-relation between processes Pr_0 and Pr_1, thus, the greatest lower bound `be`(0, 4) of the set of vector annotations {(0, 7), (1, 6), (2, 5), (3, 4)} becomes the vector annotation of bf-literal $R(p0, p1, t_1)$. Other bf-literals have the bottom vector annotation $(0, 0)$. Therefore, we have the bf-EVALP clauses,

$$R(p0, p1, t_1) : [(0, 4), \alpha],$$
$$R(p1, p2, t_1) : [(0, 0), \alpha],$$
$$R(p2, p3, t_1) : [(0, 0), \alpha].$$

At time t_2 the second process Pr_1 also has started before process Pr_0 finish. Then, two bf-annotations, `jb` (2, 5) or `ib` (3, 4) could be the final bf-relation to represent the bf-relation between processes Pr_0 and Pr_1. Thus, the greatest lower bound (2, 4) of the set of vector annotations {(2, 5), (3, 4)} has to be the vector annotation of bf-literal $R(p0, p1, t_2)$. In addition, bf-literal $R(p1, p2, t_2)$ has bf-annotation `be`(0, 4) as well as bf-literal $R(p0, p1, t_1)$ since process Pr_1 has also started before process Pr_2 starts. On the other hand, bf-literal $R(p2, p3, t2)$ has the bottom vector annotation (0, 0) since process Pr_3 has not started yet. Therefore, we have the bf-EVALP clauses,

$$R(p0, p1, t_2) : [(2, 4), \alpha],$$
$$R(p1, p2, t_2) : [(0, 4), \alpha],$$
$$R(p2, p3, t_2) : [(0, 0), \alpha].$$

At time t_3 process Pr_2 has started before both processes Pr_0 and Pr_1 finish. Then, both bf-literals $R(p0, p1, t_3)$ and $R(p1, p2, t_3)$ have the same vector annotation (2, 4) as well as bf-literal $R(p0, p1, t_2)$. Moreover, bf-literal $R(p2, p3, t_3)$ has bf-annotation be(0, 4) as well as bf-literal $R(p0, p1, t_1)$. Therefore, we have the bf-EVALP clauses,

$$R(p0, p1, t_3) : [(2, 4), \alpha],$$
$$R(p1, p2, t_3) : [(2, 4), \alpha],$$
$$R(p2, p3, t_3) : [(0, 4), \alpha].$$

At time t_4 process Pr_2 has finished before both processes Pr_0 and Pr_1 finish. Then, bf-literal $R(p0, p1, t_4)$ still has the same vector annotation (2,4) as well as the previous time t_3. In addition, bf-literal $R(p1, p2, t_4)$ has its final bf-annotation ib(3, 4). For the final bf-relation between processes Pr_2 and Pr_3 there are still two alternatives: (1) if process Pr_3 will start immediately after process Pr_2 finishes, bf-literal $R(p2, p3, t_4)$ has its final bf-annotation mb(1, 6); (2) if process Pr_3 will not start immediately after process Pr_2 finishes, bf-literal $R(p2, p3, t_4)$ has its final bf-annotation db(0, 7). Either way, at least we have the knowledge that process Pr_2 has just finished at time t_4, which can be represented by the vector annotation (0, 6) that is the greatest lower bound of the set of vector annotations {(1, 6), (0, 7)}. Therefore, we have the bf-EVALP clauses,

$$R(p1, p2, t_4) : [(2, 4), \alpha],$$
$$R(p2, p3, t_4) : [(3, 4), \alpha],$$
$$R(p3, p4, t_4) : [(0, 6), \alpha].$$

At time t_5 process Pr_0 has finished before processes Pr_1 finishes. Then, bf-literal $R(p0, p1, t_5)$ has its final bf-annotation jb(2, 5), and bf-literal $R(p2, p3, t_5)$ also has its final bf-annotation jb(0, 7) because process Pr_3 has not started yet. Therefore, we have the bf-EVALP clauses,

$$R(p1, p2, t_5) : [jb(2, 5), \alpha],$$
$$R(p2, p3, t_5) : [ib(3, 4), \alpha],$$
$$R(p3, p4, t_5) : [db(0, 7), \alpha],$$

and all the bf-relations have been determined at time t_5 before process Pr_1 finishes and process Pr_3 starts.

In Example 2, we have shown how the vector annotations of bf-literals are updated according to the start/finish information of processes in real-time. We will introduce the real-time safety verification for process order control based on bf-EVALPSN with small examples in the subsequent section.

4.3 Safety Verification in Bf-EVALPSN

First of all we introduce the basic idea of bf-EVALPSN safety verification for process order with a simple example.

Suppose that two processes Pr_0 and Pr_1 are processed repeatedly, and process Pr_1 must be processed immediately before process Pr_0 starts as shown in Fig. 15. In bf-EVALPSN process order safety verification systems, the safety for process order is verified based on the safety properties to be assured in the processing system. In order to verify the safety for the process order in Fig. 15, we assume two safety properties **SP-0** and **SP-1** for processes Pr_0 and Pr_1 as follows:

SP-0 process Pr_0 must start immediately after process Pr_1 finishes,
SP-1 process Pr_1 must start in a while after (disjoint after) process Pr_0 finishes.

Then, safety properties **SP-0** and **SP-1** should be verified immediately before processes Pr_0 and Pr_1 start, respectively.

In order to verify the bf-relation "immediate after" with safety property **SP-0**, it shoud be verified whether process Pr_1 has finished immediately before process Pr_0 starts or not, and the safety verification should be carried out immediately after process Pr_1 finishes. Then bf-literal $R(p0, p1, t)$ must have vector annotation (6, 0), which means that process Pr_1 has finished but process Pr_0 has not started yet. Therefore, safety property **SP-0** is translated to the bf-EVALPSN-clauses,

SP-0

$$R(p0, p1, t) : [(6, 0), \alpha] \wedge \sim R(p0, p1, t) : [(7, 0), \alpha] \rightarrow \mathit{Start}(p0, t) : [(0, 1), \gamma], \tag{37}$$

$$\sim \mathit{Start}(p0, t) : [(0, 1), \gamma] \rightarrow \mathit{Start}(p0, t) : [(0, 1), \beta], \tag{38}$$

where literal $\mathit{Start}(pi, t)$ represents "process Pr_i starts at time t" and the set of its vector annotations constitutes the complete lattice

$$\mathcal{T}_v(1) = \{\perp(0, 0), (0, 1), (1, 0), \top(1, 1)\}.$$

On the other hand, in order to verify bf-relation "disjoint after" with safety property **SP-1**, it should be verified whether there is a timelag between process Pr_0 finish time and process Pr_1 start time or not. Then, bf-literal $R(p1, p0, t)$ must have

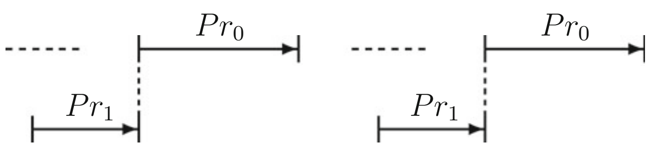


Fig. 15 Bf-EVALPSN safety verification example

bf-annotation da(7, 0). Therefore, safety property **SP-1** is translated to the bf-EVALPSN clauses,

SP-1

$$R(p1, p0, t) : [(7, 0), \alpha] \rightarrow Start(p1, t) : [(0, 1), \gamma], \tag{39}$$

$$\sim Start(p1, t) : [(0, 1), \gamma] \rightarrow Start(p1, t) : [(0, 1), \beta]. \tag{40}$$

Now, we will describe how to verify the process order safety by safety properties **SP-0** and **SP-1** in bf-EVALPSN. In order to verify the process order safety, the following safety verification cycle is applied repeatedly.

Safety Verification Cycle

1st Step (safety verification for starting process Pr_1)
Suppose that process Pr_1 has not started yet at time t_1. If process Pr_0 has already finished at time t_1, we have the bf-EVALP clause,

$$R(p1, p0, t_1) : [(7, 0), \alpha]. \tag{41}$$

On the other hand, if process Pr_0 has just finished at time t_1, we have the bf-EVALP clause,

$$R(p1, p0, t_1) : [(6, 0), \alpha]. \tag{42}$$

If bf-EVALP clause (41) is input to safety property **SP-1** {(39),(40)}, we obtain the EVALP clause,

$$Start(p1, t_1) : [(0, 1), \gamma]$$

and the safety for starting process Pr_1 is assured. On the other hand, if bf-EVALP clause (42) is input to the same safety property **SP-1**, we obtain the EVALP clause

$$Start(p1, t_1) : [(0, 1), \beta],$$

then the safety for starting process Pr_1 is not assured.

2nd Step (safety verification for starting process Pr_0)
Suppose that process Pr_0 has not started yet at time t_2. If process Pr_1 has just finished at time t_2, we have the bf-EVALP clause,

$$R(p0, p1, t_2) : [(6, 0), \alpha]. \tag{43}$$

On the other hand, if process Pr_1 has not finished yet at time t_2, we have the bf-EVALP clause,

$$R(p0, p1, t_2) : [(4, 0), \alpha]. \tag{44}$$

If bf-EVALP clause (43) is input to safety property **SP-0** {(37), (38)}, we obtain the EVALP clause,

$$Start(p0, t_2) : [(0, 1), \gamma],$$

and the safety for starting process Pr_0 is assured. On the other hand, if bf-EVALP clause (44) is input to the same safety property **SP-0**, we obtain the EVALP clause,

$$Start(p1, t) : [(0, 1), \beta],$$

then the safety for starting process Pr_0 is not assured.

Example 3 In this example we provide a more practical bf-EVALPSN safety verification for process order control with a simple brewery pipeline process order control. The brewery pipeline network consists of four tanks $\{T_0, T_1, T_2, T_3\}$, five pipe lines $\{Pi_0, Pi_1, Pi_2, Pi_3, Pi_4\}$, and two valves $\{V_0, V_1\}$ as shown in Fig. 16. We assume that four pipeline processes :

process Pr_0, a brewery process using

line-1, tank $T_0 \rightarrow$ valve $V_0 \rightarrow$ tank T_1;

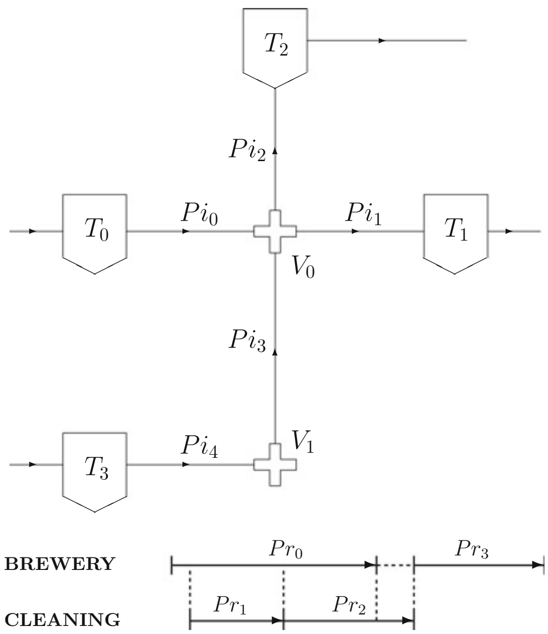


Fig. 16 Brewery pipeline and processing schedule

process Pr_1, a cleaning process by nitric acid using

line-2, tank $T_3 \rightarrow$ valve $V_1 \rightarrow$ Valve $V_0 \rightarrow$ tank T_2;

process Pr_2, a cleaning process by water in line-1;
process Pr_3, a brewery process using both line-1 and line-2 with mixing at valve V_0

are processed according to the processing schedule in Fig. 16. We also assume the following four safety properties:

safety property **SP-2**,

process Pr_0 must start before any other processes start;

safety property **SP-3**,

process Pr_1 must start immediately after process Pr_0 starts;

safety property **SP-4**

process Pr_2 must start immediately after process Pr_1 finishes;

safety property **SP-5**

process Pr_3 must start immediately after both processes Pr_0 and Pr_2 finish.

Safety property **SP-2** is translated to the bf-EVALPSN clauses,

$$\begin{aligned}&\textbf{SP-2}\\&\sim R(p0,p1,t):[(4,0),\alpha]\wedge\sim R(p0,p2,t):[(4,0),\alpha]\wedge\\&\sim R(p0,p3,t):[(4,0),\alpha]\rightarrow Start(p0,t):[(0,1),\gamma],\\&\sim Start(p0,t):[(0,1),\gamma]\rightarrow Start(p0,t):[(0,1),\beta].\end{aligned}\tag{45}$$

As well as safety property **SP-2**, other safety properties **SP-3**, **SP-4** and **SP-5** are also translated to the bf-EVALPSN clauses,

$$\begin{aligned}&\textbf{SP-3}\\&R(p1,p0,t):[(4,0),\alpha]\rightarrow Start(p1,t):[(0,1),\gamma],\\&\sim Start(p1,t):[(0,1),\gamma]\rightarrow Start(p1,t):[(0,1),\beta],\end{aligned}\tag{46}$$

$$\begin{aligned}&\textbf{SP-4}\\&R(p2,p1,t):[(6,0),\alpha]\wedge\sim R(p2,p1,t):[(7,0),\alpha]\\&\rightarrow Start(p2,t):[(0,1),\gamma],\\&\sim Start(p2,t):[(0,1),\gamma]\rightarrow Start(p2,t):[(0,1),\beta],\end{aligned}\tag{47}$$

$$\begin{aligned}
&\textbf{SP-5}\\
&R(p3,p0,t):[(6,0),\alpha]\wedge\\
&R(p3,p2,t):[(6,0),\alpha]\wedge\\
&\sim R(p3,p2,t):[(7,0),\alpha]\\
&\rightarrow Start(p3,t):[(0,1),\gamma],\\
&R(p3,p0,t):[(6,0),\alpha]\wedge\\
&R(p3,p2,t):[(6,0),\alpha]\wedge\\
&\sim R(p3,p0,t):[(7,0),\alpha]\\
&\rightarrow Start(p3,t):[(0,1),\gamma],\\
&\sim Start(p3,t):[(0,1),\gamma]\rightarrow Start(p3,t):[(0,1),\beta].
\end{aligned}\tag{48}$$

Now, we will describe the safety verification process for the process order in Fig. 16.

Initial Stage (t_0) No process has started at time t_0, we have no information in terms of all bf-relations between all processes Pr_0, Pr_1, Pr_2 and Pr_3, thus, we have the bf-EVALP clauses,

$$R(p0,p1,t_0):[(0,0),\alpha],\tag{49}$$

$$R(p0,p2,t_0):[(0,0),\alpha],\tag{50}$$

$$R(p0,p3,t_0):[(0,0),\alpha].\tag{51}$$

In order to verify the safety for starting the first process Pr_0, the bf-EVALP clauses (49), (50) and (51) are input to safety property **SP-2** (45). Then, we obtain the EVALP clause,

$$Start(p0,t_0):[(0,1),\gamma],$$

which expresses permission for starting process Pr_0, and its safety is assured at time t_0. Otherwise, it is not assured.

2nd Stage (t_1) Suppose that only process Pr_0 has already started at time t_1. Then, we have the bf-EVALP clauses,

$$R(p1,p0,t_1):[(4,0),\alpha].\tag{52}$$

In order to verify the safety for starting the second process Pr_1, the bf-EVALP clause (52) is input to safety property **SP-3** (46). Then, we obtain the EVALP clause,

$$Start(p1,t_1):[(0,1),\gamma],$$

and the safety for starting process Pr_1 is assured at time t_1. Otherwise, it is not assured.

3rd Stage (t_2) Suppose that processes Pr_0 and Pr_1 have already started, and neither of them has finished yet at time t_2. Then, we have the bf-EVALP clauses,

$$R(p2, p0, t_2) : [(4, 0), \alpha], \tag{53}$$

$$R(p2, p1, t_2) : [(4, 0), \alpha]. \tag{54}$$

In order to verify the safety for starting the third process Pr_2, if EVALP clause (54) is input to safety property **SP-4** (47), then, we obtain the EVALP clause,

$$Start(p2, t_2) : [(0, 1), \beta],$$

and the safety for starting process Pr_2 is not assured at time t_2. On the other hand, if process Pr_1 has just finished at time t_2, then, we have the bf-EVALP clause,

$$R(p2, p1, t_2) : [(6, 0), \alpha]. \tag{55}$$

If bf-EVALP clause (55) is input to safety property **SP-4** (47), then, we obtain the EVALP clause,

$$Start(p2, t_2) : [(0, 1), \gamma],$$

and the safety for starting process Pr_2 is assured.

4th Stage (t_3) Suppose that processes Pr_0, Pr_1 and Pr_2 have already started, processes Pr_0 and Pr_1 have already finished, and only process Pr_3 has not started yet at time t_3. Then, we have the bf-EVALP clauses,

$$R(p3, p0, t_3) : [(7, 0), \alpha], \tag{56}$$

$$R(p3, p1, t_3) : [(7, 0), \alpha], \tag{57}$$

$$R(p3, p2, t_3) : [(4, 0), \alpha]. \tag{58}$$

In order to verify the safety for starting the last process Pr_3, if bf-EVALP clauses (56) and (58) are input to safety property **SP-5** (48), then, we obtain the EVALP clause,

$$Start(p3, t_3) : [(0, 1), \beta],$$

and the safety for starting process Pr_3 is not assured at time t_3. On the other hand, if process Pr_2 has just finished at time t_3, then we have the bf-EVALP clause,

$$R(p3, p2, t_3) : [(6, 0), \alpha]. \tag{59}$$

If bf-EVALP clause (59) is input to safety property **SP-5** (48), then we obtain the EVALP clause,

$$Start(p3, t_3) : [(0, 1), \gamma],$$

and the safety for starting process Pr_3 is assured.

5 Reasoning in bf-EVALPSN

In this section, we summarize the before-after relation reasoning system of bf-EVALPSN, which consists of two inference rules in bf-EVALP. One of them is the basic inference rules of bf-relations according to the before-after relations of process start/finish times, and another one is the transitive inference rules that infer a bf-relation from two continuous bf-relations transitively.

5.1 Basic Reasoning for Bf-Relation

We introduce the basic inference rules of bf-relations with referring to Example 2 in Sect. 4.2, which are called *basic bf-inference rules*. Hereafter we call the inference rules as *ba-inf rules* shortly. First of all, in order to represent the basic bf-inference rules in bf-EVALPSN, we newly introduce two literals:

$st(p_i, t)$, which is intuitively interpreted that process Pr_i starts at time t, and
$fi(p_i, t)$, which is intuitively interpreted that process Pr_i finishes at time t.

Those literals are used for expressing process start/finish information and may have one of the vector annotations, $\{\perp(0,0), \mathtt{t}(1,0), \mathtt{f}(0,1), \top(1,1)\}$, where annotations $\mathtt{t}(1,0)$ and $\mathtt{f}(0,1)$ can be intuitively interpreted as "true" and "false", respectively.

We show a group of ba-inf rules to be applied at the initial stage (time t_0) for bf-relation reasoning, which are named (0, 0)-*rules*.

(0, 0)-rules
Suppose that no process has started yet and the vector annotation of bf-literal $R(p_i, p_j, t)$ is (0, 0), which shows that there is no knowledge in terms of the bf-relation between processes Pr_i and Pr_j, then the following two ba-inf rules are applied at the initial stage.

(0, 0)-rule-1

If process Pr_i started before process Pr_j starts, then the vector annotation (0, 0) of bf-literal $R(p_i, p_j, t)$ should turn to bf-annotation be(0, 8), which is the greatest lower bound of the bf-annotations,

$$\{\texttt{db}(0,12),\ \texttt{mb}(1,11),\ \texttt{jb}(2,10),\ \texttt{sb}(3,9),\ \texttt{ib}(4,8)\}.$$

(0, 0)-rule-2

If both processes Pr_i and Pr_j have started at the same time, then it is reasonably anticipated that the bf-relation between processes Pr_i and Pr_j will be one of the bf-annotations,

$$\{\texttt{fb}(5,7),\ \texttt{pba}(6,6),\ \texttt{fa}(7,5)\}$$

whose greatest lower bound is (5, 5) (refer to Fig. 12). Therefore, the vector annotation (0, 0) of bf-literal $R(p_i, p_j, t)$ should turn to (5, 5).

Ba-inf rules (0, 0)-rule-1 and (0, 0)-rule-2 may be translated into the bf-EVALPSN clauses,

$$R(p_i,p_j,t) : [(0,0),\alpha] \wedge st(p_i,t) : [t,\alpha] \wedge \sim st(p_j,t) : [t,\alpha] \\ \rightarrow R(p_i,p_j,t) : [(0,8),\alpha], \tag{60}$$

$$R(p_i,p_j,t) : [(0,0),\alpha] \wedge st(p_i,t) : [t,\alpha] \wedge st(p_j,t) : [t,\alpha] \\ \rightarrow R(p_i,p_j,t) : [(5,5),\alpha]. \tag{61}$$

Suppose that one of ba-inf rules (0, 0)-rule-1 and 2 has been applied, then the vector annotation of bf-literal $R(p_i, p_j, t)$ should be one of (0, 8) or (5, 5). Therefore, we need to consider two groups of ba-inf rules to be applied for following ba-inf rules (0, 0)-rule-1 and (0, 0)-rule-2, which are named (0, 8)-*rules* and (5, 5)-*rules*, respectively.

(0, 8)-rules

Suppose that process Pr_i has started before process Pr_j starts, then the vector annotation of bf-literal $R(p_i, p_j, t)$ should be (0, 8). We have the following inference rules to be applied for following ba-inf rule (0, 0)-rule-1.

(0, 8)-rule-1

If process Pr_i has finished before process Pr_j starts, and process Pr_j starts immediately after process Pr_i finished, then the vector annotation (0, 8) of bf-literal $R(p_i, p_j, t)$ should turn to bf-annotation mb(1,11).

(0, 8)-rule-2

If process Pr_i has finished before process Pr_j starts, and process Pr_j has not started immediately after process Pr_i finished, then the vector annotation (0, 8) of bf-literal $R(p_i, p_j, t)$ should turn to bf-annotation db(0,12).

(0, 8)-rule-3

If process Pr_j starts before process Pr_i finishes, then the vector annotation (0, 8) of bf-literal $R(p_i, p_j, t)$ should turn to (2, 8) that is the greatest lower bound of the bf-annotations,

$$\{\texttt{jb}(2, 10),\ \texttt{sb}(3, 9),\ \texttt{ib}(4, 8)\}.$$

Ba-inf rules (0, 8)-rule-1, (0, 8)-rule-2 and (0, 8)-rule-3 may be translated into the bf-EVALPSN clauses,

$$\begin{aligned} &R(p_i, p_j, t) : [(0, 8), \alpha] \wedge fi(p_i, t) : [t, \alpha] \wedge st(p_j, t) : [t, \alpha] \\ &\rightarrow R(p_i, p_j, t) : [(1, 11), \alpha], \end{aligned} \tag{62}$$

$$\begin{aligned} &R(p_i, p_j, t) : [(0, 8), \alpha] \wedge fi(p_i, t) : [t, \alpha] \wedge \sim st(p_j, t) : [t, \alpha] \\ &\rightarrow R(p_i, p_j, t) : [(0, 12), \alpha], \end{aligned} \tag{63}$$

$$\begin{aligned} &R(p_i, p_j, t) : [(0, 8), \alpha] \wedge \sim fi(p_i, t) : [t, \alpha] \wedge st(p_j, t) : [t, \alpha] \\ &\rightarrow R(p_i, p_j, t) : [(2, 8), \alpha]. \end{aligned} \tag{64}$$

(5, 5)-rules

Suppose that both processes Pr_i and Pr_j have already started at the same time, then the vector annotation of bf-literal $R(p_i, p_j, t)$ should be (5, 5). We have the following inference rules to be applied for the following ba-inf rule (0, 0)-rule-2.

(5, 5)-rule-1

If process Pr_i has finished before process Pr_j finishes, then the vector annotation (5, 5) of bf-literal $R(p_i, p_j, t)$ should turn to bf-annotation sb(5, 7).

(5, 5)-rule-2

If both processes Pr_i and Pr_j have finished at the same time, then the vector annotation (5, 5) of bf-literal $R(p_i, p_j, t)$ should turn to bf-annotation pba(6, 6).

(5, 5)-rule-3

If process Pr_j has finished before process Pr_i finishes, then the vector annotation (5, 5) of bf-literal $R(p_i, p_j, t)$ should turn to bf-annotation sa(7, 5).

Ba-inf rules (5, 5)-rule-1, (5, 5)-rule-2 and (5, 5)-rule-3 may be translated into the bf-EVALPSN clauses,

$$R(p_i,p_j,t):[(5,5),\alpha]\wedge fi(p_i,t):[t,\alpha]\wedge \sim fi(p_j,t):[t,\alpha] \\ \rightarrow R(p_i,p_j,t):[(5,7),\alpha], \tag{65}$$

$$R(p_i,p_j,t):[(5,5),\alpha]\wedge fi(p_i,t):[t,\alpha]\wedge fi(p_j,t):[t,\alpha] \\ \rightarrow R(p_i,p_j,t):[(6,6),\alpha], \tag{66}$$

$$R(p_i,p_j,t):[(5,5),\alpha]\wedge \sim fi(p_i,t):[t,\alpha]\wedge fi(p_j,t):[t,\alpha] \\ \rightarrow R(p_i,p_j,t):[(7,5),\alpha]. \tag{67}$$

If ba-inf rules, (0, 8)-rule-1, (0, 8)-rule-2, (5, 5)-rule-1, (5, 5)-rule-2 and (5, 5)-rule-3, and have been applied, bf-relations represented by bf-annotations such as `jb`(2, 10)/`ja`(10, 2) between two processes should be derived. On the other hand, even if ba-inf rule (0, 8)-rule-3 has been applied, no bf-annotation could be derived. Therefore, a group of ba-inf rules called (2, 8)-rules should be considered for the following ba-inf rule (0, 8)-rule-3.

(2, 8)-rules

Suppose that process Pr_i has started before process Pr_j starts and process Pr_j has started before process Pr_i finishes, then the vector annotation of bf-literal $R(p_i,p_j,t)$ should be (2, 8) and the following three rules should be considered.

(2, 8)-rule-1 If process Pr_i finished before process Pr_j finishes, then the vector annotation (2, 8) of bf-literal $R(p_i,p_j,t)$ should turn to bf-annotation `jb`(2, 10).

(2, 8)-rule-2 If both processes Pr_i and Pr_j have finished at the same time, then the vector annotation (2, 8) of bf-literal $R(p_i,p_j,t)$ should turn to bf-annotation `fb`(3, 9).

(2, 8)-rule-3 If process Pr_j has finished before Pr_i finishes, then the vector annotation (2, 8) of bf-literal $R(p_i,p_j,t)$ should turn to bf-annotation `ib`(4, 8).

Ba-inf rules (2, 8)-rule-1, (2, 8)-rule-2 and (2, 8)-rule-3 may be translated into the bf-EVALPSN clauses,

$$R(p_i,p_j,t):[(2,8),\alpha]\wedge fi(p_i,t):[t,\alpha]\wedge \sim fi(p_j,t):[t,\alpha] \\ \rightarrow R(p_i,p_j,t):[(2,10),\alpha], \tag{68}$$

$$R(p_i,p_j,t):[(2,8),\alpha]\wedge fi(p_i,t):[t,\alpha]\wedge fi(p_j,t):[t,\alpha] \\ \rightarrow R(p_i,p_j,t):[(3,9),\alpha], \tag{69}$$

$$R(p_i,p_j,t):[(2,8),\alpha]\wedge \sim fi(p_i,t):[t,\alpha]\wedge fi(p_j,t):[t,\alpha] \\ \rightarrow R(p_i,p_j,t):[(4,8),\alpha]. \tag{70}$$

The application orders of all ba-inf rules are summarized in Table 2.

Table 2 Application orders of basic bf-inference rules

Vector annotation	Rule	Vector annotation	Rule	Vector annotation	Rule	Vector annotation
(0, 0)	Rule-1	(0, 8)	Rule-1	(0, 12)		
			Rule-2	(1, 11)		
			Rule-3	(2, 8)	Rule-1	(2, 10)
					Rule-2	(3, 9)
					Rule-3	(4, 8)
	Rule-2	(5, 5)	Rule-1	(5, 7)		
			Rule-2	(6, 6)		
			Rule-3	(7, 5)		

5.2 Transitive Reasoning for Bf-Relations

Suppose that a bf-EVALPSN process order control system has to deal with ten processes. Then, if it deals with all the bf-relations between ten processes, forty five bf-relations have to be considered. It may take much computation cost. In order to reduce such computation cost, we consider inference rules to derive bf-relation between processes Pr_i and Pr_k from bf-relations between processes Pr_i and Pr_j and between processes Pr_j and Pr_k in bf-EVALPSN in real-time, which are called *transitive bf-inference rules*. Hereafter we call transitive bf-inference rules as *tr-inf rules* for short. We introduce how to derive some of tr-inf rules and how to apply them to real-time process order control.

Suppose that three processes Pr_0, Pr_1 and Pr_2 are processed according to the process schedule in Fig. 17 in which only the start time of process Pr_2 varies time t_3 to time t_5 and no bf-relation between processes varies. The vector annotations of bf-literals $R(p0, p1, t)$, $R(p1, p2, t)$ and $R(p0, p2, t)$ at each time $t_i (i = 1, \ldots, 7)$ are shown by the three tables in Table 3. For each table, if we focus on the vector annotations at time t_1 and time t_2, the following tr-inf rule in bf-EVALP clause can be derived:

rule-1
$R(p0, p1, t) : [(0, 8), \alpha] \wedge R(p1, p2, t) : [(0, 0), \alpha]$
$\rightarrow R(p0, p2, t) : [(0, 8), \alpha]$

which may be reduced to the bf-EVALP clause,

$$R(p0, p1, t) : [(0, 8), \alpha] \rightarrow R(p0, p2, t) : [(0, 8), \alpha]. \quad (71)$$

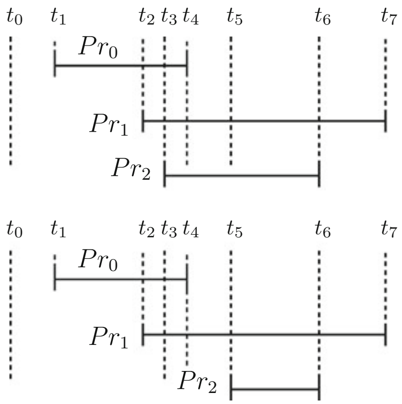


Fig. 17 Process time chart 1(*top left*), 2(*top right*), 3(*bottom left*)

Furthermore, if we also focus on the vector annotations at time t_3 and time t_4 in Table 3, the following two tr-inf rules also can be derived:

rule-2

$$R(p0, p1, t) : [(2, 8), \alpha] \wedge R(p1, p2, t) : [(2, 8), \alpha] \rightarrow R(p0, p2, t) : [(2, 8), \alpha], \tag{72}$$

rule-3

$$R(p0, p1, t) : [(2, 10), \alpha] \wedge R(p1, p2, t) : [(2, 8), \alpha] \rightarrow R(p0, p2, t) : [(2, 10), \alpha]. \tag{73}$$

As well as tr-inf rules **rule-2** and **rule-3**, the following two tr-inf rules also can be derived with focusing on the variation of the vector annotations at time t_4.

rule-4

$$R(p0, p1, t) : [(2, 10), \alpha] \wedge R(p1, p2, t) : [(2, 8), \alpha] \rightarrow R(p0, p2, t) : [(1, 11), \alpha], \tag{74}$$

rule-5

$$R(p0, p1, t) : [(2, 10), \alpha] \wedge R(p1, p2, t) : [(0, 8), \alpha] \rightarrow R(p0, p2, t) : [(0, 12), \alpha]. \tag{75}$$

Among all the tr-inf rules only tr-inf rules **rule-3** and **rule 4** have the same precedent(body)

$$R(p0, p1, t) : [(2, 10), \alpha] \wedge R(p1, p2, t) : [(2, 8), \alpha],$$

Table 3 Vector annotations of process time chart 1, 2, 3

Process time chart 1	t_0	t_1	t_2	t_3	t_4	t_5	t_6	t_7
$R(p0, p1, t)$	(0, 0)	(0, 8)	(2, 8)	(2, 8)	(2, 10)	(2, 10)	(2, 10)	(2, 10)
$R(p1, p2, t)$	(0, 0)	(0, 0)	(0, 8)	(2, 8)	(2, 8)	(2, 8)	(4, 8)	(4, 8)
$R(p0, p2, t)$	(0, 0)	(0, 8)	(0, 8)	(2, 8)	(2, 10)	(2, 10)	(2, 10)	(2, 10)
Process time chart 2								
$R(p0, p1, t)$	(0, 0)	(0, 8)	(2, 8)	(2, 8)	(2, 10)	(2, 10)	(2, 10)	(2, 10)
$R(p1, p2, t)$	(0, 0)	(0, 0)	(0, 8)	(0, 8)	(2, 8)	(2, 8)	(4, 8)	(4, 8)
$R(p0, p2, t)$	(0, 0)	(0, 8)	(0, 8)	(0, 8)	(1, 11)	(1, 11)	(1, 11)	(1, 11)
Process time chart 3								
$R(p0, p1, t)$	(0, 0)	(0, 8)	(2, 8)	(2, 8)	(2, 10)	(2, 10)	(2, 10)	(2, 10)
$R(p1, p2, t)$	(0, 0)	(0, 0)	(0, 8)	(0, 8)	(0, 8)	(2, 8)	(4, 8)	(4, 8)
$R(p0, p2, t)$	(0, 0)	(0, 8)	(0, 8)	(0, 8)	(0, 12)	(0, 12)	(0, 12)	(0, 12)

and different consequents(heads)

$$R(p0, p2, t) : [(2, 10), \alpha] \quad \text{and} \quad R(p0, p2, t) : [(1, 11), \alpha].$$

Having the same precedent may cause duplicate application of the tr-inf rules. If we take tr-inf rules **rule-3** and **rule-4** into account, obviously they cannot be uniquely applied. In order to avoid duplicate application of tr-inf rules **rule-3** and **rule-4**, we consider all correct applicable orders **order-1** (76), **order-2** (77) and **order-3** (78) for all the tr-inf rules, **rule-1**,…, **rule-5**.

$$\textbf{order-1}: \quad \textbf{rule-1} \rightarrow \textbf{rule-2} \rightarrow \textbf{rule-3} \tag{76}$$

$$\textbf{order-2}: \quad \textbf{rule-1} \rightarrow \textbf{rule-4} \tag{77}$$

$$\textbf{order-3}: \quad \textbf{rule-1} \rightarrow \textbf{rule-5} \tag{78}$$

As indicated in the above orders, tr-inf rule **rule 3** should be applied immediately after tr-inf rule **rule 2**, on the other hand, tr-inf rule **rule 4** should be done immediately after tr-inf rule **rule 1**. Thus, if we take the applicable orders (76), (77) and (78) into account, such confusion may be avoidable. Actually, tr-inf rules are not complete, that is to say there exist some cases in which bf-relations cannot be uniquely determined by only tr-inf rules.

We show an application of tr-inf rules by taking process time chart 3 in Fig. 17 as an example.

At time t_1, tr-inf rule **rule-1** is applied and we have the bf-EVALPSN clause,

$$R(p0, p2, t_1) : [(0, 8), \alpha].$$

At time t_2 and time t_3, no tr-inf rule can be applied and we still have the same vector annotation (0,8) of bf-literal $R(p0, p2, t_3)$.
At time t_4, only tr-inf rule **rule-5** can be applied and we obtain the bf-EVALP clause,

$$R(p0, p2, t_4) : [(0, 12), \alpha]$$

and the bf-relation between processes Pr_0 and Pr_2 has been inferred according to process order **order-3** (78).We could not introduce all tr-inf rules in this section though, it is sure that we have many cases that can reduce bf-relation computing cost in bf-EVALPSN process order control by using tr-inf rules. In real-time process control systems, such reduction of computing cost is required and significant in practice.

As another topic, we briefly introduce anticipation of bf-relations in bf-EVALPSN. For example, suppose that three processes Pr_0, Pr_1 and Pr_2 have started in this turn, and only process Pr_1 has finished at time t as shown in Fig. 18. Then, two bf-relations between processes Pr_0 and Pr_1 and between processes Pr_1 and Pr_2 have already determined, and we have the following two bf-EVALP clauses with final bf-annotations,

$$R(p0, p1, t) : [\mathtt{ib}(4, 8), \alpha] \quad \text{and} \quad R(p1, p2, t) : [\mathtt{mb}(1, 11), \alpha]. \tag{79}$$

On the other hand, the bf-relation between processes Pr_0 and Pr_2 cannot be determined yet. However, if we use the tr-inf rule,

rule-6

$$R(p0, p1, t) : [(4, 8), \alpha] \wedge R(p1, p2, t) : [(2, 10), \alpha] \tag{80}$$
$$\rightarrow R(p0, p2, t) : [(2, 8), \alpha],$$

we obtain vector annotation (2, 8) as the bf-annotation of bf-literal $R(p0, p2, t)$. Moreover, it is logically anticipated that the bf-relation between processes Pr_0 and Pr_2 will be finally represented by one of three bf-annotations (vector annotations), jb(2, 10), sb(3, 9) and ib(4, 8), since the vector annotation (2, 8) is the greatest lower bound of the set of vector annotations, {(2, 10),(3, 9),(4, 8)}. As mentioned above, we can systematically anticipate complete bf-annotations from incomplete

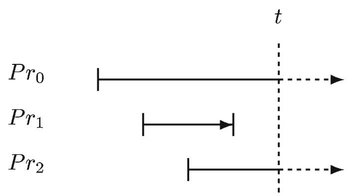


Fig. 18 Anticipation of bf-relation

bf-annotations in bf-EVALPSN. This remarkable anticipatory feature of bf-EVALPSN could be applied to safety verification and control that require such logical anticipation.

5.3 Transitive bf-Inference Rules

In this subsection we list all transitive bf-inference rules (tr-inf rules) with taking their application orders into account. For simplicity, we represent a tr-inf rule,

$$R(p_i, p_j, t) : [(n_1, n_2), \alpha] \wedge R(p_j, p_k, t) : [(n_3, n_4), \alpha] \rightarrow R(p_i, p_k, t) : [(n_5, n_6), \alpha]$$

by only vector annotations and logical connectives, ∧ and →, as follows:

$$(n_1, n_2) \wedge (n_3, n_4) \rightarrow (n_5, n_6)$$

in the list of tr-inf rules.

Transitive bf-inference Rules

TR0 $(0,0) \wedge (0,0) \rightarrow (0,0)$
TR1 $(0,8) \wedge (0,0) \rightarrow (0,8)$
TR1-1 $(0,12) \wedge (0,0) \rightarrow (0,12)$
TR1-2 $(1,11) \wedge (0,8) \rightarrow (0,12)$
TR1-3 $(1,11) \wedge (5,5) \rightarrow (1,11)$
TR1-4 $(2,8) \wedge (0,8) \rightarrow (0,8)$

TR1-4-1 $(2,10) \wedge (0,8) \rightarrow (0,12)$
TR1-4-2 $(4,8) \wedge (0,12) \rightarrow (0,8)$ (81)
TR1-4-3 $(2,8) \wedge (2,8) \rightarrow (2,8)$

TR1-4-3-1 $(2,10) \wedge (2,8) \rightarrow (2,10)$
TR1-4-3-2 $(4,8) \wedge (2,10) \rightarrow (2,8)$
TR1-4-3-3 $(2,8) \wedge (4,8) \rightarrow (4,8)$
TR1-4-3-4 $(3,9) \wedge (2,10) \rightarrow (2,10)$ (82)
TR1-4-3-5 $(2,10) \wedge (4,8) \rightarrow (3,9)$
TR1-4-3-6 $(4,8) \wedge (3,9) \rightarrow (4,8)$
TR1-4-3-7 $(3,9) \wedge (3,9) \rightarrow (3,9)$

TR1-4-4 $(3,9) \wedge (0,12) \rightarrow (0,12)$
TR1-4-5 $(2,10) \wedge (2,8) \rightarrow (1,11)$
TR1-4-6 $(4,8) \wedge (1,11) \rightarrow (2,8)$ (83)
TR1-4-7 $(3,9) \wedge (1,11) \rightarrow (1,11)$

$$\begin{array}{ll} \textbf{TR1-5} & (2,8) \wedge (5,5) \rightarrow (2,8) \\ \quad \textbf{TR1-5-1} & (4,8) \wedge (5,7) \rightarrow (2,8) \\ \quad \textbf{TR1-5-2} & (2,8) \wedge (7,5) \rightarrow (4,8) \\ \quad \textbf{TR1-5-3} & (3,9) \wedge (5,7) \rightarrow (2,10) \\ \quad \textbf{TR1-5-4} & (2,10) \wedge (7,5) \rightarrow (3,9) \end{array} \tag{84}$$

$$\begin{array}{ll} \textbf{TR2} & (5,5) \wedge (0,8) \rightarrow (0,8) \\ \quad \textbf{TR2-1} & (5,7) \wedge (0,8) \rightarrow (0,12) \\ \quad \textbf{TR2-2} & (7,5) \wedge (0,12) \rightarrow (0,8) \\ \quad \textbf{TR2-3} & (5,5) \wedge (2,8) \rightarrow (2,8) \end{array} \tag{85}$$

$$\begin{array}{ll} \textbf{TR2-3-1} & (5,7) \wedge (2,8) \rightarrow (2,10) \\ \textbf{TR2-3-2} & (7,5) \wedge (2,10) \rightarrow (2,8) \\ \textbf{TR2-3-3} & (5,5) \wedge (4,8) \rightarrow (4,8) \\ \textbf{TR2-3-4} & (7,5) \wedge (3,9) \rightarrow (4,8) \end{array} \tag{86}$$

$$\begin{array}{ll} \textbf{TR2-4} & (5,7) \wedge (2,8) \rightarrow (1,11) \\ \textbf{TR2-5} & (7,5) \wedge (1,11) \rightarrow (2,8) \end{array} \tag{87}$$

$$\begin{array}{ll} \textbf{TR3} & (5,5) \wedge (5,5) \rightarrow (5,5) \\ \quad \textbf{TR3-1} & (7,5) \wedge (5,7) \rightarrow (5,5) \\ \quad \textbf{TR3-2} & (5,7) \wedge (7,5) \rightarrow (6,6) \end{array} \tag{88}$$

Note: the bottom vector annotation (0, 0) in tr-inf rules shows that any bf-EVALP clause $R(p_j, p_k, t) : [(n, m), \alpha]$ satisfies it.

Here we indicate two important points in terms of the list of tr-inf rules.

(I) Names of tr-inf rules such as TR1-4-3 show their application orders. For example, if tr-inf rule TR1 has been applied, one of tr-inf rules TR1-1,TR1-2,… or TR1-5 should be applied at the following stage; if tr-inf rule TR1-4 has been applied after tr-inf rule TR1, one of tr-inf rules TR1-4-1,TR1-4-2,… or TR1-4-7 should be applied at the following stage; on the other hand, if one of tr-inf rules TR1-1, TR1-2 or TR1-3 has been applied after tr-inf rule TR1, there is no tr-inf rule to be applied at the following stage because bf-annotations db (0, 12) or mb(1, 11) between processes Pr_i and Pr_k have already been derived.

(II) the following eight tr-inf rules,

TR1-4-2	(81),	**TR2-2**	(85),
TR1-4-3-2	(82),	**TR2-3-2**	(86),
TR1-4-6	(83),	**TR2-5**	(87),
TR1-5-1	(84),	**TR3-1**	(88)

have no following rule to be applied at the following stage, even though they cannot derive the final bf-relations between processes represented by bf-annotations such as jb(2, 10)/ja(10, 2). For example, suppose that tr-inf rule TR1-4-3-2 has been applied, then the vector annotation (2, 8) of the bf-literal (p_i, p_k, t) just implies that the final bf-relation between processes Pr_i and Pr_k is one of three bf-annotations, jb(2, 10), sb(3, 9) and ib(4, 8). Therefore, if one of the above eight tr-inf rules has been applied, one of ba-inf rules (0, 8)-rule, (2, 8)-rule or (5, 5)-rule should be applied for deriving the final bf-annotation at the following stage. For instance, if tr-inf rule TR1-4-3-2 has been applied, ba-inf rule (2,8)-rule should be applied at the following stage.

Now we show a simple example of bf-relation reasoning by tr-inf rules taking the process time chart 3(bottom left) in Fig. 17.

Example 4 **At time** t_1, tr-inf rule TR1 is applied and we have the bf-EVALP clause,

$$R(p_i, p_k, t_1) : [(0, 8), \alpha].$$

At time t_2, tr-inf rule TR1-2 is applied, however bf-literal $R(p_i, p_k, t_2)$ has the same vector annotation (0, 8) as the previous time t_1. Therefore, we have the bf-EVALP clause,

$$R(p_i, p_k, t_2) : [(0, 8), \alpha].$$

At time t_3, no transitive bf-inference rule can be applied, since the vector annotations of bf-literals $R(p_i, p_j, t_3)$ and $R(p_j, p_k, t_3)$ are the same as the previous time t_2. Therefore, we still have the bf-EVALP clause having the same vector annotation,

$$R(p_i, p_k, t_3) : [(0, 8), \alpha].$$

At time t_4, tr-inf rule TR1-2-1 is applied and we obtain the bf-EVALP clause having bf-annotation db(0, 12),

$$R(p_i, p_k, t_4) : [(0, 12), \alpha].$$

6 Conclusions and Remarks

In this chapter, we have introduced paraconsistent annotated logic programs EVALPSN and bf-EVALPSN that was proposed most recently, which can deal with process before-after relations, and we also have introduced the safety verification method and intelligent process control based on EVALPSN/bf-EVALPSN as applications. The bf-EVALPSN safety verification based process order control method can be applied to various process order control systems requiring real-time processing.

An interval temporal logic has been proposed by Allen et al. for knowledge representation of properties, actions and events [1, 2]. In the interval temporal logic, predicates such as *Meets(m,n)* are used for representing primitive before-after relations between time intervals *m* and *n*, and other before-after relations are represented by six predicates such as *Before*, *Overlaps*, etc. It is well known that the interval temporal logic is a logically sophisticated tool to develop practical planning or natural language understanding systems [1, 2]. However, it does not seem to be so suitable for practical real-time processing because before-after relations between two processes cannot be determined until both of them finish. On the other hand, in bf-EVALPSN bf-relations are represented more minutely in paraconsistent vector annotations and can be determined according to start/finish information of two processes in real time. Moreover, EVALPSN can be implemented on microchips as electronic circuits, although it has not introduced in this chapter. We have already shown that some EVALPSN based control systems can be implemented on a microchips in [28, 37]. Therefore, bf-EVALPSN is a more practical tool for dealing with real-time process order control and its safety verification.

In addition to the suitable characteristics for real-time processing, bf-EVALPSN can deal with incomplete and paracomplete knowledge in terms of before-after relation in vector annotations, although the treatment of paracomplete knowledge has not been discussed in this chapter. Furthermore, bf-EVALPSN has inference rules for transitive reasoning of before-after relations as shortly described. Therefore, if we apply EVALPSN and bf-EVALPSN appropriately, various systems should intellectualize more.

References

1. Allen, J.F.: Towards a general theory of action and time. Artif. Intell. **23**, 123–154 (1984)
2. Allen, J.F., Ferguson, G.: Actions and events in interval temporal logic. J. Logic Comput. **4**, 531–579 (1994)
3. Apt, K.R., Blair, H.A., Walker, A.: Towards a theory of declarative knowledge. In: Minker J (ed.) Foundation of Deductive Database and Logic Programs, pp. 89–148 Morgan Kaufmann (1989)
4. Belnap, N.D.: A useful four valued logic. In: Dunn, M., Epstein, G. (eds.) Modern Uses of Multiple-Valued Logic. D.Reidel Publishing, Netherlands (1977)
5. Billington, D.: Defeasible logic is stable. J. Logic and Computation **3**, 379–400 (1993)
6. Billington, D.: Conflicting literals and defeasible logic. In: Nayak A, Pagnucco M (eds.) Proceedings of 2nd Australian Workshop Commonsense Reasoning, pp. 1–15, Australian Artificial Intelligence Institute, Sydney, 1 Dec 1997
7. Blair, H.A., Subrahmanian, V.S.: Paraconsistent logic programming. Theoret. Comput. Sci. **68**, 135–154 (1989)
8. da Costa, N.C.A., Subrahmanian, V.S., Vago, C,: The paraconsistent logics P$\mathcal{T}$. Zeitschrift für Mathematische Logic und Grundlangen der Mathematik **37**, 139–148 (1997)
9. Dressler, O.: An extended basic ATMS, In: Reinfrank, M., et al. (eds.) Proceedings of 2nd International Workshop on Non-Monotonic Reasoning, Grassau, Germany, (Lecture Notes in Computer Science LNCS 346), pp. 143–163. Springer, Heidelberg, 13–15 June 1988

10. Fitting, M.: Bilattice and the semantics of logic programming. J. Logic Program. **11**, 91–116 (1991)
11. Gelder, A.V., Ross, K.A., Schlipf, J.S.: The well-founded semantics for general logic programs. J. ACM **38**, 620–650 (1991)
12. Jaskowski, S.: Propositional calculus for contradictory deductive system (English translation of the original Polish paper). Stud. Logica. **24**, 143–157 (1948)
13. Kifer, M., Subrahmanian, V.S.: Theory of generalized annotated logic programming and its applications. J. Logic Programming **12**, 335–368 (1992)
14. Lloyd, J.W.: Foundations of Logic Programming, 2nd edn. Springer, Berlin (1987)
15. Moore, R.: Semantical considerations on non-monotonic logic. Artif. Intell. **25**, 75–94 (1985)
16. Nakamatsu, K., Suzuki, A.: Annotated semantics for default reasoning. In: Dai, R. (ed.) Proceedings of 3rd Pacific Rim International Conference Artificial Intelligence(PRICAI94), pp.180–186, International Academic Publishers, Beijin, China, 15–18 Aug 1994
17. Nakamatsu, K., Suzuki, A.: A nonmonotonic ATMS based on annotated logic programs. In: Wobcke, W., et al. (eds.) Agents and Multi-Agents Systems (Lecture Notes in Artificial Intelligence LNAI 1441), pp. 79–93. Springer, Berlin (1998)
18. Nakamatsu, K., Abe, J.M.: Reasonings based on vector annotated logic programs. Mohammadian M (ed.) Computational Intelligence for Modelling, Control & Automation (CIMCA99), (Concurrent Systems Engineering Series 55), pp. 396–403. IOS Press, Netherlands (1999)
19. Nakamatsu, K., Abe, J.M., Suzuki, A.: Defeasible reasoning between conflicting agents based on VALPSN. In: Tessier C, Chaudron L (eds.) Proceedings of AAAI Workshop Agents Conflicts, pp. 20–27. Orland, FL, AAAI Press, Menlo Park, 18 July 1999
20. Nakamatsu, K., Abe, J.M., Suzuki, A.: Defeasible reasoning based on VALPSN and its application. In: Nayak A, Pagnucco M (eds.) Proceedings of The Third Australian Commonsense Reasoning Workshop, Sydney, Australia, pp. 114–130. University of Newcastle, Sydney, Australia, 7 Dec 1999
21. Nakamatsu, K.: On the relation between vector annotated logic programs and defeasible theories. Logic and Logical Philosophy **8**, 181–205 (2000)
22. Nakamatsu, K., Abe, J.M., Suzuki, A.: A defeasible deontic reasoning system based on annotated logic programming. In: Dubois, D.M. (ed.) Proceedings 4th International Conference Computing Anticipatory Systems(CASYS2000), Liege, Belgium, (AIP Conference Proceedings 573), pp. 609–620. American Institute of Physics, New York, 7–12 Aug 2001
23. Nakamatsu, K., Abe, J.M., Suzuki, A.: Annotated semantics for defeasible deontic reasoning. In: Ziarko W, Yao Y (eds.) *Proc. 2nd Intl. Conf. Rough Sets and Current Trends in Computing(RSCTC2000)*, 16–19 October, 2000, Banff, Canada, (Lecture Notes in Artificial Intelligence LNAI 2005), pp. 432–440. Springer, Berlin (2001)
24. Nakamatsu, K., Abe, J.M., Suzuki, A.: Extended vector annotated logic program and its application to robot action control and safety verification. In: Abraham, A., et al. (eds.) Hybrid Information Systems (Advances in Soft Computing Series), pp. 665–680. Physica-Verlag, Heidelberg (2002)
25. Nakamatsu, K., Suito, H., Abe, J.M., Suzuki, A.: Paraconsistent logic program based safety verification for air traffic control, In: El Kamel A, et al. (eds.) Proceedings of IEEE International Conference on System, Man and Cybernetics 02(SMC02), Hammamet, Tunisia, IEEE SMC (CD-ROM), 6–9 Oct 2002
26. Nakamatsu, K., Abe, J.M., Suzuki, A.: A railway interlocking safety verification system based on abductive paraconsistent logic programming. In: Abraham, A., et al. (eds.) Soft Computing Systems(HIS02) (Frontiers in Artificial Intelligence and Applications 87), pp. 775–784. IOS Press, Netherlands (2002)
27. Nakamatsu, K., Abe, J.M., Suzuki, A.: Defeasible deontic robot control based on extended vector annotated logic programming. In: Dubois, D.M. (ed.) *Proc. 5th Intl. Conf. Computing Anticipatory Systems(CASYS2001)*, Liege, Belgium, (AIP Conference Proceedings 627), pp. 490–500. American Institute of Physics, New York, 13–18 Aug 2002

28. Nakamatsu, K., Mita, Y., Shibata, T.: Defeasible deontic action control based on paraconsistent logic program and its hardware application. In: Mohammadian, M. (ed.) Proceedings of International Conference Computational Intelligence for Modelling Control and Automation 2003(CIMCA2003), Vienna, Austria, IOS Press, Netherlands (CD-ROM) 12–14 Feb 2003
29. Nakamatsu, K., Seno. T., Abe, J.M., Suzuki, A.: (2003) Intelligent real-time traffic signal control based on a paraconsistent logic program EVALPSN. In: Wang, G., et al. (eds.) Rough Sets, Fuzzy Sets, Data Mining and Granular Computing(RSFDGrC2003), Chongqing, China, (Lecture Notes in Artificial Intelligence LNAI 2639), pp. 719–723. Springer, Berlin, 26–29 May 2003
30. Nakamatsu, K., Komaba, H., Suzuki, A.: Defeasible deontic control for discrete events based on EVALPSN. In: Tsumoto, S., et al. (eds.) Proceedings of 4th International Conference on Rough Sets and Current Trends in Computing(RSCTC2004), Uppsala, Sweeden, (Lecture Notes in Artificial Intelligence LNAI 3066), pp. 310–315. Springer, Berlin, 1–5 June 2004
31. Nakamatsu, K., Ishikawa, R., Suzuki. A.: A paraconsistent based control for a discrete event cat and mouse. In: Negoita, M.G.H., et al. (eds.) Proceedings 8th International Conference Knowledge-Based Intelligent Information and Engineering Systems(KES2004), Wellington, NewZealand, (Lecture Notes in Artificial Intelligence LNAI 3214), pp. 954–960. Springer, Berlin, 20–25 Sept 2004
32. Nakamatsu, K., Chung, S.-L., Komaba, H., Suzuki, A.: A discrete event control based on EVALPSN stable model. In: Slezak D, et al. (eds.) *Rough Sets, Fuzzy Sets, Data Mining and Granular Computing(RSFDGrC2005)*, Regina, Canada, (Lecture Notes in Artificial Intelligence LNAI 3641), pp. 671–681. Springer, Berlin, 31 Aug–3 Sept 2005
33. Nakamatsu, K., Abe, J.M., Akama, S.: An intelligent safety verification based on a paraconsistent logic program. In: Khosla R, et al. (eds.) Proceedings of 9th International Conference Knowledge-Based Intelligent Information and Engineering Systems(KES2005), Melbourne, Australia, (Lecture Notes in Artificial Intelligence LNAI 3682), pp. 708–715. Springer, Berlin, 14–16 Sept 2005
34. Nakamatsu, K., Kawasumi, K., Suzuki, A.: Intelligent verification for pipeline based on EVALPSN. In: Nakamatsu, K., Abe, J.M. (eds.) Advances in Logic Based Intelligent Systems (Frontiers in Artificial Intelligence and Applications 132), pp. 63–70. IOS Press, Netherlands (2005)
35. Nakamatsu, K., Suzuki, A.: Autoepistemic theory and paraconsistent logic program. In: Nakamatsu, K., Abe, J.M. (eds.) Advances in Logic Based Intelligent Systems (Frontiers in Artificial Intelligence and Applications 132), pp. 177–184. IOS Press, Netherlands (2005)
36. Nakamatsu, K., Suzuki, A.: Annotated semantics for non-monotonic reasonings in artificial intelligence—I, II, III, IV. In: Nakamatsu, K., Abe, J.M. (eds.) Advances in Logic Based Intelligent Systems (Frontiers in Artificial Intelligence and Applications 132), pp. 185–215. IOS Press, Netherlands (2005)
37. Nakamatsu, K., Mita, Y., Shibata, T.: An intelligent action control system based on extended vector annotated logic program and its hardware implementation. J. Intell. Autom. Soft Comput. **13**, 289–304 (2007)
38. Nakamatsu, K.: Paraconsistent annotated logic program EVALPSN and its application. In: Fulcher, J., Jain, C.L. (eds.) Computational Intelligence: A Compendium (Studies in Computational Intelligence 115), pp. 233–306. Springer-Verlag, Germany (2008)
39. Nute, D.: Defeasible reasoning. In: Stohr, E.A., et al. (eds.) Proceedings of 20th Hawaii Intlernational Conference on System Science(HICSS87), pp. 470–477. Kailua-Kona, Hawaii, University of Hawaii, Hawaii, 1, 6–9 Jan 1987
40. Nute, D.: Basic defeasible logics. In: del Cerro, L.F., Penttonen, M. (eds.) Intensional Logics for Programming, pp. 125–154. Oxford University Press, UK (1992)
41. Nute, D.: Defeasible logic. In: Gabbay, D.M., et al. (eds.) Handbook of Logic in Artificial Intelligence and Logic Programming 3, pp. 353–396. Oxford University Press, UK (1994)
42. Nute, D.: Apparent obligatory. In: Nute, D. (ed.) Defeasible Deontic Logic, (Synthese Library 263), pp. 287–316. Kluwer Academic Publisher, Netherlands (1997)

43. Przymusinski, T.C.: On the declarative semantics of deductive databases and logic programs. In: Minker, J. (ed.) Foundation of Deductive Database and Logic Programs, pp. 193–216. Morgan Kaufmann, New York (1988)
44. Reiter, R.: A logic for default reasoning. Artif. Intell. **13**, 81–123 (1980)
45. Shepherdson, J.C.: Negation as failure, completion and stratification. In: Gabbay, D.M., et al. (eds.) Handbook of Logic in Artificial Intelligence and Logic Programming 5, pp. 356–419. Oxford University Press, UK (1998)
46. Subrahmanian, V.S.: Amalgamating knowledge bases. ACM Trans. Database Syst. **19**, 291–331 (1994)
47. Subrahmanian, V.S.: On the semantics of qualitative logic programs. In: Proceedings the 1987 Symp. Logic Programming(SLP87), IEEE Computer Society Press, pp. 173–182. San Francisco, CA, 31 Aug–4 Sept 1987
48. Visser, A.: Four valued semantics and the liar. J. Philosophical Logic **13**, 99–112 (1987)

The New Hardware Structure of the Emmy II Robot

Cláudio Rodrigo Torres and Régis Reis

Abstract This work presents an implementation of the Emmy II Autonomous Mobile Robot [1–3] control system in a new hardware structure. The main objective of this robot is to avoid reaching any obstacle in a non-structured environment. The control system is based on the Paraconsistent Annotated Evidential Logic—Eτ. In this work, it is also detailed the mechanical platform used in the robot and the tests performed.

Keywords Paraconsistent annotated evidential logic · Autonomous mobile robot · Control system

1 Introduction

This work presents an implementation of the Emmy II autonomous mobile robot control system in a new hardware structure. The main objective of this robot is to avoid reaching any obstacle in a non-structured environment.

The project presented here is an evolution of the Emmy II robot [1–4]. The Emmy II robot is an autonomous mobile robot able to move in a non-structured environment avoiding collisions. Its control system is based on the Paraconsistent Annotated Evidential Logic Eτ.

Basically, the robot proposed here is similar to the Emmy II, but with a modern hardware structure. The objective is to build a new Emmy II robot able to receive upgrades in its functionalities.

This paper is divided as the follow: first there is a description of the Paraconsistent Annotated Evidential Logic Eτ; afterwards there is a description of

C.R. Torres (✉) · R. Reis
Methodist University of São Paulo, São Paulo, Brazil
e-mail: c.r.t@uol.com.br

R. Reis
e-mail: regis-reis@uol.com.br

J.M. Abe (ed.), *Paraconsistent Intelligent-Based Systems*,
Intelligent Systems Reference Library 94, DOI 10.1007/978-3-319-19722-7_4

the autonomous mobile robot Emmy II. At the end of the text, there is a description of the hardware structure proposed and a description of tests performed.

2 Paraconsistent Annoted Evidential Logic—Eτ

Paraconsistent Logics is a kind of logics that allows contradictions without trivialization. A branch of it, the Paraconsistent Annotated Evidential Logic Eτ, which is employed in this work, also deals with the concept of fuzziness. Its language consists on propositions (p) in the usual sense together with annotation constants: (μ, λ) where μ, λ ∈ [0, 1] (real unitary interval). Thus, an atomic formula of the Logic Eτ is of the form p(μ, λ) which can be intuitively read: the favorable evidence of p is μ and the contrary evidence of p is λ. A detailed description of the subject is found in [5–11].

The Favorable Evidence Degree (μ) is a value that represents the favorable evidence in which the sentence is true; this value is between 0 and 1.

The Contrary Evidence Degree (λ) is a value that represents the contrary evidence in which the sentence is true; this value is between 0 and 1.

Through the Favorable and Contrary Degrees, it is possible to represent the four extreme logic states, as shown in the Fig. 1.

The four extreme logic states are: True (V), False (F), Paracomplete (⊥) and Inconsistent (T).

In [12, 13] it is proposed the Para-analyzer Algorithm. By this algorithm it is also possible to represent the non-extreme logic state. The Fig. 2 shows this.

The eight non-extreme logic states are: Quasi-true tending to Inconsistent—QV → T, Quasi-true tending to Paracomplete—QV → ⊥, Quasi-false tending to Inconsistent—QF → T, 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 and Quasi-paracomplete tending to False—Q⊥ → F.

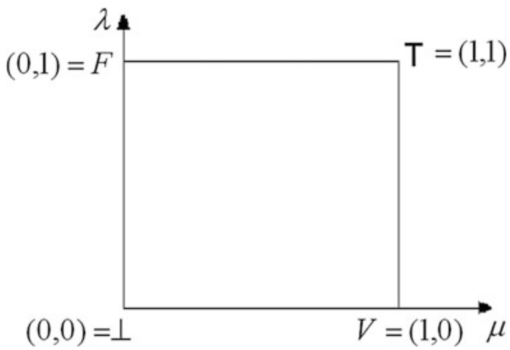


Fig. 1 The extreme logic states

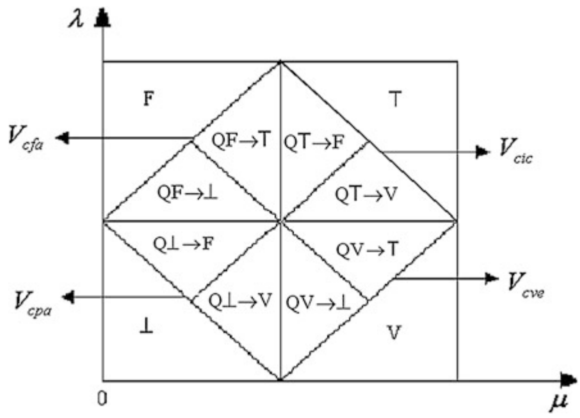


Fig. 2 The non-extreme logic states

It is also defined the Uncertainty Degree: Gun(μ, λ) = μ + λ − 1 and the Certainty Degree: Gce(μ, λ) = μ − λ (0 ≤ μ, λ ≤ 1).

Some additional control values are: Vcic = maximum value of uncertainty control, Vcve = maximum value of certainty control, Vcpa = minimum value of uncertainty control and Vcfa = minimum value of certainty control.

3 Autonomous Mobile Robot Emmy II

The Emmy II robot is an autonomous mobile robot able to avoid obstacle while it is moving in a non-structured environment.

The control system of the Emmy II uses six logic states instead of 12 logic states used in the Para-analyzer Algorithm.

Two sensors are responsible for verify whether there is any obstacle in front of the robot or not. The signals generated by the sensors are sent to a microcontroller. These signals are used to determine the favorable evidence degree (μ) and the contrary evidence degree (λ) on the proposition "The front of the robot is free". The favorable and contrary evidence degrees are used to determine the robot movements.

The signal generated by the sensor 1 is considered the favorable evidence degree and the signal generated by the sensor 2 is considered the contrary evidence degree of the proposition "The front of the robot is free". When there is an obstacle near the sensor 1 the favorable evidence degree is low and when there is an obstacle far from the sensor 1 the favorable evidence degree is high. Otherwise, when there is an obstacle near the sensor 2 the contrary evidence degree is high and when there is an obstacle far from the sensor 2 the contrary evidence degree is low. The Emmy II controller decision of what movement the robot should perform is based on the reticulated showed in Fig. 3.

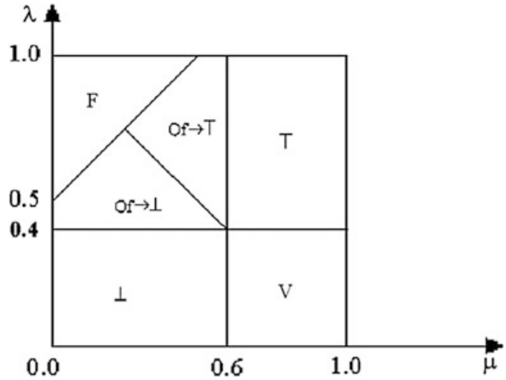


Fig. 3 Lattice of Emmy II controller

The decision for each logic state is the following:

- Robot goes ahead. DC motors 1 and 2 are supplied for spinning around forward.
- Robot goes back. DC motors 1 and 2 are supplied for spinning around backward.
- Robot turns right. Just DC motor 1 is supplied for spinning around forward.
- Robot turns left. Just DC motor 2 is supplied for spinning around forward.
- Robot turns right. Just DC motor 2 is supplied for spinning around backward.
- Robot turns left. Just DC motor 1 is supplied for spinning around backward.

The justification for each decision is the following:

When the logic state is true (V), it means that the front of the robot is free. Therefore, the robot can go ahead.

In the inconsistency (T), μ and λ are high (i.e., belong to T region). It means that the sensor 1 is far from an obstacle and the sensor 2 is near an obstacle, so the left side is more free than the right side. Then, the behavior should be to turn left by supplying only the DC motor 2 for spinning around forward and keeping the DC motor 1 stopped.

When the Paracompleteness (⊥) is detected, μ and λ are low. It means that the sensor 1 is near an obstacle and the sensor 2 is far from an obstacle, so the right side is more free than the left side. Then, the behavior should be to turn right by supplying only the DC motor 1 for spinning around forward and keeping the DC motor 2 stopped.

In the false state (F) there are obstacles near the front of the robot. Therefore the robot should go back.

In the QF → T state, the front of the robot is obstructed but the obstacle is not so near as in the false state and the left side is a little bit more free than the right side. So, in this case, the robot should turns left by supplying only the DC motor 1 for spinning around backward and keeping the DC motor 2 stopped.

In the QF → ⊥ state, the front of the robot is obstructed but the obstacle is not so near as in the false state and the right side is a little bit freer than the left side. So, in

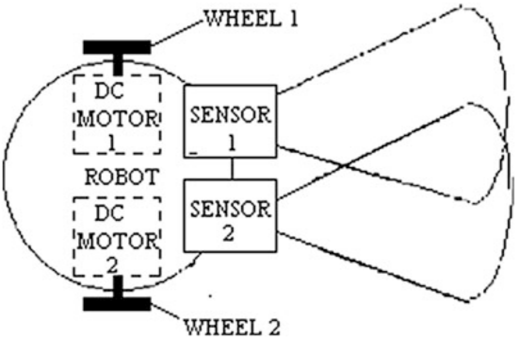


Fig. 4 The Emmy II robot basic structure

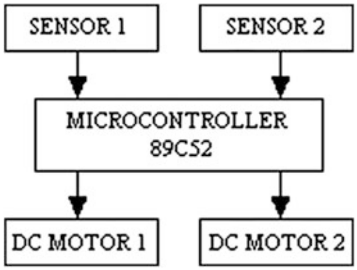


Fig. 5 Emmy II block representation

this case, the robot should turns right by supplying only the DC motor 2 for spinning around backward and keeping the DC motor 1 stopped.

The basic structure of the Emmy II robot is showed in the Fig. 4.

The Emmy II controller system uses six logic states instead of 12 logic states used in the Emmy I controller. Moreover, it may present some commands that do not exist in the Emmy I robot:

1. Velocity control: the Emmy II controller allows the robot to brake, turn and accelerate "in a smoothly way" what is not possible in the Emmy I robot.
2. The Emmy II controller allows backward motion. In some situations the robot may move backward or turns with a fixed wheel and the other spinning around backward. There are not these types of movements in the Emmy I robot.

It can be seen in the Fig. 5 a simplified block representation of Emmy II robot.

The Fig. 6 shows a picture of the robot Emmy II

It is shown in the Fig. 7 the down part of the Emmy II robot.

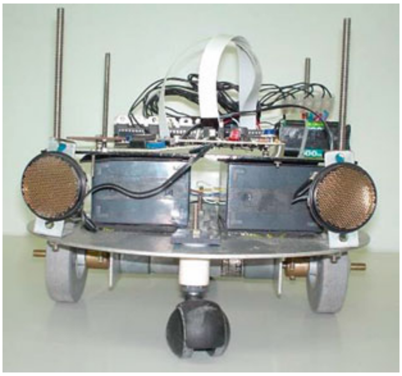


Fig. 6 The front part of the Emmy II robot

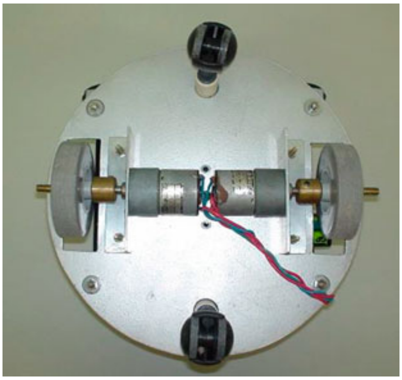


Fig. 7 The down part of the Emmy II robot

3.1 Tests of Emmy II Robot

Aiming to verify Emmy II robot functionally, it has been performed 4 tests. Basically, counting how many collisions there were while the robot moved in an environment as showed in Fig. 8.

The time duration and results for each test have been the following:

- Test 1: Duration: 3 min and 50 s. Result: 13 collisions.
- Test 2: Duration: 3 min and 10 s. Result: 7 collisions.
- Test 3: Duration: 3 min and 30 s. Result: 10 collisions.
- Test 4: Duration: 2 min and 45 s. Result: 10 collisions.

The sonar ranging modules used in the Emmy II robot can't detect obstacles closer than 7.5 cm. The sonar ranging modules transmit sonar pulses and wait for them to return (echo) so that it can determine the distance between the sonar ranging modules and the obstacles; however, sometimes the echo doesn't return,

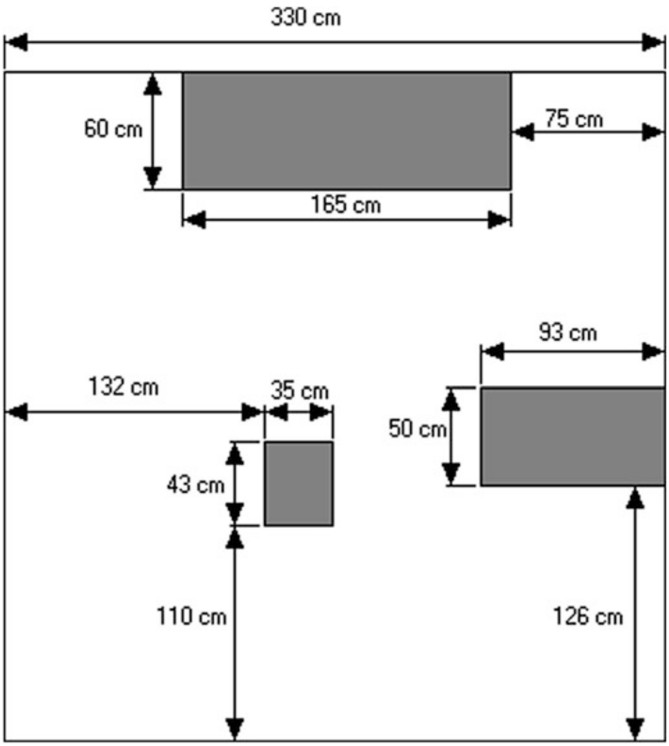


Fig. 8 Environment used to perform the Emmy II tests

because it reflects to another direction. These are the main causes for the robot collisions:

Test 1: Collisions: 13.
Collisions caused by echo reflection: 4.
Collisions caused by too near obstacles: 9.

Test 2: Collisions: 7.
Collisions caused by echo reflection: 2.
Collisions caused by too near obstacles: 5.

Test 3: Collisions: 10.
Collisions caused by echo reflection: 5.
Collisions caused by too near obstacles: 5.

Test 4: Collisions: 10.
Collisions caused by echo reflection: 4.
Collisions caused by too near obstacles: 6.

There is another robot collision possibility: when the robot is going back. As there is no sonar ranging module behind the robot, it may collide.

Fig. 9 The mechanical platform

4 The New Emmy II Hardware Structure

The mechanical platform showed in the Fig. 9 had been built in order to perform the Emmy II control system algorithm.

The platform is composed of three subsystems: Control and Planning Subsystem, Moving Subsystem and Sensing Subsystem. These subsystems work together although each one is responsible for a part of whole system. The robot system may be modelled as presented in Fig. 10.

4.1 Moving Subsystem

The Moving Subsystem is composed of a chassis, 4 wheels, 4 DC motors and potency drivers for DC motors supply. The chassis is a metallic structured aiming to

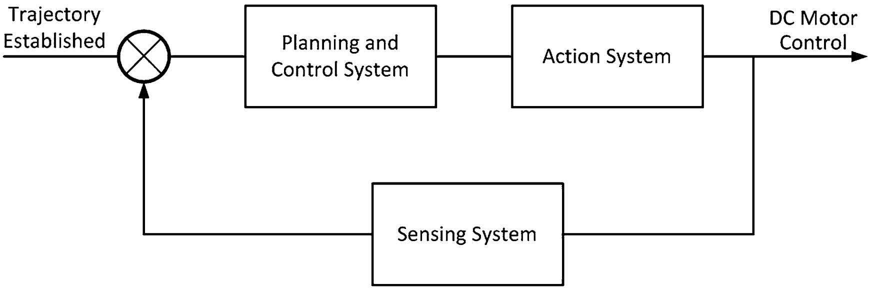


Fig. 10 The robot system model

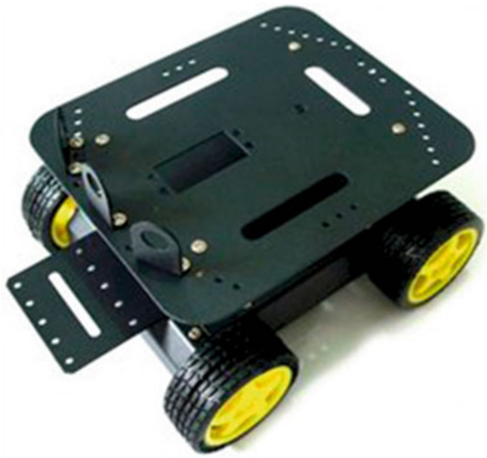


Fig. 11 The robot chassis

fix all robot devices. It is possible to see in Fig. 11 the chassis with the DC motors and wheels fixed.

The four DC Motors are of low potency and consumption because the robot is projected to move in smooth surfaces. The chosen DC motor model is the DFRobot130 with mechanical reduction by gears. This DC motor has the following operational characteristics:

- Gear ratio: 1:120.
- No-load speed (6 V): 180 rpm.
- No load current (6 V): 160 mA.
- Locked-rotor current (6 V): 2.8 A.
- Size: Long. 55 mm; Width 48.3 mm; High 23 mm.
- Weight: About 45 g.

The DC motors are supplied by signals from the Control and Planning Subsystem. These signals must be amplified because they are of low potency. The signals can be sent to the DC motors only after the amplification.

The amplification is made by two potency drivers as showed in Fig. 12. The main component of the potency driver is the integrated circuit L298N from ST Electronics. This component is composed of two H bridge encapsulated in just one involucre. And has the objective of amplify the electric current and control the DC motor rotation direction.

The electric circuitry used for supplying the DC motors of the right wheels (driver 1) is showed in Fig. 13.

It is used an electric circuitry similar to the one showed in Fig. 10 for supplying the DC motors connected to the left wheels.

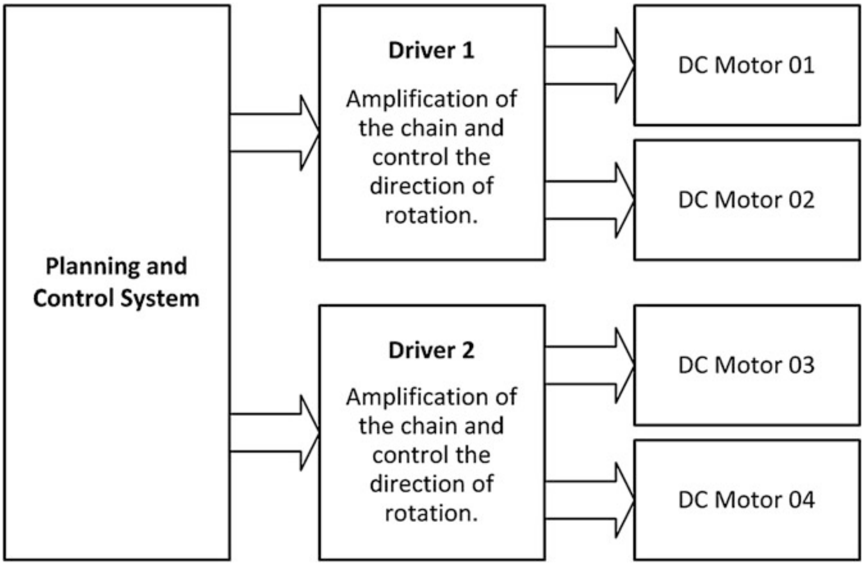


Fig. 12 DC motors supply structure

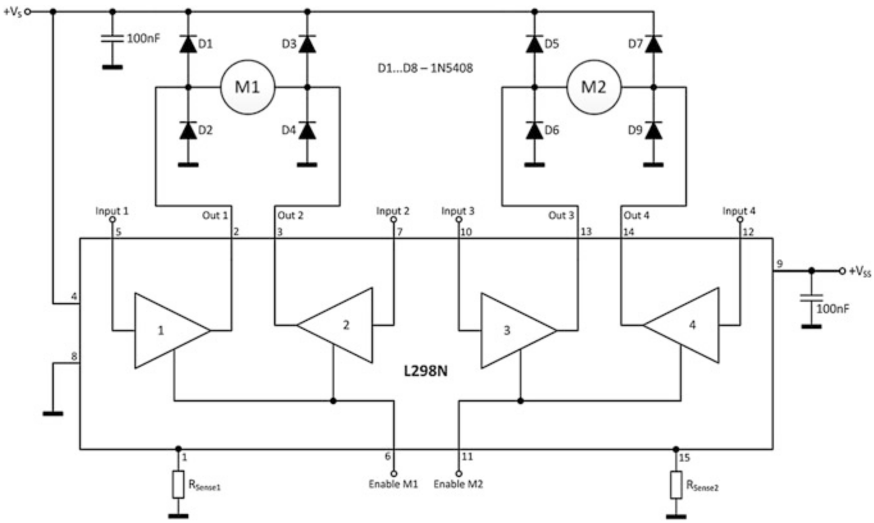


Fig. 13 Electric circuitry used for supplying the DC motors

4.2 Sensing Subsystem

The Sensing Subsystem is composed of three ultrasonic sensors and two encoders connected to the robot frontal wheels. The ultrasonic sensors are of the model PING (((from Parallax and are fixed on the frontal part of the robot. They are fixed as showed in Fig. 14.

The ultrasonic sensor may cover a range of 100º as showed in Fig. 15.

The Parallax Ping(((sensor is able to measure an interval from 2 to 300 cm precisely. The sensor emits ultrasonic waves of 40 kHz for 200 µs. A microcontroller determines when the ultrasonic wave emission starts. The sensor starts sending a signal to the microcontroller when the ultrasonic wave emission is finished. This signal sending is finished when the ultrasonic sensor receive the echo of the ultrasonic waves emitted before. Then, the microcontroller may determine the distance of the object from the sensor by the duration of the signal sent by the sensor. Figures 16 and 17 illustrate the process of obstacle detection.

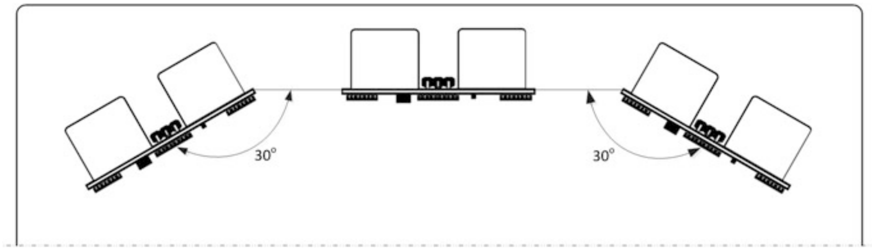


Fig. 14 Ultrasonic sensors robot position

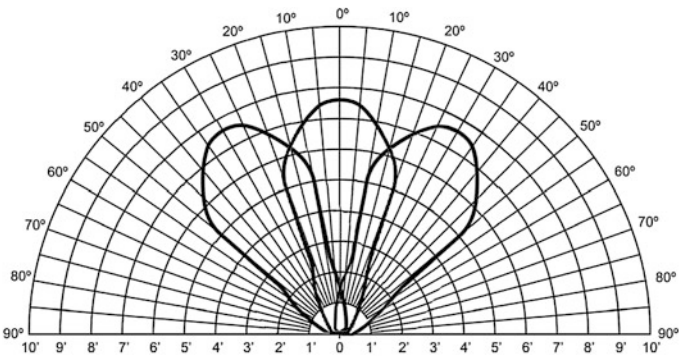


Fig. 15 Range covered by the ultrasonic sensor on the robot (*Source* Parallax Ping(((datasheet)

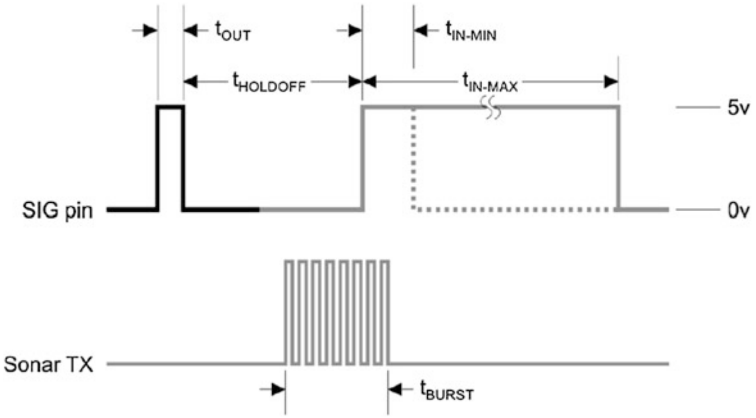


Fig. 16 Ultrasonic sensor process of wave emission and echo receiving. (*Source* Parallax Ping(((datasheet)

	Host Device	Input Trigger Pulse	t_{OUT}	2 μs (min), 5 μs typical
	PING))) Sensor	Echo Holdoff	$t_{HOLDOFF}$	750 μs
		Burst Frequency	t_{BURST}	200 μs @ 40 kHz
		Echo Return Pulse Minimum	t_{IN-MIN}	115 μs
		Echo Return Pulse Maximum	t_{IN-MAX}	18.5 ms
		Delay before next measurement		200 μs

Fig. 17 Ultrasonic sensor time intervals of the emission and receiving ultrasonic waves (*Source* Parallax Ping(((datasheet)

4.3 Control and Planning Subsystem

The control algorithm is implemented in this subsystem. The aim of this subsystem is to receive the signals from the Sensing Subsystem, processes them and sends the results of the control algorithm to the Moving Subsystem.

This subsystem is composed of a Digilent ChipKit Uno32 prototyping platform. A PIC32MX320F128 is the microcontroller which process the programs implemented in the prototyping platform. Figure 18 shows the prototyping platform.

The subsystems presented in this text are also projected to be used in the Emmy III robot in the future. For Emmy III robot, the control subsystem is going to determine a track for the robot. The Emmy III control system is based on Paraconsistent Neural Network.

Fig. 18 Digilent ChipKit Uno32 prototyping platform

5 Tests of the Emmy II New Hardware Architecture

Aiming to verify the behavior of the new Emmy II hardware structure, many tests were performed. These tests consist of the observation of the robot behavior when there is an obstacle in front of the robot while it is moving forwards. So, it was possible to verify if the robot took the right decision.

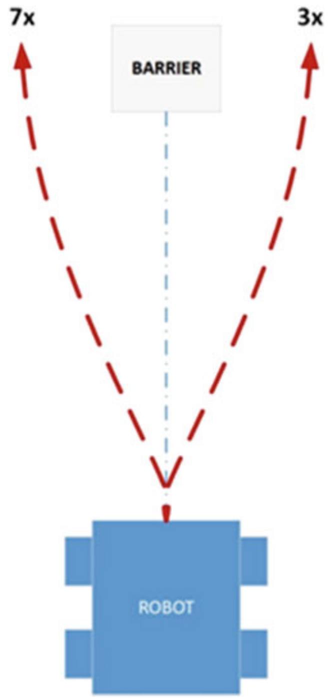


Fig. 19 Representation of the first type of test performed by the robot

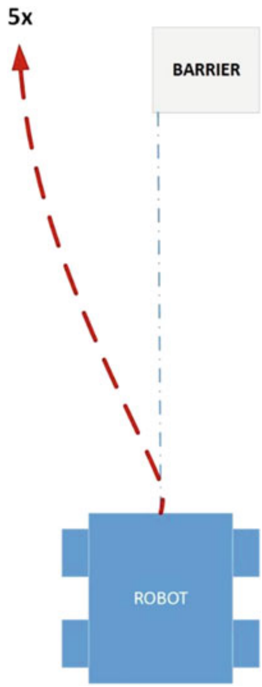


Fig. 20 Representation of the second type of test performed by the robot

At first, it was put an obstacle exactly in front of the robot as showed in Fig. 19. In this case it was expected that the evidence degrees μ and λ were the same value (both equal to 0, 5). In a situation like that, the robot must turns left.

Anyway, in a real test condition, it is must difficult to reach a situation that μ is equal to λ. It was expected that there were a small difference between them. So, the robot should turn to the left or to the right as μ or λ were bigger. This experiment were repeated 10 times and in seven occasions the robot turned to the left and in three occasions the robot turned to the right

In the second type of test, there was an obstacle in the right side of the robot as showed in the Fig. 20. In a situation like that, it is expected that the robot turn to the left. Five tests were performed and in all cases, the robot turned to the left.

In the last type of test, an obstacle was put in the right side of the robot as showed in the Fig. 21. In five occasions, the robot turned to the left as expected.

6 Conclusions

This text presents a new hardware structure for the Emmy II robot. This new hardware architecture is an evolving of the robot proposed in 2004. The Emmy II control system is based on the Paraconsistent Annotated Evidential Logic Eτ.

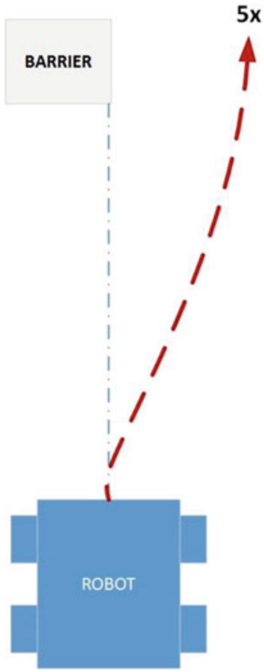


Fig. 21 Representation of the third type of test performed by the robot

The proposed robot is divided into three subsystems: Control and Planning Subsystem, Sensing Subsystem and Moving Subsystem.

The Sensing Subsystem is responsible to capture information from the environment around the robot and sent them to the Control and Planning Subsystem. The Moving Subsystem is composed of DC motors and wheels. The objective of the Control and Planning Subsystem is to receive signals from the Sensing Subsystem, process them and control the Moving Subsystem.

The functioning of this new robot was satisfactory.

The structured proposed for this robot is going to be used in future projects, as the building of the Emmy III [14–19].

References

1. Abe, J.M., Torres, C.R., Lambert-Torres, G., Nakamatsu, K., Kondo, M.: Intelligent paraconsistent logic controller and autonomous mobile robot Emmy II. Lect. Notes Comput. Sci. **4252**, 851–857 (2006)
2. Abe, J.M., Torres, C.R., Lambert-Torres, G., Nakamatsu, K., Kondo, M.: Intelligent paraconsistent logic controller and autonomous mobile robot Emmy II. In: Proceedings of the 10th International Conference on Knowledge-Based, Intelligent Information & Engineering Systems, KES2006. KES Pub., Bournemouth, UK (2006)

3. Torres, C.R., Lambert-Torres, G., da Silva, L.E.B., Abe, J.M.: Intelligent system of paraconsistent logic to control autonomous moving robots. In: Proceedings of the 32nd Annual Conference of the IEEE Industrial Electronics Society, IECON'06. IEEE Press, Paris (2006)
4. Torres, C.R.: Sistema inteligente baseado na lógica paraconsistente anotada Eτ para controle e navegação de robôs móveis autônomos em um ambiente não estruturado, in Portuguese. Ph.D. thesis, Federal University of Itajubá, Itajubá, MG, Brazil (2010)
5. Abe, J.M., Lambert-Torres, G., da Silva Filho, J.I., Torres, C.R., Martins, H.G.: Paraconsistent autonomous mobile robot Emmy III. In: Proceedings of the 6th Congress of Logic Applied to Technology—LAPTEC'2007, São Paulo, Brazil (2007)
6. Abe, J.M.: Fundamentos da lógica anotada, (Foundations of annotated logics), in Portuguese. Ph.D. thesis, University of São Paulo, São Paulo (1992)
7. Abe, J.M., da Silva Filho, J.I.: Manipulating conflicts and uncertainties in robotics. Multiple-Valued Logic and Soft Comput. **9**, 147–169 (2003). ISSN 1542–3980
8. da Silva Filho, J.I., Abe, J.M.: Para-control: an analyser circuit based on algorithm for treatment of inconsistencies. In: Proceedings of the World Multiconference on Systemics, Cybernetics and Informatics, ISAS, SCI 2001, vol. XVI, Cybernetics and Informatics: Concepts and Applications (Part I), pp. 199–203, Orlando, Florida, USA (2001). ISBN 9800775560
9. da Silva Filho, J.I., Abe, J.M.: Paraconsistent analyzer module. Int. J. Comput. Anticipatory Syst. **9**, 346–352 (2001). ISSN 1373-5411, ISBN 2-9600262-1-7
10. da Silva Filho, J.I., Abe, J.M.: Para-control: an analyser circuit based on algorithm for treatment of inconsistencies. In: Proceedings of the World Multiconference on Systemics, Cybernetics and Informatics, ISAS, SCI 2001, vol. XVI, Cybernetics and Informatics: Concepts and Applications (Part I), pp. 199–203, Orlando, Florida, USA (2001), ISBN 9800775560
11. da Silva Filho, J.I., Abe, J.M.: Paraconsistent analyzer module. Int. J. Comput. Anticipatory Syst. **9**, 346–352, (2001) ISSN 1373-5411, ISBN 2-9600262-1-7
12. da Silva Filho, J.I., Abe, J.M.: Emmy: a paraconsistent autonomous mobile robot. In: Laptec' 2001 The 2nd Congress of Logic Applied to Technology, São Paulo, Brazil. Logic, Artificial Inteligence and Robotics, vol. 1. pp. 53–61. IOS Press, Amsterdam, Holanda (2001)
13. da Silva Filho, J.I.: Métodos de Aplicações da Lógica Paraconsistente Anotada de Anotação com Dois Valores LPA2v com Construção de Algoritmo e Implementação de Circuitos Eletrônicos", in Portuguese. Ph.D. thesis, University of São Paulo, São Paulo (1999)
14. Torres, C.R., Abe, J.M., Lambert-Torres, G, da Silva Filho, J.I.: Autonomous mobile robot Emmy III. In: Gacovski, Z. (ed.) Mobile Robots—Current Trends (2011). ISBN: 978-953-307-716-1, InTech http://www.intechopen.com/articles/show/title/autonomous-mobile-robot-emmy-iii
15. Torres, C.R., Lambert-Torres, G., Abe, J.M., da Silva Filho, J.I.: The sensing system for the autonomous mobile robot Emmy III. In: Proceedings of 2011 IEEE International Conference On Fuzzy Systems, 2011, Taipei-Taiwan, pp. 2928–2933 (2011). http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=6007529
16. Torres, C.R., Lambert-Torres, G., da Silva Filho, J.I., Martins, H.G.: Results of a sensing system for an autonomous mobile robot based on the paraconsistent artificial neural network. Int. J. Reasoning Based Intell. Syst. **3**, 108–114 (2011)
17. Torres, C.R., Aplicações da Lógica Paraconsistente Anotada Evidencial Et em Robôs Móveis Autônomos In: Aspectos de Lógica e Teoria da Ciência.1 ed. São Paulo : Instituto de Estudos Avançados da Universidade de São Paulo, 2011, vol. 1, pp. 84−139. Referências adicionais: Brasil/Português. Meio de divulgação: Meio digital, ISBN: 9788563007025, http://www.iea.usp.br/textos/livroaspectosdelogica.pdf

18. Torres, C.R., Abe, J.M., Lambert-Torres, da Silva Filho, J.I., Martins, H.G.: A sensing system for an autonomous mobile robot based on the paraconsistent artificial neural network. Lect. Notes in Comput. Sci. Springer, Berlin, vol. 6278, pp. 154–163 (2010)
19. Torres, C.R., Abe, J.M.; Torres, G.L., da Silva Filho, J.I., Martins, H.G.: Autonomous mobile robot Emmy III. In: Nakamatsu, K., Phillips-Wren, G., Jain, L.C., Howlett, R.J. (Org.) New Advances in Intelligent Decision Technologies, vol. 199, 1st edn, pp. 317–327. Springer, Helderberg (2009)

A Suggestion for Sample Size Determination Using an Instrument for Collecting Medical Data Based on Discrete Items

Euro de Barros Couto Junior

Abstract This text suggests on how to calculate a sample size under the use of an instrument for collecting data formed by check-block items. The arguments for this suggestion are based on Combinatorics and Paraconsistency theories. Our purpose is to suggest a practical and simple calculation procedure to obtain an acceptable sample size to collect information, organize it, and analyze data from an application of an instrument for collecting medical data, based exclusively on discrete items (categorical items), i.e., each instrument item is considered a non-parametric variable with finite number of categories. In Bio-sciences it is very common to use survey instruments based on this type of items: clinical protocols, hospital registers, questionnaires, and other inquiring tools consider a sequence of organized categorical items.

Keywords Sample size · Discrete counting · Non-parametric variables · Paraconsistency · Combinatorics

1 Introduction

History has shown that, in all ages, Science presents some development, which, in turn, comes as part of the Human Culture, a key-point to differentiate humans from other living beings.

And the culture produced its fruit since the human being realized himself. The beginnings of Culture already aggregated knowledge that exceeded the instincts. The human soul was being made up of items less instinctive and more aware, ending in the construction of human thought and away itself from ignorance. Nature provided humans this ability: it was not a mere act of thinking, but rather the construction of thought. And so, the human race was distinguished as such, going to

E. de Barros Couto Junior (✉)
Municipal Government of São Paulo, São Paulo, Brazil
e-mail: estatisticoeuro@gmail.com

J.M. Abe (ed.), *Paraconsistent Intelligent-Based Systems*,
Intelligent Systems Reference Library 94, DOI 10.1007/978-3-319-19722-7_5

dominate their habitat and making it fertile for the development of more complex thoughts.

Hence, the observation of Nature's facts led humans to propose their registry, through symbols, and symbols live at greater length than humans [1]. This registry, in turn, provided a description thereof; in being describable, each fact can in general be seen again, and this relatively simplified sequence generated the thin thread on which emerged the basis of science, which, according to René Descartes (1596–1650) is, in its entirety, the truth and clear cognition [2]. The entirety that refers not only to Descartes, but almost all the great thinkers, concerns the uniqueness of Science, as one and indivisible body, in the broad sense of aggregation of knowledge endowed by humanity.

Thus, in a contemporary view, the integration of sciences, treated each one as fractions of a whole, is undoubtedly the very nature of the maximum human creation, which is the Knowledge. The Science and the Art, as such present themselves, are, in this respect, the faces of that knowledge, as they comprise pure and unique manifestations of the human spirit, in the same thread before mentioned. Here, it is worth emphasizing the human capacity for intellection of self and understanding of its own existence. So, humans are recognized as single records of history, and we were (and are) capable of producing art as much as science, as they are both, indeed, aspects of the same core of knowledge. If today's vision is distorted, it is because it had no stopping the rescue of the historical facts of a glorious past—and at the same time, vile—for which Humanity designed to itself in the present. Blaise Pascal (1623–1662), in his scientific treatise on the vacuum, already states that Geometry, Arithmetic, Music, Physics, Medicine, Architecture and all sciences submitted to experimentation and reason should be considered because they became perfect [2].

The formal partition of Science, under a certain point of view, provided the appearance of knowledge's specificities, but not disunited them, and not deprived it of unity: Science remains as an unity, and the varied and various aspects of Knowledge, long before, propose to be an internal method of organization of Science more than divisions of it. Thus, Science continues to be one, as ever, and as always will be.

Although those specificities of Science show in having to come to benefit the human being, the totalized vision on these conjoint problems of these specificities is presented today, as necessary, innovative and beneficial. Thus, it can be seen, clearly, the existence of component elements of Science, which are known each one as a specific science, for their specific nature. We also consider the integration of these elements as co-participants of a sustainable and objective integration. Among these component elements of Science, we consider the undeniable presence of Statistics as a helper tool (no demerits!) of a wide range of other sciences, and in this case, a tool for a long, widely applied in the Health Area (Biology, Medicine, Nursing, Physiotherapy, Occupational Therapy, Speech Therapy, Psychology, Physical Education, Veterinary etc.), which, in turn, form a vast and solid block of matters for Science itself.

Before the scientific approach in which landed Statistics, after the Renaissance, its manifestation, within civilized societies, concerned the raw count of facts and occurrences of social and governmental nature: the public money, the disease and its incidence and prevalence, the significant social events (births, deaths, marriages, population concentration and migration etc.) were the very Statistics, but it was far from a sampling approach that only emerged in the mid-seventeenth century with John Graunt (1620–1674) in England.

According to the eminent Prof. Aúthos Pagano, Brazilian statistical emeritus, nothing prevents that Statistics be applied to other sciences (…), *and in this case, which is real, it can be in the orbit of any other science* [3]. In an interview with Leslie Kish [4], on July 22nd and 23rd (1994), he declared these words as a great and renowned statistician: *(…) And now a Statistics course is necessary? I think so. HG Wells said the statistical thinking will one day as needed to form a citizen as much as having the ability to read and write—I believe this is necessary, and I said, in a letter written in 1994 on the medical research that a course like this would be very important for physicians, but a course is not enough to learn how to build an experiment or to make clinical surveys. For any level of an education in science—medical, social, physical—a Statistics course would be useful for those who could understand on what they read in articles and magazines, and these people should know enough of statistical language to be able to consult experts to help them prepare the map of their experiments. They should not, however, prepare these maps for themselves because one does not do a surgery with only a single course on this subject. (…)*

With the eyes on Statistics as a science, one can briefly consider it as composed of three parts: Sampling, Sample Description and Statistical Analysis itself.

Sampling consists of a process whose origin dates back millennia and whose interests almost always resulted in its own results, i.e. its ability to account for integer values, however, from a procedural point of view, it should consider that the sampling technique presents itself, and much less it is its interest in obtaining the account value, but rather on achieving create the method under which the sampling process should be developed. Therefore, the sampling itself reaches the procedural level and tends to be at least an equivalent tool to the other components of Statistics, as the Science itself. In this sense, modernly, we would be even glimpse the Sampling as apart from the science of Statistics, since their ways run in different directions, belonging to the fields of other sciences.

Regarding Sample Description, it should be considered the need for conjugation of sampled values and their respective reduced representations, i.e., the extensive use of summary measures characterizing the collected sample, with the proposal to summarize the collected data, producing relevant observation points. In this sense, the reducing character of Statistics arises: it has the power to describe a mass of collected data through summary measures; these measures should be arranged in lists, tables, charts, or even indicators drawings of densities, concentrations and trends.

Sir Ronald Aylmer Fisher (1890–1962) has stated that the chosen statistical method should summarize all relevant information provided by the sample.

What about statistical analysis itself, one can refer to it as necessary for the evidential character in front of a built hypothesis. In this regard, evidencing character core becomes essential for the development of analytical techniques. The analysis, as a scientific matter, and in this case, as a statistical matter, can be seen as segregated since it is, in philosophical terms, intended to be the turning point for decision-making. Science itself is seen today as having a most outstanding decision-making character than strictly procedural, but it is well known that the scientific process allows the production and the existence of a grounded decision-making and a secure content.

Even at this early exposure, it is necessary to highlight Sampling as the core of this approach since the sampling process is the most important step in the implementation of Statistics on the other component elements of scientific research: *a good sample leads to good results…*

1.1 Comments on Some Common Concepts

Currently, it is deposited on the random variable[1] the ultimate responsibility of being the defining basis of sample size because it is a statistical entity that carries in its essence the necessary characteristics to obtain prior knowledge about what will be covered in scientific research.

However, if one wants to propose (before challenging!) that the core of the random variable, represented by previously fixed parameters, but not necessarily a priori known, it is, in fact, one of several approaches which may be proved by the fact that the random variable is the most punctual element of scientific research (discounting, of course, its own parameters).

We adopted the definition of random variable as a function of mathematical nature that involves the operations of Set Theory, because in its most general sense, there must be a list of mathematical criteria to arrive at this definition, which sometimes may not be sufficient to achieve a scientific thought which overlaps the idea that the items of the survey data collection instrument are truly random variables, as suggested by this definition. On the other hand, the random variable consists in a nest of aggregated parameters because without it, it could not be represented by values, which effectively exist in the application of mathematical function to which it is associated.

The instrument for collecting data can be edited on paper, or be a virtual computer screen, for the appointment of the values observed during the process of data collection; it contains items and each item may or may not contain categories that characterize each item. Thus, a collection tool is a list of items, each with its

[1]In a brief definition, ε is an experiment and S is a sample space associated with the experiment; an X function that associates to each element $s \in S$ a real number, X(s), is called a random variable [5].

own characteristics. The main types of data collection instruments are the medical record, the common record, the questionnaire, or the protocol.

The items of the instrument for collecting data can also be of some types, and these types depend intrinsically of values to be observed (if sortable or nominal), or of textual expressions to be accepted for each value (if codified or not), or of sequenced categories which are in each item (the categories are mutually exclusive or may be multiple choice categories within the same item).

On the one hand, there is the essence of the random variable as a producer of enough information to calculate the size of the sample to be collected; on the other, it is quite intriguing that this information is, in itself, something so important as to let the instrument for collecting data apart from a decision of such consequence. Let's go back to the data collection instrument: it is certain that the word instrument may not be the happiest, but is usually used *to express the idea of a set of items laid out in a document adopted for the regular appointment of values observed during data collection* (a concept at this point).

From the text exposed above, it is suggested that the instruments should be of various types, which have graded basis on the type of variable involved in the research. Thus, the purpose of this work becomes more exciting because it will not delve to the achievement of the core of the random variable, but rather based on information more immediately apparent and identifiable, it wants to calculate the sample size required, observing the instrument composition, as an unique body of research, and that it can provide ways for this calculation, in form and content.

Backing to random variables and their parameters: in recent decades, Statistics gained considerable and timely reinforcements, due to the inclusion in its methods of so-called non-parametric statistics, field of activity and development of new techniques of analytical content, but which, in turn, provoked sensitive concussions in descriptive and sample parts of Statistics. How to come? The techniques of non-parametric analysis have been proposed, and still are proposed to be a new path to be traveled by the data analyzers, when little is known about the essence of the approached random variable. However, the analytical part of the question seems, then, even well resolved because the non-parametric analyzes find their quasi-permanent place along with techniques of parametric analysis that for requiring the random variable, still plead specific features along it. Therefore, the description of non-parametric nature will highlight, naturally, measures that previously had lower visibility, or who had specific uses, for example, the median and other percentiles (deciles, quartiles, tertiles etc.), and values such as total, maximum and minimum, whose presence in analytical-parametric content of work (in statistical terms), were less considered since there is no need for analysis of these entities.

Just as seen for Sample Description, we want to suggest that non-parametric approach is also to be explored, developed and proposed as a process. Thus, we arrive at the heart of this proposal, which will try to direct their efforts to a less parametric view of Sampling, turning to a new species, i.e., finally, the proposed instrument for the research, as a whole, and not only the random variable as unique core direction for the calculation of sample size.

We would like to emphasize that the instrument for collecting data is nothing more than a set of items and their respective categories, selected for the well-conduction of the research. The use of the word instrument in itself deserves some additional comments: the word has a general use, i.e., in many areas of knowledge, we discuss about instruments and expressions derived from this word. Therefore, one should standardize their use, here, highlighting greatly that the idea of an instrument—seen as a collection of items belonging to the interests of the research —suggests that the statistical have free participation in the making of the instrument: a big mistake! This confection is the motto originated from the researcher, and ultimately, he/she may summon the statistician to select the form of the items that will compose the instrument, not the content. It would be acceptable to consider, in participatory terms, for the presence of statistician, would be the knowledge on the part of him, the research intentions and the analytical possibilities when finishing it, considering the inherent difficulties in the investigative process to be proposed, but we insist that the participation of the statistician, at that moment, has a guiding character, and not decisive on own interest of what to be investigated.

1.2 Sampling—Concept and Some Stories

A simple definition: Part of statistical science aimed at the study of the sample, which consists of subset of the population. The population is the statistical entity from which all the elements that could be subjected to scientific research of interest to the researcher responsible for the conduct of the research. An element may be selected for the sample, with the same probability of everyone else. This concept of equal likelihood obliges the researcher to compose his own criticism of the process of collecting sample elements, and finally, on the formation of the sample itself. It is well known today that cannot be allocated, in many cases, actually equal probabilities for forming a sample; just taken of the experiments in which subpopulations of a larger population are covered by to affect the collection of the sample: if not all population elements are available, then the probability of selection will not be the same for each population element. This occurs frequently in researches in the Health Area, where the conducting of a study requires the selection of elements present in clinical services, and thus without the concern that a probability of selection is considered effectively important. In fact, this probability even gets to be estimated; it consists only in a theoretical value, far from being evaluated by the researcher, who, first, is worried about having enough sample elements to provide its results, leading them to a good conclusion. Sometimes, this conclusion could not even be considered statistical since the sample where the results generated those conclusions could also be regarded as a statistical sample solution, due to the non-likelihood selection of those elements.

This initial review draws attention of those who, with some scientific knowledge, searches, in his/her research, for the highest degree of confidence of his/her data. But in real terms, this does not occur: as mentioned, the selection of elements from the population is, in general, without this equal likelihood. Science, which has at its base, the observation of the Nature, i.e., the measurement of natural manifestations of the phenomena [6], has, by itself, limiting aspects very visible in terms of the selection of component parts for evaluations in experiments actively conducted. These limitations generally do not consist of impediments to the realization of an experiment because there is an acceptance of common sense that there is a sufficiency inherent in the experiment, and based on the elements of the collected sample, depositing, on those elements, the degree of confidence that will be accepted to conclude something about the proposed research.

The sample idea existed for millennia: certainly not under that name, but based on something more primitive, as the very repetition of observation. One of the earliest records found in Thucydides (460–395 BC), historian of the Peloponnesian Wars, who writes that in order to determine an accurate way to measure a quantity is made mister observe it in again and again [6]. So this idea, the repetition of observation, is contained in the soul of Sampling and the very soul of Statistics. So, to repeat this feature we add at the outset the feature count to organize the repetitive process. Counting is a relatively simple act that creates an environment of organization or ordering of selected objects. It is scientifically said that it is organized to recognize numerically each observed object, and this is the reason of the Statistics be the Science of Counting: in this sense, it has, at its core, not the act of counting, but how to count; counting here means not only the numerical sequence of positive integers, but the identification of ordered elements, which necessarily have to exist in order to qualify the counted objects, which, in turn, belong to the core of conducting a search of observational character.

Mathematically, counting is considered as a bijective function of each element of a set A to set an integer [7]. Thus, one can write: $f: A \rightarrow N_n$, where ‘n’ is the number of elements of the set ‘A’ and ‘N’ is the set of natural numbers. In Sampling, always we want that the entire ‘A’ contains elements (we say: sample elements) that are useful for statistical evaluations of a research.

Historically, the idea of using a sample, despite being mentioned in the ancient texts and the Bible, emerged to fill a scientific gap with Pierre Simon Laplace in 1786, when this prestigious French mathematician estimated the population of his country by using a calculation that showed birth ratios in a period of 30 French regions, and making the extent of this amount proportionally calculated for the entire geographical area covered by the French borders [8].

The first public discussion recorded on Sampling idea occurred in 1895, at the meeting of the ISI (International Statistical Institute) settled in Bern (Switzerland), in which Anders Nicolai Kiaer (1838–1919), who was director of the Norwegian Agency of Statistics, reported their experience with samples and sample surveys for the government of his country [9]. Kiaer defined that the investigation could be made based on partial surveys, based on observations collected in various regions of the investigated territory, although it obeyed a pro rata distribution of units to be investigated in each region. These units would not be chosen randomly, but according to a logical design based on a statistical level of results previously achieved.

The French geographer Pièrre Émile Levasseur (1828–1911) mentioned that there were three survey methods: (a) a full list of units; (b) A detailed description of one phenomenon; and (c) the statistical analysis, a term that allowed clarify what proposed Kiaer. It was, therefore, decided that the third method would be the agenda of the next meeting of the ISI, 4 years later, in St. Petersburg (Russian Empire).

Kiaer defended his method, at the meeting in 1899, insisting that such investigations were sufficiently representative sample, and that could be considered a photograph which shows the details of the original in its true relative proportions. He also said there would be many ways to get around the problem of representativeness, saying that *if the comparisons with census results show to be equivalent to the statistical survey for a certain studied characteristic, then it must apply to other population information, which can then be raised in an investigation with parts of the population*.

The next session of the ISI in 1903 had already been constituted a committee to assess the value of sample surveys, which, courageously, recommended that these surveys started to be made, for collecting time-saving and cost issues and they were compared to the results of the regional censuses. Also, in 1903, the statistical concept of sample and its effective use took shape with Karl Pearson (1857–1936), who, wondering about the limitations as discussed entailed by an investigation of the entire population, said that *if the entire population was taken, then it should be obtained accurate values for their constant statistics, but in practice, the researcher is only able to take a sample*.[2]

Only in the ISI meeting of 1925, its members have accepted and made official statistical sample surveys, recommending them to governments considering the good results achieved and the economic benefits of these surveys, in addition to the high degree of credibility of the outcome, based on studies made in the first decades of the twentieth century.

1.3 Sampling Importance

In a typical inferential procedure (which allows drawing conclusions about a population from the study of a sample), the sampling technique is essential [10]. These same authors consider also about the emergence of the problem of selecting a sample as representative of the total population, given the limitations of costs and loss of precision possibilities of parameter estimates. Here, we must make some caveats:

[2]The word 'sample' appears in a publication by Pearson, in 1903, in the magazine Biometrika II p. 273. A colleague of Karl Pearson, the zoologist Walter Frank Raphael Weldon (1860–1906) was using that word since 1892, to designate a *collection of observations*.

1. The expression 'the most representative' is to be understood as specifically 'representative'. The representativeness consists to search the intrinsic conformation of a sample, which, in turn, must obligatorily be representative, as 'the most representative as possible' can also consist of an insufficient factor to achieve the degree of accuracy sought in the evaluation process required;
2. As to the costs of sampling, they should always be considered, however, not only the financial costs, intrinsic to the collection procedures (hiring, investment and maintenance of equipment, printing copies of the instruments, indirect costs such as transportation and food), but also temporal costs, which determine the time markers, including a timetable for completion of scheduled tasks, and for which there may have items for the completion of the possibilities of the collect process; In both cases, there may be worn and wear to be identified, if possible, before the start of the collecting;
3. As to the accuracy loss in parameter estimation, it is inherent in the factor reducing of a sampling procedure; a sample loads, of course, a mass of inaccuracies, the less criteria and rules are previously included as inseparable members of the sampling process.

Sampling techniques are indispensably linked to the name of William Cochran that systematized them in 1953 [11]. Although the frequent use in population research, not always the analytical processing of data is appropriate for the type of procedure used for the selection of experimental units, resulting in serious interpretation bias. With this perspective, a safe object of study on Applied Statistics, in these and in the coming years, there has been and there will be the development of methods of consistent estimation and inference with the different sampling techniques. We insist that this issue has not received a due consideration and there are many examples of incorrect inferences, consequential to ordinary treatment that always is before reliable samples.

Until the first decades of the twentieth century, the idea of a random sample, which elements should be selected in a likelihood mode, and as it is seen today, it was not universally accepted, as several statisticians believed that the selection of controls to be imposed should be mandatory, and this would be (but in fact, was not) an indicator of greater precision the degree of representativeness of the samples, compared to the population of origin.

It was left, then, still in the early twentieth century, by William Sealy Gossett (known by the nickname of Student), proposing several ideas, which become more assertive rigorous studies on statistical issues and also on sampling. His core of researches developed on their interest in samples considered small. At that time, it was already known by the researchers of Statistics that samples should be large enough to represent the population from which they were collected. So it was a consensus among the most famous statisticians that small samples could not be used as good information providers, and it was common for the work with small samples were nothing more than mere speculation on population characteristics [12]. After serious studies by Gossett, small samples became the extensive subject of a lot of researches, mainly in the Health Area, where the number of specimens was (and still is!), in general, limited, not only for ethical reasons, but due to the collecting difficulty.

Although in most studies [13], the sample size does not excessively influence the results, it is known that small samples lead to errors of conclusion, therefore, to the loss of research, due the high degree of bias of the estimated parameters. In another hand, major samples minimize these biases, and almost always allow a satisfactory evaluation of the sampling process, as well as their own achievements.

1.4 A Short Literature Review

This part of the work includes, in brief, the basic references of Paraconsistent Annotated Evidential Logic Eτ, Combinatorial Analysis, and in a resumed text, some considerations on sample size calculations, highlighting some authors.

1.4.1 Paraconsistent Annotated Evidential Logic Eτ

Logic belongs to the human culture. It is the subject that investigates, formulates, and establishes principles of valid reasoning [14]. As human beings, we try to have explained the things of nature around us, through theories and models whose bases are based in logical thoughts. And as human beings, perhaps, one must consider that we have an own and individual logic: the Classical Logic, despite wanting to show its characteristic, can do nothing against the designs that subvert the human will. A simple example: an item whose possibilities consist of two categories, the yes-no type, the Classical Logic would impose only two answers as possible, and no human action should be considered for countering this charge, but it is well known that both the non-response (non-choice of 'yes' and non-choice 'no') as the double answer yes-no concomitantly are effectively answers to the pure human reasoning, therefore, no less effectively, fit in the midst of plausible and acceptable answers of human consciousness. Thus, the Classical Logic, despite being imposer of its presence, it is contained and it is surpassed by the Human Reasoning.

The principles that guide the Paraconsistent Annotated Evidential Logic Eτ allow its more acquiescent approach the Human Reasoning [15], because they take into account possibilities that go beyond the binarism proposed in an item of yes-no type, mentioned in the previous paragraph. Thus, an item consists of two categories can have four (not two answers) acceptable: yes, no, no answer and yes-no concomitantly; these four states are classified into four different logical cores: they are the true, the false, the paracomplete and inconsistent core, respectively, and allow a closer assessment of expressing the practice and the reality of a search [16]. Thus, answers not provided for Classical Logic are provided for Paraconsistent Annotated Evidential Logic Eτ.

1.4.2 Combinatorics

Although this proposed solution appears to require relatively sophisticated method, the principle to be used in calculations that involve the combinatorial analysis is very simple. Certainly, there is no way to relate the authors who have published this principle, and it is widely exposed in high school books, not requiring any addition to the use to be made of this work. Thus, we would like to mention that Combinatorics is one of the Background Art of Counting for the use of combinations to solve various problems in that a counting point to point would be exhausting [17].

Here, it is worth remembering that the Combinatorial deals with calculations concerning combinations and permutations, i.e., a set of scientific information whose scope is responsible for the formation of so-called Theory of Combinatorics, designed to cover the problems related to the counting, scoring and enumerations.

The main feature of the combination is, in fact, the selection order of the available evidence is not required, unlike the permutation in which this order is strictly considered [17]. The formula for the combination of x elements k *by* k is given by:

$$C = \frac{x!}{k! \times (x - k)!} \tag{1}$$

In a brief and equivalent mode, the above formula can be rewritten as follows:

$$C = \binom{x}{k} \tag{2}$$

Here, both the x and the k values are positive integers and $0 \leq k \leq x$, and C is the number of combinations to be calculated.

No theory of Combinatorics will be developed at this time since the definitions are universally known, and the application that will be made has a superficial character, and easily understood.

In general, Combinatorics is proposed to solve the counting-related problems, however, there are counting problems whose heart is in the previously adopted classification and used to characterize the categories you wish to examine; hence, the problems of categorization and selection of the set to be examined. For example: the cases historically recorded in two US elections, in which a candidate gave his victory for granted:

1. In 1936, Roosevelt and Landon contested the election, respectively, by the Democratic and Republican parties [18]. A research institute predicted that Landon would be elected and that Roosevelt would get 43 % of the votes, but Roosevelt won, with 62 %. Although sample of this research institute was composed of about 2.4 million people, they had been selected from a biased way because that research institute took as a basis for selecting electors, only those

who had phone and were associated clubs. Most of the voting population was poor, and so had no phone and was not associated clubs; therefore, the selection made did not include the distribution of voting intentions of the entire population of voters. Another research institute collected a sample of 3,000 voters, but taking care to shape a representative sample said: well, the institute predicted that Roosevelt would win with 56 %; the difference of six percentage points was due to the selection technique of the elements that compose the sample (selection by quotas), and, from the 1960s [19], would be subject to serious criticism, and even fallen into disuse.

2. Even in the United States, but now, in the 1948 elections, Truman (Democrat) and Dewey (Republican) vied for political strife [19]. Three research institutes, including the one that had hit the result of 1936 stated that Dewey would win with 53 %; Truman won 55 % and Dewey received 45 % of the votes. The samples were informed by the technical quotas, and therefore the fiasco of the institutes was general.

In both cases above, planned processes with disabilities for ratings were responsible for gross errors in the determination of those previous elections. In both cases, strategies to include and exclude subjects based on the categories to which they belonged consisted in the overthrow of research results. Rating a subject in this or that category is, of course, difficult when the uncertainties overlap and combine the other obstacles.

The famous writer Isaac Asimov, in a 1955 article entitled 'Franchise', mentions that in the distant future (for that time), which would be the year 2008, the United States would have been converted into an electronic democracy, where a computer named Multivac could select a single person to answer a series of questions, and so, with the answers and some personal characteristics of the chosen one could determine who would be the next president of the country, without the need to carry a real electoral process. Norman Muller, then, is chosen as 'voter of the year'; at first, he gets scared, but after voting, boasts the fact that American voters have exercised, once again, its free and unimpeded franchise [20]. This person would represent all the characterizing categories of voters!

In the Health Area, it is common to want to adjective forming the categories of an investigated item and propose, subjectively, a classification rule for the elements that will be the components of the sample to be studied. There are few cases in which reclassifications are to be built because the results of statistical content are at the mercy of that subjectivity that previously proposed initial rating. A common example is the degree of severity rating at which the patient is, 'light', 'moderate' and 'severe' constitute usual categories, and the classification based on the four crosses, or even to a binary state ('be well' and 'be bad'). The difficulty is presented to any of these possibilities.

Besides the problem of having enough sample elements in each created category, we can make the mistake of grouping together disparate elements under the same name [21]. The counting of these elements at all will not be able to meet the needs of obtaining a reliable result through research proposal. Therefore, the sampling has its time and becomes present to work towards Science, and today, more deeply, for the sake of Health Area, together with Mathematics, Physics and Logic, forming an effective set of matters to support research and scientific work.

While Statistics is the *science of counting*, Combinatorics is the *art of counting*. Initially, this art was to be represented by recreational games and play with numbers, but it did not take that scope to become serious enough to be considered a matter of unique evidence in the complex counting problems solution. Another example of this is that, in recent decades, the Genetic Science has used to solve combinatorial gene combinations in enumeration problems, and with the participation Calculation Odds, has proposed solutions to identify bullies genes that cause problems in live beings.

1.4.3 Some Brief Commentaries on Sample Size

One key item for the literature review on the calculation of sample size was published in 1991 by Gail [22]: it is an article of recommended references, and in it, this revision was made with the most recent references, published after that year.

The sample size calculations show to be effective for various types of design of experiments: they are previously known and commonly employed in clinical research as well as in other areas of knowledge [23]. The main sample size calculation methods are in common use, and provide the calculation of sample size, based on features that previously must be known and adopted.

In all conditions of statistical interest of an investigation in which a subject of Health Area to be the driver of the research axis, one must study and understand the map of the experiment to run, and based on that map, calculate the sample size. This calculation must provide a value that represents the minimum to be collected, given the experiment design on which is deposited part of the project. Thus, the sample sizes to be calculated, in general, point to minimum values. However, we must at the outset be clear that minimum values are desirable to consist in a determiner of what should be collected, but the minimum value itself is not a specific and unique limiter [24]. Certainly, there is no maximum to be calculated, but merely, adopted, or, if applicable, estimated. The maximum size of the sample to be considered is the population size itself since any sample from a population is finite, for the research in question, and it is also lower than the value that represents the size of the studied population.

In addition, some issues are commonly referred to, when we want to calculate the sample size [25]: (a) the main research question; (b) the primary variable, or variable over which we put the focus of the study; (c) the statistical analysis to be used; (d) the degree of confidence or precision to be adopted; and (e) pilot study and its features. In each of these items, there are evident limitations and for which there is not only an applicable for an immediate solution:

1. The research contains more than one question to be answered through the investigative process to which the component items of the research are being themselves submitted. Thus, it can be seen that, in general, researches answer for several questions (which cannot be many) because these issues already exist, from the planning time to the end, considering the development of the investigative process.
2. The fact that, generally, there is a variable chosen as most important, then called primary or main variable, and based on its characteristics, we should deposit full attention to calculate the sample size since the other variables of interest, belonging to the same group that is under investigation, may present with very high importance, even if minor, but enough to be able to interfere with the results to be calculated after the data collection; perceive, with some evidence that other variables often do not allow reaching a level of knowledge about them, preventing related findings to the research focus are taken with due scientific rigor.
3. Although a priori we know how to choose the statistical analyses to be used during the evaluation process of the samples. It is common for non-parametric analyzes to replace conditions at the time of planning for sample collection. This change may affect the achievement of decision-making nature of results for search since similar tests, but it does not always produce the same results.
4. An adoption is always subjective, even though it may arise from the use of auxiliary techniques; considering their subjectivity, it can be stated that the results derived from the chosen statistical tests will depend on (quite a lot) of precision that was previously adopted; the sample size, as well as its elements will be determined by this degree of precision, and obviously, all that comes from the sampling process will also be subordinate to this information at baseline of the research.
5. When some parameters to calculate the sample size are needed and are not available, we can build a pilot study, so that these values are estimated. A preliminary sample should be collected, and some initial statistics should become providers of information for the effective calculation of sample size. The sample elements collected during the pilot study may be considered part of the final sample; in such cases, one must subtract the calculated size of the sample, the number of elements already collected in the pilot study.

Briefly: under the assumption that it has become pilot samples or the results previously achieved by a similar study conducted previously, we may make use of indicative values, with the intent of calculating the sample size for a new study, by directing application of specific formulas for each situation; in other cases, no information is obtained previously, and then other means for the sample size calculation can be used [26].

1.5 Some Specific Comments on Historical Bibliographical References

Sample size determination is for a long time a constant problem in Sampling Theory. Not so many renowned authors have dedicated their efforts to this matter, but we can remember effusively the important contributions in the works by Cochran [11], Kish [19], Deming [27], and some few others. Those authors demonstrated in their respectively works that sample size is an important matter for the acceptance of the results obtained from a serious and competent survey. The development of Sampling Theory could be considered an almost well-developed product of Statistics in the twentieth century: before the last century a few scientific investigators demonstrated some preoccupation to solve problems concerned to sample size determination, and demographic studies have been conducted by scientists, not specifically statisticians, like Gauss, Laplace, Quételet, and Galton. The Golden Age of Statistics, i.e., the first decades of twentieth century, turned into evidence the names of Pearson, Yule, Fisher, and Student (Gosset), but all of them had no exclusively dedication to Sampling; their studies have allowed the development of new ideas by Kiaer, Gini, Neyman, Chuprov [28], and finally by Cochran, Kish, and Deming. They have been followed by some important names from Indian statistical school as Malahanobis, Sukhatme, and Rao, whose dedication to Sampling Theory has been demonstrated along their lives [28].

Some forms of sample size determination are useful, and consider alpha and beta types' errors, means differences, estimated proportions as prevalence or incidence of diseases or health dysfunctions etc. In Biostatistics, all of those referred sample size calculations are supported by probability distributions of the involved variables, as it is common to suppose that the distribution is quasi-Normal (or it is accepted to be Normal) to use some theory aspects to do this calculation. Nonparametric view consists in a weak tool to determine the sample size, and no nonparametric procedures are used to speculate about the methods of sample size determination [29]. Sometimes, the determination of sample size uses information of an epidemiological official source, or, no longer, some results of a previous research (a pilot-study). Our investigation is settled on a specific basis: if an instrument for collecting data is formed by categorical items, we would like to calculate a satisfactory sample size, using the structure of this instrument and its items, but not the statistical distribution aggregated to those ones. For instance, we asked to ourselves, if it would be possible, considering no previous statistical distributions, or another technical supports concerned to the traditional Statistics Theory to turn effective this calculation. Thus, the basis for this purpose could consider mathematical and logical matters as Combinatorics and Paraconsistent Annotated Evidential Logic Eτ.

The Theory of Combinatorics consists of a good matter to start a construction on how to combine discrete categories of a sequence of items. We suppose that a participant of a survey has been selected through a rigorous methodology of inclusion and exclusion rules. So, some expected combinations of responses of each participant could be enumerated, using a counting solution based on the Fundamental Counting Principle [30]. Some decades ago, only Combinatorics rules and Classical Logic could be used to determine sample sizes, and Statistics, in exception of a Bayesian view, was connected to both only matters. Besides this traditional use of Combinatorics and Logic, a development of a scientific thought has joined with the organization and formalization of new logical thoughts. Paraconsistency [31], like Fuzzy and Neutrosophic, is a special type of Non-classical Logic, based on an attempt of constructive and actual Human Thought, or in other way, on Human Reasoning.

Paraconsistent Annotated Evidential Logic Eτ permits to explain the variability of human thought, as well we can consider that some unexpected possibilities of logic responses could be unacceptable, if we have to consider only the declared categories of each item of the collecting instrument. Non-responses and multi-pointed responses can be considered under the Paraconsistent Annotated Evidential Logic Eτ, but not under the Classical Logic, because Classical Logic sustains the formal shape of the collecting instrument, and human thought is not exclusively sustained by Classical Logic. The Theory of Classical Logic and its knowledge belong specifically to an important and restrictive area of Human Thought, and an outlier Logic, as Paraconsistency, concerns to a human neared own thought [31]. In this situation, multiple responses or non-responses for each item could be submitted to a reasonable totaling account, predicted by a direct possibility of acceptable responses. In these terms, Paraconsistent Annotated Evidential Logic Eτ can consider complementary categories as acceptable responses for each item.

Logic has its formalities, and independently of philosophical positions, our purpose is to adopt formal and rigorous rules of Combinatorics and Paraconsistency theories to demonstrate an effective possibility to design a basis to calculate plausible sample sizes based on the items of an instrument for collecting medical data.

2 The Categories of the Instrument for Collecting Medical Data

In Bio-sciences, and in special for a good amount of medical investigatory procedures in surveys, there is a frequent use of instruments for collecting information from the participating subjects as a part of a scientific search. Those instruments are often prepared by specialists, and have singular characteristics. In this attempt, we will consider a special shape for this instrument: its composition consists exclusively in check-block items. In this case, an instrument for collecting data could be a set of determined and sequenced items, and each item of that instrument consists in a set of independent and mutually exclusive categories.

Mutually exclusiveness is a previous supposition for the set of categories of each purposed item of the collecting instrument. This kind of item allows an unique choice of response, and the respondent receives a previous instruction on how to fill each item of the instrument.

Each item consists of several categories predetermined by the specialist, and, for this purpose, we know that each item has minimally two categories. So, each item may have 2, 3, 4, 5 etc. categories, and they are mutually exclusive, i.e., each respondent of the instrument should choose one and only one category for each item. There is no necessity to say that each item has, in its original purpose, one unique possibility for a response, and in fact there are no correct responses: the respondent answers each item with his/her decision about the asked formulation interpreting the information of the title item.

3 Presence of the Applied Paraconsistency

Paraconsistency is concerned with a systematization method of Human Reasoning, and this is a special and formal characteristic used to an effective logic thought [31]. Instruments for collecting data have in general many types of items for requiring information; some of them have categorical items with finite categories. Under the theory of Paraconsistent Annotated Evidential Logic Eτ, we can consider all types of responses, i.e., unexpected and expected responses: multi-pointed responses, non-responses, and expected responses are the possibilities to be observed as acceptable responses that come from each respondent.

As we said in the precedent topic, it would expect that each respondent chooses one and only one category of each item, and it would also expect that all items could be answered, but we know that some items may be not answered, and another could receive two or more marks—human decision based on Human Thought permits these actual actions, but under the Classical Logic, some of these actions are unexpected because they are not predicted responses in the original set of predetermined categories of each item.

In that way, we can see in the Fig. 1 a simple scheme that due to Human Reasoning connected with paraconsistent states.

The true and false states are expected for a reasonable respondent, and are predicted by the Classical Logic Theory; in other way, the inconsistent state is a different state: it presumes more than one alternative to be chosen for the same item; finally, the paracomplete state is connected to the non-responsiveness case, or, the non-information condition, i.e., the case for which the respondent decides to not choose any available category in one item [32]. As we can consider, all these possibilities have their natural human component to proceed on a decision to respond an item, i.e., all of them are based on human comportment for taking decisions to respond each item of the whole instrument for collecting data.

All kinds of logic are basically binary, but the dimensionality through each logic space varies: Classical Logic is able to represent binary states of choosing possibilities, previously predicted by a simple decision rule; Classical Logic is by its natural shape a dual logic. For instance, Paraconsistent Annotated Evidential Logic

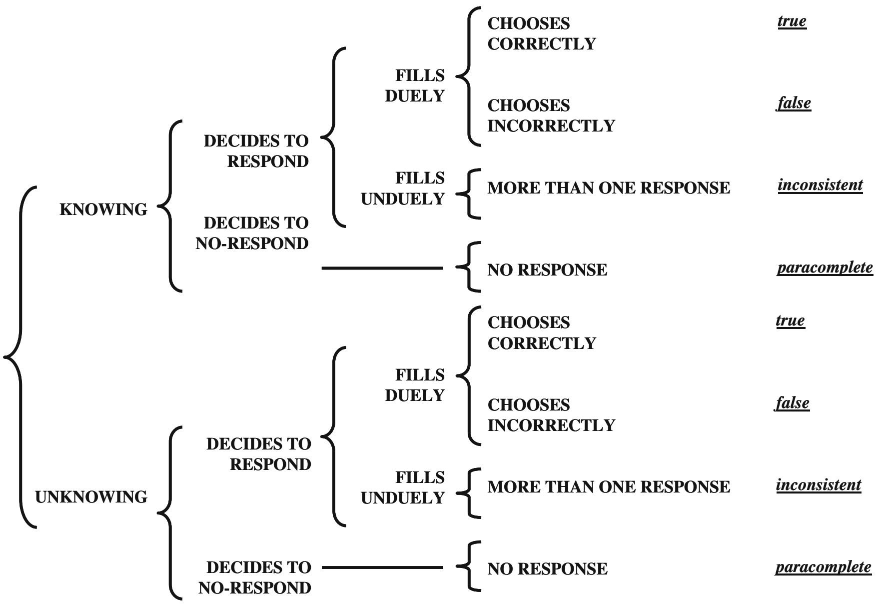


Fig. 1 Didactic-logical scheme for demonstrating the construction of Human Reasoning, and its direct interrelation within logical states presumed by Paraconsistent Annotated Evidential Logic Eτ

Eτ can represent multidimensional states of those same possibilities, as human reasoning also predicts these possibilities [31, 33]. Thinking on this way, we can see that Classical Logic, by its importance, is a basis for human thought, but this thought has become multifaceted (under its human focus), and new logical theories have also become multifaceted. Classical Logic, as dual logic, obeys to a settled-binary rule, and Paraconsistent Annotated Evidential Logic Eτ, as non-dual logic, permits more accurate possibilities neared to the Human Reasoning.

4 Application of Combinatorics

Some few decades separate our present time from the beginning of systematization of Combinatorial Analysis and its applications in whole parts of science practice. The development of techniques of counting has got diverse aspects of scientific world including a formulation of specific and important theories of counting. Historically this set of techniques has been grouped in a formal major theory under the denomination of Combinatorics, and it helps us solve a lot of problems concerning to enigmatic, difficult and sometimes early non-soluble questions that involve complex accounting processes.

We often use combinatorial techniques to give a solid theoretic basis to solve problems concerned to statistic questions [33], and by extension, in this case, with the direct application of Paraconsistent Annotated Evidential Logic Eτ, to solve a problem to determine sample sizes. In this case, all the possibilities of responses considering Paraconsistency will be able to cover a great amount of Human Thought's possibilities to make a decision, and we will be capable to calculate how many combinations of categories of all items we could get for the entire instrument for collecting data, considering all kinds of filling (including non-filling aspect) for this instrument.

In the fewest case, an item formed by two original categories could be filled as follows:

1. The decision to not respond that item (non-filling possibility, or, paracomplete state);
2. The decision to mark simultaneously two available original categories (inconsistent state); and
3. The decision to choose one and only one of the two available original categories.

Considering those above three possibilities, there are four ways to make a decision. In the case that we have three original categories in one item, there will be nine effective categories as possibilities to fill the item (non-responsiveness for the item is just one of those possible choices); in the case that we have four original categories, there will be 16 effective categories as possibilities, and so on. In extension for a generic case, we can write the following formula [33]:

$$n_w = 2^c \tag{3}$$

wherein,

n_w represents the number of ways for effective possibilities of response for an item, and c represents the number of effective categories of each item.

5 Degree of Similarity Among Respondents and the Sample Size Estimator

We are searching for a number that represents a sample size, based on the combinations of the items and categories of the collecting instrument. We will consider two types of categories: effective and original. Effective categories consist of the all possibilities of responses based on the paraconsistency, i.e., for instance, if we have a binary category (yes and no are the possibilities for response), then we will consider four effective different responses: yes, no, yes-no, and not filled (without response). Original categories are only two, in this example: yes and no. The number of effective categories can be written as 2^k, where k is the number of original categories. So, in this example, we have two original categories, and four effective categories (2^2).

We suppose that some respondents fill the items of the collecting instrument in the same way. So, the number of original categories could be considered an estimator of the degree of similarity among respondents [34].

$$1 - d^S \approx \frac{1}{c^O} \tag{4}$$

where in

c^O number of original categories of the collecting instrument
d^S degree of similarity among respondents

Then, we can suggest that the sample size based on the collecting instrument items and categories could be written as follows:

$$n = \frac{\binom{c^E}{2} - \sum_{i=1}^{k} \binom{c_i^E}{2}}{\sum_{i=1}^{k} c_i^O} \tag{5}$$

wherein

c^E number of effective categories of the collecting instrument
c_i^E number of effective categories of the i-th item
k number of collecting instrument items
c_i^O number of categories of the i-th item.

If we know the population size, we can rewrite that formula as follows [34]:

$$n = \frac{\dfrac{\binom{c^E}{2} - \sum_{i=1}^{k} \binom{c_i^E}{2}}{\sum_{i=1}^{k} c_i^O}}{1 + \frac{1}{N} \times \left[\dfrac{\binom{c^E}{2} - \sum_{i=1}^{k} \binom{c_i^E}{2}}{\sum_{i=1}^{k} c_i^O} - 1 \right]} \tag{6}$$

6 Comparison Between Methods

One traditional method for determining sample size in a medical survey considers the prevalence of the disease (event), an error margin, and a t distribution [11, 27]. In a simplified formulation, we can write:

$$n = \frac{t^2 P(1-P)}{d^2} \tag{7}$$

Using t = 1.96, P = 50 % (the value that maximizes n in formula 7), and acceptable error margin of 5 % to generate a result for the application of formula 7, we obtain n = 384.16 ≅ 385. Other values for prevalence (P) can be used, and they will produce specific results for the sample size calculation, when fixed the other parameters.

Let's suppose that the collecting instrument has a fixed number of items, and let's suppose that we can vary the prevalence value for the mentioned traditional method. Then, we could write these results in Table 1 [34].

As we can observe, traditional formula uses as basis an error margin, the value of the prevalence, and the values aggregated to a previous adopted distribution. Paraconsistent method uses only the structure of the collecting instrument, and the results of sample size calculation are quite different from one another.

Varying the parameters of both formulas, we obtain a graphical representation for each two methods, as follows in Fig. 2 [34].

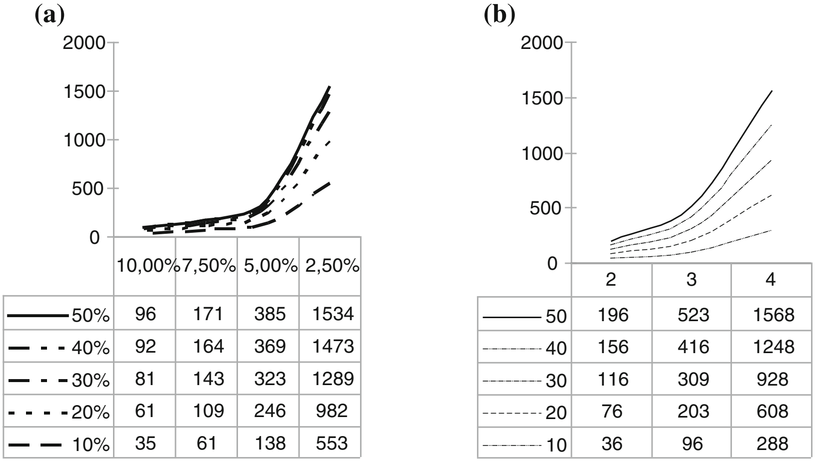
As the two methods are not directly comparable, we cannot represent both in the same graphic. As we can see, equivalences can be considered: for example, a prevalence of 10.00 % and an error margin of 10.00 % produce a sample size of 35 subjects; this is almost equivalent to an instrument based on ten items with two categories in each item, and so on.

Currently we would like to mention some immediate complementary observations on the problem to determine sample sizes:

1. The method that imposes a statistic distribution is totally independent from the content and from the shape of the collecting instrument, and it is based exclusively in the realm of the information about the prevalence of the investigated problem. Based on our suggestion, we suppose that the number of items of the collecting instrument makes an influence on the results of a sample size

Table 1 Representation of some results for the determination of sample size, using one traditional method and the paraconsistent method

Prevalence and sample size (traditional method) (%)		Sample size (paraconsistent method)				
		10 items with 2 categories	10 items with 3 categories	20 items with 2 categories	10 items with 2 categories and 10 items with 3 categories	20 items with 3 categories
50	385	36	96	76	136	203
40	369	36	96	76	136	203
30	323	36	96	76	136	203
20	246	36	96	76	136	203
10	138	36	96	76	136	203



(a)	10,00%	7,50%	5,00%	2,50%
50%	96	171	385	1534
40%	92	164	369	1473
30%	81	143	323	1289
20%	61	109	246	982
10%	35	61	138	553

(b)	2	3	4
50	196	523	1568
40	156	416	1248
30	116	309	928
20	76	203	608
10	36	96	288

Fig. 2 Graphical representation of sample sizes, considering the two methods (traditional and paraconsistent)—**a** *x* axis represents the margin error; *y* axis represents the sample size; there are five curves representing some chosen prevalence values. **b** *x* axis represents the number of categories of each item; *y* axis represents the sample size; there are five curves representing some chosen number of categories

calculation by evidence: An instrument constituted by a lot of items should require more elements than a simple instrument constituted by a few categorical items.

2. Another problem consists in the obligation to assume an approximated Normal distribution for the data. In a significant number of surveys, Normal distribution is adopted, but not efficiently proved. This acceptance can be a major estimation error, if we don't know some characteristics of the investigated population, and the origin of our collected data. In this case, it would be better not to adopt the Normal distribution. The calculation of sample size should be independent from any distributions, if we don't have a sufficient based knowledge for that. Therefore, a non-parametric view seems to be more accurate: using the survey instrument's components can turn easy and practical the sample size calculation, as will be demonstrated in the next item. Normal distribution or any other distribution can effectively be adopted, if we presume that this adoption could be accepted by theoretic assumptions. In the common practical use of a formula based on any distribution, it should be proved in its basis that the adopted distribution effectively permit the more accurate and precise determination of sample size. This is not a fact in a good number of surveys, and the sample size in a mostly times demonstrates that the collected sample produces biased estimation of parameters, implicating the results to a frequently faked conclusions.

7 Applied Calculation for Comparison Between Methods

We will consider the data available in an extensive research, conducted in the Gynecology Department of the Medicine Faculty of the University of Sao Paulo. In this survey, researchers could evaluate about 15,000 clinical protocols in almost 30 years of investigation. From that enormous amount, 14,435 could be used for our purposes in simulating a real situation, using this population as a basis to extract a representative sample, considering two ways for calculating sample size.

By the first way (the traditional method), we consider, for instance, a prevalence value for the most important variable aggregated to menopause cycle, the menopause age that shows 5,968 women have even declared to be in their menopause lifetime. So, in that research, we have a prevalence estimated of 41.344 % for that population. Immediately, we can estimate a sample size [11] by the direct application of formula:

$$n = \frac{\frac{t^2 P(1-P)}{d^2}}{1 + \frac{1}{N} \cdot \left[\frac{t^2 P(1-P)}{d^2} - 1\right]} = \frac{\frac{1.96^2 \times 0.41344 \times (1-0.41344)}{0.05^2}}{1 + \frac{1}{14{,}435} \cdot \left[\frac{1.96^2 \times 0.41344 \times (1-0.41344)}{0.05^2} - 1\right]} \cong 364 \quad (8)$$

The adopted values for t and d are, respectively, 1.96 and 0.05. As we can see, a sample minimally constituted by 364 elements is enough to attempt an investigation on that population of women.

In another way, the paraconsistent method to estimate sample size considers all present variables to be investigated in that survey. Let's see our calculation based on this same population, and considering the available variables in that clinical protocol for menopause cases. In this protocol, we can find these following original types of 93 variables: 57 continuous variables transformed into discrete variables with 2, 3, and 5 categories, under acceptable pre-established rules; and 36 discrete variables with 2, 3, 4, and 5 categories.

As we can observe, while traditional method fixes n under the basis of prevalence, or, in other common cases, under the basis of proportions differences or means differences, new suggested method permits to obtain n in connection with the collecting instrument size, i.e., the major number of considered variables, the major n.

For our purposes, the calculation of a sample size using the protocols of women population of Gynecology Department will consider the variables with a representative percent degree of time usage. The original protocol suffered not so many alterations along almost three decades, but some of them were significantly important to be considered: old medical exams have been suppressed, and substituted by new, in which the number of items differs from the original. In this specific case, from 93 variables (the amount of all items considered in almost 30 years of this protocol shape), we have available 34 variables of 50 % or more filling in a good period of effective usage, 51 variables of 25 % or more filling, and 69 variables of 10 % or more filling. For testing sample size calculation formula for paraconsistent method, we will consider a clinical protocol with those three amounts of variables. In each case, we have, respectively, n = 517 (considering those 34 variables); n = 653 (considering those 51 variables); and n = 727 (considering those 69 variables). Then, under the hypothesis of considering all 93 available variables, we could obtain n = 881 sample elements to be collected. This last hypothesis is not a true real situation: we suppose that in a special condition all items could be counted for a sample size determination, but we know that it is not the real situation.

On a good sense basis, we know that medical instruments have no more than a few dozens of items, and searchers don't plane giant instruments. In this common case, sample size calculation could also be effective under the combinatorial substrate as we demonstrated [34].

8 Some Final Observations

First, it would be interesting to mention that in so many cases, the instrument for medical surveys can have non-categorical items, i.e., free items; in that case, and when it is possible, free items could be transformed into check-block items. Thus, those new categorized items may be counted as a basis to calculate the sample size.

Secondly, our studies are carrying out to fix weights to each category of each item. We are developing this new model under the auspices of computational technology. For this, weights could be associated with each category, and we would be able to nearing our model to a more realistic condition. The basis for choosing the weights are such concerned principally to the prevalence or to the proportion expectation of each category of each item, when this information is supported by previous surveys.

The chance of observing non-responses could be considered equivalent as the summoned chance of all multiple marked categories. In this way we can consider the following sequence of chances for responding an item: (a) the higher chance is concerned to the correct filling mode; (b) the second higher chance is concerned to the non-responsiveness; and (c) we can consider the total amount of multi-pointed responses for filling with the same magnitude order of non-responsiveness chance, i.e., each one of these combinations is represented by an only less chance than non-responsiveness effective chance.

The expectation is certainly concerned to a correct filling of each item, but it is common to detect some non-responsiveness and/or incorrect filling (multi-pointed responses). Correct responses have in most cases an extremely higher chance to be observed, when the searcher or a person with elucidated criteria could help the respondent; in the cases that the respondent is in an isolated state, errors may be committed with a good chance, many times more evident than the last case. So, the status for responding to an instrument is no doubt influenced by the situation in which the respondent takes place to do this application.

Finally, we would like to point that our studies have the objective to transform this present suggestion into a more accurate purpose for the sample size calculation. We are looking for an adjustment for traditional formulas, based in the size of the collecting instrument.

References

1. Baitello Jr, N.: O homem que parou os relógios. Anablume, Sao Paulo (1997)
2. Hutchins, R.M.: Great Books of the Western World. In: Rene Descartes, Mortimer J. Adler (eds.) vol. 35, Chicago (1971)
3. Pagano, A.: Lições de Estatística(2nd ed.) São Paulo: Gráfica da Prefeitura de São Paulo (1946)
4. Frankel, M., King, B.: A conversation with Leslie Kish. Statist. Sci **11**(1), 65–87 (1996)
5. Meyer, PL.: *Probabilidade*—aplicações à Estatística (2ed.) São Paulo: LTC (1983)
6. Câmara, L.: Tecnologia da Amostragem. ENCE, Rio de Janeiro (1968)
7. Grifitts, H.B., Hilton, P.J.: *Matemática Clássica*—uma interpretação contemporânea. Edgard Blucher, EDUSP, Sao Paulo (1975)
8. Stigler, S.M.: The history of Statistics. The Belknap Press of Harvard University, Cambridge (2006)
9. O'Muircheartaigh C.: Balancing statistical theory, sampling concepts, and practicality in the teaching of survey sampling. In: ISI 55th (2005)
10. Szwarcwald, C.L., Castilho, E.A.: *Os caminhos da Estatística e suas incursões pela Epidemiologia*. Cad Saúde Pública, 8(1), Rio de Janeiro Jan/Mar (1992)
11. Cochran, W.J.: Sampling Theory, 3rd edn. Wiley, New York (1985)
12. Crouch, M., McKenzie, H.: The logic of small samples in interview-based qualitative research. Soc. Sci. Inform. **45**(4), 17 (2006)
13. Wiley, R.H.: Is there an ideal behavioural experiment? Anim. Behav. **66**, 585–588 (2003)
14. Lapedes, DN [ed].: *McGraw-Hill Encyclopedia of Science and Technology*. vol. 7, McGraw-Hill Book Company, New York (1977)
15. Abe, J.M.: *Fundamentos da Lógica Anotada* [doctorate thesis]. FFLCH—USP, 135 pp (1992)
16. Abe, J.M.: Some aspects of paraconsistent systems and applications. Logique et Anal. **157**, 83–96 (1997)
17. Grimaldi, R.P.: Discrete and Combinatorial Mathematics—an applied Introduction. Addison-Wesley Publishing Company, Reading (Massachusetts) (1986)
18. Stephan, F.F.: History of the uses of modern sampling procedures. J. Am. Stat. Assoc. **43** (241), 12–39 (1948)
19. Kish, L.: Survey Sampling. Wiley, New York (1965)
20. Asimov, I.: Earth is Room Enough. Grafton Books, Glasgow (1986)
21. Katz, D., Gutek, B.A., Kahn, R.L., Barton, E.: Bureaucratic Encounters—A Pilot Study in the Evaluation of Government Services. Michigan, Ann. Arbor. (1977)
22. Gail, M.H.: A bibliography and comments on the use of statistical models in Epidemiology in the 1980s. Stat. Med. **10**, 1819–1885 (1991)
23. Grunkemeier, G.L., Jin, R.: Power and sample size: how many patients do I need? Ann. Thorac. Surg. **83**, 1934–1939 (2007)
24. Bégaud, B., Fourrier, A., Moore, N., Moride, Y.: The delusion of reducing sample size. Eur. J. Clin. Pharmacol. **59**(8–9), 711–712 (2003)
25. Scales, D.C., Rubenfeld, G.D.: Estimating sample size in critical care clinical trials. J. Crit. Care **20**(1), 6–11 (2005)
26. Schulz, K.F., Grimes, D.A.: Epidemiology 1—sample size calculations in randomized trials: mandatory and mystical. Lancet **365**, 1348–1353 (2005)

27. Deming, W.E.: Some Theory of Sampling. Dover, New York (1966)
28. Conover, W.J.: Practical Nonparametric Statistics, 2nd edn. Wiley, New York (1980)
29. Jacobs, H.R.: Mathematics, a Human Endeavor, 3rd edn. WH Freeman and Company, New York (1998)
30. Da Costa, N.C.A., Beziau, J.Y., Bueno, O.: On the usefulness of paraconsistent logic. Logic, Thought and Action **2**, 465–478 (2005)
31. Da Costa, N.C.A.: On the theory of inconsistent formal systems. Notre Dame J. Form. Logic **15**(4), 497–515 (1974)
32. Abe, J.M.: Paraconsistent artificial neural networks: an introduction. In: Proceedings of *Knowledge-Based Intelligent Information and Engineering Systems*, pp. 942–948, vol 32(14), (2004)
33. Feller, W.: An introduction to probability theory and its applications. Wiley, New York (1968)
34. Couto Junior, E.B.: *Abordagem não-paramétrica para cálculo do tamanho da amostra com base em questionários ou escalas de avaliação na área de saúde* [doctorate thesis]. Faculdade de Medicina da Universidade de São Paulo. Departamento de Patologia, pp 130 (2009)

Paraconsistent Logic Algorithms Applied to Seasonal Comparative Analysis with Biomass Data Extracted by the Fouling Process

João Inácio da Silva Filho, Irapajy da Silva Caetano, Floriana Nascimento Pontes, Maurício Conceição Mario, Jair Minoro Abe and Fábio Giordano

Abstract In ecology, extracting information about species that live in ecosystems and to find new ways to make the analysis of this data is very important to get the levels of contamination of the marine environment and other information about ecosystems. The values of biomass taken from ecological process is called Fouling, which are considered complex, bring usually incomplete or even inconsistent information and therefore can lead to conclusions far from the reality. Recently, paraconsistent logics have emerged as an innovative proposal to make the data processing which brings contradictory or uncertain information and therefore can offer better response under these conditions. In this chapter, we present a method that uses paraconsistent annotated logic (PAL) to find degrees of evidence resulting from seasonal comparison among analytical descriptor values of biomass type in the Fouling process.

Keywords Paraconsistent annotated logic · Ecology · Marine ecology · Biodiversity · Fouling

J.I. da Silva Filho · I. da Silva Caetano · F.N. Pontes · M.C. Mario · F. Giordano (✉)
UNISANTA—Santa Cecília University, Oswaldo Cruz Street, 288, Santos, SP CEP 11045-000 Brazil
e-mail: giordano@unisanta.br

J.I. da Silva Filho
e-mail: inacio@unisanta.br

M.C. Mario
e-mail: cmario@unisanta.br

J.I. da Silva Filho · J.M. Abe
IEA-USP—Institute of Advanced Studies of the University of São Paulo, Prof. Luciano Gualberto Avenue, no 374, University, São Paulo, SP CEP 05508-900 Brazil
e-mail: jairabe@uol.com.br

J.M. Abe
Graduate Program in Production Engineering, Paulista University, São Paulo, Brazil

J.M. Abe (ed.), *Paraconsistent Intelligent-Based Systems*,
Intelligent Systems Reference Library 94, DOI 10.1007/978-3-319-19722-7_6

1 Introduction

The degradation of coastal and marine ecosystems is a fact that worries scientists and authorities in the area of sustainability and environment [1, 2]. Understanding the environment and how it correlates the species, are interesting questions that have motivated studies in order to obtain results that might indicate the stage of environmental degradation.

Since these results will serve to justify actions, and considering the complexity of the ecological study for the problem, recently great efforts to conduct research have been done [2–4].

These researches can provide monitoring and analysis methods capable of covering these gaps about the knowledge of the causes and their effects as well as the levels of impacts on the marine ecosystem [3, 5]. For these studies, it is common to use statistics, which is an important part of applied mathematics. So, statistics is used in the collecting, analysis and interpretation of observed data. However, the nature of the data collected from ecological processes, which in their majority is incomplete, ambiguous and with difficult quantitative interpretation, causes the consideration of new techniques of calculations. These new procedures are aggregated to the statistical procedures, and thus make the data treatment able to offer conclusions with a better degree of reliability [5, 6].

Paraconsistent logic belongs to the family of non-classical logics, and has as its main feature the ability to accept contradiction in its fundamental theory which does not cause conflict in its conclusive results.

The structured way of annotation of two values (PAL2v) of paraconsistent logic has presented good results in analysis and monitoring concentration levels of pollution of marine environment through data obtained with use of bioindicating techniques [6–8].

In analysis of ecological processes, the "Analytical Descriptor" is the raw data originated directly from the source of observation and that has not been treated, or manipulated by any means or mathematical tools.

The most commonly used Analytical Descriptors are: percentage covering, number of individuals (or abundance) and biomass [4–6].

The term "Fouling" used in biodiversity is used to characterize the fauna that lives directly and indirectly associated with a substrate. This term has being used to distinguish the associations of animals and plants that grow in artificial structures, from those that grow in rocks, stones and other natural substrates [4, 6]. Some authors have described "Fouling" as a biological phenomenon, in which the organisms that compose it (plants and animals) are those of sessile life that are found in shallow waters in nature.

The "Fouling" phenomenon, as well as its community of encrusting invertebrates has been widely studied, as they may contain information about the degradation of the marine environment or bring evidence of some intervention (human or natural) that can cause environmental damage. Another reason for the study of "Fouling" is the economic problems caused by the community of invertebrates,

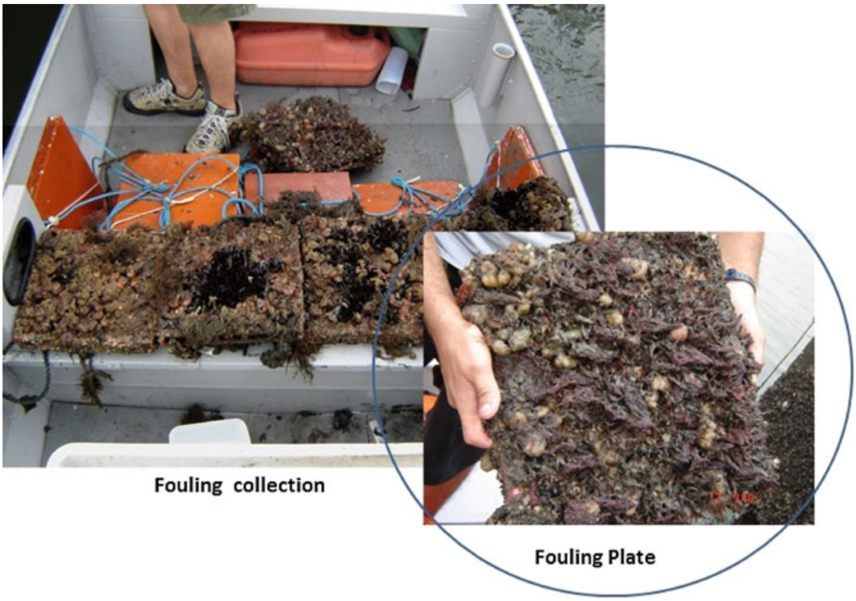


Fig. 1 Fouling process with details of a plate

such as the harmful adherence of these organisms in boats and ships, on piers, floating buoys and underwater cables [5, 6]. The Fouling plates are presented in Fig. 1.

A technique used to extract this information is the one using the biomass of "Fouling", which term refers to the fauna that lives directly and indirectly associated with a substrate. The paraconsistent method validation is done through the application of PAL on data from a secondary source that brings information of biomass with analytical descriptors: volume, wet weight, dry weight, ash-free dry weight. As the applications of the equations of PAL2v results normalized values, therefore between 0 and 1, it can make the numeric comparisons and get seasonal behavior profile. In this paper, these biomass descriptors are extracted from a seasonal "Fouling" process that compares two distinct seasons: summer and winter [6].

With the application of the algorithms of paraconsistent logic, we can make comparison between two samples of biomass and observe as the results are close to the purely statistical process. Due to innovation, these resulting values obtained in this paper work as benchmarks and will serve as a basis for future researches involving cross-species diversity analysis through the procedures of the "Fouling" studies.

In this chapter, we present a method that uses the PAL2v algorithms to analyze, in two seasons, comparative values of biomass (Analytical Descriptor) extracted from a "Fouling" process that, will be considered as a secondary data source [6] in this work. Therefore, in this application, and for this type of ecological analysis, it will be used to compose the Degrees of Evidence extracted from "Fouling" the Analytical Descriptor: biomass.

The paraconsistent annotated logic is a non-classical logic that it is evidential and propositional too [7–10].

2 The Mathematical Complex Methods and Paraconsistent Logic

The analysis and processing of data originated from Analytical Descriptors of ecosystems are characterized by its complexity. This happens because there is a great amount of ambiguous information and incomplete representative signals, uncertain and inconsistent, all of them extracted from these ecological processes. For such uncertain types of data, we may find better results when we use algorithmic tools in the area of artificial intelligence and based on non-classic methods [7].

Following this line of reasoning, where the problem of pollution in marine environment is classified as information generator of certain degree of complexity, we use as support in the treatment of information data, obtained in collects, a type of non-classical logic called paraconsistent annotated logic with annotation of two values (PAL2v).

The algorithms based on PAL2v have been effective in the analysis and interpretation of data originated from uncertain knowledge bases [7, 11].

2.1 Paraconsistent Annotated Logic (PAL)

In the paraconsistent annotated logic (PAL), propositional formulas come with annotations. Each annotation, belonging to a finite lattice, assigns values to their corresponding propositional formula or proposition (*P*), such that:

An operator is fixed $\sim: |\tau| \rightarrow |\tau|$, where the operator $\sim$ constitutes the "meaning" of the logical negation symbol ¬ of the system that will be considered.

If *P* is a basic formula operator $\sim: |\tau| \rightarrow |\tau|$ is defined as:

$$\sim [(\mu, \lambda)] = (\lambda, \mu) \text{ where } \mu, \lambda \in [0, 1] \subset \Re.$$

So it is considered (μ, λ) an annotation of *P* where:

P(μ, λ) is a paraconsistent logic signal [7].

In this case the degrees of favorable (μ) and unfavorable (λ) evidence compose an annotation that gives a logical connotation to the proposition *P*.

In Fig. 2 the lattice of paraconsistent annotated logic PAL is shown with the logical extreme states being represented in their 4 vertices [7, 12].

In this way, the association of an annotation (μ, λ) to a proposition *P*, means that the degree of favorable evidence in *P* is μ, while the degree of unfavorable evidence, or "contrary", is λ. Intuitively, such PAL lattice has:

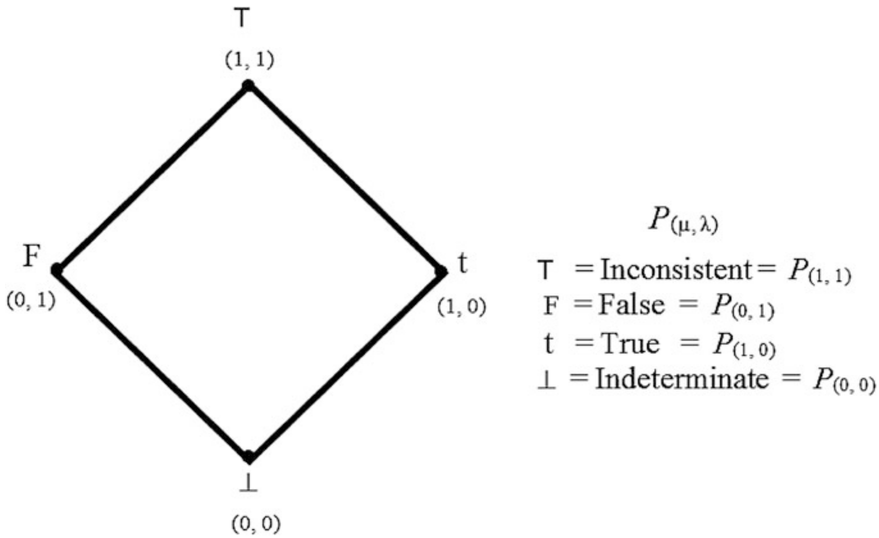


Fig. 2 Representation of PAL2v lattice and four logical states represented in their 4 vertices

$P(\mu, \lambda) = P(1, 0)$: indicating the existence of a full favorable evidence and existence of null unfavorable evidence—assigning a connotation of Truth to the proposition.

$P(\mu, \lambda) = P(0, 1)$: indicating the existence of a null favorable evidence and existence of full unfavorable evidence, assigning a connotation of Falsehood to the proposition.

$P(\mu, \lambda) = P(1, 1)$: indicating the existence of a full favorable evidence and existence of full unfavorable evidence, assigning a connotation of Inconsistency to the proposition.

$P(\mu, \lambda) = P(0, 0)$: indicating the existence of a null favorable evidence and existence of null unfavorable evidence, assigning a connotation of Indetermination to the proposition.

By the linear transformations in a unit square in the cartesian plane [7] and the representative PAL lattice, we can reach to the transformation:

$$T(x, y) = (x - y, x + y - 1) \tag{1}$$

Considering the components of the transformation $T(X, Y)$ according to the usual PAL nomenclature, where:

$$\mathrm{x} = \mu \rightarrow \text{favorable evidence degree}$$
$$\mathrm{y} = \lambda \rightarrow \text{unfavorable evidence degree}$$

- from the first term obtained in the ordered pair of the transformation equation it becomes:

$x - y = \mu - \lambda \rightarrow$ which we call degree of certainty (D_C). Therefore, the degree of certainty is calculated by:

$$D_C = \mu - \lambda \tag{2}$$

And their values, which belong to the real interval [−1, + 1] and are on the horizontal axis of the lattice [7, 10, [12], which is called the "axis of degrees of certainty".

When D_C results +1, it means that the logical state resulting from the paraconsistent analysis is True (t), and when D_C results −1, it means that the logical state resulting from the analysis is False (F).

- from the second term obtained in the ordered pair of the transformation equation, it follows that:

$x + y - 1 = \mu + \lambda - 1 \rightarrow$ which we call degree of contradiction (D_{ct}). Therefore, the degree of contradiction is obtained by:

$$D_{ct} = \mu + \lambda - 1 \tag{3}$$

And their values, which belong to the real interval [+1,−1], and they are on the vertical axis of the lattice [7, 10, 12], which is called the "axis of degrees of contradiction".

When D_{ct} results +1 it means that the logical state resulting from the paraconsistent analysis is Inconsistent (T), and when Dct results −1, it means that the logical state resulting from the analysis is Indeterminate (⊥).

Starting from a detailed study, seen in [7], we can find the real degree of certainty (Dcr) as being a value without the effect of the contradiction in the axis of the degrees of certainty of the lattice:

$$Dcr = 1 - \sqrt{(1 - |D_C|)^2 + D_{ct}^2} \quad \text{If} \quad D_C > 0 \tag{4}$$

$$Dcr = \sqrt{(1 - |D_C|)^2 + D_{ct}^2} - 1 \quad \text{If} \quad D_C < 0 \tag{5}$$

And from the Dcr we can find its normalized value, called the resulting evidence degree (μ_{ER}) [7]. Therefore:

$$\mu_{ER} = \frac{Dcr + 1}{2} \tag{6}$$

2.2 Algorithms Used in Paraconsistent Analysis

From the Eq. (2) to (6) of the paraconsistent annotated logic with annotation of two values (PAL2v) algorithms for the treatment of information signals [11, 12] are extracted. The processing of these signals is done in the form of degrees of evidence applied in paraconsistent analysis.

A paraconsistent logical treatment is defined as a mathematical procedure applied in extracted values of measurements in physical systems using the equations obtained in the LPA2v lattice.

All the procedures for a paraconsistent logical treatment will be related to the analysis made by an algorithm called PAN-paraconsistent analysis node [7, 10], as described below.

2.2.1 PAN—Paraconsistent Analysis Node Algorithm

1. Enter the Input values.
   μ */favorable evidence degree $0 \leq \mu \leq 1$
   λ */unfavorable evidence degree $0 \leq \lambda \leq 1$
2. Compute the degree of certainty.

$$D_C = \mu - \lambda$$

3. Compute the degree of contradiction.

$$D_{ct} = (\mu + \lambda) - 1$$

4. Compute the distance d.

$$d = \sqrt{(1 - |D_C|)^2 + D_{ct}^2}$$

5. Compute the normalized contradiction degree.

$$\mu_{ctr} = \frac{\mu + \lambda}{2}$$

6. Compute the interval of resulting evidence.

$$\varphi_E = 1 - |2\mu_{ctr} - 1|$$

7. Determine the output signal.
   If $\varphi_E \leq 0.25$ or if d > 1, then do: $S1 = 0.5$ and: $S2 = \varphi_{E(\pm)}$
   Consider Undefinition and go to item 11
   Otherwise go to the next item

8. Determine the real certainty degree.

$$\text{If} \quad D_c > 0 \quad \text{Calculate:} \quad Dcr = (1 - d)$$
$$\text{If} \quad D_c < 0 \quad \text{Calculate:} \quad Dcr = (d - 1)$$

9. Compute the real evidence degree resulting

$$\mu_{ER} = \frac{D_{CR} + 1}{2}$$

10. Present the results in the output

$$\text{Do} \quad S1 = \mu_{ER} \quad \text{and} \quad S2 = \varphi_{E(\pm)}$$

11. End

The PANs algorithms can be interconnected forming paraconsistent data processing.

In this work, we used an algorithm named Extractor of Contradiction Effects that it is formed by several PANs [7, 10]. The Extractor of Contradiction Effects algorithm will be described below.

2.2.2 Extractor of Contradiction Effects Algorithm

The algorithm named extractor of contradiction effects [7, 11] receives a group of signals and, regardless of any other external information, has the function of making a paraconsistent analysis on their values by subtracting the effects caused by contradiction. With that algorithm a single real resulting degree of evidence which is the representative value of the group will be presented in the output. The algorithm used in the extraction process of the effects of contradiction, is described below:

1. Present n values of evidence degrees that composes the group in study.

$$G\mu_{est} = (\mu_A, \mu_B, \mu_C, \ldots, \mu_n) * /\text{evidence degrees } 0.0 \leq \mu \leq 1.0 * /$$

2. Select the largest value among the evidence degrees of the group in study.

$$\mu_{maxA} = \text{Max}\ (\mu_A, \mu_B, \mu_C, \ldots, \mu_n)$$

3. Consider the largest value among the evidence degrees of the group in study in favorable evidence degree.

$$\mu_{maxA} = \mu_{sel}$$

4. Select the smallest value among the evidence degrees of the group in study.

$$\mu_{minA} = \text{Min}\ (\mu_A, \mu_B, \mu_C, \ldots, \mu_n)$$

5. Consider the smallest value among the evidence degrees of the group in study in unfavorable evidence degree.

6. Transform the smallest value among the evidence degrees of the group in study in unfavorable evidence degree.

$$1 - \mu_{minA} = \lambda sel$$

7. Make the paraconsistent analysis among the selected values:

$$\mu_{R1} = \mu_{maxA} \diamondsuit \lambda sel \quad */ \text{ where } \diamondsuit \text{ is a paraconsistent action of the PAN}$$

8. Increase the obtained value μ_{R1} in the group in study, excluding of this the two values μ_{max} and μ_{min}, selected previously.

$$G\mu_{est} = (\mu_A, \mu_B, \mu_C, \ldots, \mu_n, \mu_{R1}) - (\mu_{maxA}, \mu_{minA})$$

6. Return to the item 2 until that the group in study has only 1 element resulting from the analysis.

$$\text{Go to item 2 until} \quad G\mu = (\mu_{ER})$$

3 Materials and Methods

In this session an approach to the extracting of the degrees of evidence from a study of "Fouling" in which biomass were used as an analytical descriptor is presented. [13, 14].

3.1 Data Source—Secondary Source

The work from which originated the Analytic Descriptors that we use here was prepared with samples of "Fouling" coverage in Santos-St.Vincente Estuary in Sao Paulo State, Brazil, collected in 2006. The details are in [14].

Four distinct points were selected in the estuary where, in each one of them four ceramic slabs (40 cm × 40 cm) were left with the analyzed sides towards the bottom in order to prevent the incidence of light.

A list of animals and species that colonize the slabs were made, for a thorough study of the diversity and richness of the organisms in the region.

In this collecting, the analysis of biomass and volume of collected organisms was performed in order to quantify the organisms that have evolved on the slabs.

From these studies with "Fouling" in [14] in which analytical descriptors were generated it is possible to use them as secondary source of information to compose the degrees of evidence and apply the algorithms of the PAL2v. The values of the collected biomass in [14] will be used to exemplify the application of the method of analysis with the PAL2v.

3.2 The Method of Analysis with PAL2v

With the objective of developing an analysis pattern that can reproduce clear results with greater reliability, the analytical descriptors obtained in the process of study of "Fouling" are transformed into degrees of evidence. After being properly treated, the degrees of evidence are considered information signals for the paraconsistent analysis that will be made through the PAL2v algorithms [7, 11]. The sequence of the procedures for paraconsistent analysis is exposed in Fig. 3.

3.3 Extraction of Evidence Degrees Through the Analytical Descriptors

As seen, for its correct application to PAL2v information signals are required in the form of two degrees (μ) and (λ) that express evidence about the proposition relating to physical process to be analyzed. These two degrees of evidence must be representative, belonging to the real interval [0,1]. For the analysis with PAL2v a procedure called extraction of degree of evidence is first performed.

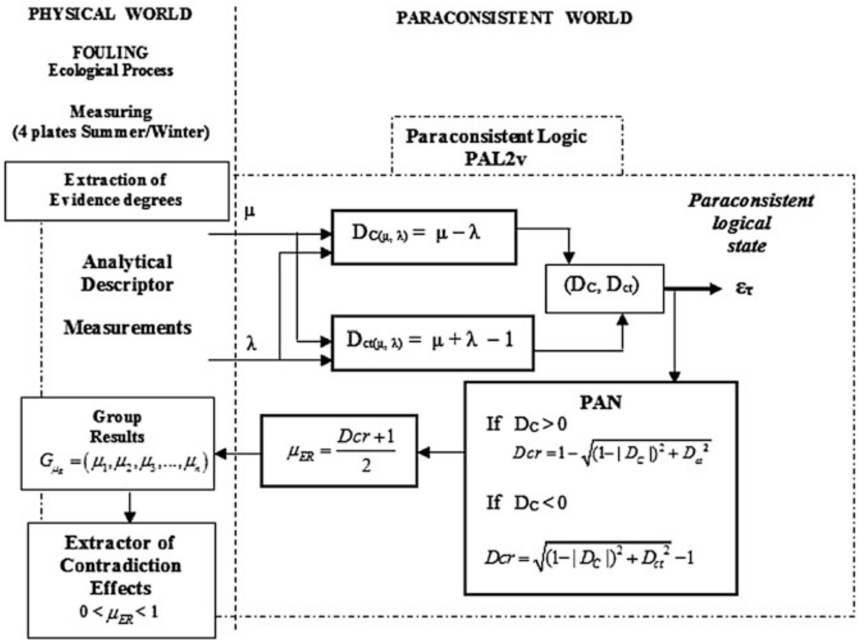


Fig. 3 Sequence of analysis and relationship between the physical world—where measurements are obtained—and the paraconsistent universe—where the paraconsistent logical states are obtained

The transformation that aims to extract degrees of evidence of physical quantities can be made in several ways. In this work, we chose to use a probabilistic form for the degrees of evidence from biomass.

In this initial procedure, the analytical descriptors of the "Fouling" process are transformed into normalized values within the interval [0,1] with equation:

$$\mu_{Plate} = \frac{\text{Analytical Descriptors X}\, MeasuringValue}{\text{Analytical Descriptors}\, MaximumValue} \quad (7)$$

So this is the equation that extracts the degrees of evidence.

3.4 Calculation Using PAL2v Equations

The topology of the paraconsistent network of data analysis can be formed in several ways. In this work, the calculations are made for seasonal comparison analysis between biomass extracted from "Fouling" on two different seasons of the year. Therefore, after the process that makes the extraction of the degrees of evidence we use the topology of the contradiction effect extracting algorithm which is composed of PAN algorithms. The topology is showed in Fig. 4.

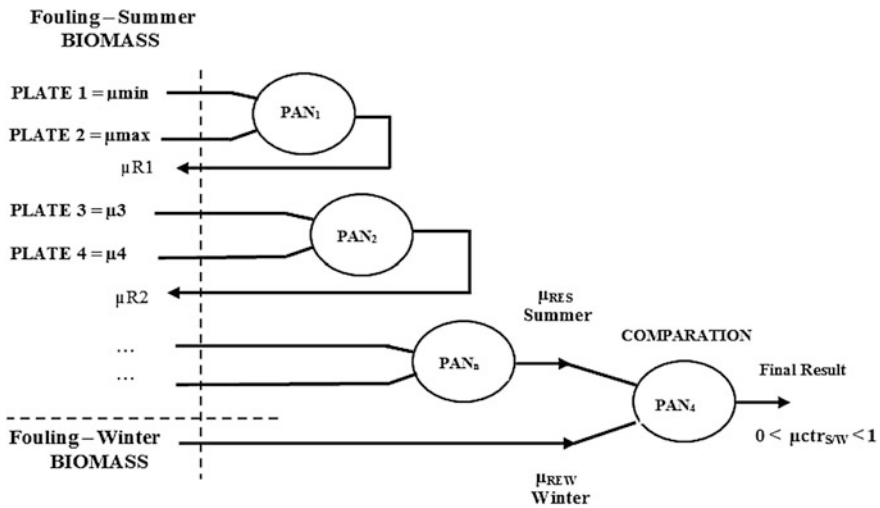


Fig. 4 Network topology configured with PANs in which the contradiction effect extracting algorithm makes the analysis of biomass

3.5 Comparison of the Results Between the Two Seasons

The comparison of the results between the two seasons of the year is made to get a seasonal analysis using the PAL2v. The application of a PAN that will produce the normalized Degree of Contradiction is used, where:

$$\mu_{ctr} = \frac{D_{ct} + \lambda}{2} \quad (8)$$

The greater the difference between biomass analytic descriptors obtained between the two seasons, the farthest from the value 0.5 the normalized degree of contradiction will be. So, to obtain a value between 0 and 1 an algorithm which works with two line equations is used. The final score algorithm is exposed as follows:

3.5.1 Final Score Algorithm

1. Enter the normalized degree of contradiction: μ_{CtrX}
2. If $\mu_{CtrX} \leq 0.5$ from Eq. (8), compute the degree of final contradiction for:

$$\mu_{Final} = \frac{\mu_{CtrX} - 0.5}{-0.5}$$

Otherwise: $$\mu_{Final} = \frac{0.5 - \mu_{CtrX}}{0.5}$$

3. End.

4 Applications of PAL2v in the Analysis of Ecosystems

From these studies we use the results as secondary source information to compose the degrees of evidence and apply the algorithms of the PAL2v. Therefore, the values of the collected biomass in [14] will be used to exemplify the seasonal comparative analysis method with the algorithms of paraconsistent annotated logic [15].

4.1 Computations and Results

The analytical descriptors of biomass used to generate the degrees of evidence are: volume, wet weight, dry weight and ash-free dry weight. These values will be removed from the Tables 1 and 2, in the summer season, and removed from the Tables 3 and 4 in the winter season.

Table 1 Volume—Summer—location 1. *Source* [14]

Plate (n)	Volume (ml)
Plate 1	630
Plate 2	1,055
Plate 3	500
Plate 4	700

Table 2 Fresh weight, dry weight and ash weight—Summer—location 1. *Source* [14]

Plate (n)	Fresh weight (g)	Dry weight (g)	Ash weight (g)
Plate 1	702	274.68	209.80
Plate 2	1,101	443.20	326
Plate 3	623	372.90	216
Plate 4	866	293.30	234.20

Table 3 Volume—Winter location 1. *Source* [14]

Plate (n)	Volume (ml)
Plate 1	20
Plate 2	600
Plate 3	300
Plate 4	400

Table 4 Fresh weight—dry weight and ash weight—Winter location 1. *Source* [14]

Plate (n)	Fresh weight (g)	Dry weight (g)	Ash weight (g)
Plate 1	27.60	7.48	5.90
Plate 2	680.66	173.20	134
Plate 3	348.40	98.68	75.50
Plate 4	446.17	128.08	102.60

4.2 Extraction of Degrees of Evidence-Summer

The group of values is removed from the Table 1:

$$\text{Group}_{R1vol\text{Summer}} = \left\{Value_{1plate1}, Value_{1plate2}, Value_{1plate3,} Value_{1plate4}\right\}$$

Where: Analytical Descriptors $MaximumValue = 1{,}055$

By applying the equation of extraction degrees of evidence the corresponding degrees of each plate are found. By Eq. (7):

$$\mu_{Platen} = \frac{\text{X } MeasuringValue}{VolumeMaximumValue} = \frac{Value_{1platen}}{1{,}055}$$

4.2.1 PAL2v Analysis—Summer

The extraction degrees of evidence process results in:

$$\mathrm{Group}_{R1vol\mathrm{Summer}} = \{\mu_{1plate1}, \mu_{1plate2}, \mu_{1plate3}, \mu_{1plate4}\}$$
$$\mathrm{Group}_{R1vol\mathrm{Summer}} = \{0.597156; 1.0; 0.4739336; 0.6635071\}$$

The contradiction effect algorithm is applied for obtaining the degree of evidence of volume—Local 1 in the summer.

$$\mathrm{Group}_{R1vol\mathrm{Summer}} = \{0.597156; 1.0; 0.4739336; 0.6635071\}$$
$$\mathrm{Group}_{R1vol\mathrm{Summer}} = \{0.597156; 0.628014; 0.6635071\}$$
$$\mathrm{Group}_{R1vol\mathrm{Summer}} = \{0.6288458; 0.628014\}$$
$$\mu_{R1vol\mathrm{Summer}} = 0.628430148795892$$

For the values of Table 2 the same procedures are made:

4.2.2 Extraction of Degrees of Evidence—Summer—Fresh Weight

$$\mu_{1plate1} = \frac{702}{1,101} = 0.6376021 \quad \mu_{1plate2} = \frac{1,101}{1,101} = 1.0$$
$$\mu_{1plate3} = \frac{623}{1,101} = 0.5658492 \quad \mu_{1plate4} = \frac{866}{1,101} = 0.7865576$$

4.2.3 PAL2v Analysis—Summer

The contradiction effect extracting algorithm is applied for obtaining the degree of evidence of fresh weight in Location 1—Summer.

$$\mathrm{Group}_{R1PFSummer} = \{\mu_{1plate1}, \mu_{1plate2}, \mu_{1plate3}, \mu_{1plate4}\}$$
$$\mathrm{Group}_{R1PFSummer} = \{0.6376021; 1.0; 0.5658492; 0.7865576\}$$
$$\mathrm{Group}_{R1PFSummer} = \{0.6376021; 0.693009; 0.7865576\}$$
$$\mathrm{Group}_{R1PFSummer} = \{0.693009; 0.702603046\}$$
$$\mu_{R1PFSummer} = 0.697767964478114$$

This value indicates the evidence of the contribution of Fresh Weight biomass on the “Fouling” in summer season.

4.2.4 Extraction of Degrees of Evidence—Summer—Dry Weight

$$\mu_{1plate1} = \frac{274.68}{443.20} = 0.6197653 \quad \mu_{1plate2} = \frac{443.20}{443.20} = 1.0$$
$$\mu_{1plate3} = \frac{372.90}{443.20} = 0.84138086 \quad \mu_{1plate4} = \frac{293.30}{443.20} = 0.6617779$$

4.2.5 PAL2v Analysis—Summer

The contradiction effect extracting algorithm for obtaining the degree of evidence of dry weight in Location 1—Summer season.

$$\begin{aligned}
\text{Group}_{R1PSSummer} &= \{\mu_{1plate1}, \mu_{1plate2}, \mu_{1plate3,}\ \mu_{1plate4}\} \\
\text{Group}_{R1PSSummer} &= \{0.61976; 1.0; 0.8413; 0.66177\} \\
\text{Group}_{R1PSSummer} &= \{0.84138086; 0.73113346; 0.661777\} \\
\text{Group}_{R1PSSummer} &= \{0.7358464; 0.7311334651\} \\
\mu_{R1psSummer} &= 0.733479528184588
\end{aligned}$$

This value indicates the evidence of the contribution of dry weight biomass on the "Fouling" in summer season.

4.2.6 Extraction of Degrees of Evidence—Summer—Ash Weight

$$\mu_{1plate1} = \frac{209.80}{326} = 0.64355828 \quad \mu_{1plate2} = \frac{326}{326} = 1.0$$
$$\mu_{1plate3} = \frac{216}{326} = 0.662576687 \quad \mu_{1plate4} = \frac{234.20}{326} = 0.7184049$$

4.2.7 PAL2v Analysis—Summer

The contradiction effect extracting algorithm is applied for obtaining the degree of evidence of ash weight in Location 1—Summer season.

$$\begin{aligned}
\text{Group}_{R1PashSummer} &= \{\mu_{1plate1}, \mu_{1plate2}, \mu_{1plate3,}\ \mu_{1plate4}\} \\
\text{Group}_{R1PashSummer} &= \{0.64355828; 1.0; 0.662576687; 0.7184049\} \\
\text{Group}_{R1PashSummer} &= \{0.7493891; 0.662576687; 0.7184049\} \\
\text{Group}_{R1PashSummer} &= \{0.7027961; 0.7184049\} \\
\mu_{R1PashSummer} &= 0.710495307174549
\end{aligned}$$

This value indicates the evidence of the contribution ash weight biomass on the "Fouling" in the summer.

4.2.8 PAL2v Analysis—Summer—Total Biomass

The contradiction effect extracting algorithm is applied for obtaining the degree of evidence of biomass in Location 1—Summer.

$$\begin{aligned}
\text{Group}_{R1TotalBioSummer} &= \left\{\mu_{R1volSummer}, \mu_{R1pfSummer}, \mu_{R1psSummer}, \mu_{R1PashSummer}\right\} \\
\text{Group}_{R1TotalBioSummer} &= \{0.62843014;0.6977679;0.733479;0.7104953\} \\
\text{Group}_{R1TotalBioSummer} &= \{0.676659;0.6977679;0.7104953\} \\
\text{Group}_{R1TotalBioSummer} &= \{0.69311094;0.6977679\} \\
\mu_{R1BioSummer} &= 0.695430519882402
\end{aligned}$$

This value indicates the total evidence of the contribution of biomass on the "Fouling" in Summer season.

4.3 *Extraction of Degrees of Evidence-Winter*

4.3.1 Extraction of Degrees of Evidence

$$\mu_{1plate1} = \frac{20}{1{,}055} = 0.0189573 \quad \mu_{1plate2} = \frac{600}{1{,}055} = 0.568720$$
$$\mu_{1plate3} = \frac{300}{1{,}055} = 0.284360 \quad \mu_{1plate4} = \frac{400}{1{,}055} = 0.379146$$

4.3.2 PAL2v Analysis—Winter

The contradiction effect extracting algorithm is applied for obtaining the degree of evidence of volume in Location 1—Winter.

$$\begin{aligned}
\text{Group}_{R1volWinter} &= \left\{\mu_{1plate1}, \mu_{1plate2}, \mu_{1plate3,}\mu_{1plate4}\right\} \\
\text{Group}_{R1volWinter} &= \{0.0189573;0.568720; 0.284360;0.379146\} \\
\text{Group}_{R1volWinter} &= \{0.4023691;0.284360;0.379146\} \\
\text{Group}_{R1volWinter} &= \{0.3483974;0.379146\} \\
\mu_{R1volWinter} &= 0.364096449118789
\end{aligned}$$

This value indicates the evidence of the contribution of biomass volume on the "Fouling" in the winter compared to the summer.

4.3.3 Extraction of Degrees of Evidence Winter—Fresh Weight

$$\mu_{1plate1} = \frac{27.60}{1{,}101} = 0.025068 \quad \mu_{1plate2} = \frac{680.66}{1{,}101} = 0.6182198$$
$$\mu_{1plate3} = \frac{348.40}{1{,}101} = 0.316439 \quad \mu_{1plate4} = \frac{446.17}{1{,}101} = 0.405240$$

4.3.4 PAL2v Analysis—Winter

The contradiction effect extracting algorithm is applied for obtaining the degree of evidence of fresh weight in Location 1—Winter.

$$\begin{aligned} \text{Group}_{R1PFWinter} &= \{\mu_{1plate1}, \mu_{1plate2}, \mu_{1plate3}, \mu_{1plate4}\} \\ \text{Group}_{R1PFWinter} &= \{0.025068;0.6182198;0.316439;0.405240\} \\ \text{Group}_{R1PFWinter} &= \{0.43751037;0.316439;0.405240\} \\ \text{Group}_{R1PFWinter} &= \{0.3818042;0.405240\} \\ \mu_{R1pfWinter} &= 0.393696547508046 \end{aligned}$$

This value indicates the evidence of the contribution of fresh weight biomass on the “Fouling” in the winter compared to the summer.

4.3.5 Extraction of Degrees of Evidence—Winter—Dry Weight

$$\mu_{1plate1} = \frac{7.48}{443.20} = 0.016877 \quad \mu_{1plate2} = \frac{173.20}{443.20} = 0.3907942$$
$$\mu_{1plate3} = \frac{98.68}{443.20} = 0.222653 \quad \mu_{1plate4} = \frac{128.08}{443.20} = 0.288989$$

4.3.6 PAL2v Analysis—Winter

The contradiction effect extracting algorithm is applied for obtaining the degree of evidence of dry weight in Location 1—Winter.

$$\begin{aligned} \text{Group}_{R1PSWinter} &= \{\mu_{1plate1}, \mu_{1plate2}, \mu_{1plate3}, \mu_{1plate4}\} \\ \text{Group}_{R1PSWinter} &= \{0.016877;0.3907942;0.222653;0.288989\} \\ \text{Group}_{R1PSWinter} &= \{0.27659;0.222653;0.288989\} \\ \text{Group}_{R1PSWinter} &= \{0.27659;0.25796\} \\ \mu_{R1psWinter} &= 0.267438750189945 \end{aligned}$$

This value indicates the evidence of the contribution of dry weight biomass on the “Fouling” in the winter compared to the summer.

4.3.7 Extraction of Degrees of Evidence—Winter—Ash Weight

$$\mu_{1plate1} = \frac{5,90}{326} = 0.018098 \quad \mu_{1plate2} = \frac{134}{326} = 0.4110429$$
$$\mu_{1plate3} = \frac{75.50}{326} = 0.231595 \quad \mu_{1plate4} = \frac{102.60}{326} = 0.3147239$$

4.3.8 PAL2v Analysis—Winter

The contradiction effect extracting algorithm is applied for obtaining the degree of evidence of ash weight in Location 1—Winter.

$$\begin{aligned}
\text{Group}_{R1PashWinter} &= \left\{\mu_{1plate1}, \mu_{1plate2}, \mu_{1plate3,}\mu_{1plate4}\right\} \\
\text{Group}_{R1PashWinter} &= \{0.018098;0.4110429;0.231595;0.3147239\} \\
\text{Group}_{R1PashWinter} &= \{0.29093;0.231595;0.3147239\} \\
\text{Group}_{R1PashWinter} &= \{0.29093;0.2763036\} \\
\mu_{R1pashWinter} &= 0.283714121123985
\end{aligned}$$

This value indicates the evidence of the contribution of Ash Weight biomass on "Fouling" in the winter compared to the summer.

4.3.9 PAL2v Analysis—Winter—Total Biomass

The contradiction effect extracting algorithm is applied for obtaining the degree of evidence of biomass in Location 1—Winter.

$$\begin{aligned}
\text{Group}_{R1TotalBioWinter} &= \left\{\mu_{R1volWinter}, \mu_{R1pfWinter}, \mu_{R1psWinter}, \mu_{R1PashWinter}\right\} \\
\text{Group}_{R1TotalBioWinter} &= \{0.364096;0.393696;0.267438;0.28371\} \\
\text{Group}_{R1TotalBioWinter} &= \{0.364096;0.3365409;0.28371\} \\
\text{Group}_{R1TotalBioWinter} &= \{0.3365409;0.326387\} \\
\mu_{R1BioWinter} &= 0.331502973431612
\end{aligned}$$

This value indicates the total evidence of the contribution of biomass on the "Fouling" in the winter compared to the summer.

4.4 Comparison Between the Two Biomass Results

It is applied the PAN algorithm for obtaining the Degree of Contradiction of biomass, from Local 1, between the two seasons: Summer and Winter.

$$\mu_{CtrBioSummer/Winter} = 0.68196355$$

This value indicates the contradiction that exists in relation to biomass in the "Fouling" between Summer and Winter in location 1.

It is obtained the Score between 0 and 1, by doing:

$$\text{as:}\quad \mu_{CtrBioSummer/Winter} \geq 0.5 \quad \text{then: } \mu_{CtrBio} = \frac{\mu_{CtrBioSummer/Winter} - 0.5}{0.5} \rightarrow$$
$$\mu_{CtrBio} = \frac{0.68196355 - 0.5}{0.5} \quad \rightarrow \quad \mu_{CtrBio} = 0.363871$$

If we wish to make comparisons between the secondary values it is possible, in the same way, use the PAN algorithm.

For example: The PAN algorithm is applied for obtaining the degree of contradiction of the dry weight in Location 1 and between the two seasons (summer and winter).

$$\mu_{CtrPSSummer/Winter} = 0.733020125$$
$$\text{as:}\quad \mu_{CtrPSSummer/Winter} \geq 0.5 \quad \text{then:}\quad \mu_{CtrBio} = \frac{\mu_{CtrPSSummer/Winter} - 0.5}{0.5} \rightarrow$$
$$\mu_{CtrBio} = \frac{0.733020125 - 0.5}{0.5} \quad \rightarrow \quad \mu_{CtrBio} = 0.46604025$$

We can verify that the dry weight analytical descriptor has a seasonal variation bigger than biomass, taking into consideration the fact that the "Fouling" community species are predominantly filter-feeding. They quickly incorporate water biomass with residues, according to the season of the year.

5 Conclusions

In this chapter, we presented a method that uses the foundations of paraconsistent logic to analyze and extract evidence comparing seasonal changes in species from marine ecosystems based on variations in biomass obtained by the "Fouling" studies. Three algorithms of the paraconsistent annotated logic with annotation of two values (PAL2v) were applied in biomass analytical descriptors taken from collectings made in 2006. To find new approaches—especially computational ones—for the treatment of these data is very important, since every day the process of studies of ingrained faunas ("Fouling") has become important in ecology. Great efforts are being made to find ways of deeply studying the changes in marine ecosystem due to factors that can be natural or by contamination of the environment by humans or through industrial processes. It was possible to list the degrees of evidence and contradiction from the values found which represent changes between the two seasons of the year: summer and winter.

In this work, it was seen that applying the algorithms of the paraconsistent annotated logic with annotation of two values (PAL2v) in analytical descriptors of "Fouling" ecological processes; it is possible to find degrees of evidence to create a metric for comparing two separate collectings seasonally. The method of analysis of using the PAL2v brings some advantages in relation to purely statistical methods. We can mention as the main advantage, the facility of the final results visualization, as well the easy access to all calculations and their internal results which might, through their values, highlight what are the most influential analytic descriptors in the final results. This identification is important because in many processes of ecosystem modifications there is the need to establish quick actions and with greater accuracy, directly and efficiently. The method also has the advantage of being algorithmic, which provides conditions for development of computational tools, capable of providing real-time monitoring in critical ecological processes applications.

The results obtained in this work are benchmarks and present values that can serve as a basis for conclusions regarding analyses of the marine environment. This will be possible through the seasonal study of fauna comparing its biomass and its behavior, which may be impaired by changes, or contaminants in the ecosystem.

References

1. Santos, I.R.; Friedrich, A.C., Ivar do Sul, J.A. Marine debris contamination along undeveloped tropical beaches from northeast Brazil Environ. Monit Assess. vol. 148, no. 1–4, pp. 455–462, January 2009
2. Spengler, A., Costa, M.F.: Methods applied in the studies of benthic marine debris. Mar. Pollut. Bull. **56**, 226–230 (2008)
3. Dufault, S., Whitehead, H.: Floating marine pollution in 'the Gully' on the continental slope, Nova Scotia Canada. Mar. Pollut. Bull. **28**, 489–493 (1994)
4. Nollkaemper, A.: Land-based dicharges of marine debris: from local to global regulation. Mar. Pollut. Bull. **28**(11), 649–652 (1994)
5. Giordano, F., Colonização de placas de *fouling* no estuário de Santos (Santos-SP): análises transicional e de sensibilidade complementando a abordagem baseada em recobrimentos específicos e diversidade. Doctorate Thesis (in Portuguese) - Instituto de Biociências da Universidade de São Paulo. Departamento de Ecologia. 141 p. São Paulo—ESP- Brasil (2001)
6. Da Silva Filho, J.I. et al.: An expert system structured in paraconsistent annotated logic for analysis and monitoring of the level of sea water pollutants. In: Vizureanu, P. (ed.) Expert Systems for Human, Materials and Automation, 392 p. InTech, October 2011. ISBN 978-953-307-334-7
7. Da Silva Filho, J.I., Lambert-Torres, G., Abe, J.M.: Uncertainty Treatment Using Paraconsistent Logic—Introducing Paraconsistent Artificial Neural Networks, p. 328. IOS Press, Amsterdam (2010). doi:10.3233/978-1-60750-558-7-i
8. Abe, J.M., Da Silva Filho, J.I.: "Inconsistency and Electronic Circuits". In: Alpaydin, E. (ed.) Proceedings of EIS'98 international ICSC symposium on engineering of intelligent systems. Artifi. Intell., vol. 3, pp. 191–197. ICSC Academic Press, Rochester (1998)
9. Da Silva Filho, J.I.: Algorithms based on paraconsistent annotated logic for applications in Expert systems. In: Segura, J.M., Reiter, A.C., Expert System Software: Engineering, Advantages and Applications. Nova Science Publishers, Inc, 400 Oser Avenue, Suite 1600, Hauppauge, NY 11788-3619, USA (2011). ISBN: 978-1-61209-114-3
10. Abe, J.M., Lopes, H.F.S., Anghinah, R.: Paraconsistent artificial neural networks and alzheimer disease . Preliminary Study Dement. Neuropsychologia **3**, 241–247 (2007)

11. Da Silva Filho, J.I., Lambert-Torres, G.; Ferrara, L.F.P., Mario, M.C., Santos, M.R., Onuki, A.S., Camargo, J.M., Rocco, A.: Paraconsistent algorithm extractor of contradiction effects—paraextrctr. J. Softw. Eng. Appl. vol. 4, p. 579–584 (2011)
12. Da Silva Filho, J.I.: Treatment of uncertainties with algorithms of the paraconsistent annotated logic. J. Intell. Learn. Syst. Appl. vol. 4, no. 2, pp. 144–153 (2012). doi:10.4236/jilsa.2012.42014
13. Woods Hole Oceanography Institute (WHOI). Marine fouling and its prevention. Annapolis, Maryland, U.S. Naval Institute (1952)
14. Borges, R.P., Fiori, E.F., Santos, J.A.P., Zaroni, L.P., Giordano, F., Kodama, L.K., Moya, G. C., Rosso, S., Ribeiro, R.B.: Zoobenthos of the santos estuarine system. In: Neves, R., Baretta, J.W., Mateus, M. (eds.) Perspectives on Integrated Coastal Zone Management in South America, p. 175–182. IST Press, Lisbon (2008)
15. Mario, M.C., Abe, J.M., Ortega, N.R., Del Santo, M.: Paraconsistent artificial neural network as auxiliary in cephalometric diagnosis. Artif. Organs **34**(7), E215–E221 (2010). doi:10.1111/j.1525-1594.2010.00994.x

An Algorithmic Method Supported by Paraconsistent Annotated Logic Applied to the Determination of Friction Factors for Turbulent Flow in Smooth Pipes

Maurício Conceição Mário, Marcílio Dias Lopes, Cláudio Luís Magalhães Fernandes, Dorotéa Vilanova Garcia, João Inácio Da Silva Filho, José Carlos Morilla, Clóvis Misseno Da Cruz, Jair Minoro Abe, Claudio Rodrigo Torres and Deovaldo de Moraes Júnior

Abstract The high complexity of the study of fluid flow is due to the existence of an excessive number of formulas to determine analytically the friction factor in pipelines. Currently, with more than a dozen formulas and the obligation of using graphics with readings on logarithmic scales for this purpose, the results are obtained with some degree of uncertainty. Recent work, with treatment of uncertainties, suggests that these complex calculations can be better performed with the basis of non-classical logic, such as the paraconsistent annotated logic (PAL) which has as a fundamental property the acceptance of contradictions. In this chapter we present a method that uses algorithms of PAL to make analysis in tests of fluid flow in smooth pipes. The PAL algorithms select and classify various results originating

M.C. Mário (✉) · M.D. Lopes · C.L.M. Fernandes · D.V. Garcia · J.I. Da Silva Filho
J.C. Morilla · C.M. Da Cruz · C.R. Torres · D. de Moraes Júnior
UNISANTA - Santa Cecília University, Oswaldo Cruz Street 288 CEP, Santos City, SP 11045-000, Brazil
e-mail: cmario@unisanta.br

D.V. Garcia
e-mail: dora@unisanta.br

J.I. Da Silva Filho
e-mail: inacio@unisanta.br

D. de Moraes Júnior
e-mail: deovaldo@unisanta.br

J.I. Da Silva Filho · J.M. Abe
IEA-USP- Institute of Advanced Studies of the University of São Paulo,
Prof. Luciano Gualberto Avenue nº 374 CEP, São Paulo City, SP 05508-900, Brazil
e-mail: jairabe@uol.com.br

J.M. Abe (ed.), *Paraconsistent Intelligent-Based Systems*,
Intelligent Systems Reference Library 94, DOI 10.1007/978-3-319-19722-7_7

from the various equations for the obtaining of friction factor and, according to the Reynolds number, they optimize the calculation application of hydraulic projects in smooth pipes.

Keywords Paraconsistent logic · Reynolds number · Load loss · Friction factor · Algorithm

1 Introduction

Researches related to hydraulic pipes treat the practical aspects that involve the analysis of incompressible fluid flow in forced and uniform pipelines in permanent regime [1, 2]. In these studies, we verify the phenomenon of loss of energy called "pressure loss", or "load loss", whenever a liquid seeps inside a tube from one point to another [1, 3].

Since long ago, the laws that are capable of ruling load loss in ducts are reason for in-depth studies and researches. The work made by researchers in this area over the years established laws supported by various empirical formulas. Some of these formulas were used in some degree of certainty in many engineering applications [2, 4–6]. Among these formulas we can mention the Hazen-Williams Formula, the Manning Formula and the Flamant Formula. However, new considerations and results of work of several researchers have shown that the use of these equations out of range of validity may cause errors. It is observed that the inaccuracy is bigger when the application domain of these formulas is broader, as intended by the authors of [7]. Therefore, in practice, the correct choices of the equations have a certain level of complexity. Recent researches have shown that this level of complexity can be attenuated by other means, which may be new empirical methods or by using techniques based on non-classical logics [8, 9].

In this work, we use the concepts of paraconsistent annotated logic (PAL) [9, 10], which is a non-classical logic, whose main feature is the ability of treating contradictions in their fundamental theory. With that in mind, the PAL [9] can offer means and methods for a comparative analysis serving as a support for the process of selection of results obtained by various formulas.

For comparison of methods on the extraction of the friction factor in smooth pipes, it was applied initially 16 different equations based on the visual results of Moody Chart [11–15]. Then, trials were made with a computational tool in which, through a simulating program, it was possible to make a selection of results applied to a situation established in the project.

The simulating program is designed with algorithms based in the paraconsistent annotated logic (PAL) [9] and its main paraconsistent algorithm is called "extractor of contradiction effects". In this work, a classification was possible in the range established by the Reynolds number, indicating which, among these 16 formulas, have the greater effectiveness in the calculation of load loss in smooth ducts [7, 13, 15]. As it is seen, through the concepts of the PAL, within certain conditions, we can analyze the accuracy and take advantage of the qualities of these formulas applied together.

1.1 Paraconsistent Annotated Logic PAL—Fundamental Concepts

Paraconsistent Logic (PL) belongs to the family of non-classical logics and has as its main feature the ability to treat contradictions in its fundamental theory. In the PL application process the conflict of information does not invalidate the conclusive results [9, 16, 17].

In paraconsistent annotated logic (PAL) annotations are assigned to propositional formulas. Each annotation, belonging to a finite lattice τ, assigns values to their corresponding propositional formula or proposition (*P*), such that:

$$\tau = \{(\mu, \lambda)\,|\mu, \lambda \in [0, 1] \subset \Re\}.$$

Let $\tau = < |\tau|, \leq, \sim >$ be a finite lattice (Fig. 1a) with a fixed operator. Such lattice is called *lattice of truth-values* and the operator $\sim$ constitutes the "meaning" of the negation symbol ¬ of the logic system which will be considered (Fig. 1b).

If *P* is a basic formula, the operator $\sim : |\tau| \rightarrow |\tau|$ is defined as:

$$\sim [(\mu, \lambda)] = (\lambda, \mu) \quad \text{where } \mu, \lambda \in [0, 1] \subset \Re.$$

It is considered that (μ, λ) is an annotation of *P* where: P(μ, λ) is a paraconsistent logic signal [9]. In this case the degrees of favorable evidence (μ) and unfavorable evidence (λ) compose an annotation that gives a logical connotation to the proposition *P*. Doing so, the association of an annotation (μ, λ) to a proposition *P* means that the degree of favorable evidence in *P* is μ, while the degree of unfavorable evidence is λ [7, 9].

By linear transformations in a unit square in the cartesian plane (USCP) [9] and the representative PAL lattice τ we can achieve the transformation.

$$T(X, Y) = (x - y, x + y - 1) \tag{1}$$

considering the components of the transformation Eq. (1) as the usual PAL terminology, where:

x = μ favorable evidence degree
y = λ unfavorable evidence degree

From the first term obtained in the ordered pair of the transformation Eq. (1), we have:

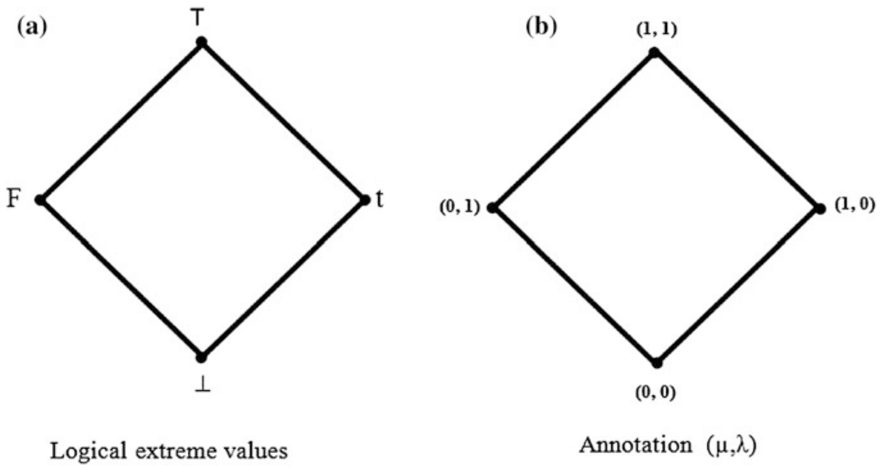


Fig. 1 Paraconsistent annotated logic representative lattice τ

$x - y = \mu - \lambda \rightarrow$ which we call degree of certainty (D_C). Therefore, the degree of certainty is calculated by:

$$D_C = \mu - \lambda \tag{2}$$

And their values, that belong to the set of real numbers in closed interval [+1, −1], and they are on the horizontal axis of the lattice τ [9, 10, 16], which is called the axis of degrees of certainty.

When D_C is +1, it means that the logical state resulting from the paraconsistent analysis is true (t), and when D_C is −1, it means that the logical state resulting from the analysis is false (F).

From the second term obtained in the ordered pair of the transformation equation, we can write:

$x + y - 1 = \mu + \lambda - 1 \rightarrow$ which is called degree of contradiction (D_{ct}). Therefore, the degree of contradiction is obtained by

$$D_{ct} = \mu + \lambda - 1 \tag{3}$$

and their values, which belong to the set of real numbers in the interval [−1, +1] are on the vertical axis of the lattice [9, 10, 16], which is called the axis of degrees of contradiction.

When D_{ct} is +1, it means that the logical state resulting from paraconsistent analysis is inconsistent (T), and when D_{ct} is −1, it means that the logical state resulting from the analysis is indeterminate (⊥).

From a detailed studydone in [9, 18], we can find the real certainty degree (D_{CR}) as being a D_C value without the effect of the contradiction. The D_{CR} value is on the axis of degrees of certainty in the PAL-Lattice and it is computed by:

If $D_C > 0$

$$D_{CR} = 1 - \sqrt{(1 - |D_C|)^2 + D_{ct}^2} \tag{4}$$

If $D_C < 0$

$$D_{CR} = \sqrt{(1 - |D_C|)^2 + D_{ct}^2} - 1 \tag{5}$$

From D_{CR} we can find its normalized value, called the resulting evidence degree (μ_{ER}) [9]. Therefore:

$$\mu_{ER} = \frac{D_{CR} + 1}{2} \tag{6}$$

2 Algorithms Used in Paraconsistent Analysis

From the equations of the paraconsistent annotated logic with annotations of two values (PAL2v) are extracted algorithms for the treatment of information signals [9, 16]. The processing of these signals is done in the form of degrees of evidence applied in paraconsistent analysis.

The first data processing that receives the signals is a transformation of the values measured in degrees of evidence. For this initial processing, we use a normalization algorithm called evidence extractor. Into the evidence extracting algorithm it is adjusted a universe of discourse (or interval of interest) that will determine the values of the degrees of evidence for any measurements made on physical quantity under analysis [9].

2.1 Evidence Degree Extracting Algorithm

The algorithm that transforms the measurements of physical quantities in degrees of evidence, with a directly proportional variation in the universe of discourse, is shown below.

1. Present the maximum boundary-value to form the universe of discourse.

$$\text{Valuemax} \ = \ldots\ldots\ldots\ldots\ldots$$

2. Present the minimum boundary-value to form the universe of discourse.

$$\text{Valuemin} \ = \ldots\ldots\ldots\ldots\ldots$$

3. Present the value measured of the physical quantities.
   Value Quantities $X = \ldots\ldots\ldots\ldots$ This value came from the measure of one of the transmitters.
4. Compute the favorable evidence degree through the equations:

$$\mu_1 = \begin{cases} \frac{Value_{QuantitiesX} - Value_{\min}}{Value_{\max} - Value_{\min}} & if \quad Value_{QuantitiesX} \in [Value_{\max}, Value_{\min}] \\ 1 \quad if \quad Value_{QuantitiesX} \geq Value_{\max} \\ 0 \quad if \quad Value_{QuantitiesX} \leq Value_{\min} \end{cases}$$

5. Compute the unfavorable evidence degree by complementing the favorable evidence degree.

$$\lambda = 1 - \mu_1$$

2.2 PAN—Paraconsistent Analysis Node

All the procedures related to the paraconsistent logical treatment will be made by an algorithm called PAN—paraconsistent analysis node [9, 16, 18]. The flux diagram and the symbol of the PAN is showed in Fig. 2.

The PANs' algorithms can be interconnected forming paraconsistent data processing networks. In this work, we used an algorithm named extractor of contradiction effects that it is formed by several PANs [9, 16]. We will give a brief description of this algorithm in the following section.

2.3 Contradiction Effect Extracting Algorithm

The contradiction effect extracting algorithm [9, 16, 18] receives a group of signals in evidence degree format and, regardless of any other external information, has the main function of making a paraconsistent analysis on their values by subtracting the effects caused by the contradiction. This is done through a network of PANs and the complete algorithm used in the extraction process of the effects of contradiction is shown in Fig. 3 by a block diagram.

The contradiction effect extracting algorithm presents in the output a single real resulting degree of evidence which is the representative value of the group [18].

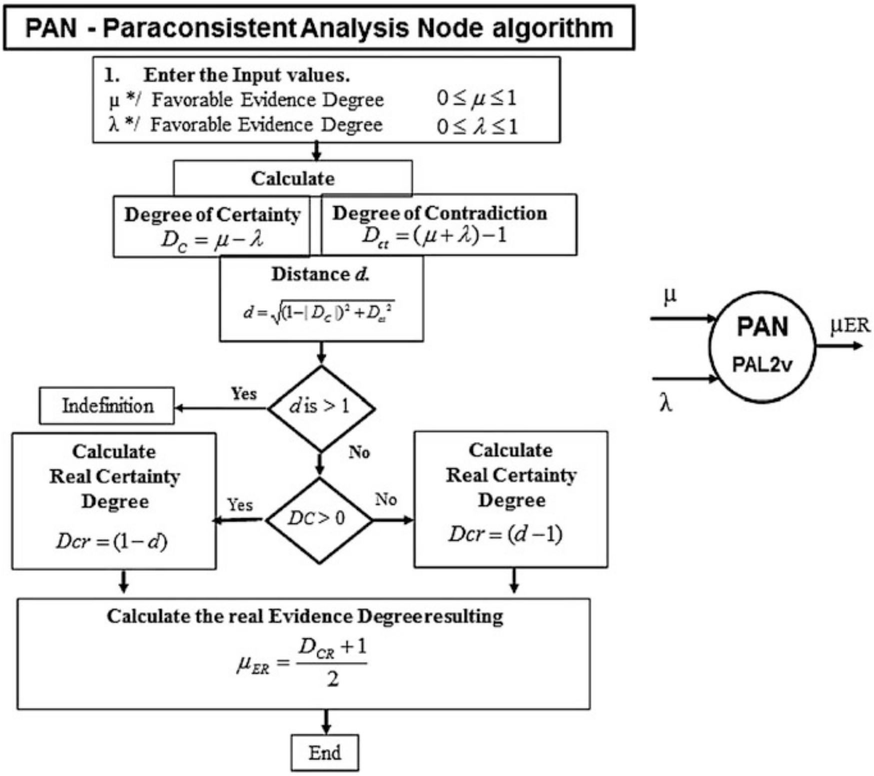


Fig. 2 The flux diagram and symbol of the PAN-Paraconsistent Analysis Node

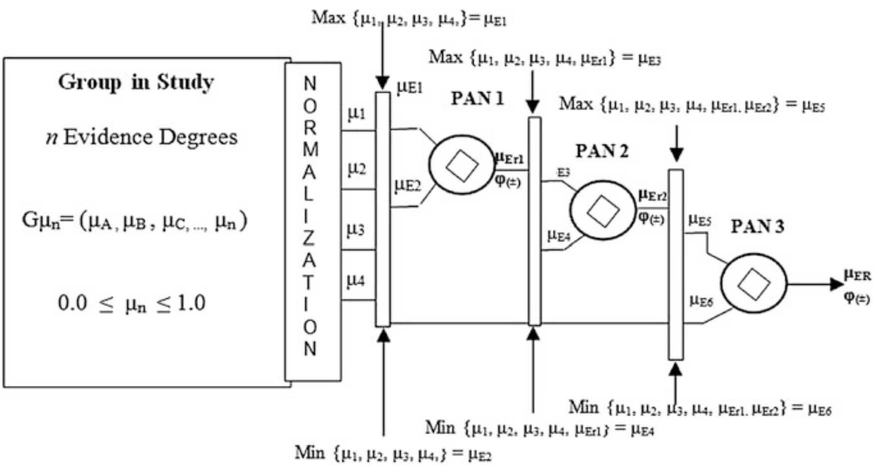


Fig. 3 The block diagram of the contradiction effect extracting algorithm

2.4 The Calculation of Load Loss

In the study of a hydraulic installation it is necessary to take into consideration some aspects of hydraulic scaling of forced pipeline, which are usually called tubes or pipes.

The calculations about the scaling of pipes are done more specifically by the determination of the type of flow and its directly proportional relation with the load loss.

It is known that by draining through a forced duct, the fluid is subject to pressure variations due to variation in the elevation of the pipe, the flow speed and also the fluid friction against the inner side of the duct wall and its accessories. When a liquid seeps from one point to another into a tube, it always generates a loss of energy, called pressure loss (or load loss). This power loss happens due to friction against the wall of the tube and is also caused by the viscosity of the flowing liquid. Thus, the higher the roughness of the pipe wall and the more viscous liquid is, the greater the load loss is. In pipes, the variation in flow rate is related not only to different areas of cross sections of the tube, as occurs in the reductions and expansions, but also to the degree of roughness and regularity of its inner surface. In both cases, this variation in flow speed causes load loss which can be divided into:

(a) Localized loss (due to singularities, such as expansions, reductions, bends, valves, etc.);
(b) Distributed loss (due to the friction of the fluid against the walls of the duct, along its whole length, with constant cross-sectional area).

The researches that aim to establish laws that might govern the loss in ducts were made a long time ago. Nowadays, a formula that was proposed in 1845, known as equation of *Darcy-Weisbach* [4] is the most accurate expression and is used universally for flow analysis in pipes.

The *Darcy-Weisbach* equation for determining losses is exposed in the following:

$$h_f = f \frac{L}{D} \frac{v^2}{2g} \tag{7}$$

where:

h_f is the load loss along the length of the tube (mca)
f is the friction factor of Darcy-Weisbach (dimensionless)
L is the length of pipe (m)
v is the velocity of the fluid inside the tube (m/s)
D is the inner diameter of the tube (m)
g is the local acceleration due to gravity (m/s^2)

For the study of load loss it is common to use the Experience of Reynolds, where it was obtained the classification of incompressible flow in permanent regime. Thus in hydraulic projects the Reynolds Number (Re) is used to identify the types of flow in pipes.

2.5 The Reynolds Number

The Reynolds Number, (abbreviated as Re, was named so in honor of Osborne Reynolds, Irish physicist and hydraulic engineer) is a dimensionless number used in fluid mechanics to calculate the flow regime of a given fluid in a tube or on a surface [2, 4].

Similarly, the Reynolds Number is used in the calculations for hydraulic installations as the ones that treat of centrifugal pumps in chemical and petrochemical industries for hydraulic processes which involves transportation of liquids. Its physical meaning is a quotient between the forces of inertia and viscosity, such that:

$$\mathrm{Re} = \frac{\rho \mathrm{v} D}{\mu} \tag{8}$$

where:

ρ is the fluid density (kg/m^3)
v is the average velocity of the fluid (m/s)
D is the diameter for the flow in the tube (m)
μ is the dynamic viscosity of the fluid (kg/m)

2.6 The Reynolds Number in Computing the Friction Factor

Despite of the success in applying the *Darcy-Weisbach* equation, it has the disadvantage that a secure way to determine the friction factor is needed. The most well-known equation for the friction factor was proposed by Colebrook-White [3, 4, 6]:

$$\frac{1}{\sqrt{f}} = -2\log_{10}\left(\frac{k}{3.7D} + \frac{2.51}{\mathrm{Re}\sqrt{f}}\right) \tag{9}$$

Being

K is the equivalent roughness of the pipe wall (m)
Re is the Reynolds number (dimensionless)

2.7 Types of Flow and the Reynolds Number

According to [6], a flowing is defined as turbulent when the structure of flow lines develops three-dimensional random movements, in which the velocity vectors of the particles have three-dimensional random components, in addition to average speed.

When the calculation is for incompressible fluid flow along pipes, the nature of the flow (laminar or turbulent) is determined by the parameter called the Reynolds Number [5]. However, the transition from laminar to turbulent flowing can occur according to various numbers of Re because the transition depends on the degree of flow disturbance and may be affected by vibrations in pipelines and roughness of the input region. Therefore, it is not possible to define accurately the corresponding values of Reynolds Numbers that indicate whether the flow is laminar, transitional or turbulent.

In engineering projects the following values are used for flow in tubes:

Laminar flow: $Re \leq 2000$
Transitional flow: $2000 < Re \leq 4000$
Turbulent flow: $Re \leq 4000$

Despite its use, the study of the flow of a real fluid is today somehow empirical. In many cases, mainly for high Reynolds Numbers, the theoretical computations do not always correspond to the results observed in practice.

2.8 The Calculation Formulas of the Friction Factor

It is seen in Eq. (9) of *Colebrook-White* that the term *f,* that appears twice represents the Darcy Friction factor, also known as the coefficient of friction. Thus, the calculation of the coefficient *f* is not immediate and there is not a single formula to calculate it in all possible situations. Currently, the Eq. (9) of *Colebrook-White* has been regarded as the most accurate law of flow resistance and has been used as a reference standard. However, it brings the inconvenience that the friction factor *f* occurs in both members of the equation with no possibility of being expressed in terms of the other dimensions. So, its resolution requires an iterative process that causes inconsistencies in the resolution and in the results obtained when propagating errors for calculations of load loss.

These difficulties led many researchers to strive to find explicit equations which could be used as alternatives to the Eq. (9) of *Colebrook-White*. In the course of time, in order to solve this problem, some more compact and more simple equations appeared. Therefore, they are easier to be memorized, but they have large deviations. On the other hand, other formulas, which are less compact and complex, appeared but with minor deviations. Many other formulas are published in the specialized literature, combining simplicity and precision with reduced errors compared to the friction factor computed with the *Colebrook-White* equation.

2.9 The Moody Chart

In practice, the friction factor *f* can be obtained in the Moody Chart (Fig. 4) which is the graphical representation in double logarithmic scale of the friction factor according to the Reynolds Number and the relative roughness of a pipe. However, the display of the values in the diagram becomes difficult to read due to the logarithmic scales, which gives rise to interpretations with different values, leading to errors in the projects calculations.

Despite this, due to its wide acceptance and practical use, we decided to do the reading on the Moody Chart as a reference for the computational model that will define the selection and classification of formulas in experiments of this work. The Moody Chart is shown in the following figure.

In consultations held in specialized literature, in order to find the equations of the friction factor, it was found that some have the friction factor as a function of the Reynolds Number and the relative roughness (ε/D) of pipe [7]. In this paper the smooth pipe feature was adopted with the value of the relative roughness considered ε/D = 0.0000001.

It was also verified that in some papers [5, 11, 19, 20] the friction factor is also known as the *Moody* Factor and some authors still relate the *Fanning* factor with the Reynolds Number. To establish paradigms in the construction of software in this first implementation of the algorithms, for these cases, the following correlation was used: *Moody* Factor = 4 × *Fanning* Factor.

Among the universe of formulas for the obtaining of the friction factor, used in studies related to fluid mechanics, were selected, for this paper, the ones considered most significant due to its use in practice.

2.10 The Selected Formulas for Study with Paraconsistent Logic

Initially, for this analysis, we used a group of 16 formulas for determining the selected factor of friction among those most commonly used in practice. They were developed by different authors for calculation of the load loss in smooth pipes. Then the results for the different equations went through a comparative test with readings in the Moody chart, connecting each formula to a precision error. Based on the value of the Friction factor, that establishes the slightest precision error obtained by equations according to the Reynolds number range, a classifier paraconsistent

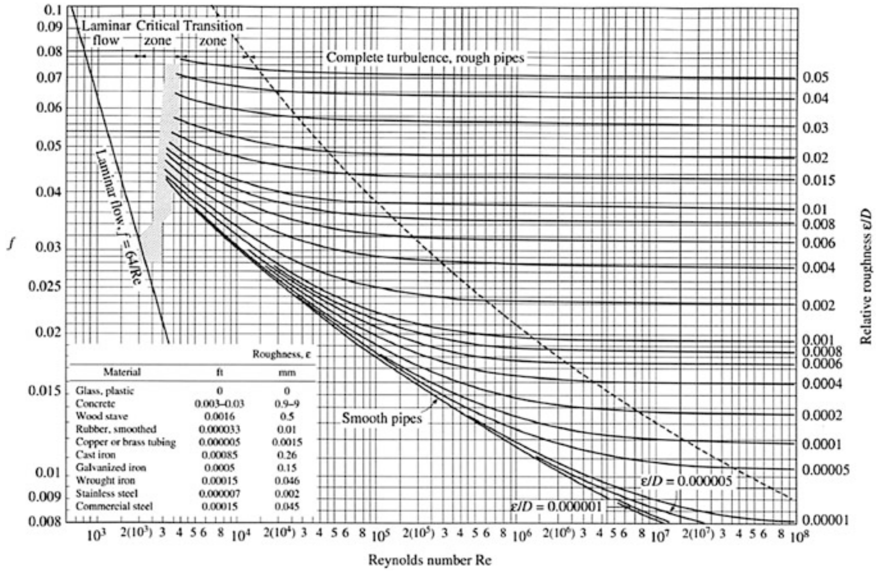


Fig. 4 Moody chart

computational program, which presents an optimization of these 16 results, was developed.

Following this empirical criterion for this test, 16 mathematical equations which correlate the calculation of the Friction factor f with the Reynolds number, were chosen, as set out below:

1. Blasius

$$f = 0.3164 \cdot \mathrm{Re}^{-0.25} \tag{10}$$

2. Drew

$$f = 0.0056 + 0.5\mathrm{Re}^{-0.32} \tag{11}$$

3. Bhatti

$$f = 0.00512 + 0.4572\mathrm{Re}^{-0.311} \tag{12}$$

4. Filonenko

$$\frac{2}{\sqrt{f}} = 1.58\ln(\mathrm{Re}) - 3.28 \tag{13}$$

5. Colebrook

$$\frac{2}{\sqrt{f}} = 1.5635 \ln\left(\frac{\text{Re}}{7}\right) \tag{14}$$

6. Techo

$$\frac{2}{\sqrt{f}} = 1.7372 \ln\left(\frac{\text{Re}}{1.964 \cdot \ln(\text{Re}) - 3.8215}\right) \tag{15}$$

7. Jain-Swami

$$\frac{2}{\sqrt{f}} = -1.737 \ln\left(\frac{5.72}{\text{Re}^{0.9}} + 2.7 \cdot 10^{-7}\right) \tag{16}$$

8. Round.

$$\frac{2}{\sqrt{f}} = 1.563 \ln\left(\frac{\text{Re}}{1.39 \cdot 10^{-7} \cdot \text{Re} + 6.5}\right) \tag{17}$$

9. Moody

$$f = 0.0055\left(1 + \left(0.02 + \frac{10^6}{\text{Re}}\right)^{\frac{1}{3}}\right) \tag{18}$$

10. Chen

$$\frac{2}{\sqrt{f}} = -1.737 \ln\left(2.7 \cdot 10^{-7} - \frac{2.1911}{\text{Re}} \ln\left(7.76 \cdot 10^{-8} + \left(\frac{7.149}{\text{Re}}\right)^{0.8981}\right)\right) \tag{19}$$

11. Zirrang.

$$\frac{2}{\sqrt{f}} = -1.737 \ln\left(2.7 \cdot 10^{-7} - \frac{2.18}{\text{Re}} \ln\left(2.7 \cdot 10^{-7} - \frac{2.18}{\text{Re}} \ln\left(2.7 \cdot 10^{-7} + \frac{14.5}{\text{Re}}\right)\right)\right) \tag{20}$$

12. Konakov

$$\frac{1}{\sqrt{f}} = 1.8 \log(\text{Re}) - 1.5 \tag{21}$$

13. Altshul

$$\frac{1}{\sqrt{f}} = 1.82 \log\left(\frac{\text{Re}}{100}\right) + 2 \tag{22}$$

14. Jain

$$\frac{1}{\sqrt{f}} = 2 \log\left(\frac{\text{Re}^{0.9}}{5.72}\right) \tag{23}$$

15. Nikuradse

$$f = 0.0032 + 0.221 \cdot \mathrm{Re}^{-0.237} \tag{24}$$

16. Churchill.

$$f = 8\left(\left(\frac{8}{\mathrm{Re}}\right)^{12} + \frac{1}{\left(\left(-2.457 \cdot \ln\left(\left(\frac{7}{\mathrm{Re}}\right)^{0.9} + 2.7 \cdot 10^{-7}\right)\right)^{16} + \left(\frac{37530}{\mathrm{Re}}\right)^{16}\right)^{1.5}}\right)^{\frac{1}{12}} \tag{25}$$

3 The Computational Process

The main goal of this paper is to develop an application of the algorithm of PAL [10, 21] in a hydraulic flow system to optimize results in several calculation formulas. To achieve this goal, the algorithms must choose, among the numerous equations used in the calculation of the Friction factor, the most convenient equation for the current proposal situation. Then, through a process of extraction of the effects of the contradiction, the final analysis will submit a corresponding value to the optimization of answers. The computational procedure for this analysis is presented in the block diagram of Fig. 5.

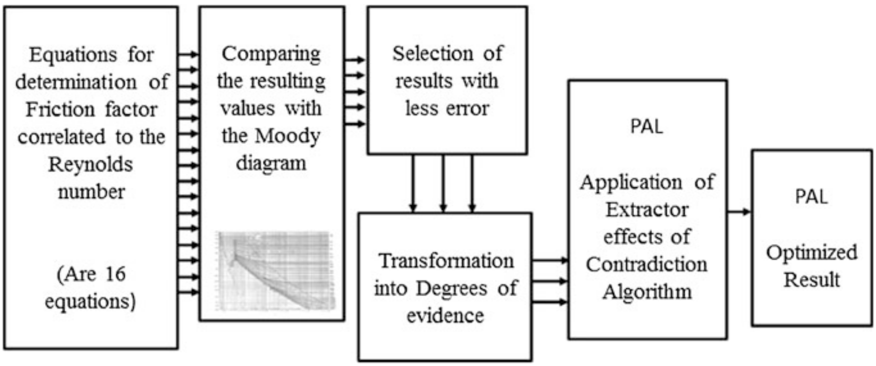


Fig. 5 Block diagram of optimization process for selecting the equation of the optimum result

3.1 Selection and Classification of Formulas

To select the best equation among the 16 presented ones, the Paraconsistent Computer System, through its algorithms, computes the Friction factor by applying each one of the equations. This procedure is evaluated by the comparison of the results with the query made to the *Moody* Chart. In this way, the paraconsistent computational system selects among the equations or the group of equations those that cause less precision error. At the end of the process, the Computer System makes the classification offering optimized value and serving as support tool in determining the value of the friction factor on real hydraulic systems projects.

3.2 Example of Performance of the Computational System Classifier

To build the model in real conditions, 16 equations were studied using the *Moody* Chart (Fig. 4) for the graphics reading, obtaining comparative data of references. Under these conditions 17 readings from the Reynolds Number were held using values of the order of 10^3–10^7.

The performance of the formula classifying system can be understood through the example of the application described below:

Initially, the values of friction factor graphically obtained were tabulated to form a comparative database on the model for the real conditions. Later, the previously selected mathematical equations built to the calculation of the Friction Factor were assembled in a spreadsheet giving the results for various simulated situations.

In these considerations, the calculated friction factor for each one of the equations were compared with the value found graphically through visual query in the *Moody* Chart by the classifying algorithm. With these values, the precision error was calculated for each result extracted from the 16 different equations.

In this classification it is seen that for every range of Reynolds Number there is a different equation that behaves best at calculating the friction factor when compared to visual analysis with the *Moody* Chart. These conditions can be observed in the Fig. 6 showing the precision error of the friction factor calculated by formulas originated from various authors.

From the values obtained in this step, the classifier system chooses only those formulas that present the best efficiency. These selected formulas will be distributed to the various ranges of Reynolds Number and their values are analyzed through the contradiction effect extracting algorithm to present a single response analysis.

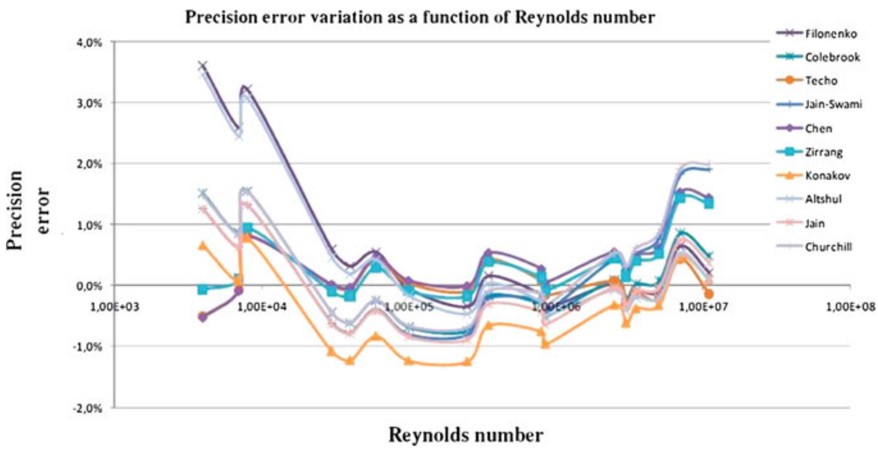


Fig. 6 Graph of the precision error of the calculated friction factor for various authors

Table 1 The range of variation of the Reynolds Number. (indicating the recommended equation)

Reynolds number	Recommended equation
Re < 2,000	Laminar Flow, use $f = 64/Re$
2,000 < Re < 4,000	Transitional flow, f is not calculated
4,000 < Re < 10,000	*Zirrang*
10,000 < Re < 1,000,000	*Zirrang, Techo and Chen*
1,000,000 < Re < 10,000,000	*Techo, Colebrook, Jain and Filonenko*
Re > 10,000,000	*Konanov and Altshul*

The comparison of the values obtained by the process described above allows the classifying system to make a classification using the contradiction effect extracting algorithm. Thus, with the result of the formula selected for the presented condition in various ranges of Reynolds Number, the contradiction effect extracting algorithm makes the analysis and offers optimized formula results, i.e. makes its indication to the range in which the use of the selected equation is recommended.

Table 1 shows the range of variation of the Reynolds Number indicating the recommended equation by the classifying system for the use in calculating the friction factor.

Through the description of the process of selection and classification shown previously, the classifying system establishes the parameters for the performance of the algorithms of paraconsistent logic. Thereby the Classifying system uses the contradiction effect extracting algorithm and selects the best formula for the condition presented in smooth pipes project.

3.3 Classifying System Programming

Software with the algorithm of Extracting Evidence

Begin

Function receives arrangements with the calculated values of the equations

If only one value of equation is provided, so there is no contradiction and the value is returned as response

Extracted the maximum and minimum values of the arrangement

Normalization of elements of the arrangement in the range [0,1]

Loop – To converge the elements of the arrangement to a degree of paraconsistent Evidence

Begin

Extract the maximum and minimum values of the normalized arrangement

Determines the maximum value and the positive evidence

Determines the minimum value and the negative evidence

Calculation of the degree of certainty and degree of contradiction

Calculation of the real degree of evidence

End

Undo normalization of the degree of evidence calculated

Returns the real degree of evidence

End

If you selected a group of formulas represented by various degrees of evidence the classifying software using the contradiction effect extracting algorithm indicates for the given range of the value of the Reynolds Number and the equation more accurately.

Classifier Software using the Contradiction effect extracting algorithm

```
Begin
    Enter with value of the flow
    Enter with value of the pipe diameter
    Enter with value of viscosity
    Enter with value of density
    Calculation of the area
    Calculation of velocity
    Calculation of the Reynolds Number
    Calculation of the friction factor using the Colebrook equation
    Calculation of the friction factor using the equation of Techo
    Calculation of the friction factor using the equation of Jaim-Swami
    Calculation of the friction factor using the equation of Chen
    Calculation of the friction factor using the Konanov equation
    Calculation of the friction factor using the Zirrang equation
    Calculation of the friction factor using the equation of Jain
    Calculation of the friction factor using the equation of Churchill
    Calculation of the friction factor using the equation of Filonenko
    Calculation of the friction factor using the Altshul equation

    If (Reynolds Number <= 2000)
            Begin
                    friction factor = 64/ Reynolds Number
                    Write (“The flow is Laminar”, friction factor)
            End
            Otherwise
    If (Reynolds Number > 2000) and (Reynolds Number <= 4000)
            Begin
                    friction factor = 64/ Reynolds Number
                    Write (“Pipe flow is Laminar. The flow in the pipe is not
                    rowdy. The friction factor is calculated in critical system
                    and has its value undefined.”)
            End
    Otherwise


    If (Reynolds Number > 4000) and  (Reynolds Number <= 10000)
            Begin
                    Write (“The most appropriate friction factor is through the
                    Zirrang equation ", result of the equation of Zirrang)
                    paraconsistent friction factor = contradiction extractor
                    method (Zirrang)
                    Write (“Using the extractor of contradiction of the PAL:
                    PAL friction factor, "Paraconsistent friction factor)
            End
```

```
Otherwise
If (Reynolds Number > 10000) and ( Reynolds Number <= 1000000)
    Begin
        Write ("the most appropriate friction factor is through the
        Zirrang, Techo and Chen equations", result of the equation
        of Zirrang, result of the equation of Techo, result of the
        equation of Chen,)
        paraconsistent friction factor = contradiction extractor
        method (Zirrang, Techo and Chen)
        Write ("Using the extractor of contradiction of the PAL:
        PAL friction factor," Paraconsistent friction factor)
    End
Otherwise
If (Reynolds Number>1000000) and (Reynolds Number <= 10000000)
    Begin
        Write ("the most appropriate friction factor is through the
        equations of Techo,Colebrook, Jain and Filonenko", result
        of the equation of Techo result of the equation of Cole-
        brook, result of the equation of Jain, result of the equation
        of Filonenko)
        paraconsistent friction factor = extractor method of con-
        tradiction (Techo,Colebrook, Jain and Filonenko)
        Write ("Using the contradiction effect extracting algo-
        rithm: LPA friction factor," Paraconsistent friction factor)
    End
Otherwise
If ( Reynolds Number > 1000000)
    Begin
        Write ("the most appropriate friction factor is through the equa-
        tions of Konanov and Altshul ", result of the equation of Konanov
        result of the equation of Altshul)
        paraconsistent friction factor = extractor method of contradic-
        tion (Konanov and Altshul)
        Write ("Using the extractor of contradiction Effects of the APL:
        APL friction factor," paraconsistent friction factor)
    End
    Calculation of the Friction factor for the equations of Konanov and Altshul.
    raconsistent friction factor = extractor method of contradiction (Konanov
    and Altshul)
End
```

As an example of application of the classifying software, Table 2 shows the results of its action steps. This simulation was made in the situation with the Reynolds Number in the range between 10,000 and 1,000,000.

Table 2 Simulating situation with Reynolds Number in the range **10,000** and **1,000,000**

Input values		
	Power flow in (m^3/h) 25	Pipe diameter (inches) 3
	Value of viscosity in cp 5.5	Density value in (t/m^3) 1.4
Calculations		
	Velocity Calculated in m/s 1.522782	Reynolds Number 29536.429960
Friction factors calculated		
COLEBROOK	**TECHO**	**JAIN and SWAMI**
0.023483	0.023586	0.023441
CHEN	**ZIRRANG**	**JAIN**
0.023591	0.023563	0.023438
KONAKOV	**FILONENKO**	**CHURCHIL**
0.023333	0.023729	0.023482
ALTSHUL		
0.023697		
Conclusions	*Calculations with the Contradiction effect extracting algorithm*	*The values calculated in accordance with Zirrang, Techo and Chen*
The factors of friction most suitable for this situation are • ZIRRANG the indicated value is 0.023563. • TECHO the indicated value is 0.023586 • CHEN the indicated value is 0.023591	**Maximum = 0.023591** **Minimum = 0.023563** **Normalized = ARRAY 0.83225** **Normalized ARRAY = 1** **Normalized ARRAY = 0**	**PAL Friction Factor 0.023552**
We can also use the Extractor of Contradiction Effects of LPA2v for all values calculated in accordance with 10 equations.	In this case, the friction factor is = 0.023116	

The results obtained through paraconsistent analysis show that computer programs based on non-classical logics, like the paraconsistent logic, can be applied in the classification of formulas working as an important tool in support of applied engineering.

4 Discussions

Due to inclusion of paraconsistent algorithms we saw that the classifying system made the selection and classification of the formulas in the most efficient way, suitable for the actual conditions presented.

With the computational resources presented in paraconsistent classifying system everything indicates that these paraconsistent analyses will bring the highest reliable results for the drainage project in smooth pipes. It was found that the equations proposed in this study, by being selected as optimizers for the calculation of the friction factor, presented precision errors of the order of 0.09–0.37 %.

In relation to the values that worked in practice, the results that express precision errors are extremely low, which makes the formulas selected reliable enough by validating the process applied in this work as support for applied research.

In a behavioral analysis of formulas, in the example below, it is shown that in certain ranges of variation of the Reynolds Number, authors such as *Colebrook* and *Churchill*, which are often used, presented lower results comparing with other authors. These tests showed the optional possibility for that classified equations to show the slightest mistake as compared to the initial condition. In future papers the number of formulas designed for the obtaining of the Reynolds Number will be increased, in order to allow adjusting the precision error.

5 Final Considerations

We presented in this paper an algorithmic method based on paraconsistent annotated logic able to do optimization and the classification of formulas for the calculation of the friction factor in drainage projects in smooth pipes. In order to make the classification of equations and perform the comparative analysis and other procedures for the calculation of load loss, we used three main algorithms: the extractor of degrees of evidence, paraconsistent analysis node and extractor of contradiction effects.

These three algorithms, all structured in paraconsistent annotated logic, work together to analyze and make the classification based on the efficiency of application of the 16 formulas created for the calculation of the friction factor. As proof of the efficiency of this method, it has been developed a computational paraconsistent classifying program which is able to cover paraconsistent analysis at intervals specified by the Reynolds Number. The correct choice and selection made computationally shows efficiency in supporting projects of smooth pipes drainage because it prevents the designer to have the need to consult frequently the *Moody* Chart which may take him to unnecessary fatigue and impact on a read error with negative consequences for the project. In this work, the tests were made with some restrictions as: use of minimum roughness, considerations only for smooth pipes, etc. However, the results obtained allow that in the future we can explore the possibilities of developing more robust paraconsistent expert systems for this type of application.

The results found with the tests are excellent when it is known that this is a first simulation process through the analysis with the paraconsistent logic. It was found that, through conditions established in the project, it is possible with paraconsistent analysis to get the selection of the most suitable equation to be used for a particular situation encountered in practice. One of the issues studied in the future will be the possibility of using the algorithms of the PAL for the investigation of the behavior of the precision error for several relative roughness. These conditions are also found in real projects.

References

1. Munson, B.R., Young, D.F., Okiisshi, T.H.: Fundamentals of Fluid Mechanics. Wiley, New York (1998)
2. Tullis, J.P.: Hydraulics of Pipelines. Wiley, New York (1989)
3. Colebrook, C.F.: Turbulent flow in pipes with particular reference to the transition region between the smooth and rough pipe laws. Proc. Inst. Civil Eng. **12**, 393–422 (1939)
4. Darcy, H.: Recherches expérimentales relatives au mouvement de l'eau dans les tuyaux, Mallet-Bachelier, Paris. 268 pages and atlas (1857) (in French)
5. Olujic, Z.: Compute friction factors fast for flow in pipes. Chem. Eng. **88**, 91–93 (1981)
6. Colebrook, C.F., White, C.M.: Experiments with fluid-friction in roughened pipes. Proc. R. Soc. London **161**, 367–381 (1937)
7. Serghides, T.K.: Estimate friction factor accurately. Chem. Eng. **91**, 63–64 (1984)
8. Abe, J.M., Da Silva Filho, J.I.: Inconsistency and electronic circuits. In: Alpaydin, E. (ed.) Proceedings of EIS'98 International ICSC Symposium on Engineering of Intelligent Systems, vol. 3, pp. 191–197. Artificial Intelligence, ICSC Academic Press, Rochester, 1998
9. Da Silva Filho, J.I., Lambert-Torres, G., Abe, J.M.: Uncertainty Treatment Using Paraconsistent Logic—Introducing Paraconsistent Artificial Neural Networks, 328 pp. IOS Press, Amsterdam, Netherlands, 2010. doi:10.3233/978-1-60750-558-7-i
10. Da Silva Filho, J.I.: Algorithms Based on Paraconsistent Annotated Logic for Applications in Expert Systems. In: Segura, J.M., Reiter, A.C. (eds.) Expert System Software: Engineering, Advantages and Applications. Nova Science Publishers, Inc. 400 Oser Avenue, Suite 1600, Hauppauge, NY 11788-3619, USA, 2011. ISBN: 978-1-61209-114-3
11. Nakayama, Y., Boucher, R.F.: Introduction to Fluid Mechanics. Butterworth Heinemann, Oxford (1999)
12. Cengel, Y.A., Cimbala, J.M.: Fluid Mechanics. McGraw Hill, New York (2006)
13. McDonough, J.M.: Lectures in Elementary Fluid Dynamics: Physics, Mathematics and Applications. University of Kentucky, Lexington (2004)
14. Gulyani, B.B.: Simple equations for pipe flow analysis. Hydrocarbon Process. **78**, 67–70 (1999)
15. Sonnad, J.R., Goudar, C.T.: Turbulent flow friction factor calculation using a mathematically exact alternative to the Colebrook-White equation. J. Hydraul. Eng. **132**(8), 863–867 (2006)
16. Da Silva Filho J.I.: Treatment of uncertainties with algorithms of the paraconsistent annotated logic. J. Intell. Learn. Syst. Appl. **4**(2), 144–153 (2012). doi:10.4236/jilsa.2012.42014
17. Abe, J.M., Lopes, H.F.S., Anghinah, R.: Paraconsistent artificial neural networks and Alzheimer disease—a preliminary study. Dement. Neuropsychol. **3**, 241–247 (2007)
18. Da Silva Filho, J.I., Lambert-Torres, G., Ferrara, L.F.P., Mc, Mario, Mr, Santos, As, Onuki, Jm, Camargo, Rocco, A.: Paraconsistent algorithm extractor of contradiction effects—Paraextrctr. J. Softw. Eng. Appl. **4**, 579–584 (2011)
19. Sonnad, J.R., Goudar, C.T.: Explicit friction factor correlation for pipe flow analysis. Hydrocarbon Process. **84**, 103–105 (2005)
20. Ludwig, E.E.: Applied Process Design for the Chemical and Petrochemical Plants, Third edn, vol. 1. Gulf Professional Publishing (1999)
21. Mario, M.C., Abe, J.M., Ortega, N.R., Jr Del Santo, M.: Paraconsistent artificial neural network as auxiliary in cephalometric diagnosis. Artif. Organs **34**(7), 215–221 (2010). doi:10.1111/j.1525-1594.2010.00994.x

Paraconsistent Logic Study of Image Focus in Cylindrical Refraction Experiments

Paulo Henrique Ferraz Masotti and Roberto Navarro de Mesquita

Abstract Automation techniques have increased their applications in different areas of knowledge areas. Digital Image Processing is one of the most important application areas. Image processing algorithms have been developed to automate autofocus in digital cameras, to evaluate focus quality, and many other industrial automation tasks. In scientific use, image fidelity is determinative as blurred pictures may induce erroneous conclusions on imaged-object size, position, shape, and volume evaluation. For this reason, plenty of algorithms have been created to avoid these mistakes and to ensure a precise focus. However, these new algorithms' uprising has produced some contradictory results. To solve these inconsistencies, the use of Paraconsistent Logic (PL) can be an important method to provide parameters to measure lack of information, indicating a paracomplete condition. Images with cylindrical refraction effects are important examples of how PL can be applied to solve focus inconsistencies. This work analyses experimental acquired images from objects inside glass cylindrical tube typically used in a natural circulation facility. This experiment is used as basis to exemplify the importance of using PL to evaluate different focus measurements in order to obtain good flow parameters estimation. Some intelligent algorithms are used to predict and to correct these possible inconsistencies on optical distortion evaluation, which is directly related to focus definition and estimation. As a result, object dimensions estimation can have its accuracy enhanced.

1 Introduction

The recent developments of advanced digital technologies have caused image concept to enlarge in the last years. Traditionally an optical field, imaging through digital equipment is progressively using a wider range of physical properties of

P.H.F. Masotti (✉) · R.N. de Mesquita
Nuclear Engineering Center, Instituto de Pesquisas Energéticas e Nucleares, IPEN/CNEN-SP, São Paulo, Brazil
e-mail: pmasotti@ipen.br

J.M. Abe (ed.), *Paraconsistent Intelligent-Based Systems*,
Intelligent Systems Reference Library 94, DOI 10.1007/978-3-319-19722-7_8

imaged objects. These imaging systems comprise many scientific areas related to medical physics, material analysis, microscopy, astronomy, satellite scanning systems, and many others.

The continuous decrease of digital equipment costs during the last decades is responsible for an increasing number of imaging system users and vice versa. This growth in number of users has enabled many scientists from different areas to acquire and analyze digital images related to their own research environment. Most of these new studies are related with phenomena that were previously observed, but could not be appropriately registered because of appropriate equipment's high cost.

Multiphase flow phenomena is an important area in which there was a significant number of recent publications related to digital visualization [1–7]. Thermo-hydraulic studies of multiphase flow using digital visualization have specific problems mostly related with image deformation resulting from cylindrical refraction effects [8]. Image deformations also imply focus imprecision that can cause misinterpretation of estimated parameters. This work will analyze some of these difficulties to show how Paraconsistent Logic (PL) can appropriately deal with them.

Inconsistencies on focus-measure quality indexes applied to acquire digital images are sometimes misinterpreted as part of optical focus phenomena. These inconsistencies are much dependent on each experimental condition, and therefore there is no universal quality index. Alternatively, the use of complex logic systems as PL can enable a systematic, flexible and impartial analysis of the best index to unfold a precise focus, or other image quality property of interest.

This chapter will show how the use of Paraconsistent Logic may improve a correct focus evaluation. An example of appropriate focus imprecision and uncertainty quantification through a Paraconsistent Fuzzy Logic System (PFLS) is expounded.

2 Focus and Image Quality

An important portion of recent imaging studies presents image quality metrics developments [9–16]. It is important to state some digital image definitions in order to describe this problem more appropriately.

2.1 Digital Image Concept

Digital image is any image representation that can be stored as a binary code and consequently be stored and processed by computers.

The most common representation includes bit words (called bytes) that can be organized on matrixes proportional to original captured information and this representation is usually stored in binary files with specific format. Visual information capture process is realized by electronic micro-devices, usually image sensors, where the most common are called Charged Coupled Device (CCD) [17], Complementary Metal-Oxide-Semiconductor (CMOS) [18] and Foveon X3® sensor [19]. Electronic and computation complex systems are used to make digital information available to be evaluated by algorithms for posterior scientific analysis.

Therefore, digital imaging systems include a variety of not-film-based image capture mechanisms. Even if these systems are not 'photographic' in a strict sense, many of them use optical systems, have a limited spectral range and usually store images as digital data to be posteriorly analyzed. Some of the main imaging systems are: thermal imaging, computerized tomography, positron emission tomography, magnetic resonance imaging and ultrasound imaging [20].

2.2 Image Quality

New image compression techniques were developed in order to enable large amounts of image information transmission and storage. A quantitative measure of image quality was necessary to compare distortion and degradation due to lossy compression algorithms [21]. Many different applications need to stablish quality standards related to each image information content as can be seen on x-ray images quality measures [22]. A classical work on this matter was published by Eskicioglu and Fisher [9] where they propose Hosaka plots to be applied to reconstructed images. The importance of statistical measures is more general and can be applied in any application, as they take into account all image processing involved [12].

An important landmark in this research was the wavelet-based measure by Kautsky et al. [10] which was aimed at astronomical imaging. Following this development, a new sharpness metric base on local kurtosis, edge and energy information was developed [15] in order to be applied on high-quality image capture control.

Other measure technique proposal using spatial frequency response was proposed by Williams and Burns [16]. An important evaluation of sharpness measures can be found in more recent conference [23].

2.3 Focus Evaluation

Development of image quality measures has a parallel development of image focus metrics. Both areas have related efforts on new parameter findings. An important evaluation of focus measures applied in multi-focus imaging fusion is described on [14].

Ferzli and Karam [13] proposed another important image sharpness metric landmark called Just Noticeable Blur. A recent method where blur motion parameters are estimated based on a radial basis function neural network is

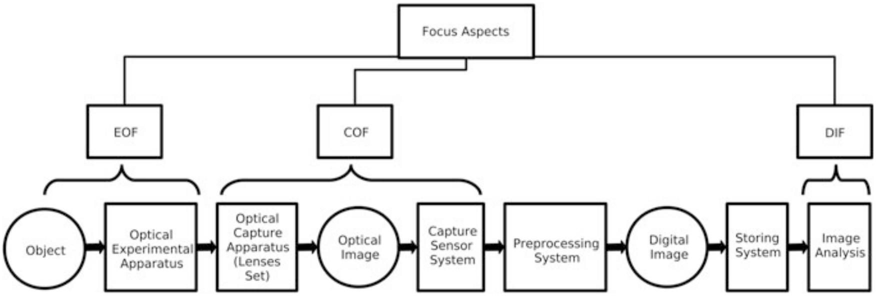


Fig. 1 Digital image acquisition, processing and storing phases and correspondent focus aspects used in this text (Experimental Optical Focus *EOF*, Capture Optical Focus *COF* and Digital Inferred Focus *DIF*)

described on [11]. These techniques are important to cope with degradation problems derived from misfocus, atmospheric turbulence, camera or object motion, and many others [22]. Resulting developments of focus-quality evaluation measurements have also been published. They are usually associated with depth maps and with estimation through focusing automation of servo-controlled acquisition mechanism improvements [24–27].

Focus study must include a set of sub-concepts that describes important image features related to optical capture conditions, digital processing mechanisms and storing resources used (Fig. 1). The word 'focus' is sometimes used to describe superimposed concepts, especially in scientific usage. In order to avoid misunderstanding, this text will use three different definitions to describe different focus aspects:

1. Experimental Optical Focus (EOF): this focus aspect is related to the optical system of the experimental object being studied (Fig. 1). Many times, the experimental apparatus has proper optical features that may include multiple refractions, reflexions, diffractions, and other optical conditions such as photomultipliers in neutron imaging. These optical conditions require specific optical studies in order to evaluate focus distances involved;
2. Capture Optical Focus (COF): this focus aspect is related to all optical properties involved in the capture hardware system (Fig. 1). This system includes the optical capture apparatus (lenses set or objective), the optical capture sensor and the digital camera mechanisms used to control and optimize capture. Usually, COF is mainly determined through the lenses optical center evaluation. The available commercial and scientific objectives usually present more than ten different lenses arrangement, which enables zoom adjustments associated with optical center determination changes;
3. Digital Inferred Focus (DIF): this focus aspect is related to evaluation systems used during image analysis (Fig. 1) which are aimed to obtain EOF estimations. This evaluation usually take into account most of EOF and COF properties including appropriate coupling conditions between them. Some misunderstanding

may arise when DIF is simply called 'focus' since DIF is essentially an inference result. Image analysis evaluates DIF based on stored image properties. These properties depend both on which electronic hardware, and on which processing software is used to retain acquired image information. Different hardware and software combinations may imply in different pixel resolution or special resolution, which can strongly influence DIF. On the other hand, EOF is estimated based on image acquisition, illumination and optical apparatus parameters. Ideally, DIF should be equal to EOF, but lack of information on acquisition conditions and other imprecisions may imply significant deviations.

Analyzed images enclose some important features that can be used as basis for a quality evaluation. Quality evaluation measures are used to control image acquisition and have been included in new automatic focusing mechanisms. Recover the *z* distance from object $P(x, y, z)$ to the center of the lens, based on focal distance, is a common task in Image Analysis.

Automatic focusing developments are aimed at a servo-controlled focus ring. This problem can be described as: "Given the projection $P' = (u, v)$ onto the focal plane of an object point $P = (x, y, z)$ (z unknown), what focal length f produces the sharpest definition of P' ? " [28].

Robot focusing ability [24] rises two important focus determination issues: how to command a servo-controlled focus ring to produce best image focus and how to estimate the object distance from lens.

The focal depth improvement (described on Sect. 3.1) in light microscopy through digital image processing techniques [25] is intrinsically related to many works who try to fuse multi-focus image [29–34].

Depth map estimation for 3d shape recovery is other common application [27]. Remote sensing and astronomy have similar problems [10].

Microscopic Particle Image Velocimetry is a recently developed technique that has produced many out-of-focus effects which need to have new experimental parameters and algorithms to deal with [26].

A common need in all these studies is a proper standardization of focus quality metrics. Knowledge about focus implies a better efficiency in different imaging phases: acquisition, pre-processing, storing and analysis (Fig. 1).

3 Optical Refraction and Focus Inaccuracy

3.1 Focus on Scientific Image Acquisition

Focus determination is very important to scientific imaging acquisition process as the more obtained image focus you are able to obtain the more information (since noise is properly discarded) about the 'object of interest' can be put to use. A first step in order to obtain good image focus is the experiment planning. Therefore is important to appropriately understand the photographic optics involved [35].

The main objective in scientific imaging is to be able to evaluate a specific parameter of interest and its associated error range [20]. The parameter of interest may be one of the geometrical properties like length, width, height or depth, but it can also be related to texture properties like rugosity [36, 37].

Focus range evaluation can be of great use for scientific parameter estimations in situations where the information of depth is of crucial importance.

Krotkov [24, 28] describes a classical scheme based on a thin lens arrangement for the distance evaluation from object to the optical center of lens, which he calls d_{out}. Based on simple geometric properties Krotkov concludes that this depth can be written as: $d_{out} \pm DOF$, where these parameters are defined [28] by Eqs. 1 and 2:

$$d_{out} = \left(\frac{d_{in} \times f}{d_{in} - f} \right), \tag{1}$$

$$DOF = \frac{2 d_{out}\, afc(d_{out} - f)}{a^2 f^2 - c^2 (d_{out} - f)^2}, \tag{2}$$

where d_{out} is the object distance, a is the aperture diameter, and c the smallest dimension of the sensor.

The depth-of-field (DOF) and depth-of-focus can be estimated based on aperture (a), blur circle diameter (c), d_{in} (image plane) and d_{out} (object plane) distances as can be seen on Fig. 2.

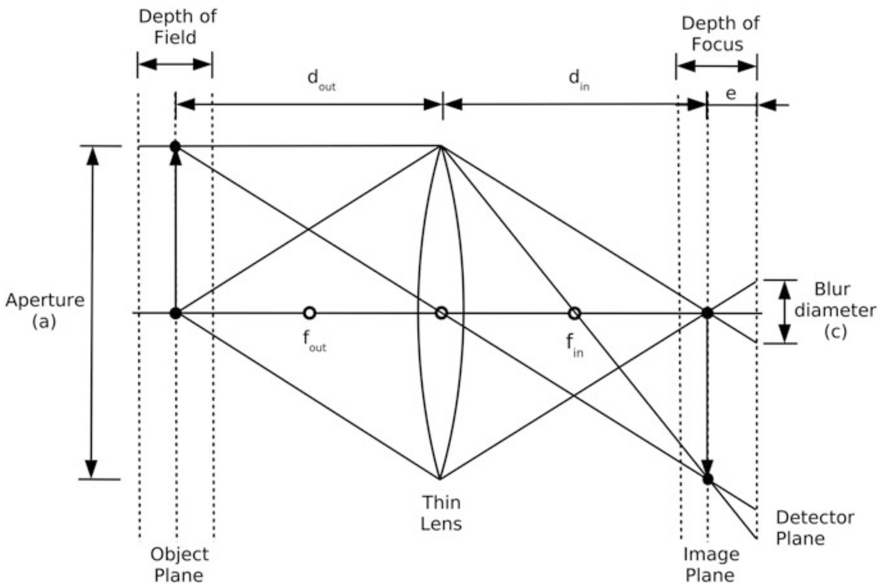


Fig. 2 Geometry scheme used by Krotkov [24] to estimate object depth and correspondent depth-of-field (DOF). Distances between lens center and the planes defined by object and image, are called d_{out} and d_{in}. Corresponding focus distances are f_{out} and f_{in}

Using the thin lens formula, a focus error (e) relative to the distance between image plane and detector plane can be expressed as in Eq. 3 [28]:

$$\frac{1}{d_{out}} + \frac{1}{d_{in}} - \frac{1}{f} = e. \tag{3}$$

Therefore, it is possible to determine the best Capture Optical Focus (COF) if f_{out} or d_{out} can be appropriately estimated. However, in many scientific experiments, f_{out} and d_{out} can only be estimated through experimental optics (EOF), which may be independent of optical capture apparatus (Fig. 1). The evaluation of focus error (e), which is associated with image blur (c) is of extreme importance in these experiments. In the next section, some metrics to evaluate Digital Inferred Focus (DIF) based on acquired digital image information will be described.

3.2 Selected Focus Metrics for Use in PFLS

A pair of focus metrics was selected to exemplify the implementation of a PFLS to obtain a focus diagnosis appropriately. The system will be centered in the question (proposition): 'is the imaged object in focus?'. This question has two main aspects to be analyzed. The first is if there is coherence in DIF evaluation. Focus metrics are important tools to be applied on digital images in order to evaluate the focus quality. This metric alone though, is unable to measure how close DIF is from EOF. Based solely on focus metrics, sometimes is impossible to obtain the object distance (z), or to estimate shape from focus. Shape from focus is a specific field of study, which will not be treated on this chapter.

Even considering only DIF measures, incoherence and discrepancies arises between different metrics, as each one is based on specific statistical properties of overall images. Evaluation of focus may lead to a set of image features that change and gain more importance for each specific metric.

The first selected metric to be evaluated by PFLS was the Gradient Magnitude Maximization method that is described by Krotkov [24] to be firstly cited by Tennebaum in 1970 in his Stanford thesis, 'Accommodation in Computer Vision'. The method is known as Tenengrad since 1983 [38], and is based in measuring edge characteristics change. The gradient $\nabla I(x, y)$ at each image point (x, y) is evaluated and magnitudes values (Eq. 4) greater than a threshold are summed.

$$|\nabla I(x, y)| = \sqrt{I_x^2 + I_y^2}, \tag{4}$$

where the partials can be estimated by Sobel operator convolution [39]. The Sobel operators are classical derivative masks (shown on Eqs. 5 and 6) to be applied through a convolution algorithm to digital images:

$$G_x = \begin{vmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{vmatrix}, \quad (5)$$

$$G_y = \begin{vmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{vmatrix}. \quad (6)$$

The gradient magnitude [24] is computed as shown on Eq. 7:

$$S(x, y) = \sqrt{|G_x * I(x, y)|^2 + |G_y * I(x, y)|^2}. \quad (7)$$

Tenengrad focus measurement (fm_{TN}) is evaluated by summing all $S(x, y)^2$ obtained from image that are greater than a stablished T threshold value. This chapter considered a zero T value for calculations. This is a conservative basis as $T = 0$ condition includes, in this evaluation, all the present noise contained in the studied bi-dimensional image $I(x, y)$.

The second selected metric used in PFLS evaluation was the Discrete Cosine Transform (DCT) [21, 40]. DCT is otherwise, a metric based on frequency domain properties, and is known to be less sensible to noise presence. The ratio between the DC energy (E_{DC}) and the AC energy (E_{DC}) from the DCT of the image (or part of the image) is considered as the main focus measure (*fm)* parameter described on Eq. 8:

$$fm_{CDT} = \frac{E_{AC}}{E_{DC}}. \quad (8)$$

This parameter is considered a focus measure as it quantifies the high-frequency components that constitute the image details [41].

Whether both techniques were used to analyze some object of interest inside digital image, traditional logic would lead to the simple following possibilities shown on Table 1:

Paraconsistent Logic maybe used as a tool to obtain more information from diagnosis results as presented on Table 1. Through PL treatment, more reliable and refined results are possible.

Table 1 Possible diagnosis for two focus metric techniques applied to an *object of interest*

1st technique		2nd technique	
Diagnosis	Truth value	Diagnosis	Truth value
Is focused	1	is focused	1
Is focused	1	out of focus	0
Out of focus	0	is focused	1
Out of focus	0	out of focus	0

4 Paraconsistent Logic (PL)

Paraconsistent Logic precursors were the Polish logician J. Łukasiewicz and the Russian philosopher N.A. Vasil'év. Both have developed their ideas independently and simultaneously by 1910. Łukasiewicz student, the Polish S. Jaskowski, first proposed a paraconsistent logical system. His proposal was published in 1948 including his ideas about logic and contradiction, and a system known as "Discursive Propositional Calculus". The logician Newton C.A. da Costa made an independent development from 1954 onwards with the first construction of a paraconsistent first-order predicate calculi and paraconsistent high-order logics [53].

This section is based on the work by researchers Abe and Da Silva Filho [43–53], where the *Paraconsistent Annotated Evidential Logic Eτ* [53] will be used as basis to PFLS. The basic definitions and conventions are as follows: for a given proposition *P*, a pair of values (μ, λ) is associated, where $0 \leq \mu \leq 1$ is the favorable evidence degree (f_e) in *P* and $0 \leq \lambda \leq 1$ is the unfavorable evidence degree (u_e) in *P*. This pair defines a domain that is called Hasse reticulate [42, 53]. This reticulate presents four extreme values defined by:

1. (1, 0) intuitively indicates 'total favorable evidence',
2. (0, 1) intuitively indicates 'total unfavorable evidence',
3. (1, 1) intuitively indicates 'totally inconsistent evidence',
4. (0, 0) intuitively indicates 'evidence absence', paracomplete condition.

An alternative representation is stated on Table 2:

These same values can be represented in Hasse reticulate as shown o Fig. 3:

The μ and λ values are used to define two variables: the *Degree of uncertainty*, D_{uc}, and the *Degree of certainty*, D_c. These variables are associated with *P* proposition. The transformation of the initials variables is performed by Eqs. 9 and 10:

$$D_{ct}(\textit{degree of uncertainty}) = \mu + \lambda - 1, \textit{where} - 1 \leq D_{uc} \leq 1, \tag{9}$$

$$D_c(\textit{degree of certainty}) = \mu - \lambda, \textit{where} - 1 \leq D_c \leq 1. \tag{10}$$

Table 2 Hasse reticulate extreme values

Annotated value		Logic state
f_e (μ)	u_e (λ)	Evidence diagnosis
0	0	Paracomplete
0	1	False
1	0	True
1	1	Inconsistent

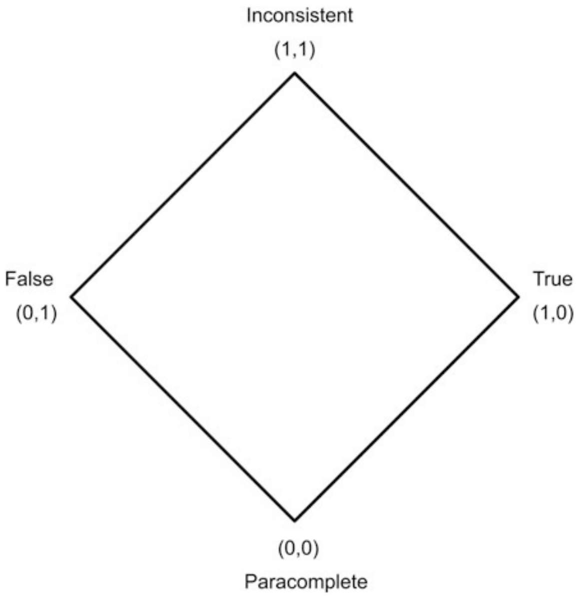


Fig. 3 Extreme logic states represented on Hasse diagram

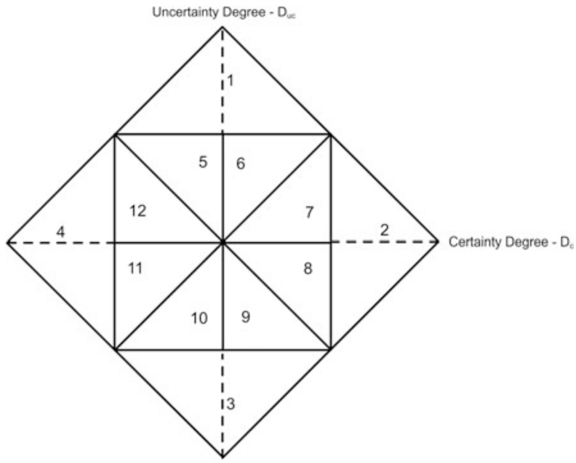


Fig. 4 D_c and D_{uc} logic dominion representation

Figure 4 shows a typical example of the reticulated Paraconsistent Annotated Logic subdivided into 12 regions, using the variables *degree of certainty* and *degree of uncertainty* [42, 54]:

The diagram shown on Fig. 4 represents twelve logic states:

1. Inconsistent,
2. True,
3. Paracomplete,

4. False,
5. Inconsistent tending to False,
6. Inconsistent tending to True,
7. Quasi-True tending to Inconsistent,
8. Quasi-True tending to Paracomplete,
9. Paracomplete tending to True,
10. Paracomplete tending to False,
11. Quasi-False tending to Paracomplete,
12. Quasi-False tending to Inconsistent.

In the theory of Fuzzy Sets, an x element of the universe of discourse set X is associated with the fuzzy set A by the $\mu_{A(x)}$ membership function, with values that vary in the [0, 1] range. This continuous A set can be represented by:

$$A = \int_X x_A(x_i)/x_i. \tag{11}$$

Considering that, using Paraconsistent Logic, a pair of membership functions (μ_A (x), μ_B (x)) characterizes a given proposition P; it can be shown that the fuzzy set A can be represented by:

$$A = \int_X (\mu_A(x_i)/x_i) + \mu_B(x_i)/x_i, \tag{12}$$

where the ‘plus’ signal on Eq. 12 may represent logical AND or OR operation. This last equation can be implemented and evaluated through Fuzzy Toolbox from Matlab [56].

4.1 PFLS Implementation

Image acquisition on cylindrical tube has some important optical difficulties due to multiple refractions that happen through cylindrical shape. That implies in variation on focus determination, as for each angle and position relative to the glass interface, image is formed in different distances. In practical terms, the main difficulty in this experiment is to evaluate if the object inside the tube (a vapor bubble inside liquid water) is ‘in focus’ (EOF) (Fig. 11 on Sect. 3).

A correct evaluation of which region of the acquired image is correctly focused (DIF), and which region may be considered blurred, is an important task in order to make possible a good image analysis. This correct evaluation may turn possible a more reliable estimation of flow parameters as void fraction.

One way to apply PL to evaluate if the image, or a part of it, is focused, is applying two different focus measure algorithms in order to obtain relative numeric values. These values can be used (as a universe of discourse) to construct two

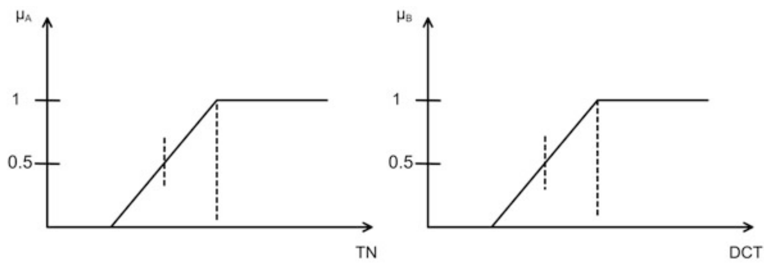


Fig. 5 Tenengrad and Direct Cosine Transform measures used as universe of discourse for μ_A and μ_B evaluation

membership functions that 'fuzzyfies' the 'crisp' focus measures in order to find μ_A and μ_B, as shown on Fig. 5.

The most reliable technique is used as *favorable evidence degree* (f_e). Considering the current image application example, f_e will be the Tenengrad metric. The *unfavorable evidence degree* (u_e) will be the DCT technique result. Both f_e and u_e are defined trough evaluated μ_A and μ_B. These techniques were chosen based on literature information on its efficiency, but for this example, any other metrics could have been chosen.

The following step is to evaluate μ and λ. This is done by using relations shown on Eqs. 13 and 14:

$$\mu = \mu_A, \tag{13}$$

$$\lambda = 1 - \mu_B \tag{14}$$

The λ evaluation is illustrated on Fig. 6:

Based on μ and λ values, D_c and D_{uc} can be evaluated based on Eqs. 9 and 10. Finally, these D_c and D_{uc} values will be used as input to inference rules that will result in a truth-value for proposition P that in present example is 'is the image on

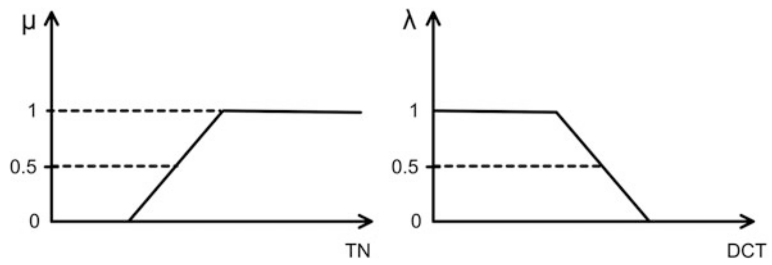


Fig. 6 Evaluation of μ and λ (degree of favorable and unfavorable evidence) using TN and DCT focus measures respectively

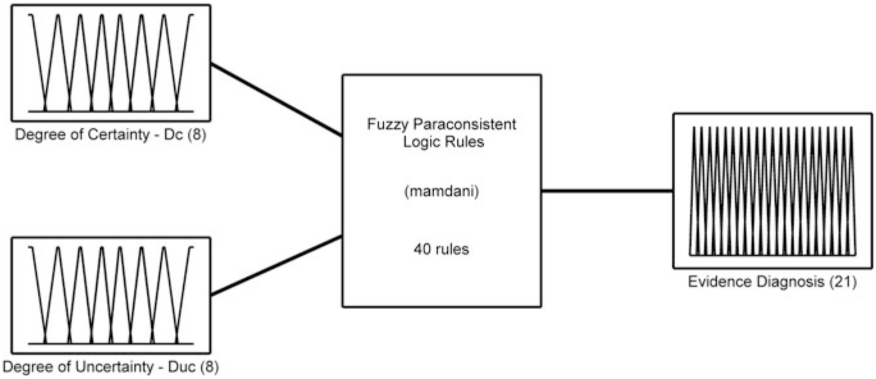


Fig. 7 PFLS block diagram showing two input variables (D_c and D_{uc}) with eight values each, base of rule with 40 rules and output variable (Diagnosis) with 21 possible values

focus?'. The Matlab Fuzzy Toolbox [56] is used to implement all needed fuzzy rules. Figure 7 shows a block diagram representing PFLS implemented.

The membership functions are detailed on Figs. 8 and 9.

Where the *Degree of Certainty* input variable presents the following eight possible fuzzy values (labels):

1. T = True,
2. QT = Quasi-True,
3. LT = Low True,
4. IT = Incipient True,

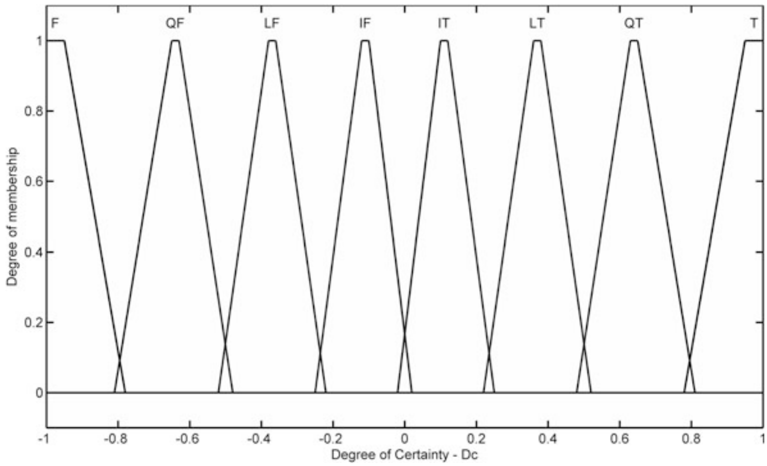


Fig. 8 Degree of certainty input variable composed of eight possible fuzzy values (labels) and their respective degrees of membership

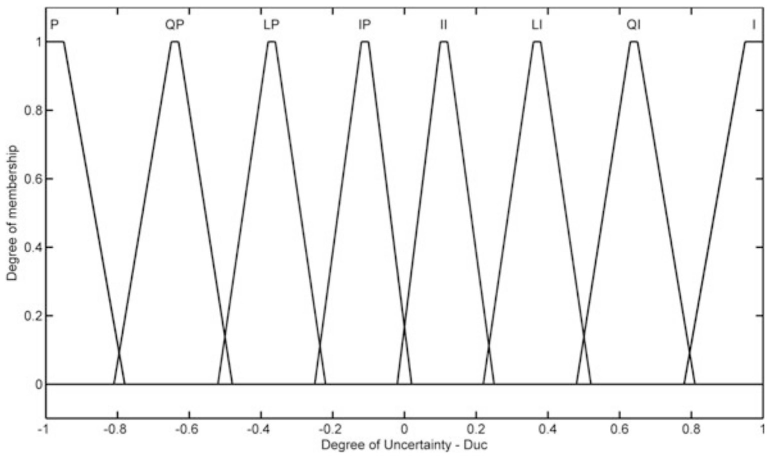


Fig. 9 Degree of uncertainty input variable composed of eight possible fuzzy values (labels) and their respective degrees of membership

5. IF = Incipient False
6. LF = Low False,
7. QF = Quasi-False,
8. F = False.

The *Degree of Uncertainty* input variable presents the following eight possible fuzzy values (labels):

1. I = Inconsistent,
2. QI = Quasi-Inconsistent,
3. LI = Low Inconsistent,
4. II = Incipient Inconsistent,
5. IP = Incipient Paracomplete,
6. LP = Low Paracomplete,
7. QP = Quasi-Paracomplete,
8. P = Paracomplete.

Table 3 exhibits all possible input combinations and their respective output results (*Evidence Diagnosis* logic states and their corresponding labels). It is a similar structure of a logic classic ‘truth table’, notwithstanding using Paraconsistent Logic’s states. These labels and relations are elaborated similarly as example shown on Fig. 4. The output membership functions were constructed to characterize the output variable *Diagnosis* that presents 20 different possible values or labels (Fig. 10) where I1 and I2 are the same output value. The PFLS implementation using Matlab Fuzzy Toolbox [56] impose the creation of two variable instances. These values qualify and quantify the logic state of a proposition P: ‘the image or part of it is in focus’. This is equivalent to give a *Paraconsistent Logic*’s answer to this question. The labels are:

Table 3 Logic values (output diagnosis) for each input combination

		Degree of certainty—D_c							
		F	QF	LF	IF	IT	LT	QT	T
Degree of uncertainty—D_{uc}	I	–	–	–	I	I	–	–	–
	QI	–	–	QIF	QIF	QIT	QIT	–	–
	LI	–	QFI	LFLI	LFLI	LTLI	LTLI	QTI	–
	II	F	QFI	LFLI	IFII	ITII	LTLI	QTI	T
	IP	F	QFP	LFLP	IFIP	ITIP	LTLP	QTP	T
	LP	–	QFP	LFLP	LFLP	LTLP	LTLP	QTP	–
	QP	–	–	QPF	QPF	QPT	QPT	–	–
	P	–	–	–	P	P	–	–	–

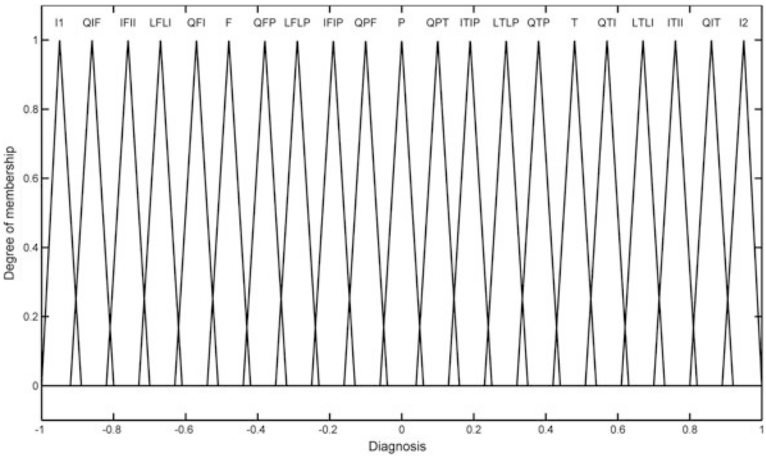


Fig. 10 Diagnosis output variable representation with 17 different possible values. Each value is represented by a triangular membership function, which is identified by a label

1. Inconsistent—I1,
2. Quasi-Inconsistent tending to False—QIF,
3. Incipient False and Incipient Inconsistent—IFII,
4. Low False and Low Inconsistent—LFLI,
5. Quasi-False tending to Inconsistent—QFI,
6. False—F,
7. Quasi-False tending to Paracomplete—QFP,
8. Low False and Low Paracomplete—LFLP,
9. Incipient False and incipient Paracomplete—IFIP,
10. Quasi-Paracomplete tending to False—QPF,
11. Paracomplete—P,
12. Quasi-Paracomplete tending to True—QPT,
13. Incipient True and incipient Paracomplete—ITIP,

14. Low True and Low Paracomplete—LTLP,
15. Quasi-True tending to Paracomplete—QTP,
16. True—T,
17. Quasi-True tending to Inconsistent—QTI,
18. Low True and Low Inconsistent—LTLI,
19. Incipient True and Incipient Inconsistent—ITII,
20. Quasi-Inconsistent tending to True—QIT,
21. Inconsistent—I2.

5 Image Acquisition on Cylindrical Tubes

Cylindrical tube visualization is an important and common technique used in many different multiphase flow studies.

5.1 Multiphase Flow Applications

Related applications can be found on petroleum extraction industry on profound waters where there is high temperature gradient between internal and external environment. This high gradient favors the existence of multiple phases inside the tube during transportation to the surface (topside facility). These called line risers usually have large diameter, although experiments are confined to much smaller diameters where visualization is important to check some of simulations results [1].

Microchip cooling beds are being tested based on micro-channels containing refrigerant fluid where critical heat flux and other parameters are investigated [2–4]. Most of these tubes are cylindrical shaped.

Nuclear applications also lead to refrigerant heat transfer studies, where most models use cylindrical tubes. Water refrigerated reactor projects commonly study two-phase flow composed of vapor and liquid phases flow through refrigerant tubes. Two-phase flow patterns are commonly studied based on image acquisition and analysis [5–7].

5.2 Experimental Acquisition

Specific experimental and theoretical problems related to flow patterns visualization will be used on this work to illustrate focus quality importance on proper image acquisition and analysis.

A proposed solution using Paraconsistent Logic enables comparison of different focus metrics based on experimental knowledge-based rules. Intrinsic difficulties on image acquisition and optics are described based on cylindrical visualization sections.

Cylindrical shape is present in most tubes, including stainless steel and other constituent materials. Experimental laboratories use transparent visualization sections in order to study phenomena using images. Neutron imaging and other systems are also used [6, 7] to study this area.

Therefore, this work analyses the difficulties to acquire optical images through cylindrical glass visualization sections. One of the main difficulties is optical deformation due cylindrical geometry leading to complex refraction optic effects [8]. A correction lens with cubic geometry around cylindrical tubes is used in different experiment configurations in order to attenuate multiple refraction effects. This lenses, however, also brings some additional difficulties on focus precise determination. Precise focus is important for implementing automatic acquisition and analysis procedures.

During acquisition procedure, focus control is one of the main parameters to assure precision on border estimates on posterior image analysis. Many different experimental parameters are usually affected directly by environment factors as spurious vibrations, luminosity, humidity and others. This group of factors can significantly alter final estimated focus and consequently lead to imprecision on border estimation from acquired image analysis and consequent poor control over camera settings during acquisition.

Determining the correct separation between two flow regions may be very important on characterizing flow patterns correctly.

5.3 Pattern Recognition

This chapter authors (R.N. de Mesquita and P.H.F. Masotti) have participated on a previous work on pattern recognition applied to Natural Circulation two-phase flow [5]. Typical images of two-phase flow inside cylindrical tubes are shown on Fig. 11.

In Fig. 11 is possible to observe that this kind of image presents different regions with different focus quality measures. Usually the cylindrical glass tube is filled with water. The optical conditions for image acquisition on these experiments include multiple refractions with different geometries and refractive indexes. Focus measure is smaller (blurred edges) for deepest bubbles as COF is adjusted to be near to the front tube interface. Image on Fig. 11 was acquired with camera and objective adjustments to restrict depth of field (DOF) in order to enable object distance estimation from camera. Based on these estimations is possible to distinguish which bubbles are nearer and which are more 'profound' (distant from camera). This distinction is important in order to evaluate individual bubble sizes and therefore be able to estimate void fraction values. Void fraction in this case is defined as the volume fraction of bubbles relative to total volume.

With closer observation it is possible to note on Fig. 11 that some bubble are 'out of focus' and some are perfectly focused. Most of these discrepancies occur because

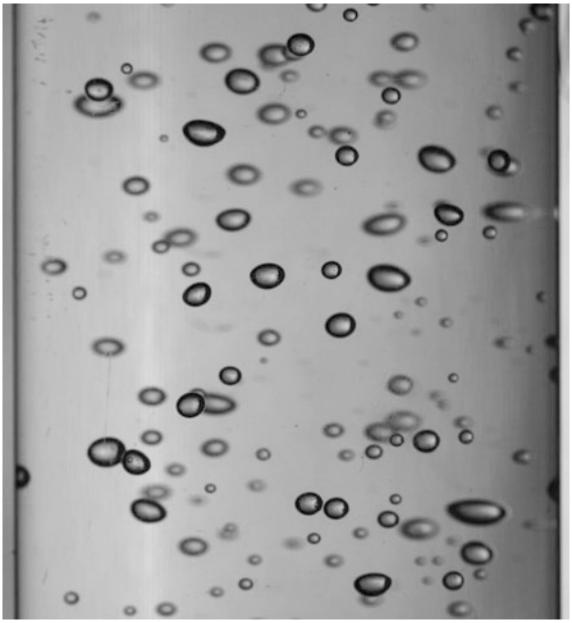


Fig. 11 Typical upward two-phase flow (air-water) image. Air bubbles present *ellipsoidal* shapes with different eccentricities. Some of these shapes are due to cylindrical refraction

these bubbles are located at different depths. Restricted COF camera adjustments imply that only the bubbles whose borders are in DOF range (COF ≤ DOF) will present good focus measure values.

Analyzing Fig. 11 is also possible to observe larger optical cylindrical aberrations occurring mainly on bubbles near the right and left tube sides. Cylindrical and spherical refraction aberrations are mostly important for object points that have larger angles from central optical axis, so bubble nearest to side surfaces are more deformed as can be observed. It is important to say that bubbles with bigger depths (farthest from camera) are also ellipsoidal shaped by cylindrical refraction, but in our experiment, are mostly ‘out of focus’.

Natural circulation phenomenon has been included in recent nuclear power plant projects as a heat removal mechanism for specific accidents mitigation [44]. The same mechanism is also used in chemical processes, refrigeration, electronics, etc.

Most of natural circulation test facilities are aimed to simulate conditions of heat removal on these reactors and study changes in flow pattern and hydro-dynamics. Two-phase flow behavior and different instabilities are well known for traditional nuclear power plants.

New two-phase flow pattern features have become available due recent developments on image acquisition and processing techniques. Flow patterns, pressure drop from each phase are correlated with void fraction and flow parameters in most of these studies [45].

5.4 Natural Circulation Facility (NCF)

An important application of focus quality evaluation is acquired images from flow patterns analysis. Authors have been studying [5] these patterns in a Natural Circulation Facility installed at *Instituto de Pesquisas Enérgeticas e Nucleares*, *IPEN/CNEN*, São Paulo, Brazil. This facility is an experimental circuit designed to provide data from one and two-phase flow behavior under natural circulation conditions. This loop is presently conceded to *IPEN/CNEN* by the Chemical Department of *Escola Politécnica da Universidade de São Paulo* (*USP*).

NCF is a rectangular assembly (with 2600 mm height and 850 mm width) of borosilicate glass tubes that are temperature resistant, with 38.1 mm internal diameter and 4.42 mm wall thickness each. The loop has a heated section, also made of glass tube with 76.2 mm internal diameter and 880 mm length. This section has two Ni–Cr alloy electric heaters (H1 and H2) in U form and stainless steel cladded. This heaters can deliver up to 8000 W. Approximately 12 l of demineralized water are used to fill the circuit.

Many thermocouples are distributed along the circuit to measure fluid and ambient temperatures. Two Validyne differential pressure transducers are used to measure the relative pressure at the heaters outlet and the water level in the expansion tank.

A data acquisition system is used to acquire sensor data. Visualization is possible in all regions of the circuit, and a visualization section can be mounted with one or more digital cameras and usually use backlight illumination.

Temperature measurements and image acquisition can be concomitantly done in order to characterize phase transition patterns and correlate them with different flow features observable with time synchronism [46].

5.5 Cylindrical Refraction Optics

A typical visualization section for cylindrical tube image acquisition is shown on Figs. 12b and 13 in two views.

A digital camera is represented on the optical axis relative to cylindrical tube in Fig. 13. The tube usually is filled with water in different flow conditions that may include multiple phases (two-phase flow), different flow speeds and flow dynamics.

In Fig. 14 two different conditions of refraction are represented. The focus distance varies due to multiple refractions through different medium. Two different focus f1 and f2 are represented to illustrate possible three (water-glass-air) refractions. These possible light trajectories are shown to reach the camera digital sensor plane enabling sharp capture if the camera DOF is appropriately set. Cylindrical geometry introduces a focus distance behavior that can be very complex and may induce misunderstanding situations.

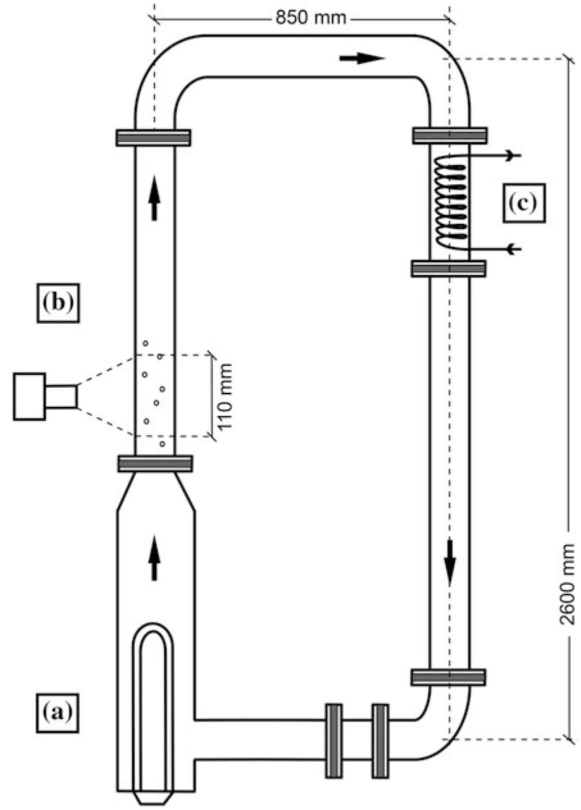


Fig. 12 Natural circulation facility (NCF) with: **a** electrical heating section, **b** visualization section and **c** cooling section

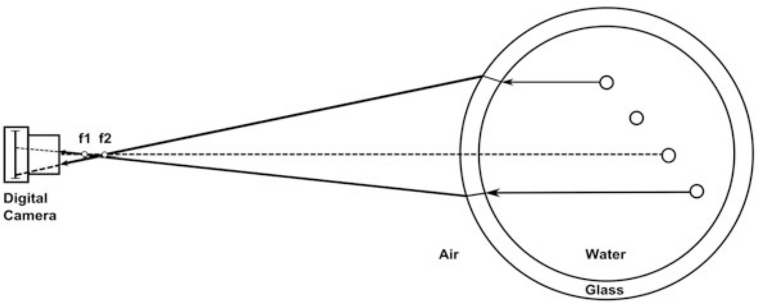


Fig. 13 Cylindrical refraction during digital image acquisition. Focus f1 and f2 differ due multiple refractive indexes and interface shapes

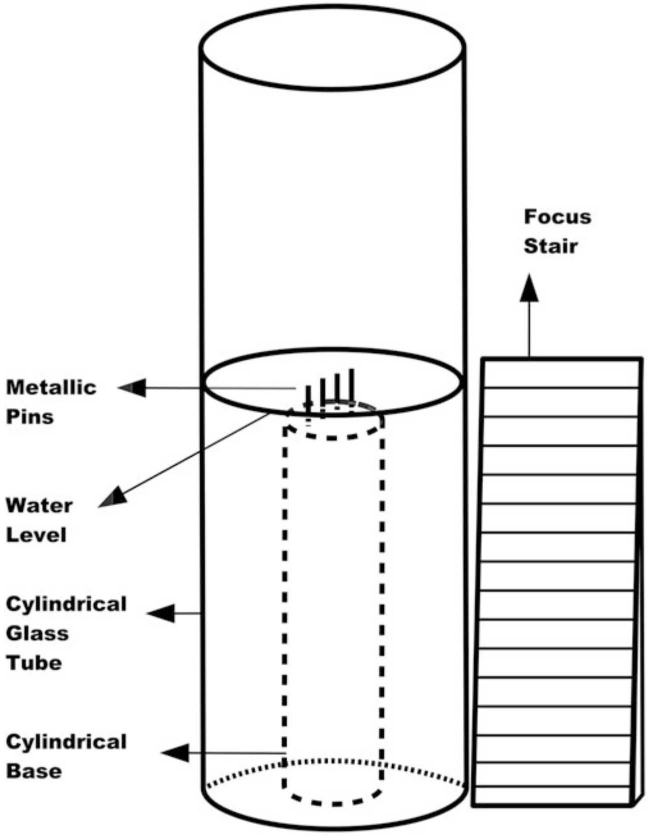


Fig. 14 Experiment to demonstrate focus determination with variable focus distances

Many times, refractive boxes are not available due to experimental conditions or to acquisition costs involved. In pattern recognition tasks is desirable to have a predictable behavior and correction maps are elaborated [8] for deformations due to cylindrical shape. However, for possible identification of bubble depth, that model presented by Thome et al. [8] cannot be applied. In the absence of refractive box, it would be interesting to evaluate where image plane and focus plane are formed for each acquired image point.

6 Experimental Focus Imprecision Measurement

In this section will be described some experiments where PL application can help to determine and compare different focus measures.

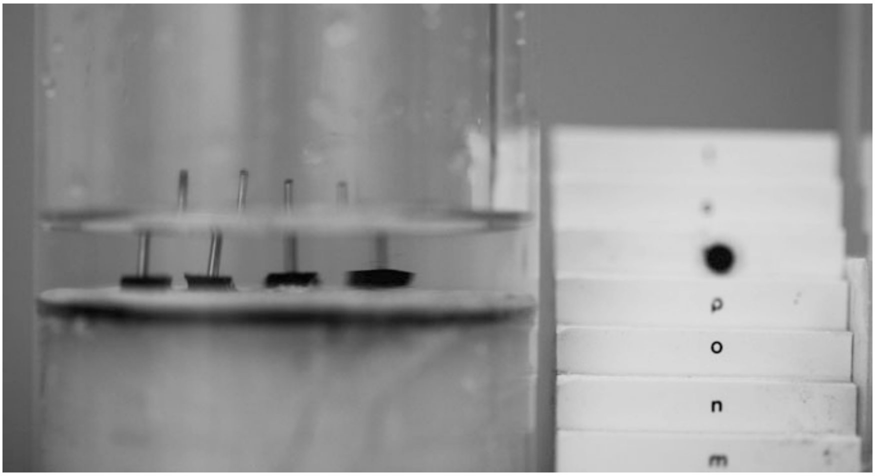


Fig. 15 Frontal image showing focus on the second deeper metallic pin. A focus stair is positioned besides the tube

6.1 Experimental Setup

A set of two experiments were done using an arrangement that makes possible image acquisition using different focus distances (Fig. 14). In this arrangement, a cylindrical glass tube of 46.7 mm external diameter and 3.85 mm of wall thickness is partially filled with water. Inside the glass tube, an opaque and rigid polyethylene cylindrical piece of 36.4 mm diameter is used as support for four metallic pins, which are fixed at different depths (5 mm distant from each other) as shown on Figs. 13, 14 and 15.

A CMOS full-frame digital camera with a 100 mm macro lens is mounted on a table trail that allows distance from object to be controlled. The metallic pins are partially immersed by the water that fills the glass tube, as shown of Figs. 14 and 15. It is possible to observe that, in Fig. 15, the upper part of pins do not suffer cylindrical distortion and are horizontally displaced relative to lower part. The unique exception is the third pin, which is on cylindrical optical axis and therefore suffers no significant refraction effect.

6.2 Image Acquisition with Different Focus Distances (COFs) Using a Focus Stair

In this first experiment, a macro lens was adjusted for best focus on consecutive stair steps, as shown on Figs. 14 and 15. Each consecutive step is labelled with consecutive letters in the following order: *m*, *n*, *o*, *p*, a reference circle (called '*r*' in this text) and letters *s* and *t*. From one step to another, there is a depth difference of 5 mm. A set of seven photographs were made, where each was focused in each labelled step.

The camera was adjusted to an aperture of f/2.8, using speed of 1/320 s, and ISO 3200. The distance from CMOS sensor and object was 49 cm for the last focus stair step (labelled *t*), which was aligned (same distance from sensor) with the farthest glass tube external wall.

6.3 Image Acquisition with Different Focus Distances (COFs) Adjustments for Each Metallic Pin

In the second experiment, a set of images was acquired adjusting focus distances based on the first experiment results. This time focus distances (COFs) were adjusted to the lower parts of the consecutive metallic pins (submersed). The metallic pins also have 5 mm depth difference between each other.

As can be easily verified, images under water suffer many different refractions due the convergent lens effect (Fig. 13), which changes focus distance for images projected onto digital camera sensor. It is possible to have multiple focuses for the same object, as there are beam lights with small incidence angle to the tube that presents different refraction angles, but still are projected over CMOS surface. As a result, we have blur conditions all over the acquired image, with very small regions where objects present good focus quality.

7 PL Application for a Comparative Evaluation of Focus Quality Measure Indexes

Different ROIs (*Regions of Interest*) were extracted from acquired images in order to estimate focus measures for each of these parts. Figure 16 shows the chosen regions for cropping. The image analysis was divided into two phases.

7.1 Image Analysis of Stair ROIs

Firstly, the ROI focus measures were compared one to the other, for different focus distances. For a comparative study, different ROIs were extracted from acquired images in order to measure the DIF for each of these image parts, as can be seen on Fig. 16. Figure 17 shows an example of cropped ROIs for each step on the same image (image 3, which is the third from the seven taken). The figure also shows, on its lower-right side, a cropped image from the second metallic pin (*p2* on Fig. 16). For each step ROI, Tenengrad (TN) and Discrete Cosine Transform (DCT) metrics

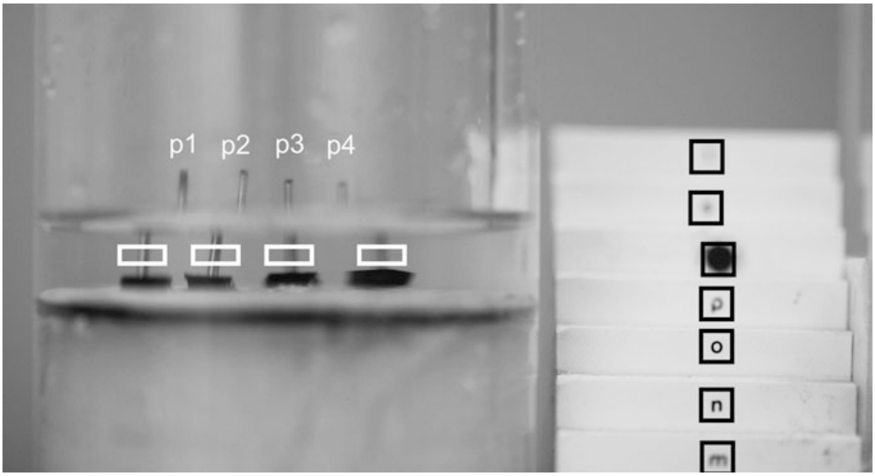


Fig. 16 Selected ROIs from acquired frontal images for analysis. The *white rectangles* are ROIs for each metallic pin (*p1*, *p2*, *p3* and *p4*) bellow water level. *Black rectangles* define ROIs for each focus stair step (*m*, *n*, *o*, *p*, *r*—reference circle, *s*, and *t*)

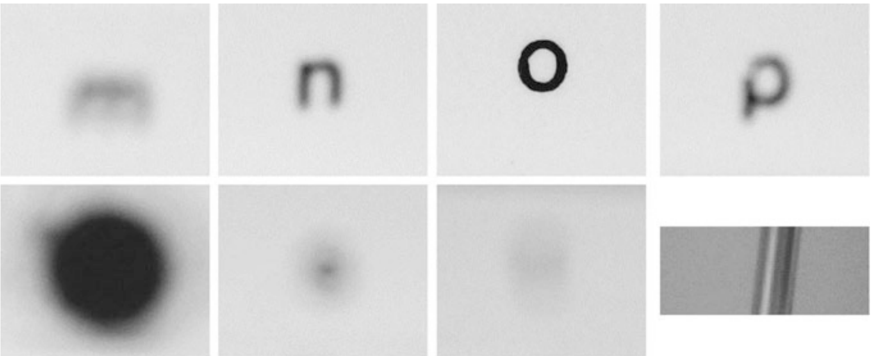


Fig. 17 Example of image cropped ROIs on each 'focus stair' step, with focus distance varying 5 mm from one ROI to the next. The *lower right* ROI shows the second nearer metallic pin (p2) which is at the same focus distance of step labelled with letter 'o'

were applied through algorithms described on Sect. 3.2. Figure 18 shows the TN and DCT results for each letter-labelled step shown on Fig. 17.

Table 4 presents the TN and DCT values shown on Fig. 18, and shows the D_c and D_{ct} values with the respective PFLS output values (Diagnosis) for each step. These values show that the step with label 'o' was considered 'in focus'. The True (T) PFLS output affirms that solely this step was 'in focus', while none of the other steps were considered True for proposition: 'is in focus?'.

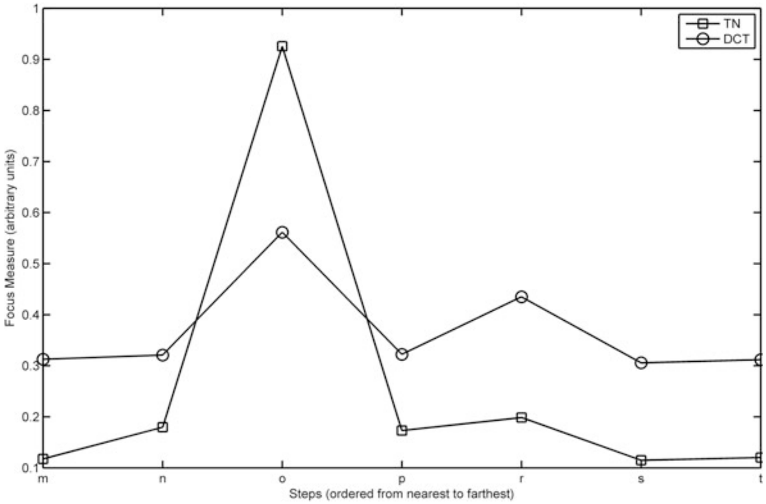


Fig. 18 Two selected focus measures evaluated for each ROI shown on Fig. 17

Table 4 TN and DCT focus measures and paracomplete diagnosis for third image

Image 3						
Label	TN	DCT	Gc	Gct	Out	Diagnosis
m	3.12E+07	1.64E+03	−0.78	0.06	−0.56	QFI
n	4.77E+07	1.69E+03	−0.53	0.22	−0.56	QFI
o	2.46E+08	2.95E+03	1.00	0.00	0.48	T
p	4.60E+07	1.69E+03	−0.54	0.19	−0.58	QFI
r	5.28E+07	2.29E+03	0.08	−0.26	0.28	LTLP
s	3.06E+07	1.61E+03	−0.82	0.08	−0.48	F
t	3.20E+07	1.64E+03	−0.77	0.07	−0.56	QFI

This first part of analysis confirmed that both metrics could detect the best focus object for all seven images. A direct correspondence between COF and EOF was verified for all images of ‘focus stair’ as these images don’t have any refraction optics occurring.

7.2 Image Analysis for Each Metallic Pin

In the second analysis, images from metallic pins were extracted and focus measures were applied. Here it is possible to note that the deeper or farther the pin is more inconsistencies on focus measure arises. Most of these inconsistencies are due to illumination conditions and EOF critical optics. Small-angled light beams can

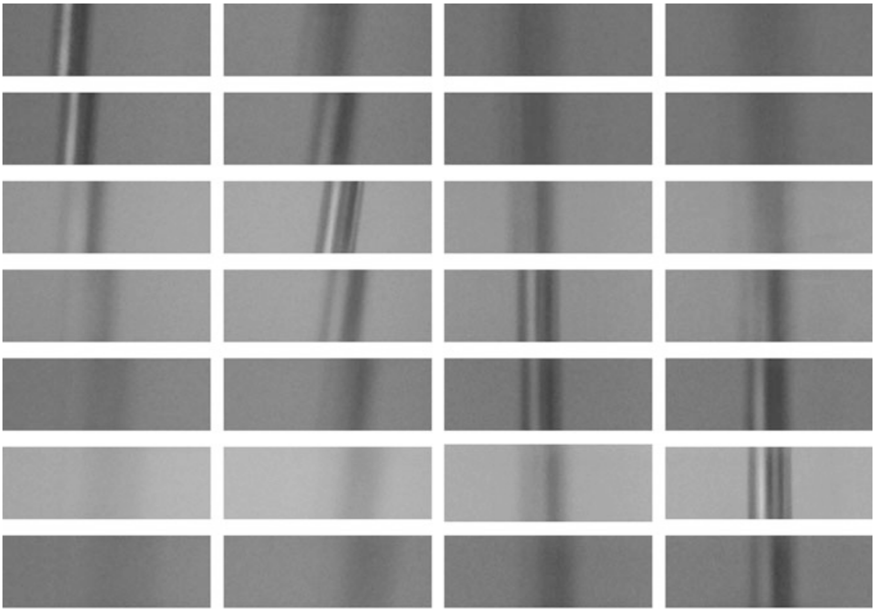


Fig. 19 ROIs from the four metallic pins (p1, p2, p3 and p4 represented on four column) for seven images (represented in seven lines) with focus distances varying 5 mm from each other

propagate with different large-angled trajectories through the multiple refractions (considering water-glass and glass-air consecutive cylindrical interfaces). This optical effect causes EOF fluctuations and consequent difficulty on EOF estimation based on DIF measures.

Figure 19 shows ROIs extracted from the seven images taken. The four pins are shown with changing focus distance.

Tables 5, 6, 7 and 8 shows the DCT, TN and correspondent PLFS inputs (Gc and Gct) and outputs (Out and Diagnosis) obtained values for the seven images. Each pin image was compared to find which was 'in focus'. In this part of the analysis, it is possible to observe that the four pins (corresponding to the four columns on Fig. 19) have a True (T) or Quasi-True (QT) Diagnosis output value for the best-focused pins. This result shows that PLFS could correctly diagnose the focus quality for these pins, although there were inconsistencies on TN and DCT measures.

On Table 5, it is possible to observe that DCT measures were grouped in only three values (1.62×10^3, 1.50×10^3 and 1.48×10^3) differently from TN values that have detected seven different levels of focus quality. However, TN values for *image* 6 shows a relative good evaluation although this pin image has the worst focus quality as can be seen on first column of Fig. 19.

On Table 6 the same behavior of DCT measure method. There is also only three different values for the seven images of second pin (second column of Fig. 19). TN

Table 5 DCT and TN focus measures for first pin (p1) and correspondent PLFS inputs and outputs

1st pin						
Image	DCT	TN	Dc	Duc	Out	Diagnosis
1	1.62E+03	3.19E+07	1.00	0.00	0.48	T
2	1.62E+03	3.25E+07	1.00	0.00	0.48	T
3	1.50E+03	2.27E+07	0.17	0.25	0.68	LTLI
4	1.50E+03	2.13E+07	0.03	0.06	0.76	ITII
5	1.50E+03	1.79E+07	−0.41	−0.34	−0.30	LFLP
6	1.47E+03	2.05E+07	−0.38	0.26	−0.68	LFLI
7	1.48E+03	1.74E+07	−0.65	−0.23	−0.38	QFP

Table 6 DCT and TN focus measures for second pin (p2) and correspondent PLFS inputs and outputs

2nd pin						
Image	DCT	TN	Dc	Duc	Out	Diagnosis
1	1.53E+03	2.23E+07	0.50	−0.17	0.32	LTLP
2	1.55E+03	2.44E+07	0.93	−0.07	0.48	T
3	1.53E+03	3.37E+07	0.78	0.22	0.56	QTI
4	1.50E+03	2.17E+07	0.07	0.13	0.76	ITII
5	1.50E+03	1.97E+07	−0.12	−0.18	−0.20	IFIP
6	1.46E+03	2.02E+07	−0.47	0.28	−0.68	LFLI
7	1.49E+03	1.97E+07	−0.25	−0.06	−0.28	LFLP

measure method has a similar problem to evaluate the poor focus of the *image* 6 of second pin, where the TN value was relatively high.

Table 7 shows that both focus measures techniques failed to evaluate correctly the focus quality gradual increase. TN measure (which was chosen to be the favorable evidence parameter) presented incoherent evaluations for four of the seven images of third pin. DCT measure has also presented incoherent results.

However, the PFLS was able to manage the different evaluations from both metrics, diagnosing the incoherence problems. On *image* 7 of third pin, DCT value is relatively high, indicating a very good focus quality, although this is not observed. PFLS-Diagnosis output detects some tendency to paracomplete condition.

Table 8 shows that TN evaluates the best-focused image (image 6) with the best focus value. However, DCT evaluations present disagreeing values for the same *image* 6. This incoherence is diagnosed as a True value with an inconsistency tendency (QTI—Quasi-True Tending to Inconsistent). This image can be viewed on Fig. 19 (Sect. 7.2), sixth line and fourth column.

Table 7 DCT and TN focus measures for third pin (p3) and correspondent PLFS inputs and outputs

3rd pin						
Image	DCT	TN	Dc	Duc	Out	Diagnosis
1	1.54E+03	1.99E+07	0.28	−0.54	0.10	QPT
2	1.50E+03	1.74E+07	−0.39	−0.48	−0.30	LFLP
3	1.47E+03	1.83E+07	−0.64	−0.02	−0.38	QFP
4	1.56E+03	2.85E+07	1.00	0.00	0.48	T
5	1.58E+03	2.61E+07	1.00	0.00	0.48	T
6	1.45E+03	1.76E+07	−0.89	0.06	−0.48	F
7	1.57E+03	2.32E+07	0.78	−0.22	0.38	QTP

Table 8 DCT and TN focus measures for fourth pin (p4) and correspondent PLFS inputs and outputs

4th pin						
Image	DCT	TN	Dc	Duc	Out	Diagnosis
1	1.52E+03	1.87E+07	−0.03	−0.52	−0.10	QPF
2	1.53E+03	1.80E+07	−0.08	−0.65	−0.10	QPF
3	1.46E+03	1.69E+07	−0.88	−0.12	−0.48	F
4	1.53E+03	2.38E+07	0.69	0.01	0.54	QTI
5	1.59E+03	2.84E+07	1.00	0.00	0.48	T
6	1.52E+03	3.61E+07	0.67	0.33	0.56	QTI
7	1.58E+03	2.52E+07	1.00	0.00	0.48	T

8 Conclusions

The example presented on this chapter shows that PFLS can be useful to manage inconsistencies of different focus metrics to evaluate scientific cylindrical refraction experiments. This system provides reliable and objective method to compare focus metrics for each part of the image, allowing the detection of these inconsistencies.

The critical refraction distortions registered on captured images may lead to incoherent focus metrics results. PFLS makes possible the use of different pairs of focus metrics to check inconsistencies, and diagnose conflicting results. The two chosen metrics to be compared, in the current example, have been used in many different applications on Digital Image Analysis field. However, any other pair of metrics could have been chosen depending on the experimental conditions.

Digital Image Analysis has a wide range of possible applications for Paraconsistent Logic use, especially on monitoring and diagnosis of defects, diseases and artifacts detection.

Focus determination is a developing field within Digital Image Analysis, who presents many different modalities. Recent multiphasic flow studies have been based on visualization techniques and pattern recognition tasks. Both depend on a correct focus evaluation.

Therefore, this chapter demonstrates one important application of Annotated Logic to directly deal with experimental inconsistencies on scientific digital-image-focus measures.

Acknowledgments Authors would like to thank the support of *Fundação de Inovação e Pesquisa* (*FINEP*), project 01.10.0248.00, with the *Comissão Nacional de Energia Nuclear* (*CNEN*).

References

1. Ali, S.F., Yeung, H.: Experimental investigation and numerical simulation of two-phase flow in a large-diameter horizontal flow line vertical riser. Petrol. Sci. Technol. 37–41 (2010)
2. Serizawa, A., Feng, Z., Kawara, Z.: Two-phase flow in microchannels. Exp. Therm. Fluid Sci. **26**, 703–714 (2002)
3. Arcanjo, A.A., Tibiriçá, C.B., Ribatski, G.: Evaluation of flow patterns and elongated bubble characteristics during the flow boiling of halocarbon refrigerants in a micro-scale channel. Exp. Therm. Fluid Sci. **34**, 766–775 (2010)
4. Liu, W.-C., Yang, C.-Y.: Two-phase flow visualization and heat transfer performance of convective boiling in micro heat exchangers. Exp. Therm. Fluid Sci. **57**, 358–364 (2014)
5. De Mesquita, R.N., Masotti, P.H.F., Penha, R.M.L., Andrade, D.A., Sabundjian, G., Torres, W.M., Macedo, L.A.: Classification of natural circulation two-phase flow patterns using fuzzy inference on image analysis. Nucl. Eng. Des. **250**, 592–599 (2012)
6. Zboray, R., Adams, R., Cortesi, M., Prasser, H.-M.: Development of a fast neutron imaging system for investigating two-phase flows in nuclear thermal–hydraulic phenomena: a status report. Nucl. Eng. Des. **273**, 10–23 (2014)
7. Mantle, M.D., Sederman, A.J., Gladden, L.F.: Single- and two-phase flow in fixed-bed reactors : MRI flow visualisation and lattice-Boltzmann simulations. Chem. Eng. Sci. **56**, 523–529 (2001)
8. Thome, J.R., Hajal, J.El: Two-phase flow pattern map for evaporation in horizontal tubes: latest version. Heat Transf. Eng. **24**, 3–10 (2003)
9. Eskicioglu, A.M., Fisher, P.S.: Image quality measures and their performance. Commun. IEEE Trans. **43**, 2959–2965 (1995)
10. Kautsky, J., Flusser, J., Zitová, B., Šimberová, S.: A new wavelet-based measure of image focus. Pattern Recogn. Lett. **23**, 1785–1794 (2002)
11. Dash, R., Majhi, B.: Motion blur parameters estimation for image restoration. Opt. Int. J. Light Electron Opt. **125**, 1634–1640 (2014)
12. Kumar, V., Gupta, P.: Importance of statistical measures in digital image processing. IJATAE **2**, 56–62 (2012)
13. Ferzli, R., Karam, L.J.: A no-reference objective image sharpness metric based on the notion of Just Noticeable Blur (JNB). IEEE Trans. Image Process. **18**, 717–728 (2009)
14. Huang, W., Jing, Z.: Evaluation of focus measures in multi-focus image fusion. Pattern Recogn. Lett. **28**, 493–500 (2007)
15. Caviedes, J., Oberti, F.: A new sharpness metric based on local kurtosis, edge and energy information. Signal Process. Image Commun. **19**, 147–161 (2004)
16. Williams, D., Burns, P.: Measuring and managing digital image sharpening. In: Proceedings of IS&T 2008 Archiving Conference, pp. 89–93. Bern, Switzerland (2008)
17. Gruev, V., Perkins, R., York, T.: CCD polarization imaging sensor with aluminum nanowire optical filters. Opt. Express **18**, 19087–19094 (2010)

18. Fossum, E.R., Member, S.: CMOS image sensors : electronic camera-on-a-chip. IEEE Trans. Electron Devices **44**, 1689–1698 (1997)
19. Riutort-Mayol, G., Marqués-Mateu, A., Seguí, A.E., Lerma, J.L.: Grey level and noise evaluation of a Foveon X3 image sensor: a statistical and experimental approach. Sensors (Basel) **12**, 10339–10368 (2012)
20. Ray, S.F.: Scientific Photography and Applied Imaging. Focal Press, Oxford (1999)
21. Eskicioglu, A.M., Fisher, P.S.: A survey of quality measures for gray scale image compression. In: Proceedings of 1993 Space and Earth Science Data Compression Workshop, pp. 49–61. NASA (1993)
22. Tapiovaara, M.J.: Review of relationships between physical measurements and user evaluation of image quality. Radiat. Prot. Dosimetry. **129**, 244–248 (2008)
23. Moreno, P., Calderero, F.: Evaluation of sharpness measures and proposal of a stop criterion for reverse diffusion in the context of image deblurring. In: 8th International Conference on Computer Vision Theory and Applications (2013)
24. Krotkov, E.: Focusing. Int. J. Comput. Vis. **237**, 223–237 (1987)
25. Goldsmith, N.T.: Deep focus; a digital image processing technique to produce improved focal depth in light microscopy. Image Anal. Stereol. **19**, 163–167 (2000)
26. Olsen, M.G., Adrian, R.J.: Out-of-focus effects on particle image visibility and correlation in microscopic particle image velocimetry. Exp. Fluids **29**, s166–s174 (2000)
27. Malik, A, Choi, T.: A novel algorithm for estimation of depth map using image focus for 3D shape recovery in the presence of noise. Pattern Recogn. **41**, 2200–2225 (2008)
28. Krotkov, E., Martin, J.-P.: Range from focus. In: Proceedings of the 1986 IEEE International Conference on Robotics and Automation, vol. 3 (1986)
29. Eltoukhy, H.A., Kavusi, S.: A computationally efficient algorithm for multi-focus image reconstruction. Proc. SPIE Electron. Imag. **5017**, 332–341 (2003)
30. Groen, F.C.A., Young, I.T., Ligthart, G.: A comparison of different focus functions for use in autofocus algorithms. Cytometry **6**, 81–91 (1985)
31. Huang, W., Jing, Z.: Multi-focus image fusion using pulse coupled neural network. Pattern Recogn. Lett. **28**, 1123–1132 (2007)
32. Li, S., Kwok, J.T., Wang, Y.: Multifocus image fusion using artificial neural networks. Pattern Recogn. Lett. **23**, 985–997 (2002)
33. Wang, Z., Ma, Y., Gu, J.: Multi-focus image fusion using PCNN. Pattern Recogn. **43**, 2003–2016 (2010)
34. Mast, T.D., Nachman, A.I., Waag, R.C.: Focusing and imaging using eigenfunctions of the scattering operator. J. Acoust. Soc. Am. **102**, 715–725 (1997)
35. Ray, S.F.: Applied Photographic Optics. Focal Press, Oxford (1994)
36. Li, S., Kwok, J.T., Zhu, H., Wang, Y.: Texture classification using the support vector machines. Pattern Recogn. **36**, 2883–2893 (2003)
37. Jähne, B., Haussecker, H., Geissler, P. ed: Handbook of Computer Vision and Applications. Academic Press, New York (1999)
38. Schlag, J., Sanderson, A., Neuman, C., Wimberly, F.: Implementation of automatic focusing algorithms for a computer vision system with camera control. Carnegie Mellon University (1983)
39. Gonzalez, R.C., Woods, R.E., Eddins, S.L.: Digital Image Processing Using Matlab—Gonzalez Woods & Eddins.pdf. Prentice Hall, Upper Saddle River (2004)
40. Baina, J., Dublet, J.: Automatic focus and iris control for video cameras. In: Fifth International Conference on Image Processing and Its Applications, pp. 232–235 (1995)
41. Shen, C., Chen, H.H.: Robust focus measure for low-contrast images. In: Proceedings of IEEE International Conference on Consumer Electronics, Digest Technical Papers, pp. 69–70 (2006)
42. Costa, N.C.A., Abe, J.M., Carlos Murolo, A., Filho, J.I.D.S., Fernando S. Leite, C.: Lógica Paraconsistente Aplicada. Editora Atlas S.A., São Paulo (1999)
43. Da Silva Filho, J.I., Abe, J.M.: Para-fuzzy logic controller II—a hybrid logical controller indicated for treatment of fuzziness and inconsistencies. In: Proceedings of the international

ICSC, congress on computational intelligence methods and applications, CIMA 99, Rochester, NY, USA (1999)
44. Da Silva Filho, J.I., Abe, J.M.: Para-fuzzy Logic Controller I—A Hybrid Logical Controller Indicated for Treatment of Inconsistencies Designed with Junction of the Paraconsistent and Fuzzy Logic. Proceedings fo the International ICSC, Congress on Computational Intelligence Methods and Applications, CIMA 99., Rochester, NY, USA (1999)
45. Da Silva Filho, J.I., Abe, J.M.: Fundamentos das Redes Neurais Artificiais Paraconsistentes. Editora Arte & Ciência, São Paulo (2001)
46. Da Silva Filho, J.I.: Métodos de Aplicações da Lógica Paraconsistente Anotada de Anotação com Dois Valores-LPA2v com Construção de Algoritmo e Implementação de Circuitos Eletrônicos, http://paralogike.com.br/site/links/ver/32 (1999)
47. Da Silva Filho, J.I.: Treatment of uncertainties with algorithms of the paraconsistent annotated logic. J. Intell. Learn. Syst. Appl. **04**, 144–153 (2012)
48. Da Silva Filho, J.I.: Métodos de Aplicações da Lógica Paraconsistente Anotada de anotação com dois valores-LPA2v. Rev. Seleção Doc. **1**, 18–25 (2006)
49. Da Silva Filho, J.I.: Paraconsistent differential calculus (Part I): first-order paraconsistent derivative. Appl. Math. **5**, 904–916 (2014)
50. Da Silva Filho, J.I.: Lógica Para Fuzzy – Um método de Aplicação da Lógica Paraconsistente e Fuzzy em Sistemas de Controle Híbridos. Rev. Seleção Doc. **2**, 16–24 (2009)
51. Abe, J.M.: Introdução à Lógica Paraconsistente Anotada. Paralogike, Editora (2006)
52. Abe, J.M., Lopes, H.F.S., Nakamatsu, K.: Paraconsistent artificial neural networks and. **17**, 99–111 (2013)
53. Abe, J.M.: Remarks on paraconsistent annotated evidential logic E τ. Unisanta Sci. Technol. **3**, 25–29 (2014). http://periodicos.unisanta.br/index.php/sat
54. Costa, N.C.A.: On the theory of inconsistent formal systems. Notre Dame J. Form. Log. **XV**, 497–510 (1974)
55. Masotti, P.H.F.: Metodologia de Monitoração e Diagnóstico Automatizado de Rolamentos utilizando Lógica Parconsistente, Transformada de Wavelet e Processamento de Sinais Digitais, http://www.teses.usp.br/teses/disponiveis/85/85133/tde-28052007-165556/pt-br.php (2006)
56. Mathworks: Matlab version 8.3.0.532, (2014)
57. Nayak, A.K., Sinha, R.K.: Role of passive systems in advanced reactors. Prog. Nucl. Energy **49**, 486–498 (2007)
58. Hervieu, E., Seleghim, P.: An objective indicator for two-phase flow pattern transition, (1998)
59. Mesquita, R.N., Sabundjian, G., Andrade, D.A., Umbehaun, P.E., Torres, W.M., Conti, T.N., Macedo, L.A.: Two-phase flow patterns recognition and parameters estimation through natural circulation test loop image analysis. In: ECI International Conference on Boiling Heat Transfer, pp. 3–7 (2009)

Paraconsistent Artificial Neural Networks and Aspects of Pattern Recognition

Sheila Souza and Jair Minoro Abe

Abstract In this chapter we discuss about the use of Paraconsistent Artificial Neural Network on computer pattern recognition. Computer pattern recognition is one of the most important Artificial Intelligence tools present in numerous knowledge areas with applications in several themes, including character recognition. Our focus is the investigation of an automated computational process able to recognize numeric characters furnishing a technical basis to recognize digital images and documents. The methodology employed for the task is based on Paraconsistent Artificial Neural Networks for being a tool with the ability to work with imprecise, inconsistent and paracomplete data without trivialization.

Keywords Paraconsistent annotated logic · Paraconsistent aritificial neural networks · Pattern recognition · Character recognition · Handwritten character recognition · Magnetic ink character recognition

1 Introduction

The aim of this chapter is to present the investigation of an automated computational process—Paraconsistent Recognition Process (PRP)—able to recognize Magnetic Ink Character Recognition (MICR) used on Brazilian bank checks (Fig. 1) and handwritten numeric characters furnishing a technical basis to recognize different kinds of signals. Although there are several studies on character

S. Souza (✉)
PRODESP—Data Processing Company of São Paulo State, Taboão da Serra, Brazil
e-mail: sheinara@gmail.com

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

J.M. Abe (ed.), *Paraconsistent Intelligent-Based Systems*,
Intelligent Systems Reference Library 94, DOI 10.1007/978-3-319-19722-7_9

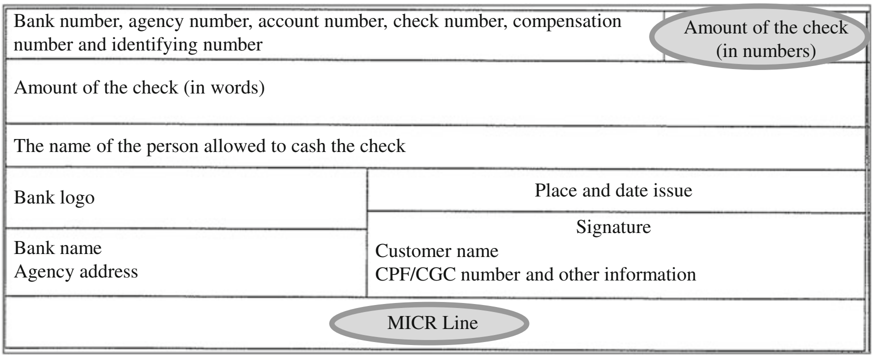


Fig. 1 Brazilian bank check example

recognition, we have chosen this theme due to its intrinsic importance and constant improvement.

The PRP process [1] is performed from some previously selected character features based on some Graphics techniques and, the analysis of these features as well as the character recognition are performed through the Paraconsistent Artificial Neural Networks (PANN).

PANN [2], in turn, are based on Paraconsistent Annotated Evidential Logic Eτ [3, 4] which is able to manipulate some of the most challenging concepts in pattern recognition such as imprecision, inconsistency and paracompleteness. Nowadays there are numerous research works in Artificial Intelligence area based on PANNs such as [5–8], and others.

Taking into consideration that the character recognition technique classifies characters from their features, the major difficulty for pattern recognition systems concentrates on determining the feature set capable of being extracted due to interferences from errors and noise. Thus, the higher quality the original document/image has and the more able to deal with imprecise, conflicting and paracomplete data the system is, the better the system will perform. Hence the use of PANN tool is important and relevant to applications that involve data with such features.

2 PRP Process

PRP process is a computational technique based on Paraconsistent Annotated Evidential Logic Eτ [3] to recognize characters from digital images. The process should read a digital image, extract its features, recognize them and, finally, identify the standard characters that represent the recognized and discarded characters.

As the recognition process is performed by comparing a digital image and a standard character image set, PRP needs an image database composed by standard character images and digital image samples. Here we will need a digital image set

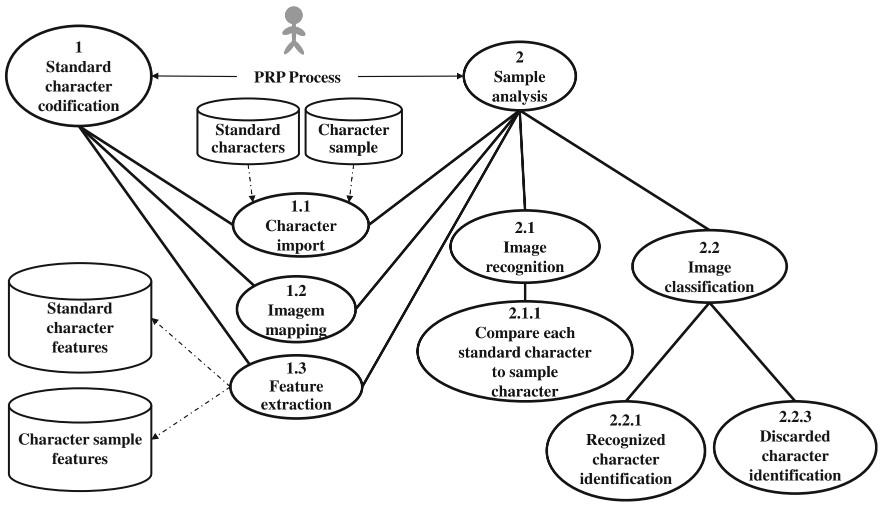


Fig. 2 PRP process macro vision

for each type of character that wishes to acknowledge. Let's assume that all database image was previously preprocessed, binarized and segmented.

The process also needs a feature database to store the standard character features and sample features.

Initially, according to Fig. 2, PRP imports the standard characters from its database, extracts the feature of each character and then stores them into standard character feature database. This configuration must be done prior to any character recognition. After that, PRP is able to start the recognition process of any character type previously configured.

Figure 2 presents the PRP process macro vision.

The recognition process is divided into four steps according to Fig. 3: Image mapping; Feature extraction; Image recognition and; Image classification. Each of them will be detailed on the next section.

According to Fig. 2, the two first steps are the same for standard character codification and sample analysis. However, for the sample analysis we have more two steps to compare the features between standard and sample character and identify the standard characters which represent the recognized and discarded characters.

3 PRP Recognition Process

The Fig. 3 presents the four steps of the PRP recognition process. The two first features "Image mapping" and "Feature extraction" are based on Graphoscopy and Graphology techniques, which are important techniques used by criminal experts to

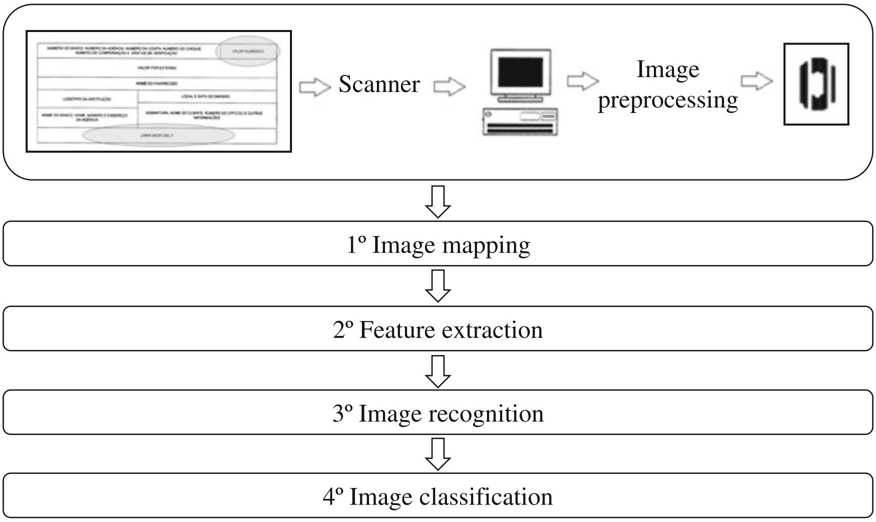


Fig. 3 PRP recognition process steps

create handwritten signature analysis reports. The Graphoscopy was developed to clarify criminal issues and it is responsible for verifying document authenticity through its graphical features while Graphology distinguishes one author from another through the features that determine the author's psychological profile.

The first step (Image mapping) is responsible for mapping a binary image in evidence degrees. Let's assume that each evidence degree is a value belonging to the real interval [0, 1], and that 0 is a white pixel and 1 is a black pixel.

The second step (Feature extraction) extracts some interest features from the image based on the image mappings obtained in the first step.

The third step (Image recognition) analyzes the extracted features comparing each standard character features to sample character features and calculates a single recognition evidence degree for each standard character through the use of ParaExtr_{ctr} algorithm [9] (detailed on Sect. 3.3).

The fourth and last step (Image classification) identifies the recognized and discarded characters.

3.1 Image Mapping

The image mapping step performs several mapping types based on the binary image matrix. This process analyzes the matrix pixels in different ways and calculates an evidence degree for the pixels. The quantity of mapping types depends on each type of character that wishes to acknowledge.

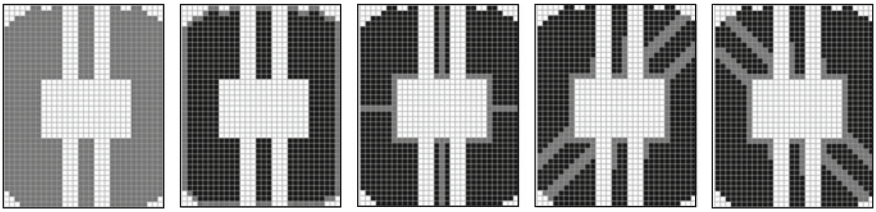


Fig. 4 PRP image mapping types

On this chapter we present the following mapping types:

- External border;
- Internal border;
- Pixels of the image;
- Diagonal internal border (right to left);
- Diagonal internal border (left to right);
- Image color considering all the pixels of the image and;
- Image color by column.

Figure 4 presents the five first mapping types where the gray pixels represent the conducted mapping.

Considering that the evidence degree must be a real value within the interval [0, 1], we assume that 0 is a white pixel and 1 is a black pixel.

External Border. We assign an evidence degree for each black pixel found on the image external border.

Internal Border. We assign an evidence degree for each black pixel found on the image internal border.

Both mapping types are divided into four parts:

1. Left side;
2. Right side;
3. Upper side and;
4. Lower side.

On the "External border" mapping, we use the evidence degree normalization among the image columns as a base to map the left and right side, in other words, we use the Eq. 1 to calculate the evidence degree for each black pixel found on the left and right side of the image external border.

$$\text{evidence degree} = \frac{[\text{column of the black pixel found}]}{[\text{total of columns}]} \tag{1}$$

For the upper and lower side we change the base, that is, according to Eq. 2 we use the evidence degree normalization among the image lines to calculate the evidence degree for each black pixel found on the upper and lower side of the image external border.

$$\text{evidence degree} = \frac{[\text{line of the black pixel found}]}{[\text{total of lines}]} \tag{2}$$

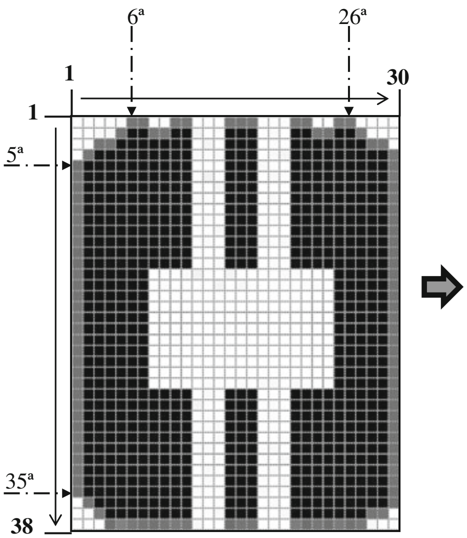
For each image side the calculated evidence degrees are organized into a list. So, this mapping is represented by four evidence degree lists. Figure 5 presents an example to calculate the evidence degree for each image side during the “External border” mapping.

On the “Internal border” mapping, the same as in the “External border” mapping, we use the same equations to calculate the evidence degree for each black pixel found, that is, we use the Eq. 1 for left and right side and Eq. 2 for upper and lower side.

Figure 6 presents an example to calculate the evidence degree for each image side during the “Internal border” mapping.

Similarly, the evidence degrees obtained for each image side are organized into a list. So, this mapping is also represented by four evidence degree lists.

Pixels of the Image. We assign 0 as a white pixel, and for each black pixel found on each image line we use the Eq. 1 to calculate the evidence degree based on the image columns. Figure 7 presents an example to calculate the evidence degree for each image pixel during this image mapping type. The calculated evidence degrees are organized into a single list following the column sequence to represent this mapping.



E.g. Left side
Column of the black pixel found = 6
Total of columns = 30
Evidence degree = 6 / 30 = 0.2

E.g. Right side
Column of the black pixel found = 26
Total of columns = 30
Evidence degree = 26 / 30 = 0.87

E.g. Upper side
Line of the black pixel found = 5
Total of lines = 38
Evidence degree = 5 / 38 = 0.13

E.g. Lower side
Line of the black pixel found = 35
Total of lines = 38
Evidence degree = 35 / 38 = 0.92

Fig. 5 External border mapping example

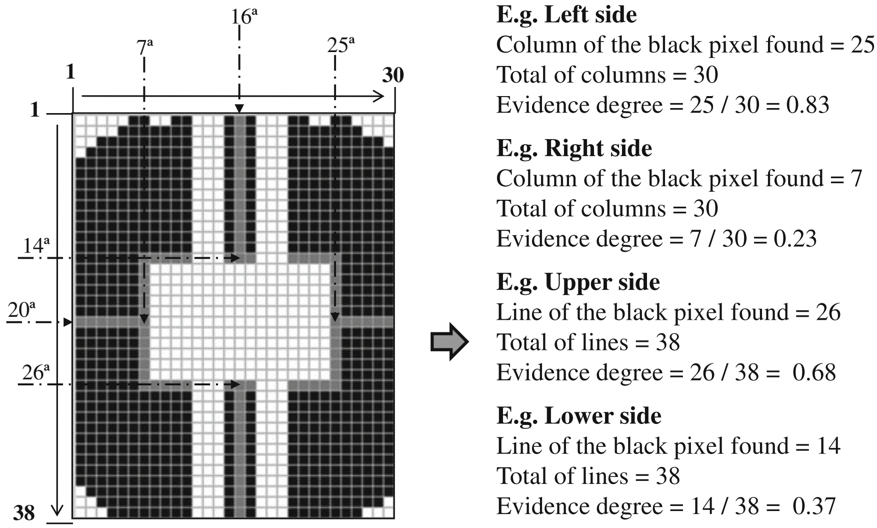


Fig. 6 Internal border mapping example

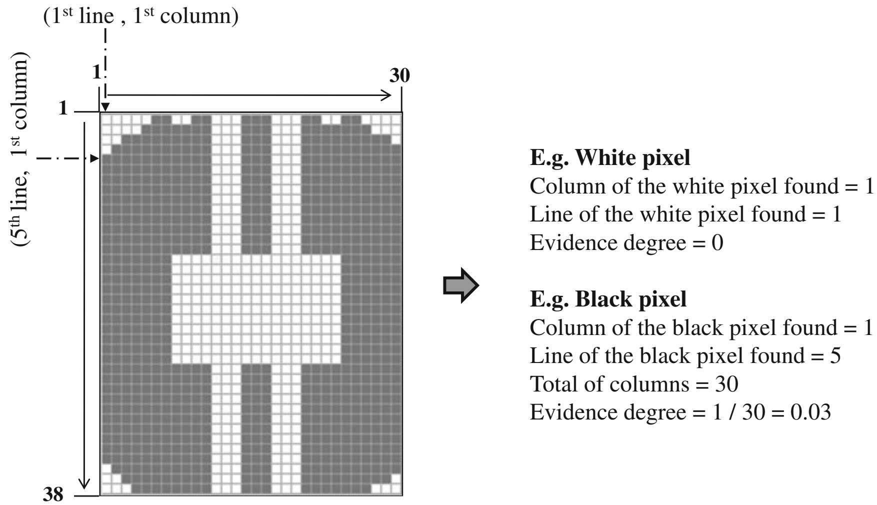


Fig. 7 "Pixels of the image" mapping example

Diagonal Internal Border (Right to Left). We assign an evidence degree for each black pixel found on the inner diagonal border considering two imaginary diagonal lines named "superior" and "inferior" presented on Fig. 8. Here the mapping process is divided into two parts:

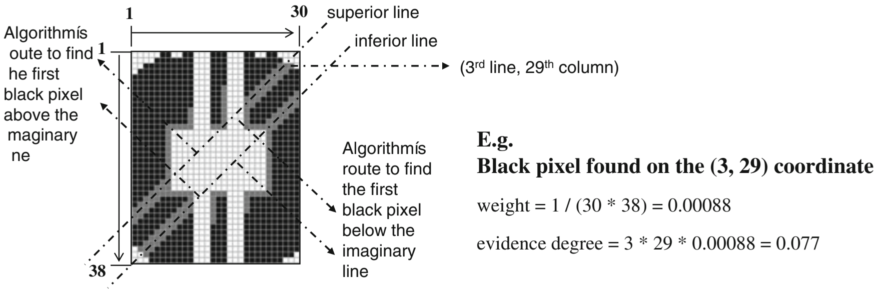


Fig. 8 Diagonal internal border (*right* to *left*) mapping example

1. Upper and lower border mapping from the "superior" imaginary line and;
2. Upper and lower border mapping from the "inferior" imaginary line;

The gray pixels on Fig. 8 represent the mappings above mentioned. Both use the same algorithm to find the black pixels on the upper and lower border from the respective imaginary line. The algorithm runs along the imaginary line looking for the first black pixel in every diagonal line and calculates the evidence degree taking into consideration the (x, y) pixel coordinates and a "weight" (a value calculated based on the image pixels total according to Eq. 3). Then, each evidence degree is calculated using the Eq. 4.

$$weight = \frac{1}{\text{image pixels total amount}} \tag{3}$$

$$evidence\,degree = x\,coordinate * y\,coordinate * weight \tag{4}$$

For each imaginary diagonal line the calculated evidence degrees are organized into two lists, one to upper border and other to lower border. So, this mapping is represented by four evidence degree lists.

Diagonal Internal Border (Left to Right). Similar to "Diagonal internal border (right to left)" we assign an evidence degree for each black pixel found on the inner diagonal border considering two imaginary diagonal lines named "superior" and "inferior" and we use the same equations (Eqs. 3 and 4) to calculate de evidence degree for each black pixel found on the inner diagonal border of the imaginary lines. And, for each imaginary diagonal line the calculated evidence degrees are also organized into two lists, the same as in "Diagonal internal border (right to left)" mapping, one list to upper border and other to lower border. So, this mapping is represented by four evidence degree lists too.

Figure 9 presents the imaginary lines, the gray pixels that represent the two parts of the mapping and an example.

Image Color Mapping. We assign two evidence degrees, one to the percentage of black pixels and another to the percentage of white pixels, in other words, if the image has 66 % black pixels and 34 % white pixels then the corresponding

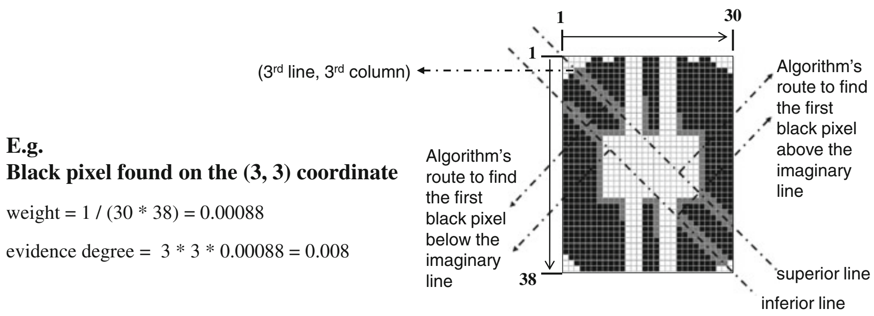


Fig. 9 Diagonal internal border (*left* to *right*) mapping example

evidence degrees will be 0.66 and 0.34 respectively. This evidence degrees are organized into a single list composed first by the percentage of the black pixels and then the percentage of the white pixels.

Image Color by Column. We assign one evidence degree for the percentage of black pixels in each image column. These values are organized into a single list where 1 is 100 % and 0 is 0 %.

E.g.: The evidence degrees calculated for the Fig. 8 are: [0.82, 0.87, 0.92, 0.95, 0.97, 1.00, 1.00, 0.68, 0.68, 0.71, 0.71, 0.00, 0.00, 0.00, 0.71, 0.71, 0.71, 0.00, 0.00, 0.00, 0.71, 0.71, 0.68, 0.68, 1.00, 1.00, 0.97, 0.92, 0.92, 0.87].

At the end of this step we have evidence degree lists which represent the mapping types and these lists will be the base to the next step.

3.2 Feature Extraction

Feature extraction step is responsible for grouping some character qualities and turn them into evidence degree datasets, so that they can be transmitted and received by the Image recognition step, that for, it's an abstraction that characterizes the character and, in general, distinguishes one character from another.

Taking into consideration that the recognition process is performed from some character qualities, let's define some features based on the study of some Graphoscopy and Graphology techniques [10, 11] such as "curvilinear values", "height", "width", "direction" and "regularity".

Let's define the following features to compose the feature extraction process:

- Vertical line segments;
- Horizontal line segments;
- Histogram: All pixels of the image;
- Histogram: External upper border;
- Histogram: External border;
- Histogram: Internal Border;

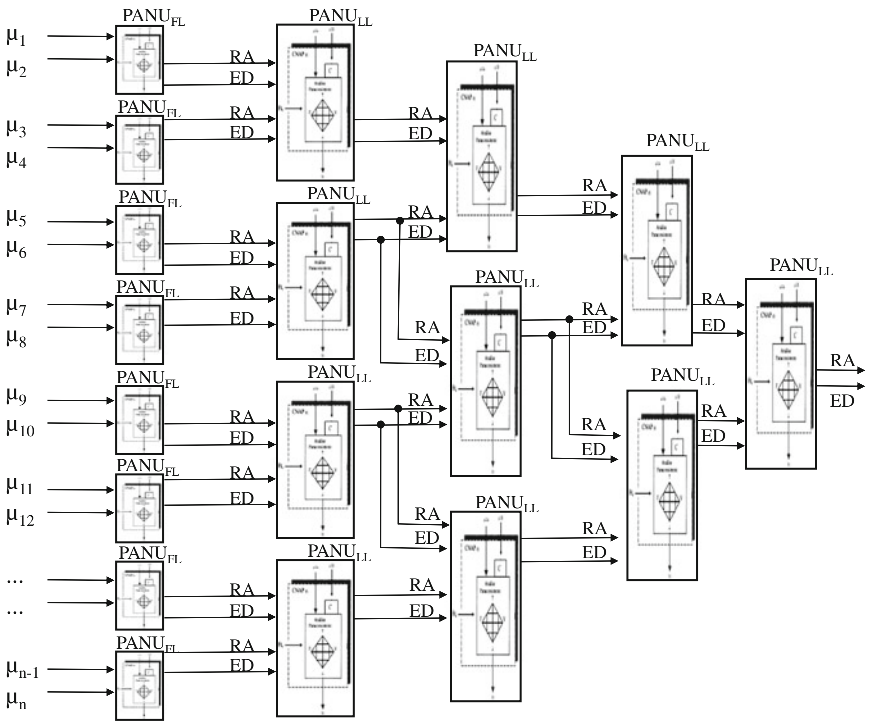
- Histogram: Diagonal internal border (right to left);
- Histogram: Diagonal internal border (left to right);
- Histogram: Image color;
- Histogram: Image color by column.

Vertical Line Segments and Horizontal Line Segments. The first two features analyze the character based on the vertical and horizontal line segments found on the image external border.

For Vertical line segments we create two evidence degree lists to represent the extracted features, one for the image right side and the other for the image left side.

For Horizontal line segments we create two evidence degree lists to represent the extracted features, one for the image upper side and the other for the image lower side.

Both features uses the PANN presented on Fig. 10 to detect the vertical and horizontal line segments of the character on the image where μ is an evidence



μ_n: evidence degree from a specific side external border mapping
$PANU_{FL}$: Paraconsistent Artificial Neural Unit of the First Layers
$PANU_{LL}$: Paraconsistent Artificial Neural Unit of the Last Layers
RA: Real Analytical Paraconsistent Artificial Neural Cell's output – RaPANC's output
ED: Analytical Paraconsistent Artificial Neural Cell of Equality Detection's output – $PANC_{ED}$'s output

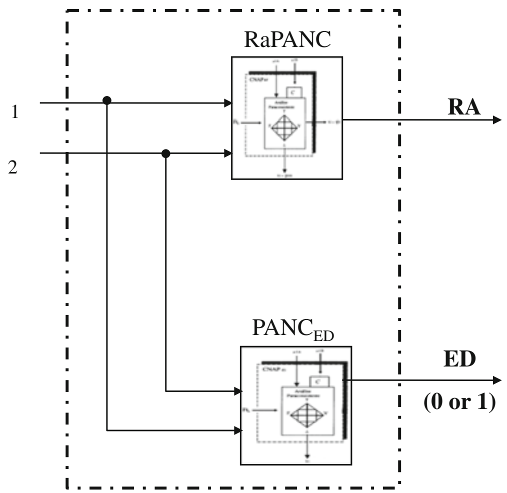
Fig. 10 VHLN architecture

degree, $PANU_{FL}$ is a Paraconsistent Artificial Neural Unit (PANU) [2] named PANU of the Fist Layer, $PANU_{LL}$ is also a PANU named PANU of the Last Layer, RA is an evidence degree calculated from a Real Analytical Paraconsistent Artificial Neural Cell (RaPANC) [2] and ED is also an evidence degree calculated from a PANC of Equality Detection ($PANC_{ED}$) [2] which is 0 or 1. Let's named this PANN as Vertical and Horizontal Line segment Network (VHLN).

VHLN takes the evidence degree list from a specific side external border mapping as input—e.g. right side, identifies the line segments on its character side and creates a line segment list as output. Each line segment is composed by an evidence degree set.

On the first layer, each PAN_{FL} is composed by two PANC types, Real Analytical PANC (RaPANC) and PANC of Equality Detection ($PANC_{ED}$) according to Fig. 11. RaPANC (Real analytical PANC) calculates and returns a real evidence degree which decreases the input inconsistency while $PANC_{ED}$ checks whether the inputs are equal and returns the values 0 or 1 [2]. $PANU_{FL}$'s output is composed by two values, RA and ED where RA represents the RaPANC's output and ED the $PANC_{ED}$'s output.

On VHLN next layers, each $PANU_{LL}$ is composed by eight PANCs according to Fig. 12:



1: 1st evidence degree
2: 2nd evidence degree
RaPANC: Real analytical Paraconsistent Artificial Neural Cell
$PANC_{ED}$: Paraconsistent Artificial Neural Cell of Equality Detection
RA: RaPANC's output
DE: $PANC_{ED}$'s output

Fig. 11 $PANU_{FF}$ architecture

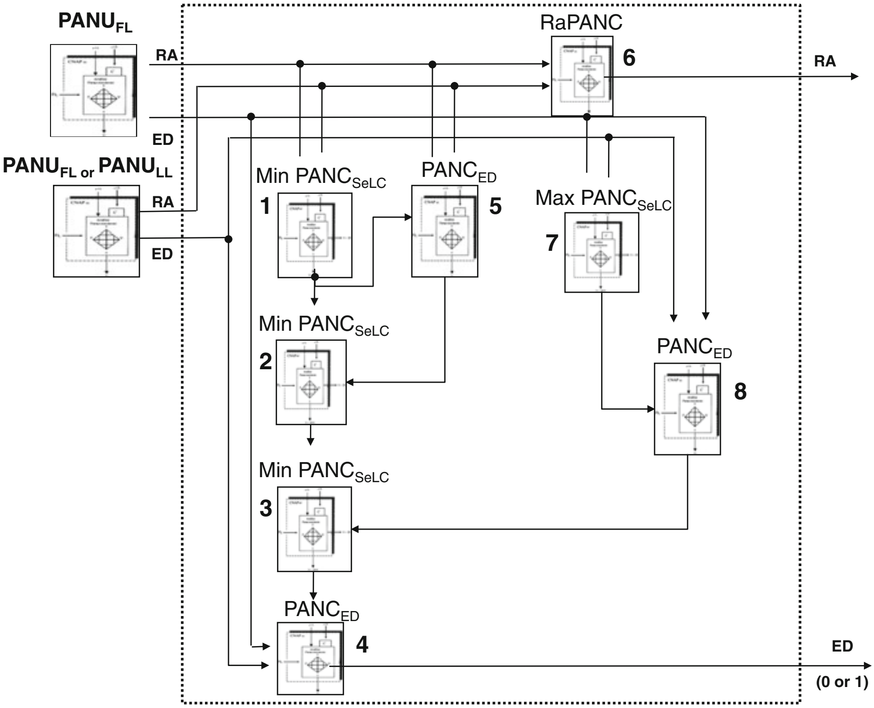


Fig. 12 $PANU_{LL}$ architecture

- Three minimization PANC of Selective Logical Connection—$PANC_{SeLC}$;
- One maximization $PANC_{SeLC;}$
- Three $PANC_{ED}$ and;
- One RaPANC.

There are two types of $PANC_{SeLC}$ (maximization and minimization). They receive two values as input and return the maximum or minimum value according to the $PANC_{SeLC}$ type [2].

For Vertical line segments extraction we use two VHLN, one to identify right vertical line segments and another to identify left vertical line segments. Each identified line segment is a list composed by four evidence degrees organized into the following sequence which represent:

- Direction (1 = left side and 0 = right side);
- Minimum line segment length;
- Column position where the line segment is;
- Line position where the line segment is.

For Horizontal line segments extraction we use two VHLN, one to identify upper horizontal line segments and another to identify lower horizontal line segments. Each identified line segmented is also a list composed by four evidence degrees representing:

- Direction (1 = upper side and 0 = lower side)
- Minimum line segment length;
- Line position where the line segment is;
- Column position where the line segment is.

Figure 13 presents an example of the left vertical line segment extraction. Line segment identification depends on the ED output analysis of each PANU on the VHLN from the last layer. There are three situations featuring an identified line segment:

1. The PANU of the last layer has ED output = 1;
2. The PANU has ED output = 1 and both PANU neighbors has ED output = 0 or;
3. The PANU is the first or the last layer element, and it has ED output = 1 and its only neighbor has also ED output = 0.

Minimum line segment length is a value in a real interval [0, 1] calculated by Eq. 5 if the layer index is 0 or Eq. 6 if the layer index is not 0.

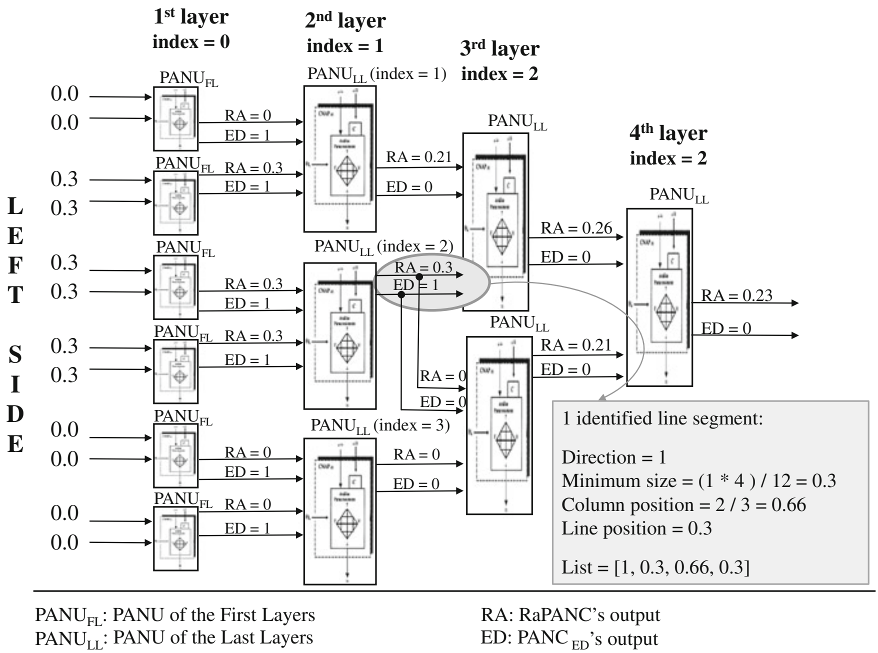


Fig. 13 Example of the *left vertical line* segment extraction

$$(\text{layer index} * 4) \ / \ \text{amount of evidence degrees on the input list} \tag{5}$$

$$(\text{layer index} * 2) \ / \ \text{amount of evidence degrees on the input list} \tag{6}$$

Column position where the line segment is, also represents a value in a real interval [0, 1] calculated by Eq. 7, where UIL is PANU index in the layer, AUL is the amount of PANUs in the layer

$$\text{Column position where the line segment is} = \text{UIL} \ / \ \text{AUL} \tag{7}$$

Line position where the line segment is, also represents a value in a real interval [0, 1] obtained by RA output of the PANU that represents the identified line segment, for example, in Fig. 13 the evidence degree which represents this line position is 0.3.

At the end of these feature extractions, the line segments are organized into four lists:

- Vertical line segments of the left side;
- Vertical line segments of the right side;
- Horizontal line segments of the upper side and;
- Horizontal line segments of the lower side.

Histogram: All Pixels of the Image. The process creates a histogram using the list obtained in "Pixels of the image" mapping to represent this feature.

Histogram: External Upper Border. The process creates a histogram using the upper side list obtained in the "External border" mapping.

Histogram: External Border. The process creates a histogram composed by the evidence degrees obtained in the "External border" mapping obeying the following sequence: left side, lower side, upper side and right side.

Histogram: Internal Border. The process creates a histogram composed by the evidence degrees obtained in the "Internal border" mapping obeying the following sequence: left side, lower side, upper side and right side.

Histogram: Diagonal Internal Border (Right to Left). The process creates a histogram composed by the evidence degrees obtained in the "Diagonal internal border (right to left)" mapping obeying the following sequence: upper border of the superior imaginary line, lower border of the superior imaginary line, upper border of the inferior imaginary line and lower border of the inferior imaginary line.

Histogram: Diagonal Internal Border (Left to Right). The process creates a histogram composed by the evidence degrees obtained in the "Diagonal internal border (left to right)" mapping obeying the following sequence: upper border of the superior imaginary line, lower border of the superior imaginary line, upper border of the inferior imaginary line and lower border of the inferior imaginary line.

Histogram: Image Color. The process creates a histogram using the list obtained in the "Image color" mapping.

Histogram: Image Color by Column. The process creates a histogram using the same list obtained in the "Image color by column" mapping.

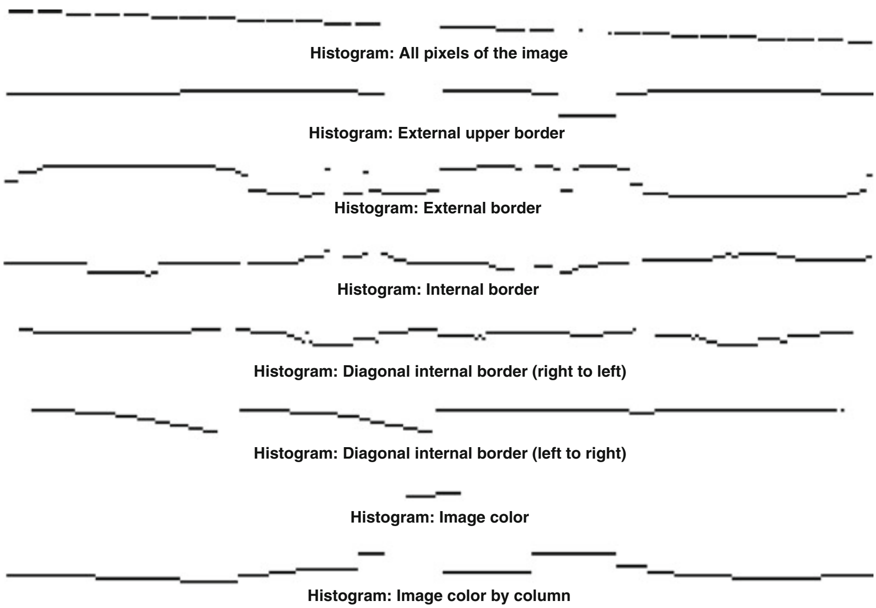


Fig. 14 Histograms created in PRP Feature extraction

Figure 14 presents an example of each created histogram for the last eight feature extraction types.

At the end of this step, we have evidence degree lists to represent each extracted feature and then compose the input of the Image recognition step.

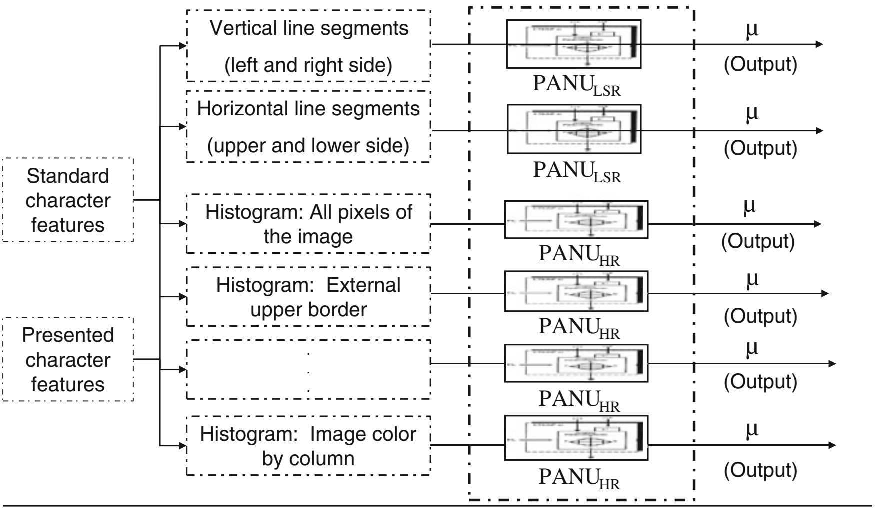
3.3 Image Recognition

Image recognition step performs a feature analysis between the features obtained into the Feature extraction step and the features of each standard character.

For each standard character, the analysis performs the following procedures: (1) compares the features between presented and standard character features; (2) calculates a single recognition evidence degree for each feature and then; (3) calculates a single recognition evidence degree for the standard character.

The first procedure is performed according to the Fig. 15. Here the process performs a feature recognition for each standard character using a PANU of Line segment recognition ($PANU_{LSR}$) to recognize each line segment feature and a PANU of Histogram recognition ($PANU_{HR}$) to recognize each histogram feature. The PANU's output represents the recognition evidence degree of the corresponding feature.

$PANU_{LSR}$ is responsible to perform the feature recognition of the "Vertical line segment" and "Horizontal line segment" features according to Fig. 16. It is



μ: PANU's output in evidence degree form

PANU$_{LSR}$: PANU of Line segment recognition

PANU$_{HR}$: PANU of Histogram recognition

Fig. 15 Image recognition—comparison procedure architecture

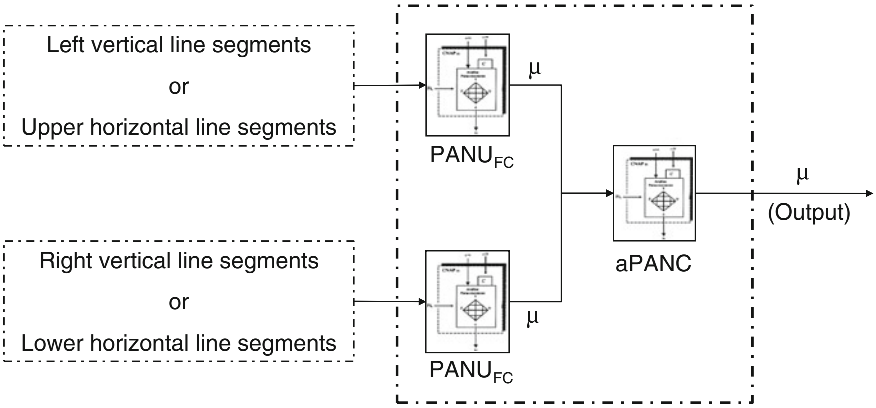
composed by two PANU of Feature comparison (PANU$_{FC}$) and a Analytical PANC (aPANC).

PANU$_{LSR}$'s input depends on the feature to be analyzed. If the analyzed feature is "Vertical line segments" the inputs will be the left and right line segments, on other hand, if the analyzed feature is "Horizontal line segments" then the inputs will be the upper and lower line segments.

In PANU$_{LSR}$ architecture, PANU$_{FC}$ compares evidence degree pairs. Each evidence degree pair is composed by one evidence degree from standard character and one evidence degree from presented character according to the analyzed feature. And, aPANC analyses the inputs and performs the interconnection between them [2]. Its inputs are composed by the PANU$_{FC}$'s output and its output represents the recognition evidence degree of the analyzed feature.

In turn, PANU$_{FC}$, presented on Fig. 17, compares, in fact, the evidence degrees. PANU$_{FC}$ architecture receives two evidence degrees lists, standard character features and presented character features.

PANU$_{FC}$ first layer is composed by Paraconsistent Analysis Nodes (PANs). PAN is an algorithm able to treat and control signs of imprecise and contradictory information [2]. Each PAN receives two evidence degrees as input, μ_n and (standard) μ_n, where μ_n is an evidence degree from presented character feature list and (standard) μ_n is an evidence degree from standard character feature list, calculates the interval between the inputs and then sends the output to the next layer.



μ: PANU's output or aPANC's output in evidence degree form

PANU$_{FC}$: PANU of Feature comparison

aPANC: Analytical PANC

Left or right vertical line segments: evidence degree list

Upper or lower horizontal line segments: evidence degree list

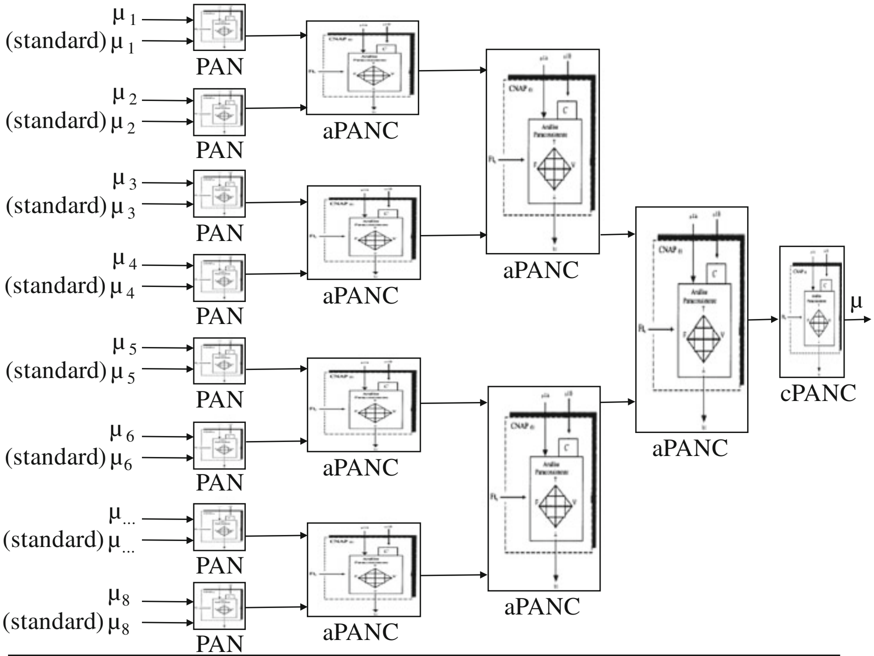
Fig. 16 PANU$_{LSR}$ architecture

PANU$_{FC}$ next layers, except the last one, are composed by aPANCs. Each aPANC receives two evidence degrees from previous layer, calculates the resulting evidence degree and sends its value to the next layers.

The last layer is composed by other PANC type, the Crossing PANC (cPANC), which is able to channel signals into the PANN [2]. Its output represents the recognition evidence degree of the analyzed feature (Vertical line segment or Horizontal line segment).

Returning to the architecture present on Fig. 15, PANU$_{HR}$ performs the histogram feature recognition. It receives two evidence degrees lists, one from standard character features and other from presented character features, uses a PAN to calculate the evidence interval between the standard and presented character features and then calculates a single recognition evidence degree through the ParaExtr$_{ctr}$ algorithm according to Fig. 18.

The ParaExtr$_{ctr}$ algorithm, is an algorithm able to decrease, gradually, contradiction effects on information signals from uncertain knowledge databases employing PANs [9]. From an evidence degree list represented by G_{ei}, it selects the maximum and minimum evidence degrees (ei_{max} and ei_{min}) to compose the PAN inputs as showed in Fig. 18. Here, into the ParaExtr$_{ctr}$ algorithm, the PANs are responsible for calculating the interval between μ and λ inputs to extract contradiction effect: "$\mu = ei_{max}$" and "$\lambda = 1 - ei_{min}$". Then, the selected evidence degrees from G_{ei} are replaced by the PAN output into the same list (G_{ei}). This process is repeated until only one evidence degree remains into the G_{ei} list.



μ_n: Evidence degree from presented character features

(standard) μ_n: Evidence degree from standard character features

PAN: Paraconsistent Analysis Node

aPANC: Analytical PANC

cPANC: Crossing PANC

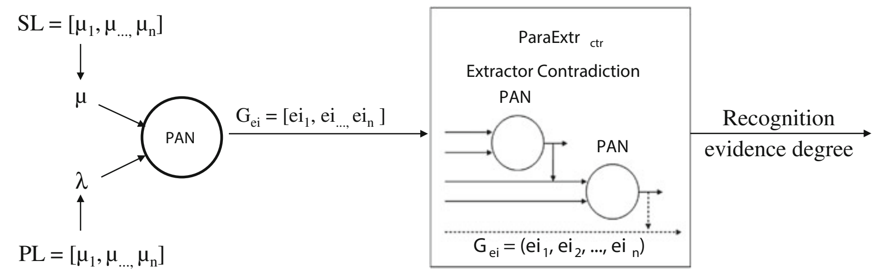
Fig. 17 $PANU_{FC}$ architecture

At the end of this process we have one recognition evidence degree of each histogram feature. Now, the recognition evidence degree of the standard character in analyses is calculated according to the Fig. 19.

At the end of this step we have one recognition evidence degree for each analyzed standard character.

3.4 Image Classification

Image classification step performs the analysis among the recognition evidence degrees obtained for each standard character into the Image recognition step. It receives a single recognition evidence degree for each standard character and identifies the maximum and minimum value.



SL: Evidence degree from the list of standard character features

PL: Evidence degree from the list of presented character features

μ: One value from SL

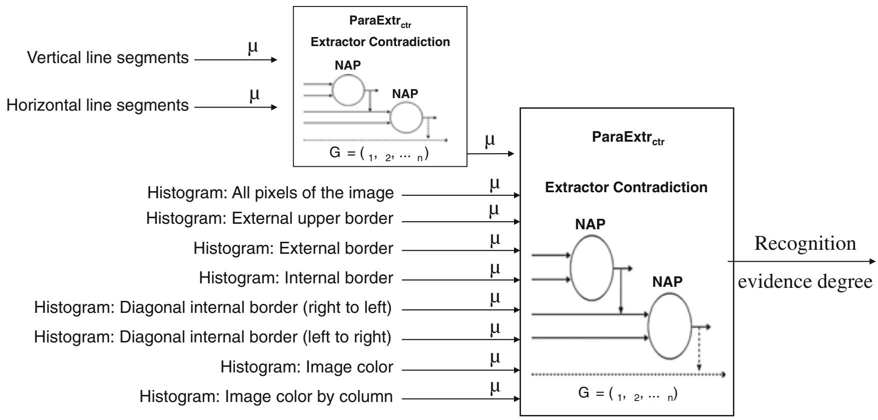
λ: 1 – (One value from PL)

PAN: Paraconsistent Analysis Node

ei_n: PAN's output

G_{ei}: List of evidence interval obtained from PAN's output

Fig. 18 Histogram recognition architecture



μ: Recognition evidence degree of the feature PAN: Paraconsistent Analysis Node

G_μ: List of recognition evidence degree from the features

Fig. 19 Standard character recognition architecture

The standard character with the maximum value represents the character recognized by the PRP process and the standard character with the minimum value represents the character discarded by the PRP process.

At the end of this step we have the recognized and discarded character.

4 Practical Applications

In this section we present some tests used to analyze the PRP performance with MICR characters and handwritten characters.

As we saw on the section above, the PRP process can be configured according to the sample type and the characters which will be recognized. Therefore, we have chosen different feature sets for each sample type to configure the PRP process.

4.1 MICR Recognition

MICR characters are used on Brazilian bank checks to codify data from customer's bank account. These characters are organized in line form as presented in Fig. 20.

To represent the MICR characters we will use the corresponding characters according to Fig. 21 to represent the ten MICR digits and the three MICR special characters.

For the tests we selected 2,092-element sample composed by ten distinct digits and three special characters from actual Brazilian bank checks and, two different feature sets to configure the Feature extraction step:

1. "Histogram: Internal border" and "Histogram: Diagonal internal border (right to left)";
2. "Histogram: Internal border" and "Histogram: Diagonal internal border (left to right)".

Figure 22 presents the PRP architecture for MICR recognition.

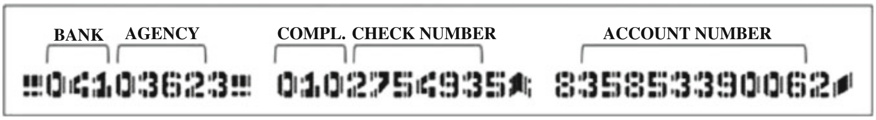


Fig. 20 MICR line example

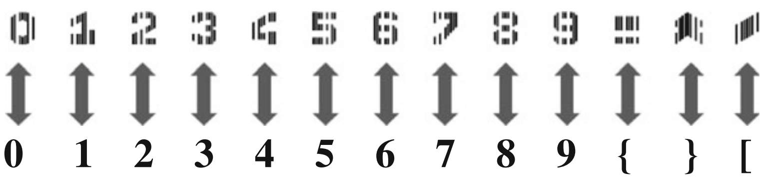


Fig. 21 MICR corresponding characters

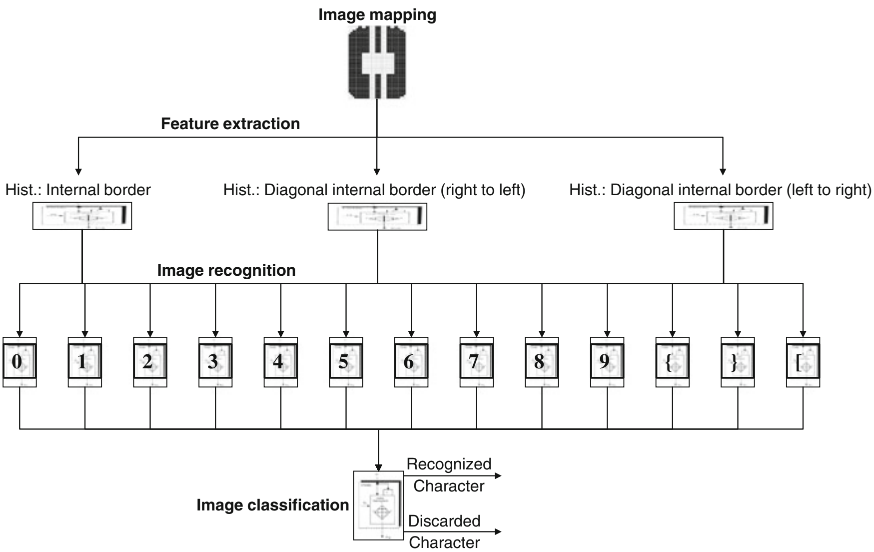


Fig. 22 PRP architecture for MICR recognition

Both feature sets presented results with 97.8 % correct recognition rate, 97.4 % sensitivity, 99.8 % specificity, 98.6 % efficiency and 99.7 % accuracy.

Table 1 presents the results obtained on the tests for both feature sets aforementioned. The table columns represent:

- Sample: sample type (use Fig. 21 to find the corresponding MICR character);
- Size: amount of elements
- Correct recognition: amount of correct recognition;
- % Sensitivity: sensitivity rate;
- % Specificity: specificity rate;
- % Efficiency: efficiency rate;
- % Accuracy: accuracy rate.

4.2 *Handwritten Numeric Character Recognition*

Handwritten numeric characters are used in numerous documents as bank checks, protocols, patient records and others.

For the tests we selected 1,050-element sample composed by ten distinct handwritten digits and the feature set composed by "Histogram: External border" and "Histogram: Diagonal internal border (right to left)" to configure the Feature extraction step.

Table 1 Results obtained in tests with MICR characters considering "Histogram: Internal border" and "Histogram: Diagonal internal border (right to left)" features or "Histogram: Internal border" and "Histogram: Diagonal internal border (left to right)" features

Sample	Size	Correct recognition	% Sensitivity	% Specificity	% Efficiency	% Accuracy
0	436	434	99.5	99.9	99.7	99.9
1	264	261	98.9	99.9	99.4	99.8
2	122	122	1.00	99.9	1.00	99.9
3	153	152	99.3	99.8	99.6	99.8
4	152	151	99.3	1.00	99.7	99.9
5	190	179	94.2	1.00	71.0	99.5
6	111	109	98.2	99.7	99.0	99.7
7	98	94	95.9	1.00	98.0	99.8
8	146	143	97.9	99.8	98.9	99.7
9	102	96	94.1	99.7	96.9	99.4
{	138	133	96.4	99.9	98.2	99.7
}	90	87	96.7	99.0	97.9	98.9
[	90	86	95.6	99.9	97.7	99.7
Total	2,092	97.8 %	97.4	99.8	96.6	99.7

Figure 23 presents the PRP architecture for handwritten numeric character recognition.

Table 2 presents the results obtained in tests.

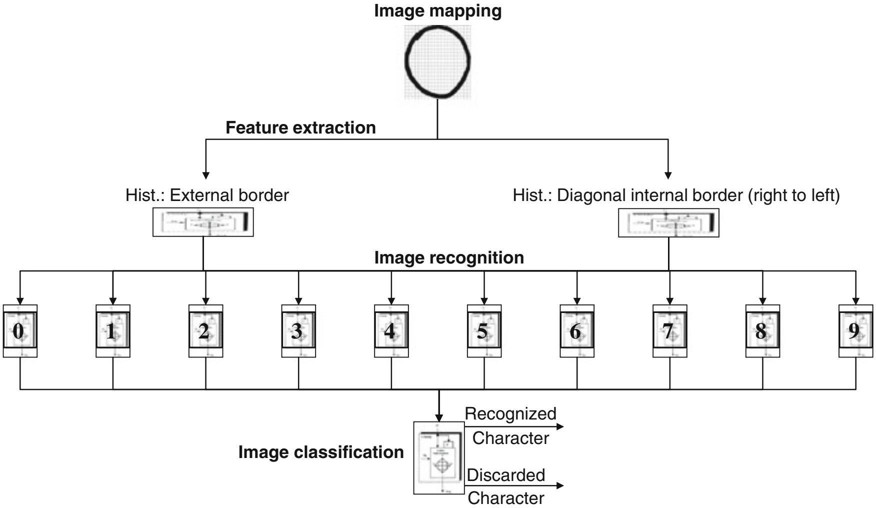


Fig. 23 PRP architecture for handwritten numeric character recognition

Table 2 Results obtained in tests with Handwritten numeric characters considering "Histogram: External border" and "Histogram: Diagonal Internal border (right to left)" features

Sample	Size	Correct recognition	% Sensitivity	% Specificity	% Efficiency	% Accuracy
0	109	109	1.00	99.4	99.7	99.4
1	92	62	67.4	99.6	83.5	96.8
2	121	118	97.5	98.6	98.1	98.5
3	134	123	91.8	99.3	95.6	98.4
4	85	77	90.6	99.7	95.1	98.9
5	103	81	78.6	99.6	89.1	97.5
6	104	103	99.0	97.5	98.2	97.6
7	88	87	98.9	98.7	98.8	98.8
8	116	109	94.0	98.5	96.2	98.0
9	98	93	94.9	99.8	97.3	99.3
Total	1,050	91.6 %	91.3	99.1	95.2	98.3

The feature set aforementioned presented the best results with 91.6 % correct recognition rate, 91.3 % sensitivity, 99.1 % specificity, 95.2 % efficiency and 98.3 % accuracy.

4.3 Additional Experiments

In addition to MICR character and Handwritten numeric character tests we selected three types of unaccented alphabetic characters to carry out some experiments. Figure 24 presents the PRP architecture for this type of character.

Printed Alphabetic Character—Arial 12. For the tests we selected 1,783-element sample composed by fifty-two distinct alphabetic character and the feature set composed by "Histogram: External border" and "Histogram: Diagonal internal border (right to left)" to configure the Feature extraction step. The results presented 82.4 % correct recognition rate, 91.0% sensitivity, 99.7 % specificity, 95.3 % efficiency and 99.3 % accuracy.

Printed Alphabetic Character—Times New Roman 12. For the tests we selected 264-element sample composed by fifty-two distinct alphabetic character and the feature set composed by "Histogram: Internal border" and "Histogram: Diagonal internal border (left to right)" to configure the Feature extraction step. The results presented 95.1 % correct recognition rate, 95.2 % sensitivity, 99.9 % specificity, 97.5 % efficiency and 99.8 % accuracy.

Handwritten Alphabetic Character. For the tests we selected 284-element sample composed by 52 distinct alphabetic character and the feature set composed by "Histogram: External border" and "Histogram: Diagonal internal border (right to left)" to configure the Feature extraction step. The results presented 46.5 % correct

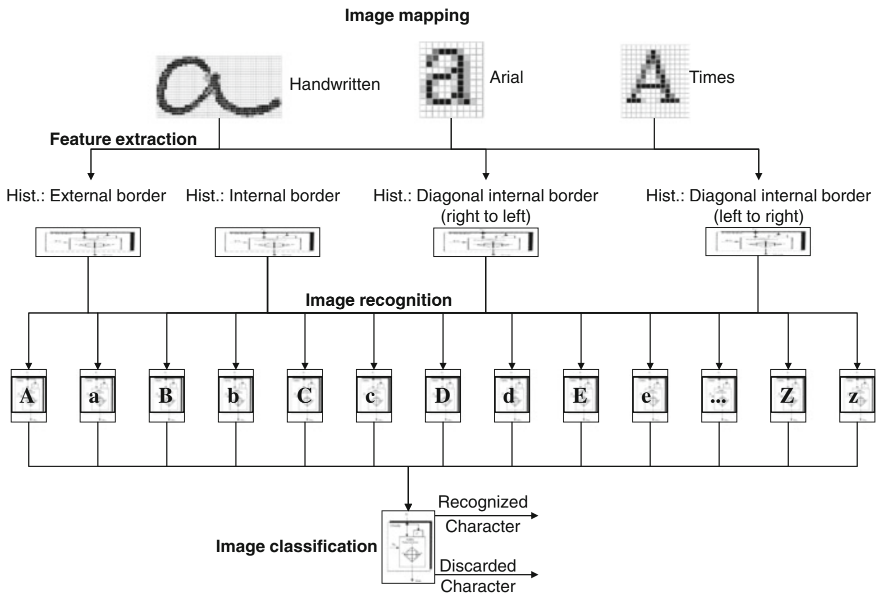


Fig. 24 PRP architecture for unaccented alphabetic character recognition

recognition rate, 47.5 % sensitivity, 98.9 % specificity, 73.2 % efficiency and 97.9 % accuracy.

According to the results obtained on these tests with alphabetic characters, the PRP process proved able to recognize them, however some additional adaptations are required to improve the results.

5 Conclusion

Although the PRP process was considered to recognize numeric characters with a previously determined pattern and handwritten numeric characters based on PANNs, it can also be applied to several medical areas to pattern recognition such as medical report management systems, which uses a barcode to identify the patient; automatic recognition of handwritten medical report; Laboratory Information System, which uses a specific protocol to identify the patient on the collection tube labels and others.

The method considered to collect the evidence degrees, extracted the features and recognized them, that is, the PRP process construction, was able to effectively treat ambiguous and uncertain aspects inherent to noisy data analysis directly and without trivialization. Thus, this process proved to be fully operational and applicable to MICR and handwritten numeric character recognition.

The obtained experimental results to recognize MICR, numeric and alphabetic characters demonstrate that the PRP process can be improved to recognize either other character types or other image types. And then, the PRP process application into Biomedicine area sounds interesting and relevant due to its intrinsic importance on clinic diagnosis segment, image interpretation and analysis, signal interpretation and analysis, drug development and other important themes within Medicine area.

References

1. Souza, S.: Handwritten numeric character recognition system based on paraconsistent artificial neural network, master thesis, University of São Paulo, Brazil (in Portuguese) (2013)
2. Da Silva Filho, J.I., Torres, G.L., Abe, J.M.: Uncertainty treatment using paraconsistent logic: introducing paraconsistent artificial neural networks. IOS Press, Netherlands (2010)
3. Abe, J.M.: Foundations of annotated logics, Doctor thesis, University of São Paulo, Brazil (in Portuguese) (1992)
4. Abe, J.M.: Some aspects of paraconsistent systems and applications. Logique et Analyse. **157**, 83–96 (1997)
5. Prado, J.P.A.: A IA architecture based on paraconsistent logic, Doctor Thesis, University of São Paulo, Brazil (in portuguese) (1996)
6. Prado, J.C.A.: The phonic constellation and artificial neural network: computational analysis of speech production's applicability, Doctor Thesis, University of São Paulo, Brazil (in Portuguese) (2007)
7. 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. Systems. **34**(6), 1–9 (2009)
8. Mário, M.C.: Analysis of craniometric variables throughout paraconsistent logic neural network, Doctor Thesis, University of São Paulo, Brazil (in Portuguese) (2006)
9. Da Silva, J.I.: Paraconsistent algorithm extractor of contradiction's effects—$\text{ParaExtr}_{\text{ctr}}$ (in portuguese). Seleção Documental **15**, 21–25 (2009)
10. Fujisawa, Y., Shi M., Wakabayashi T., Kimura F.: Handwritten numeral recognition using gradient and curvature of gray scale image. 5th International Conference on Document Analysis and Recognition, Bangalore (1999)
11. Amend, K., Ruiz, M.S.: Handwriting Analysis: The Complete Basic Book. NJ, Franklin Lakes (1980)

Paraconsistent Logic in Decision Making: Paraconsistent Decision Method (PDM)

Fábio Romeu de Carvalho

Abstract This chapter introduces the Paraconsistent Decision-Making Method (PDM) based on Paraconsistent Annotated Evidential Logic E_τ, an alternative to classical logic and detracting from the principle of non-contradiction, that is, accept this principle without becoming trivial. In addition, an application example is presented in viability analysis, along with a comparative study of this method with the Statistical Method of Decision-Making and the referenced bibliography. In the application example, it analyzes the viability of launching a new product, examining carefully the situation of influencing factors with actual conditions of the consumer market. Comparing the PDM with the Statistical Method of Decision-Making (SMD) shows the compatibility of the two methods and consistency of the results obtained by both. Lastly, the bibliography used as reference as well as supplementary resources for consulting and research are presented.

Keywords Paraconsistent logic · Decision making method · Rule of decision · Para-analyzer algorithm

1 Paraconsistent Decision Method (MPD)

1.1 General Considerations

Throughout the study years for a Master's Degree (spent devoted to logic) [17] and PhD (to Decision Making) [23,24], a lot of theory has been read regarding decision making, particularly those decision processes used by Management [34, 41], in Organizations [6, 40] and in Production Engineering processes [28, 29, 39, 45].

F.R. de Carvalho (✉)
Universidade Paulista, UNIP, Rua Dr. Bacelar, nº 1.212, São Paulo CEP 04026-002, Brazil
e-mail: fabioromeu@unip.br

J.M. Abe (ed.), *Paraconsistent Intelligent-Based Systems*,
Intelligent Systems Reference Library 94, DOI 10.1007/978-3-319-19722-7_10

During this period, it has been noticed that practically all processes are concerned with using objective data, gathered throughout the period and catalogued in a certain manner. It also has been noticed that the decision process is aimed at using more intangible information stemmed from the knowledge, expertise and sensitivity etc. of experts (specialists) on different subjects.

Such information—usually not catalogued—is of great relevance when it comes to decision making in corporations. So much so that in a good number of cases, the chairman of the company relying almost exclusively on experience makes the decisions based on their knowledge and intuition acquired in past years [8]. Therefore when needed, the chairman will tap into this information that is stored away internally, which is not catalogued anywhere else.

Then the intention of finding a way to use this information was set (knowledge, expertise, experience and expert sensitivity) for decision-making but in such a way that they can be used without the expert's direct participation, interference or presence. The idea was to use information resulting from the knowledge and intuition of experts with experience in a specific area in order to help others in making decisions.

However, how does one store a person's knowledge, experience, sensitivity and intuition so that others may use them as ingredients in their own decision-making? After all, these are rather intangible values.

After a lot of thought and research, the possibility of using paraconsistent annotated evidential logic Eτ was contemplated. It allows for valuable evidence (opinions, diagnosis, etc.) of experts be stored in the form of numbers. By doing so, such evidence stored in the form of numbers, may be used by non-expert decision makers.

Some aspects of this idea have proven to be relevant. Paraconsistent annotated evidential logic Eτ enables the translation of an expert's background by means of degrees of favorable evidence (or belief) and degrees of contrary evidence (or disbelief) all done by way of numbers; it also permits the manipulation of such data even if inconsistent, contradictory or para-complete. Another relevant aspect is that once the experts' opinions are gathered through degrees of evidence, such data then becomes available to other people for a considerable amount of time, with no need for intervention of an expert, thus saving them the trouble and inconvenience of being asked to intervene at any time. It is practically a perpetuation of these opinions, which may help in decision-making for a long time [22].

Without emphasizing too much detail, this chapter is an attempt to present what has been known as the Paraconsistent Decision Method (PDM).

1.2 Notions of Paraconsistent Annotated Evidential Logic Eτ

Paraconsistent Logic, whose recent discovery has been attributed to Brazilian logician Newton C.A. da Costa, Ph.D., is an adversary of classical logic [30, 32, 35, 36] since it derogates from the principle of non-contradiction, that is, it accepts contradictions (propositions of the form ($A \wedge \neg A$) in its structure without trivializing itself, which is the exact opposite of classical logic [11, 12, 13, 15].

As part of this type of logic, paraconsistent annotated logic arose from the research papers of Da Costa et al. [9], who developed the first syntactic and semantic of this logic that was completed by Abe [1, 2]. The latter, along with his research team made significant advances which later resulted in the introduction of the paraconsistent annotated evidential logic Eτ.

In this logic, a proposition **p** is represented by $\mathbf{p}_{(\boldsymbol{a}\ ;\ \boldsymbol{b})}$, with $\boldsymbol{a}$ and $\boldsymbol{b}$ varying on the closed interval [0, 1] of real numbers. Therefore, the pair ($\boldsymbol{a}$; $\boldsymbol{b}$) belong to the Cartesian product [0, 1] × [0, 1]. The real number $\boldsymbol{a}$ translates the degree of favorable evidence in **p**, and $\boldsymbol{b}$, the degree of contrary evidence in **p** ($\boldsymbol{a}$ and $\boldsymbol{b}$ are also called the degree of belief and degree of disbelief in p respectively). The pair $\mu = (\boldsymbol{a}\ ;\ \boldsymbol{b})$ is referred to as the constant of annotations [3, 4, 14].

So, we have as extreme values: the pair (1; 0), which will translate the logical state known as Truth (V); (0; 1) doing the same for falsity (F); (1; 1) for inconsistency (T), and the pair (0; 0) for the logical state known as para-completeness (⊥).

The set $|\tau| = [0, 1] \times [0, 1]$ with the order relation $\leq^*$ is the annotation lattice ($\leq^*$ is defined by $((\boldsymbol{a}_1; \boldsymbol{b}_1), (\boldsymbol{a}_2; \boldsymbol{b}_2)) \in \leq^* \Leftrightarrow \boldsymbol{a}_1 \leq \boldsymbol{a}_2$ and $\boldsymbol{b}_1 \leq \boldsymbol{b}_2$), where $\leq$ is the order relation on the set of real numbers.

The annotation lattice defines the unit square represented in the Cartesian plane (Fig. 1).

For a certain annotation constant $\mu = (\boldsymbol{a}; \boldsymbol{b})$, are defined: $G(\boldsymbol{a}; \boldsymbol{b}) = \boldsymbol{a} + \boldsymbol{b} - 1$, known as the **degree of uncertainty**, and $H(\boldsymbol{a}; \boldsymbol{b}) = \boldsymbol{a} - \boldsymbol{b}$, known as the **degree of certainty**. Please notice that $-1 \leq G \leq 1$ and $-1 \leq H \leq 1$.

The segment CD, for which G = 0, is known as a perfectly defined line (PDL); AB, for which H = 0, is known as a perfectly undefined line (PIL). Other noticeable lines can thus be defined accordingly as follows:

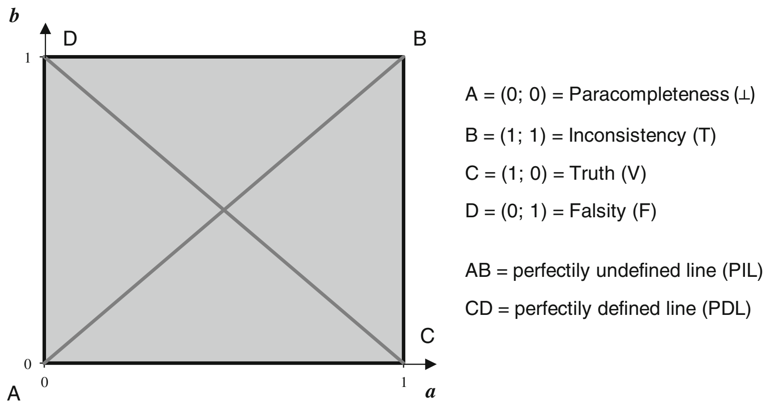


Fig. 1 Cartesian Unit Square (CUS)

Para-completeness borderline: straight line MN in such a way that $G = -k_1$, with $0 < k_1 < 1$;

Inconsistency borderline: straight line RS, in such a way that $G = +k_1$, with $0 < k_1 < 1$;

Falsity borderline: straight line TU, in such a way that $H = -k_2$, with $0 < k_2 < 1$;

Truth borderline: straight line PQ, in such a way that $H = +k_2$, with $0 < k_2 < 1$.

Usually, $k_1 = k_2 = k$ is adopted to give symmetry to the diagram such as in Fig. 2, in which you have $k_1 = k_2 = k = 0.60$.

The unit square of the Cartesian plane can be divided into regions translating the logical states with different characteristics. A division that attributes to the lattice that it represents an interesting and convenient characterization is the one obtained through PDL, PIL and limit lines (Fig. 2), partitioning it into twelve regions.

From these twelve regions, four extreme regions are featured: region of truth (CPQ), region of falsity (DTU), region of para-completeness (AMN) and region of inconsistency (BRS).

The k_2 value will be called the **level of requirement**, because it represents the minimum value for $|H|$ so that the point $\mathbf{X} \equiv (\boldsymbol{a}; \boldsymbol{b})$ belongs to either the region of falsity or truth. In Fig. 2 there are four extreme regions and one central region.

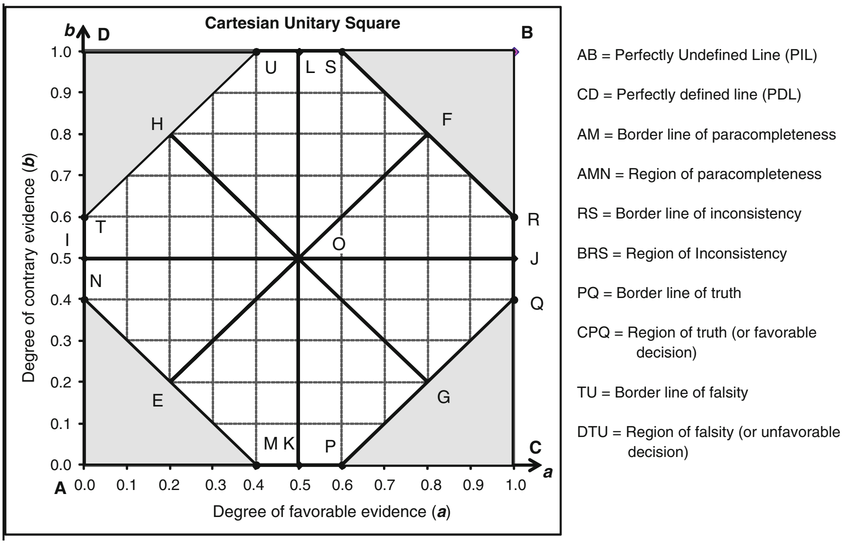


Fig. 2 Cartesian Unitary Square (CUS) divided in twelve regions

1.3 Operators of Evidential Annotated Paraconsistent Logic: NOT, MAX and MIN

The NOT operator is defined by $\text{NOT}(a;b) = (b;a)$.

For example: $\text{NOT}(0.8;\ 0.3) = (0.3;\ 0.8)$.

So: $\neg p_{(0.8;0.3)} = p_{(0.3;0.8)} = p_{[\sim(0.8;0.3)]}$.

Notice that: $\text{NOT}(\text{T}) = \text{T}; \text{NOT}(\perp) = \perp; \text{NOT}(\text{V}) = \text{F}$ and $\text{NOT}(\text{F}) = \text{V}$.

Operator **MAX** (from here on forward called maximizing) is defined as being applied to a group of n annotations (n ≥ 1); it acts in such a way as to maximize the degree of certainty ($H = \boldsymbol{a} - \boldsymbol{b}$) in this group of annotations by selecting the best favorable evidence (bigger value of $\boldsymbol{a}$) and the best contrary evidence (smaller value of $\boldsymbol{b}$). It is defined as follows [27]:

$$\textbf{MAX}\{(\boldsymbol{a}_1;\boldsymbol{b}_1),(\boldsymbol{a}_2;\boldsymbol{b}_2),\ldots,(\boldsymbol{a}_n;\boldsymbol{b}_n)\} = (\max\{\boldsymbol{a}_1,\boldsymbol{a}_2,\ldots,\boldsymbol{a}_n\};\ \min\{\boldsymbol{b}_1,\boldsymbol{b}_2,\ldots,\boldsymbol{b}_n\})$$

Operator MIN (from now on called minimizing) is also defined to be applied to a group of n annotations (n ≥ 1); it acts in such a way as to minimize the degree of certainty ($H = \boldsymbol{a} - \boldsymbol{b}$) in this group of annotations by selecting the worst favorable evidence (smaller value of $\boldsymbol{a}$) and the worst contrary evidence (bigger value of $\boldsymbol{b}$). It is defined as follows [27]:

$$\textbf{MIN}\{(\boldsymbol{a}_1;\boldsymbol{b}_1),(\boldsymbol{a}_2;\boldsymbol{b}_2),\ldots,(\boldsymbol{a}_n;\boldsymbol{b}_n)\} = (\min\{\boldsymbol{a}_1,\boldsymbol{a}_2,\ldots,\boldsymbol{a}_n\};\max\{\boldsymbol{b}_1,\boldsymbol{b}_2,\ldots,\boldsymbol{b}_n\})$$

If $\boldsymbol{\mu}_1 = (\boldsymbol{a}_1;\boldsymbol{b}_1), \boldsymbol{\mu}_2 = (\boldsymbol{a}_2;\boldsymbol{b}_2)$ and $\boldsymbol{a}_1 \le \boldsymbol{a}_2$ and $\boldsymbol{b}_1 \le \boldsymbol{b}_2$, it follows that

$\textbf{MAX}\{\boldsymbol{\mu}_1,\boldsymbol{\mu}_2\} = \textbf{MAX}\{(\boldsymbol{a}_1;\boldsymbol{b}_1),(\boldsymbol{a}_2;\boldsymbol{b}_2)\} = (\boldsymbol{a}_2;\boldsymbol{b}_1)$ and

$\textbf{MIN}\{\boldsymbol{\mu}_1,\boldsymbol{\mu}_2\} = \textbf{MIN}\{(\boldsymbol{a}_1;\boldsymbol{b}_1),(\boldsymbol{a}_2;\boldsymbol{b}_2)\} = (\boldsymbol{a}_1;\boldsymbol{b}_2)$.

The operator **MAX** must be applied in situations where the items considered are not all determinant, and it is sufficient that one of them presents a favorable condition.

Operator **MIN** has the purpose of minimizing the degree of certainty to a set of annotations. Therefore, it must be applied in situations where the considered items are all determinant.

Once the experts are separated into groups, operator **MAX** must be applied inside each group (intragroup) and then, operator **MIN** among the results obtained from the groups (between groups).

For instance, a set of four specialists distributed among two groups: A, by specialists E_1 and E_2, and B, by specialists E_3 and E_4.

The application of the rules in this case is as follows:

$$\mathbf{MIN}\{\mathbf{MAX}[(E_1),\ (E_2)];\ \mathbf{MAX}[(E_3),(E_4)]\}\ \text{ or }\ \mathbf{MIN}\{[G_A];\ [G_B]\}$$

This way of applying the rules of maximization and minimization for decision-making is known as the min/max principle or optimistic decision, because it minimizes the greater degree of certainty [40].

1.4 Decision Regions and Decision Rule [21, 37]

Figure 2 shows a Cartesian plane unit square divided into twelve regions. Among them, four external regions stand out.

In the AMN and BRS regions, the module of the degree of uncertainty is high (close to 1) and the module of the degree of certainty is low (close to zero).

Figure 2 represents $|G| \geq 0.6$ and $|H| < 0.6$. Therefore, the X points = (***a***; ***b***) of these two regions translate logical states of high uncertainty (inconsistency /contradictory or para-completeness) and of little certainty. So, they do not provide for decision making since they only acknowledge that the data leading to the pair (***a***; ***b***) show high uncertainty.

Regions CPQ and DTU are the exact opposite: the module of the degree of uncertainty is low (close to zero) and the module of the degree of certainty is high (close to 1). Figure 2 shows $|G| < 0.6$ and $|H| \geq 0.6$. Therefore, the X points = (***a***; ***b***) of these two regions translate logical states of low uncertainty (contradictory/inconsistency or para-completeness), but of high certainty (truth or falsity). So they can be used in decision making since they translate a high level of certainty in the enterprise being analyzed.

The region CPQ, in which the degree of certainty is close to 1, is called the region of truth, while the other, DTU, in which the degree of certainty is close to −1, is called the region of falsity.

There is a value of the module of the degree of certainty (|H|) defining the regions of truth and falsity. In the case of Fig. 2, such a value is 0.6. Therefore, if a degree of certainty should be greater than or equal to 0.6 ($H \geq 0.6$), then the logical resulting state X = (***a***; ***b***) will be close to point C; thus resulting in a favorable decision (the enterprise is feasible/viable).

Otherwise, if the degree of certainty is less than or equal to –0.6 ($H \leq -0.6$), then the logical resulting state X = (***a***; ***b***) will be close to point D; hence resulting in an unfavorable decision (the enterprise is not feasible/viable).

The module of the degree of certainty that defines the decision regions (|H|) is called **level of requirement** (LR).

Therefore, the decision rule can be expressed as follows:

$H \geq LR \Rightarrow$ favorable decision (enterprise is feasible);

$H \leq -LR \Rightarrow$ unfavorable decision (enterprise is not feasible);

$-LR < H < LR \Rightarrow$ inconclusive analysis.

Table 1 Summary of the analysis of the twelve regions of Cartesian Unitary Square

Region	***a***	***b***	G	H	Description	Representation
AMN	[0; 0.4]	[0; 0.4]	[−1; −0.6]	[−0.4; 0.4]	Indetermination or paracompleteness	$\bot$
BRS	[0.6; 1]	[0.6; 1]	[0.6; 1]	[−0.4; 0.4]	Inconsistency	$\top$
CPQ	[0.6; 1]	[0; 0.4]	[−0.4; 0.4]	[0.6; 1]	Truth	V
DTU	[0; 0.4]	[0.6; 1]	[−0.4; 0.4]	[−1; −0.6]	Falsity	F
OFSL	[0.5; 0.8]	[0.5; 1]	[0; 0.6[	[−0.5; 0]	Quasi-inconsistency tending to falsity	$Q\top \to F$
OHUL	[0.2; 0.5]	[0.5; 1]	[0; 0.5]	[−0.6; 0]	Quasi-falsity tending to inconsistency	$QF \to \top$
OHTI	[0; 0.5]	[0.5; 0.8]	[–0.5; 0]	[−0.6; 0]	Quasi-falsity tending to indetermination	$QF \to \bot$
OENI	[0; 0.5]	[0.2; 0.5]	[−0.6; 0]	[−0.5; 0]	Quasi-indetermination tending to falsity	$Q\bot \to F$
OEMK	[0.2; 0.5]	[0; 0.5]	[−0.6; 0]	[0; 0.5]	Quasi-indetermination tending to truth	$Q\bot \to V$
OGPK	[0.5; 0.8]	[0; 0.5]	[−0.5; 0]	[0; 0.6]	Quasi-truth tending to indetermination	$QV \to \bot$
OGQJ	[0.5; 1]	[0.2; 0.5]	[0; 0.5]	[0; 0.6]	Quasi-truth tending to inconsistency	$QV \to \top$
OFRJ	[0.5; 1]	[0.5; 0.8]	[0; 0.6]	[0; 0.5]	Quasi-inconsistency tending to truth	$Q\top \to V$

The unit square of the Cartesian plane divided into regions as in Fig. 2, for example, is known as the para-analyzing algorithm [16]. In fact, each region in Fig. 2 translates a set of logical states that determines the tendency of the analyzed situation, as summarized in Table 1.

1.5 The Paraconsistent Decision Method (PDM)

Every reasonable decision must be based on a variety of factors that may have an influence on the enterprise being analyzed. Each of these factors will influence the enterprise in their own unique way, indicating feasibility (favorable decision) or non-feasibility (unfavorable decision) of the enterprise, or still it may not be conclusive and not indicate neither favorably nor unfavorably, or not even contrary. This can be clearly noticed when the para-analyzing algorithm is used, that is, when the values of the degrees of favorable evidence ($\boldsymbol{a}_{i,R}$) and the degrees of unfavorable evidence ($\boldsymbol{b}_{i,R}$) for each factor are plotted in such a way that each factor is represented by an X point = ($\boldsymbol{a}$; $\boldsymbol{b}$) of the lattice τ.

However, it is not practical to work with a great number of factors because the method would prove to be quite exhausting and expensive. So, the proposition would be to narrow down the list of factors to only those that are most important, that is, to the ones having the most influence on the decision, within of course a limit of rationality as professed by Simon, "who works with a simplified real-life model, taking into consideration the fact that many aspects of the reality are substantially irrelevant in any determined moment; he chooses based on the rhythm of the actual situation, considering only a few more relevant and crucial factors" [41].

Usually, examining separately the influence of each factor is not necessary. What really matters in the viability analysis of an enterprise is the combined influence of all selected factors, which are translated into a final logical state known as barycenter (W). It is represented by a **W** point in the lattice τ, whose coordinates ($\boldsymbol{a}_W$ and $\boldsymbol{b}_W$) are determined by the weighted average of the coordinates of points $X_i = (\boldsymbol{a}_{i,R}, \boldsymbol{b}_{i,R})$ of τ, that translates the resulting influence of each factor separately.

1.5.1 Steps of the Paraconsistent Decision Method

The Paraconsistent Decision Method (PDM) consists of eight steps of which only a brief idea will be outlined at first, while the rest of details will come along shortly down the chapter.

(1) Set the level of requirement (LR) of the decision to be made.
(2) Select the most important factors (F_i) that most influence the decision.
(3) Define sections (S_j) for each factor (Three, four, five or more sections can be set depending on the case and the level of detail desired).
(4) Build the database, which is composed of the weights (P_i) assigned to factors (for instance to distinguish them by importance) and by the values of favorable evidence (or degree of belief) (***a***) and the contrary evidence (or degree of disbelief) (***b***) assigned to each factor in one of the sections; the weights and values of evidences are assigned by experts conveniently selected to give their opinion (The database can also be built with stored statistical data obtained from previous experiences in similar enterprises).
(5) Perform field survey (or research) to find out in which section (condition) each factor is placed.
(6) Obtain the value of the degree of favorable evidence ($\boldsymbol{a}_{i,R}$) and the value of the degree of contrary evidence ($\boldsymbol{b}_{i,R}$) for each of the chosen factors (F_i), with $1 \leq i \leq n$, in sections found in the survey (S_{pj}) by applying the maximizing (MAX operator) and minimizing (MIN operator) techniques of logic (Eτ).
(7) Obtain the degree of favorable evidence ($\boldsymbol{a}_W$) and the degree of contrary evidence ($\boldsymbol{b}_W$) of the barycenter of the points representing the selected factors in the lattice (τ).
(8) Make the decision by applying the decision rule or the para-analyzing algorithm.

1.5.2 Detailed Steps of the Paraconsistent Decision Method (PDM)

To make an analysis of the feasibility of a project for a decision, the planning should be assigned and designated to a particular person (the business owner, an engineer, a consultant etc.). This individual would be required to handle the data in such a way as to translate them into Eτ language, thus enabling a proper plotting for tools of analysis of this kind of logic.

Setting Up the Level of Requirement

Firstly, one should set up the level of requirement (LR) for the decision to be made that depends on the level of safety desired for the decision as well as the responsibility it entails, the size of the investment at stake, the involvement and the risks to human lives, or to environment, etc.

When the level of requirement (LR) is set, the decision regions are automatically defined and, consequently, so is the decision rule and the para-analyzer algorithm. For example, take for instance a situation where the requirement level is set at 0.70. The decision rule is:

$H \geq 0.70 \Rightarrow$ favorable decision (enterprise is feasible);
$H \leq -0.70 \Rightarrow$ unfavorable decision (enterprise is not feasible);
$-0.70 < H < 0.70 \Rightarrow$ inconclusive analysis.

See Fig. 3 for the para-analyzer algorithm.

Choice of the Factors of Influence

Secondly, one should find out the factors that may influence in the success (or failure) of the enterprise. This is done by consulting with people who work in similar organizations, or with experts on the subject matter or on projects of the same nature, or even reading specialized literature, etc.

Once the factors that may influence in success (or failure) of the enterprise are found, one should choose the **n** factors F_i ($1 \leq i \leq \mathbf{n}$) that are more important and more influential, that is, those whose conditions would mostly affect the feasibility of the enterprise. Whether the chosen factors may affect in various ways or whether they present different importance in the decision, such differences may be compensated by assigning different weights to each chosen factor.

Setting Up Sections for Each Factor

The next step is to set up the sections $S_{i,j}$ ($1 \leq j \leq \mathbf{s}$), that translate the conditions in which each factor can be found. Then, depending on the level of refinement intended for the analysis, more (or fewer) sections can be assigned.

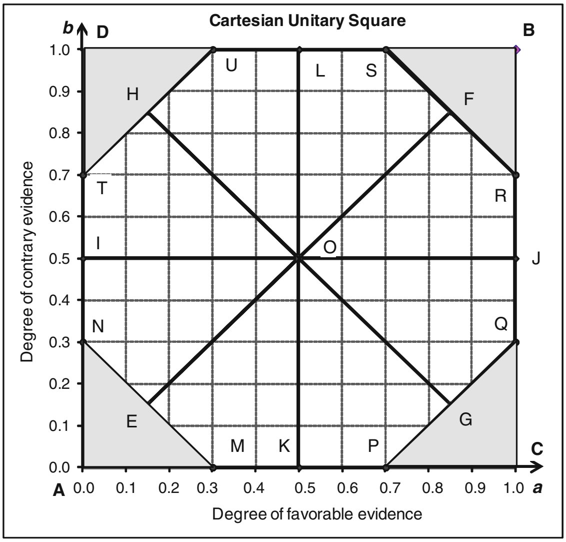


Fig. 3 Decision rule and para-analyzer algorithm for a level of requirement equal to 0.70

Should one choose to assign three sections, they would be:

S_1 factor is in **favorable** condition to the enterprise;
S_2 factor is in **neutral** condition to the enterprise;
S_3 factor is in **unfavorable** condition to the enterprise.

Should one chose to assign five sections, they would be:

S_1 factor is in **very favorable** condition to the enterprise;
S_2 factor is in **favorable** condition to the enterprise;
S_3 factor is in **neutral** condition to the enterprise;
S_4 factor is in **unfavorable** condition to the enterprise;
S_5 factor is in **very unfavorable** condition to the enterprise.

Building Up the Database

Constructing the database is a very important task and to do so **m** experts E_k ($1 \leq k \leq$ **m**) **in** the area—or relating area—must be selected. The selection of experts should look for people with different backgrounds, so that the assignment of values is not a result of one single line of thought.

One should notice that the process displays great versatility once it enables the choice of more (or fewer) factors of influence. It also enables the assignment of three or more sections for each factor, the use of a larger (or smaller) number of experts and the use of weights to differentiate between the factors and/or experts. Although the process may allow it, it is not advisable to use less than four experts so that the outcome is not too subjective.

Firstly, experts should indicate if—among the chosen factors—there is a distinction regarding importance. If there is not, then a weight equal to 1 (one) should be assigned to all of them; on the other hand if there is, then each expert should assign a weight ($q_{i,k}$) to the factor they deem fit and should take into consideration the importance of the factor in relation to the other regarding the decision to be made.

$$q_{i,k} = \text{weight assigned by expert k to factor i.}$$

In assigning the weights, some conditions may be applied, for example the weights must be whole or integer numbers and belonging to the interval [1, 10]. Once all invited experts assign weights to all factors, the final weight, P_i, of the factors will be adopted, and is calculated as the arithmetic mean of the weights assigned by the experts.

$$P_i = \frac{\sum_{k=1}^{m} q_{i,k}}{m} \tag{1}$$

Please note that there is a possibility of the experts being distinguished according to their background (practice, experience, knowledge), thereby assigning different weights (r_k) to them. In this case, the final weight, P_i, of each factor would not be an arithmetic average but instead a weighted average.

$$P_i = \frac{\sum_{k=1}^{m} r_k q_{i,k}}{\sum_{k=1}^{m} r_k} \tag{2}$$

$$r_k = \text{weight assigned by the knowledge engineer to expert k.}$$

Highlighted here is just one of the features of the method, showing its versatility and the variety of options the method offers users.

The next step towards building the database is to ask experts to assign the degree of favorable evidence (***a***) and the degree of contrary evidence (***b***) to each factor present in the conditions in a to be found and which are characterized by the defined sections.

Each ordered pair ($\boldsymbol{a}_{i,j,k}$; $\boldsymbol{b}_{i,j,k}$) formed by the values of the degrees of favorable and contrary evidence, assigned by an expert E_k to the factor F_i according to the condition defined by a section S_j, constitutes an annotation symbolized by $\boldsymbol{\mu}_{i,j,k}$.

The database consists of the matrix of weights [P_i], a column matrix of **n** rows formed by the weights P_i of the factors and by the matrix of annotations $M_A = \boldsymbol{\mu}_{i,j,k}$ (bivariate annotations) with **n** × **s** rows and **m** columns, that is: a total of

n × **s** × **m** elements. This last matrix is composed of all annotations that the **m** experts assigned to each of the **n** factors under the conditions defined by the **s** sections.

The matrix $M_A = [\boldsymbol{\mu}_{i,j,k}]$ may be represented by $[(\boldsymbol{a}_{i,j,k}; \boldsymbol{b}_{i,j,k})]$, since each annotation $\boldsymbol{\mu}_{i,j,k}$ is an ordered pair of the form $(\boldsymbol{a}_{i,j,k}; \boldsymbol{b}_{i,j,k})$.

For example, in a situation with four experts (m = 4), five factors (n = 5) and three sections for each factor (s = 3), the matrix of the weights, M_P, will be a column matrix of 5 rows (n = 5) and the matrix of annotations, M_A, will be a matrix of 15 rows and 4 columns (n × s = 5 × 3 = 15 e m = 4) as indicated in Tables 2 and 3.

Field Survey

Now that the decision-making device is complete, one is able to apply the method and reach the final decision, using information that will be collected through research on the condition (defined by section) of each influence factor. So, the next step will be to perform the field survey and find out the real condition of each of the influence factors, that is, to discover in which section $S_{i,j}$ lies each factor F_i.

Table 2 Calculation table with indication of the bivalued annotations

	M_P	M_{pq}	M_{Dpq}				M_{G1}	M_{G2}	M_R
F_i	P_i	S_{pi}	E_1	E_2	E_3	E_4	MAX [E_1, E_4]	MAX [E_2, E_3]	MIN {G_1, G_2}
F_1	P_1	S_{p1}	$\lambda_{1,1}$	$\lambda_{1,2}$	$\lambda_{1,3}$	$\lambda_{1,4}$	$\rho_{1,g1}$	$\rho_{1,g2}$	$\omega_{1,R}$
F_2	P_2	S_{p2}	$\lambda_{2,1}$	$\lambda_{2,2}$	$\lambda_{2,3}$	$\lambda_{2,4}$	$\rho_{2,g1}$	$\rho_{2,g2}$	$\omega_{2,R}$
F_3	P_3	S_{p3}	$\lambda_{3,1}$	$\lambda_{3,2}$	$\lambda_{3,3}$	$\lambda_{3,4}$	$\rho_{3,g1}$	$\rho_{3,g2}$	$\omega_{3,R}$
F_4	P_4	S_{p4}	$\lambda_{4,1}$	$\lambda_{4,2}$	$\lambda_{4,3}$	$\lambda_{4,4}$	$\rho_{4,g1}$	$\rho_{4,g2}$	$\omega_{4,R}$
F_5	P_5	S_{p5}	$\lambda_{5,1}$	$\lambda_{5,2}$	$\lambda_{5,3}$	$\lambda_{5,4}$	$\rho_{5,g1}$	$\rho_{5,g2}$	$\omega_{5,R}$

Table 3 Calculation table with indication of the values of favorable (**a**) and contrary (**b**) evidences

	M_P	M_{pq}	M_{Dpq}								M_{G1}: MAX		M_{G2}: MAX		M_R: MIN	
F_i	P_i	S_{pi}	E_1		E_2		E_3		E_4		[E_1, E_4]		[E_2, E_3]		{G_1, G_2}	
F_1	P_1	S_{p1}	$a_{1,1}$	$b_{1,1}$	$a_{1,2}$	$b_{1,2}$	$a_{1,3}$	$b_{1,3}$	$a_{1,4}$	$b_{1,4}$	$a_{1,g1}$	$b_{1,g1}$	$a_{1,g2}$	$b_{1,g2}$	$a_{1,R}$	$b_{1,R}$
F_2	P_2	S_{p2}	$a_{2,1}$	$b_{2,1}$	$a_{2,2}$	$b_{2,2}$	$a_{2,3}$	$b_{2,3}$	$a_{2,4}$	$b_{2,4}$	$a_{2,g1}$	$b_{2,g1}$	$a_{2,g2}$	$b_{2,g2}$	$a_{2,R}$	$b_{2,R}$
F_3	P_3	S_{p3}	$a_{3,1}$	$b_{3,1}$	$a_{3,2}$	$b_{3,2}$	$a_{3,3}$	$b_{3,3}$	$a_{3,4}$	$b_{3,4}$	$a_{3,g1}$	$b_{3,g1}$	$a_{3,g2}$	$b_{3,g2}$	$a_{3,R}$	$b_{3,R}$
F_4	P_4	S_{p4}	$a_{4,1}$	$b_{4,1}$	$a_{4,2}$	$b_{4,2}$	$a_{4,3}$	$b_{4,3}$	$a_{4,4}$	$b_{4,4}$	$a_{4,g1}$	$b_{4,g1}$	$a_{4,g2}$	$b_{4,g2}$	$a_{4,R}$	$b_{4,R}$
F_5	P_5	S_{p5}	$a_{5,1}$	$b_{5,1}$	$a_{5,2}$	$b_{5,2}$	$a_{5,3}$	$b_{5,3}$	$a_{5,4}$	$b_{5,4}$	$a_{5,g1}$	$b_{5,g1}$	$a_{5,g2}$	$b_{5,g2}$	$a_{5,R}$	$b_{5,R}$

Upon completion of the survey, one obtains a set of **n** sections resulting from the survey, $S_{i,jp,}$ with $1 \le i \le$ **n**, one for each factor and translating the actual conditions of the factors (jp translates the particular value of j, $1 \le i \le$ **s**, that was obtained from the research pertaining to factor F_i). These **n** values of the resulting sections of the survey constitute a column matrix of **n** rows (M_{pq}). With this result it is possible to look up in the database what the opinions of the experts are on the feasibility of the enterprise in the conditions of the factors.

Therefore, the database can stand out as another matrix, a subset of M_A, that can be named as the matrix of surveyed data $M_{Dpq} = [\boldsymbol{\lambda}_{i,k}]$, of **n** rows and **m** columns, made from the rows of M_A.

Calculation of the Resulting Annotations

At this point, a task needs to be carried out: divide the experts into groups according to the criteria of the engineer that directs the decision making process.

When forming the groups of experts to apply **MAX** and **MIX** operators in the study of real cases in order to assist in decision making, some details must be adhered to.

The operator **MAX** should be applied to situations in which the favorable opinion of just one of them is enough to consider the group result as satisfactory. The operator **MIN** should be applied to situations where the opinions of two or more experts (or surveyed items) are all determinant and it must be mandatory that all are favorable so that the result of the analysis is considered satisfactory.

The following is an example that may clarify some more how the groups are formed. Imagine the four components of a soccer team: the goalkeeper (a player with the number 1), the defense (four players numbered 2–5), the mid-field (three players numbered 6–8) and the offense (three players numbered from 9 to 11). This is what a soccer understood would call the 4-3-3 tactic.

Every coach knows that in order to build an excellent team he must have a great player in each sector, that is, a formidable goalkeeper, a great defense player, a terrific mid-fielder and a tremendous attacker. Therefore, each sector (group) is judged by their best player, suggesting that maximization is applied to each group.

Therefore in the team's viability analysis, the groups are already naturally formed. The goalkeeper, who is the only one in the sector, makes up one group (A); The four defense players make up another group (B), bearing in mind that only one great player is enough to meet the requirements of the team. Similarly, the three mid-fielders constitute the third group (C) and the three attackers, the fourth group (D).

On the other hand, if all team sectors are excellent, the team will be "excellent"; whereas if one sector is not excellent, but good, this good sector will define the team status "good", despite the other three excellent sectors; If medium, the team will be "medium" and so on, thus suggesting the application of the minimization rule among the groups (sectors).

Based on the above, the distribution of the groups and the application of the **MAX** and **MIN** operators are defined as follows:

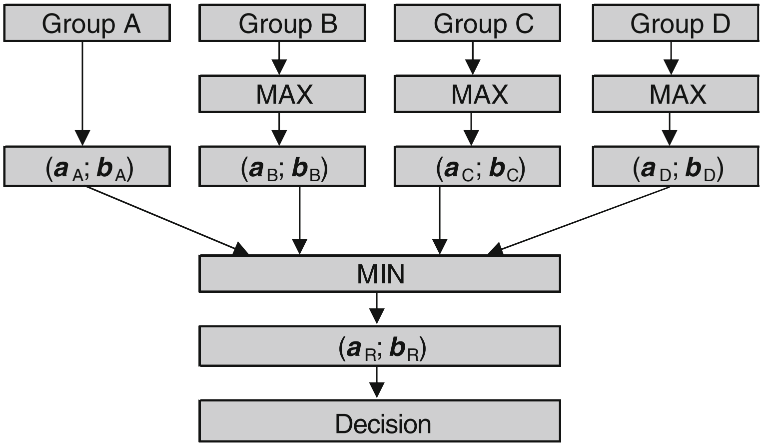


Fig. 4 Operators **MAX** and **MIN** scheme application

$$\mathbf{MIN}\{[\text{Group A}], [\text{Group B}], [\text{Group C}], [\text{Group D}]\} \text{ or}$$
$$\mathbf{MIN}\{[1], \mathbf{MAX}[2,3,4,5], \mathbf{MAX}[6,7,8], \mathbf{MAX}[9,10,11]\} \text{ or}$$
$$\mathbf{MIN}\{[(\boldsymbol{a}_A;\boldsymbol{b}_A)], [(\boldsymbol{a}_B;\boldsymbol{b}_B)], [(\boldsymbol{a}_C;\boldsymbol{b}_C)], [(\boldsymbol{a}_D;\boldsymbol{b}_D)]\},$$

represented by the schematic in Fig. 4.

It should be noted that the goalkeeper's influence is very high because he is the only one responsible for the result in group A.

The application of these operators provides a way to determine the values of favorable evidence ($\boldsymbol{a}_{i,R}$) and of contrary evidence ($\boldsymbol{b}_{i,R}$), results for each factor F_i ($1 \le i \le \mathbf{n}$) in the section $S_{i,jp}$ found in the survey.

Suppose that the **m** experts are distributed among **p** groups G_h, with $1 \le h \le \mathbf{p}$, each one with g_h expert being $\sum_{h=1}^{p} g_h = \mathbf{m}$.

Thus, the group G_h will be composed of the following g_h experts: E_{1h}, E_{2h},…, E_{ghh}. Then, the application of the rule of maximizing within the group G_h (intra-group) can be summarized as follows:

$$\mathbf{MAX}\left[(E_{1h}), (E_{2h}), \ldots (E_{ghh})\right] \text{ or}$$
$$\mathbf{MAX}\left[(\boldsymbol{a}_{i,1h};\boldsymbol{b}_{i,1h}), (\boldsymbol{a}_{i,2h};\boldsymbol{b}_{i,2h}), \ldots, (\boldsymbol{a}_{i,ghh};\boldsymbol{b}_{i,ghh})\right]$$

The result of the maximization is the ordered pair ($\boldsymbol{a}_{i,h}$; $\boldsymbol{b}_{i,h}$), in which

$$\boldsymbol{a}_{i,h} = \max\{\boldsymbol{a}_{i,1h}, \boldsymbol{a}_{i,2h}, \ldots, \boldsymbol{a}_{i,\,ghh}\} \text{ and } \boldsymbol{b}_{i,h} = \min\{\boldsymbol{b}_{i,\,1h}, \boldsymbol{b}_{i,\,2h}, \ldots, \boldsymbol{b}_{i,\,ghh}\}$$

Since there are **n** factors, **n** ordered pairs are obtained in this way, resulting in the group G_h, matrix $M_{Gh} = [(\boldsymbol{a}_{i,h}; \boldsymbol{b}_{i,h})]$, with **n** rows, since $1 \le i \le \mathbf{n}$, and one column.

It can be inferred that since there are **p** groups, **p** similar column matrices are obtained.

Returning to the example of **n** = 5 factors, **s** = 3 sections and **m** = 4 specialists and, assuming that the four experts were distributed among two groups (**p** = 2), the first, G_1, by specialists E_1 and E_4 and the second, G_2, by specialists E_2 and E_3, the application of the rule of maximizing would be as follows:

$$\begin{aligned}
&\text{Within } G_1 \text{ group: } \mathbf{MAX}[(E_1),(E_4)];\\
&\text{Within } G_2 \text{ group: } \mathbf{MAX}[(E_2),(E_3)] \text{ or}\\
&\mathbf{MAX}\left[(\boldsymbol{a}_{i,1};\boldsymbol{b}_{i,1}),(\boldsymbol{a}_{i,4};\boldsymbol{b}_{i,4})\right], \text{ giving } (\boldsymbol{a}_{i,g1};\boldsymbol{b}_{i,g1}) \text{ for group } G_1 \text{ and}\\
&\mathbf{MAX}\left[(\boldsymbol{a}_{i,2};\boldsymbol{b}_{i,2}),(\boldsymbol{a}_{i,3};\boldsymbol{b}_{i,3})\right], \text{ giving } (\boldsymbol{a}_{i,g2};\boldsymbol{b}_{i,g2}) \text{ for group } G_2 \text{ such that}\\
&\boldsymbol{a}_{i,g1} = \max\{\boldsymbol{a}_{i,1},\boldsymbol{a}_{i,4}\}; \qquad \boldsymbol{b}_{i,g1} = \min\{\boldsymbol{b}_{i,1},\boldsymbol{b}_{i,4}\} \text{ and}\\
&\boldsymbol{a}_{i,g2} = \max\{\boldsymbol{a}_{i,2},\boldsymbol{a}_{i,3}\}; \qquad \boldsymbol{b}_{i,g2} = \min\{\boldsymbol{b}_{i,2},\boldsymbol{b}_{i,3}\}.
\end{aligned}$$

Therefore, **p** = 2 column matrices are obtained with **n** = 5 rows as a result of the application of the maximization rule within groups G_1 and G_2 (intra-groups). They are:

$$M_{G1} = \left[(\boldsymbol{a}_{i,g1};\boldsymbol{b}_{i,g1})\right] = \left[\boldsymbol{\rho}_{i,g1}\right] \quad \text{and} \quad M_{G2} = \left[(\boldsymbol{a}_{i,g2};\boldsymbol{b}_{i,g2})\right] = \left[\boldsymbol{\rho}_{i,g2}\right],$$

and can be represented in another way as in Tables 2 and 3.

Once the maximization rules (**MAX** operator) are applied within the groups (intra-groups), the next step will be the application of the minimization rule (**MIN** operator) in the groups (between groups) that can be as follows:

$$\begin{aligned}
&\mathbf{MIN}\{[G_1],[G_2],\ldots[G_h],\ldots[G_p]\} \text{ or}\\
&\mathbf{MIN}\{(\boldsymbol{a}_{i,g1};\boldsymbol{b}_{i,g1}),\ (\boldsymbol{a}_{i,g2};\boldsymbol{b}_{i,g2}),\ldots(\boldsymbol{a}_{i,gh};\boldsymbol{b}_{i,gh}),\ldots,(\boldsymbol{a}_{i,gp};\boldsymbol{b}_{i,gp})\},
\end{aligned}$$

Hence obtaining for each factor F_i the resulting annotation ($\boldsymbol{a}_{i,R}$; $\boldsymbol{b}_{i,R}$), in which

$$\begin{aligned}
\boldsymbol{a}_{i,R} &= \min\{\boldsymbol{a}_{i,g1},\boldsymbol{a}_{i,g2},\ldots,\boldsymbol{a}_{i,gh},\ldots,\boldsymbol{a}_{i,gp}\} \text{ and}\\
\boldsymbol{b}_{i,R} &= \max\{\boldsymbol{b}_{i,g1},\boldsymbol{b}_{i,g2},\ldots,\boldsymbol{b}_{i,gh},\ldots,\boldsymbol{b}_{i,gp}\}.
\end{aligned}$$

Since there are **n** factors, these results will constitute a matrix column with **n** rows, which will be called resulting matrix $M_R = [(\boldsymbol{a}_{i,R}; \boldsymbol{b}_{i,R})] = [\boldsymbol{\omega}_{i,R}]$.

Going back to the example of **n** = 5 factors, **s** = 3 sections and **m** = 4 experts, the application of the minimization rule would be reduced to **MIN**{[G_1], [G_2]}.

$$\boldsymbol{a}_{1,\mathrm{R}} = \min\{\boldsymbol{a}_{1,\mathrm{g1}}, \boldsymbol{a}_{1,\mathrm{g2}}\}\mathrm{e} \quad \boldsymbol{b}_{1,\mathrm{R}} = \max\{\boldsymbol{b}_{1,\mathrm{g1}}, \boldsymbol{b}_{1,\mathrm{g2}}\};$$
$$\boldsymbol{a}_{2,\mathrm{R}} = \min\{\boldsymbol{a}_{2,\mathrm{g1}}, \boldsymbol{a}_{2,\mathrm{g2}}\}\mathrm{e} \quad \boldsymbol{b}_{2,\mathrm{R}} = \max\{\boldsymbol{b}_{2,\mathrm{g1}}, \boldsymbol{b}_{2,\mathrm{g2}}\};$$
$$\boldsymbol{a}_{3,\mathrm{R}} = \min\{\boldsymbol{a}_{3,\mathrm{g1}}, \boldsymbol{a}_{3,\mathrm{g2}}\}\mathrm{e} \quad \boldsymbol{b}_{3,\mathrm{R}} = \max\{\boldsymbol{b}_{3,\mathrm{g1}}, \boldsymbol{b}_{3,\mathrm{g2}}\}$$
$$\boldsymbol{a}_{4,\mathrm{R}} = \min\{\boldsymbol{a}_{4,\mathrm{g1}}, \boldsymbol{a}_{4,\mathrm{g2}}\}\mathrm{e} \quad \boldsymbol{b}_{4,\mathrm{R}} = \max\{\boldsymbol{b}_{4,\mathrm{g1}}, \boldsymbol{b}_{4,\mathrm{g2}}\}$$
$$\boldsymbol{a}_{5,\mathrm{R}} = \min\{\boldsymbol{a}_{5,\mathrm{g1}}, \boldsymbol{a}_{5,\mathrm{g2}}\}\mathrm{e} \quad \boldsymbol{b}_{5,\mathrm{R}} = \max\{\boldsymbol{b}_{5,\mathrm{g1}}, \boldsymbol{b}_{5,\mathrm{g2}}\}$$

The resulting column matrix (M_R) of 5 rows along with the previous are represented in Tables 2 and 3.

The application of the rules of maximization (**MAX**) and minimization (**MIN**) to the example in analysis can be summarized as follows:

$$\mathbf{MIN}\{\mathbf{MAX}[(\mathrm{E}_1)\,(\mathrm{E}_4)], \mathbf{MAX}[(\mathrm{E}_2)\,(\mathrm{E}_3)]\} \text{ or } \mathbf{MIN}\{[\mathrm{G}_1]\,[\mathrm{G}_2]\}.$$

In applications, some of the matrices seen (matrix of weights, $[P_i]$, of surveyed sections, M_{pq}, of surveyed data, M_{Dpq}, matrix of the groups, M_{Gh}, and resulting matrix, M_R) will be displayed as columns in the calculations table and would have the same format in Tables 2 or 3. These tables take into consideration, for example, a situation with four experts (**m** = 4), five factors (**n** = 5) and three sections for each factor (**s** = 3), used as an example.

The values of the resulting favorable evidence ($\boldsymbol{a}_{i,R}$) and contrary evidence ($\boldsymbol{b}_{i,R}$) obtained for all factors, aid in determining what the influence of each factor is in terms of feasibility of the enterprise.

Determining the Barycenter

Usually, there is not much interest in discovering the influence of each factor separately. However, it is crucial to know the combined influence of all factors on the feasibility of the enterprise, once it leads to the final decision.

The combined influence of factors is determined by the analysis of the barycenter (**W**) of the points representing them in the Cartesian plane (in lattice τ). In order to determine the barycenter, one must calculate its coordinates, that are the degrees of favorable ($\boldsymbol{a}_W$) and contrary ($\boldsymbol{b}_W$) evidence. The degree of favorable evidence of the barycenter ($\boldsymbol{a}_W$) is equal to the weighted average of the degrees of favorable evidence results ($\boldsymbol{a}_{i,R}$) for all the factors, by taking as coefficients the weights (P_i) assigned by experts to the factors. In like manner, the degree of contrary evidence of the barycenter ($\boldsymbol{b}_W$) is calculated.

$$a_W = \frac{\sum_{i=1}^{n} P_i a_{i,R}}{\sum_{i=1}^{n} P_i} \qquad b_W = \frac{\sum_{i=1}^{n} P_i b_{i,R}}{\sum_{i=1}^{n} P_i} \tag{3}$$

In the case where all factors have equal weights (P_i), the weighted averages above will turn into arithmetic means and the barycenter of the points representing the factors will turn into the geometric center of those points. In this case the Eq. 3 becomes as follows:

$$a_W = \frac{\sum_{i=1}^{n} a_{i,R}}{n} \qquad b_W = \frac{\sum_{i=1}^{n} b_{i,R}}{n} \tag{4}$$

Decision-Making

Since the favorable ($\boldsymbol{a}_W$) and contrary ($\boldsymbol{b}_W$) evidence values of the barycenter are determined, the final decision is now ready to be made by utilizing the para-analyzer algorithm.

To do this, just plot the ordered pair ($\boldsymbol{a}_W$; $\boldsymbol{b}_W$) in the Cartesian plane and find out in which region of the lattice τ does the barycenter **W** belongs. If it belongs to the region of truth, then the decision will be favorable, i.e., the analysis implies that the enterprise is feasible. If it belongs to the region of falsity, then the decision will be unfavorable, i.e. the analysis implies that the enterprise is not feasible. However, if it is found in any other different region of the lattice (τ) the analysis is deemed not conclusive. In such a case, the feasibility of the enterprise is not stated.

Another way of reaching a final decision is the application of the decision rule. In this case, just calculate the degree of certainty of the barycenter ($H_W = \boldsymbol{a}_W - \boldsymbol{b}_W$) and apply the decision rule. If $H_W \geq LR$, then the decision is favorable and the implementation of the enterprise is recommended (feasible); if $H_W \leq -LR$, then the decision is unfavorable and the implementation of the enterprise is not recommended (not feasible) and, if $-LR < H_W < LR$, the analysis is inconclusive.

It is important to note, therefore, that the degree of certainty of the barycenter (H_W) is the well determined final number that will enable the decision-making, and that the entire process will lead to this very important number.

All the operations described above can be carried out with the aid of a computer program such as Microsoft's Excel Software Package. For simplification purposes, this program will be referred to as Calculation Program (CP).

In order to illustrate the application of the PDM, one example will be presented in the next paragraph.

2 PDM in Analysis of Viability

To set an example, we are going to apply PDM in a problem that marketing professionals often face, and that is a thorough study involving the launching of a new product [20]. There is a great number of factors influencing such a decision.

Basically, the idea is to isolate the factors of major influence on these decisions, establish five sections for each one and, with the assistance of specialists, obtain annotations for each factor in each section, attributing a degree of favorable evidence (***a***) and a degree of contrary evidence (***b***) to all of them [18, 19].

After that, applying the operators (**MAX**) and (**MIN**) one obtains resultant degrees of favorable evidence ($\boldsymbol{a}_{i,R}$) and contrary evidence ($\boldsymbol{b}_{i,R}$) for each factor. These, when plotted on the Cartesian Unit Square, CUS, will facilitate in finding out how viability was influenced by each factor. This is the para-analyzer algorithm.

For the final decision making, it is necessary to know the combined influence of all analyzed factors. This may be determined by the barycenter **W** of the points that represent each factor separately.

The degree of favorable evidence ($\boldsymbol{a}_W$) of **W** is the arithmetic mean of the resulting degrees of favorable evidence for all the factors, and the degree of contrary evidence ($\boldsymbol{b}_W$) is the arithmetic mean of the resulting degrees of contrary evidence for all the factors. With such values one can calculate the degree of certainty of **W** and apply the rule of decision.

2.1 Choosing Factors of Influence and Establishing Sections

We have come up with ten factors (F_{01} to F_{10}) that may influence the viability of launching a new product.

For each of these factors five sections were established (S_1 to S_5), so that S_1 represents a very favorable situation, S_2 represents a favorable situation, S_3 represents a neutral situation, S_4 represents an unfavorable situation and S_5 is a very unfavorable situation in terms of launching a new product. After that, specialists (E_1 to E_4) will be required to attribute the degree of favorable evidence (***a***) and the degree of contrary evidence (***b***) in relation to the viability of the product in each of the sections for all of the factors. Their results will constitute the database.

The chosen factors and the established sections are:

F_{01}: necessity and utility of the product—translated by the percentage of the population that uses the product—S_1: more than 90 %; S_2: between 70 and 90 %; S_3: between 30 and 70 %; S_4: between 10 and 30 %; S_5: less than 10 %.

F_{02}: number of features or functions of the product—measured by comparing the average M of features or functions of similar market product—S_1: more than 1.5 M; S_2: between 1.2 and 1.5 M; S_3: between 0.8 and 1.2 M; S_4: between 0.5 and 0.8 M; S_5: less than 0.5 M.

F_{03}: competition—translated by the quality and quantity of competitors in the same region—S_1: very little; S_2: little; S_3: average; S_4: strong; S_5: very strong.

F_{04}: clients potential—translated by the size and purchasing power of the region's population—S_1: very big; S_2: big; S_3: average; S_4: small; S_5: very small.

F_{05}: acceptance of product or similar product existing in the market—translated by the percentage of the population using the product—S_1: more than 90 %; S_2:

between 70 and 90 %; S_3: between 30 and 70 %; S_4: between 10 and 30 %; S_5: less than 10 %.

F_{06}: product price in the market—translated in relation to the average market price P of the product (or a similar product)—S_1: less than 70 %P; S_2: between 70 and 90 %P; S_3: between 90 and 110 %P; S_4: between 110 and 130 %P; S_5: more than 130 %P.

F_{07}: product estimated cost—translated in relation to the market average price P (or a similar product)—S_1: less than 20 %P; S_2: between 20 and 40 %P; S_3: between 40 and 60 %P; S_4: between 60 and 80 %; S_5: more than 80 %P.

F_{08}: product life cycle (C)—measured by one time unit T—S_1: more than 10 T; S_2: between 8 and 10 T; S_3: between 4 and 8 T; S_4: between 2 and 4 T; S_5: less than 2T.

F_{09}: Deadline for project development and product implementation—measured in terms of life cycle (C)—S_1: less than 10 %C; R2: between 10 and 30 %C; S_3: between 30 and 70 %C; S_4: between 70 and 90 %C; S_5: more than 90 %C.

F_{10}: Investment for project development and product implementation—Measured in terms of net result (RES) expected in the product life cycle—S_1: less than 20 %RES; S_2: between 20 and 40 %RES; S_3: between 40 and 60 %RES; S_4: between 60 and 80 %RES; S_5: more than 80 %RES.

2.2 Database Construction

Below is an assumption of the opinions obtained from four specialists (E_1: marketing professional; E_2: economist; E_3: production engineer; E_4: business manager). They are given in Table 4.

2.3 Working Out the PDM

Once the database is built, we will proceed to analyze the viability of product X in Region Y. To do so, we must conduct a survey in Region Y with respect to product X, so as to determine in which section each factor is encountered. The result of this survey can be summarized in columns 1 and 2 of Table 5.

This means that researchers must check in Region Y for each of the factors F_i, ($1 \leq i \leq 10$) and in which section S_j ($1 \leq j \leq 5$) product X is found. Column 2 of Table 5 must be filled in with the values S_j. With these results we can then extract from the database (Table 4) the specialist's opinions on the conditions of product X in Region Y. They are summarized in columns 3–10 in Table 5.

After that, we can apply the operators **MAX** and **MIN** of the Paraconsistent Annotated Evidential Logic. For this application it is necessary to form the groups of specialists according to the opinion of the engineer. For example, in the given frame of specialists it is reasonable to have: in Group A—a professional of

Table 4 Database (degrees of favorable and contrary evidences attributed by specialists in each section for all factors)

F_i	S_i	Specialist 1		Specialist 2		Specialist 3		Specialist 4	
		$a_{i,1}$	$b_{i,1}$	$a_{i,2}$	$b_{i,2}$	$a_{i,3}$	$b_{i,3}$	$a_{i,4}$	$b_{i,4}$
F_{01}	S_1	0.88	0.04	0.94	0.14	0.84	0.08	0.78	0.03
	S_2	0.63	0.19	0.79	0.23	0.73	0.14	0.59	0.24
	S_3	0.48	0.43	0.53	0.44	0.58	0.39	0.48	0.41
	S_4	0.23	0.77	0.41	0.61	0.33	0.73	0.29	0.53
	S_5	0.01	0.94	0.13	0.88	0.14	1.00	0.17	0.91
F_{02}	S_1	1.00	0.05	0.95	0.15	1.00	0.10	0.85	0.00
	S_2	0.75	0.25	0.85	0.25	0.85	0.30	0.73	0.35
	S_3	0.55	0.45	0.55	0.45	0.65	0.40	0.45	0.55
	S_4	0.35	0.65	0.31	0.79	0.29	0.70	0.24	0.83
	S_5	0.00	0.95	0.15	0.75	0.15	0.85	0.25	1.00
F_{03}	S_1	0.92	0.08	0.98	0.18	0.88	0.12	0.82	0.07
	S_2	0.67	0.23	0.83	0.27	0.77	0.18	0.63	0.28
	S_3	0.52	0.47	0.57	0.48	0.62	0.43	0.52	0.45
	S_4	0.17	0.73	0.24	0.65	0.37	0.67	0.33	0.64
	S_5	0.05	0.98	0.17	0.83	0.18	0.02	0.21	0.95
F_{04}	S_1	0.95	0.11	1.00	0.21	0.91	0.15	0.85	0.10
	S_2	0.70	0.26	0.86	0.30	0.80	0.21	0.66	0.31
	S_3	0.55	0.50	0.60	0.51	0.65	0.46	0.55	0.48
	S_4	0.30	0.76	0.48	0.68	0.22	0.70	0.28	0.60
	S_5	0.08	1.00	0.20	0.86	0.21	0.05	0.24	0.98
F_{05}	S_1	1.00	0.88	0.06	0.10	0.95	0.85	0.04	0.00
	S_2	0.70	0.20	0.80	0.30	0.80	0.20	0.70	0.30
	S_3	0.50	0.50	0.60	0.50	0.60	0.40	0.50	0.40
	S_4	0.30	0.0	0.33	0.69	0.30	0.70	0.26	0.73
	S_5	0.00	1.00	0.10	0.80	0.90	0.08	1.00	0.15
F_{06}	S_1	0.90	0.10	1.00	0.10	0.90	0.00	1.00	0.00
	S_2	0.80	0.30	0.80	0.20	0.70	0.30	0.70	0.20
	S_3	0.60	0.50	0.60	0.40	0.50	0.40	0.50	0.50
	S_4	0.40	0.60	0.40	0.70	0.30	0.60	0.30	0.70
	S_5	0.10	0.80	0.20	0.90	0.13	1.00	0.00	1.00
F_{07}	S_1	0.95	0.15	1.00	0.10	0.85	0.00	1.00	0.05
	S_2	0.85	0.25	0.85	0.30	0.73	0.35	0.75	0.25
	S_3	0.55	0.45	0.65	0.40	0.45	0.55	0.55	0.45
	S_4	0.40	0.65	0.35	0.75	0.24	0.78	0.35	0.65
	S_5	0.05	0.88	0.15	0.85	0.12	1.00	0.00	0.95
F_{08}	S_1	0.98	0.18	0.88	0.12	0.82	0.07	0.92	0.08
	S_2	0.83	0.27	0.77	0.18	0.63	0.28	0.67	0.23
	S_3	0.57	0.48	0.62	0.43	0.52	0.45	0.52	0.47
	S_4	0.45	0.65	0.37	0.85	0.33	0.57	0.27	0.86
	S_5	0.08	0.83	0.18	0.95	0.21	0.95	0.05	0.98

(continued)

Table 4 (continued)

F_i	S_i	Specialist 1		Specialist 2		Specialist 3		Specialist 4	
		$a_{i,1}$	$b_{i,1}$	$a_{i,2}$	$b_{i,2}$	$a_{i,3}$	$b_{i,3}$	$a_{i,4}$	$b_{i,4}$
F_{09}	S_1	1.00	0.21	0.91	0.15	0.85	0.10	0.95	0.11
	S_2	0.86	0.30	0.80	0.21	0.66	0.31	0.70	0.26
	S_3	0.60	0.51	0.65	0.46	0.55	0.48	0.55	0.50
	S_4	0.39	0.76	0.30	0.70	0.36	0.60	0.30	0.76
	S_5	0.10	0.86	0.15	0.93	0.24	0.98	0.08	1.00
F_{10}	S_1	0.94	0.14	0.84	0.08	0.78	0.03	0.88	0.04
	S_2	0.79	0.23	0.73	0.14	0.59	0.24	0.63	0.19
	S_3	0.53	0.44	0.58	0.39	0.48	0.41	0.48	0.43
	S_4	0.41	0.69	0.33	0.63	0.29	0.53	0.23	0.69
	S_5	0.13	0.79	0.14	0.90	0.17	0.91	0.01	0.94

marketing (E_1) along with an economist (E_2); in Group B—a production engineer (E_3) with a business manager (E_4). Therefore to apply the maximization (**MAX**) and minimization (**MIN**) rules to the specialist's opinions, we will do the following:

$$[(E_1)\ \mathbf{MAX}\ (E_2)]\ \mathbf{MIN}\ [(E_3)\ \mathbf{MAX}\ (E_4)]\ \text{or}$$
$$\mathbf{MIN}\ \{\mathbf{MAX}[(\mathbf{E_1}),\ (\mathbf{E_2})],\ \mathbf{MAX}[(\mathbf{E_3}),\ (\mathbf{E_4})]\}$$

In Table 5, the result of the application of the operator **MAX** to groups A and B (intra-groups) are in columns 11–14. The result of the application of the operator **MIN** between groups A and B (inter-groups) is shown in columns 15 and 16.

We are now going to analyze the final results with the para-analyzer algorithm. To do so, we are going to plot them together in the Cartesian plane (Fig. 5), assuming as boundary liner for truth and falsity the straight lines determined by $|H| = 0.60$ and as inconsistency and indetermination boundaries, the straight lines determined by $|G| = 0.60$, which means we are adopting 0.60 as the **level of requirement** for decision making, that is, we will make decisions with at least 0.60 or 60 % of certainty. The decision rule with such value is as follows:

$$\mathbf{H} \geq \mathbf{0.60} \Rightarrow \textbf{viable};$$
$$\mathbf{H} \leq -\mathbf{0.60} \Rightarrow \textbf{unviable}; \textbf{ and}$$
$$-\mathbf{0.60} < \mathbf{H} < \mathbf{0.60} \Rightarrow \textbf{inconclusive}.$$

This analysis determines the influence of each factor (F_1 to F_{10}) for the viability of launching product X in Region Y and also the combined influence of all factors through the barycenter **W**.

In the present case of study, the viability analysis for product X in Region Y, the analysis of the points obtained in the CUS has shown us that four factors (F_{02}, F_{03}, F_{05} and F_{09}) recommend the launching of the product with level of requirement

Table 5 Surveyed sections, degrees of favorable and contrary evidences, application of operators **MAX** and **MIN**, calculation and analysis of results

1	2	3	4	5	6	7	8	9	10	11	12
		Group A				Group B				A	
		E_1		E_2		E_3		E_4		E_1 MAX E_2	
F_i	S_{pj}	$a_{i,1}$	$b_{i,1}$	$a_{i,2}$	$b_{i,2}$	$a_{i,3}$	$b_{i,3}$	$a_{i,4}$	$b_{i,4}$	$a_{i,gA}$	$b_{i,gA}$
F_{01}	S_5	0.01	0.94	0.13	0.88	0.14	1.00	0.17	0.91	0.13	0.88
F_{02}	S_1	1.00	0.05	0.95	0.15	1.00	0.10	0.85	0.00	1.00	0.05
F_{03}	S_1	0.92	0.08	0.98	0.18	0.88	0.12	0.82	0.07	0.98	0.08
F_{04}	S_2	0.70	0.26	0.86	0.30	0.80	0.21	0.66	0.31	0.86	0.26
F_{05}	S_1	1.00	0.88	0.06	0.10	0.95	0.85	0.04	0.00	1.00	0.10
F_{06}	S_5	0.10	0.80	0.20	0.90	0.13	1.00	0.00	1.00	0.20	0.80
F_{07}	S_4	0.40	0.65	0.35	0.75	0.24	0.78	0.35	0.65	0.40	0.65
F_{08}	S_4	0.45	0.65	0.37	0.85	0.33	0.57	0.27	0.86	0.45	0.65
F_{09}	S_1	1.00	0.21	0.91	0.15	0.85	0.10	0.95	0.11	1.00	0.15
F_{10}	S_2	0.79	0.23	0.73	0.14	0.59	0.24	0.63	0.19	0.79	0.14

1	2	11	12	13	14	15	16	17	18	19
		A		B		A MIN B		Level of requirement = 0.60		
		E_1 MAX E_2		E_3 MAX E_4				Conclusions		
F_i	S_{pj}	$a_{i,gA}$	$b_{i,gA}$	$a_{i,gB}$	$b_{i,gB}$	$a_{i,R}$	$b_{i,R}$	H	G	Decision
F_{01}	S_5	0.13	0.88	0.17	0.91	0.13	0.91	−0.78	0.04	Unviable
F_{02}	S_1	1.00	0.05	1.00	0.00	1.00	0.05	0.95	0.05	Viable
F_{03}	S_1	0.98	0.08	0.88	0.07	0.88	0.08	0.80	−0.04	Viable
F_{04}	S_2	0.86	0.26	0.80	0.21	0.80	0.26	0.54	0.06	Inconclusive
F_{05}	S_1	1.00	0.10	0.95	0.00	0.95	0.10	0.85	0.05	Viable
F_{06}	S_5	0.20	0.80	0.13	1.00	0.13	1.00	−0.87	0.13	Unviable
F_{07}	S_4	0.40	0.65	0.35	0.65	0.35	0.65	−0.30	0.00	Inconclusive
F_{08}	S_4	0.45	0.65	0.33	0.57	0.33	0.65	−0.32	−0.02	Inconclusive
F_{09}	S_1	1.00	0.15	0.95	0.10	0.95	0.15	0.80	0.10	Viable
F_{10}	S_2	0.79	0.14	0.63	0.19	0.63	0.19	0.44	−0.18	Inconclusive
Baricenter W: averages of the resultant degrees						0.62	0.40	0.21	0.02	Inconclusive

equal to 0.60 since they belong to the truth region (viability); two factors (F_{01} and F_{06}) do not suggest the launching of the product since they belong to the falsity region (unviability).

The other factors fell in the inconclusive region, thus indicating that the product launch is neither viable nor inviable. F_{04} fell in the semi-truth region that tends to inconsistency; F_{10} fell in the semi-truth region also tending to para-completeness or indetermination; and F_{07} e F_{08} is in the semi-falsity region that tends to para-completeness or indetermination.

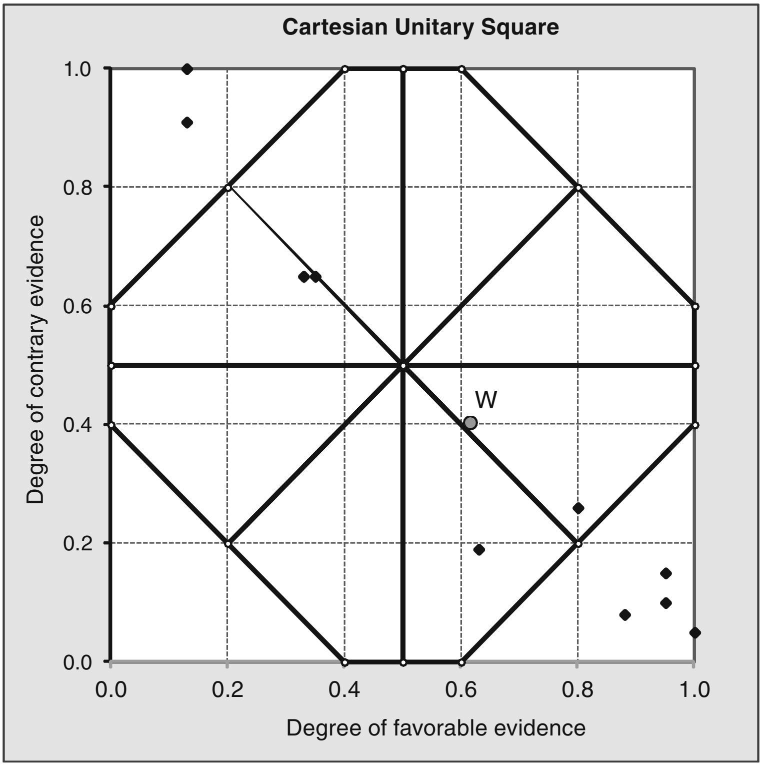


Fig. 5 Analysis of the results by para-analyzer algorithm

However, the collective influences of all unviability factors in launching product X in region Y can be summarized by point **W**. This is the barycenter of the ten points and translates the combined influence of the ten analyzed factors. Since **W** is in the semi-truth region tending to inconsistency, we can say that the analysis result is inconclusive. That is, the analysis does not recommend the launching of product X in region Y, but it does not say otherwise either. It simply suggests that new surveys should be conducted in an attempt to increase the evidences.

The analysis of influence for each factor in relation to the product viability performed by the para-analyzer algorithm, can be done numerically by calculating the resulting degree of certainty, $H_i = \boldsymbol{a}_{i,R} - \boldsymbol{b}_{i,R}$ for each of the factors and by the application of the rule of decision (columns 17–19 from Table 5). The influence of all factors combined can be analyzed likewise. The only thing to do is to calculate the barycenter's degree of certainty **W**, $H_W = \boldsymbol{a}_W - \boldsymbol{b}_W$.

Since ($H_W = \boldsymbol{a}_W - \boldsymbol{b}_W = 0.211$) and ($-0.60 < 0.211 < 0.60$), the result is **inconclusive,** that is, it is not possible to assume the viability of product X launch in region Y, nor its unviability.

It is important to notice that once the survey is conducted, i.e., since column 2 of Table 5 has been filled out, all other operations translated by columns 3–19 can be automatically performed by a small computer program, based on Excel.

In order to perform a fidelity test of the method and exercise its application, we would suggest that the reader conduct a viability analysis to launch a product X′ in a Y′ region, assuming that in field research, all the factors fell into section S_1, in other words, all the factors were highly in favor of the launching of product X′ in region Y′. In this case, evidently, it is expected a highly favorable viability analysis for product X′ in region Y′.

In fact, by applying the PDM to this case (and this is the expected drill) we have $\boldsymbol{a}_W = 0.93$ and $\boldsymbol{b}_W = 0.09$. This enables the calculation $H_W = \boldsymbol{a}_W - \boldsymbol{b}_W = 0.93 - 0.09 = 0.84$. Since $0.84 \geq 0.60$, the rule of decision affirms the **viability** for product X′ launch in region Y′ (Fig. 6).

On the contrary, if all factors are in section S_5, by the PDM we have $\boldsymbol{a}_W = 0.15$ and $\boldsymbol{b}_W = 0.90$ (please verify this result as an exercise). This leads to the calculation $H_W = \boldsymbol{a}_W - \boldsymbol{b}_W = 0.15 - 0.90 = -0.75$. Since $-0.75 \leq -0.60$, the rule of decision is claiming the **unviability** of product X″ launch in region Y″ (Fig. 7).

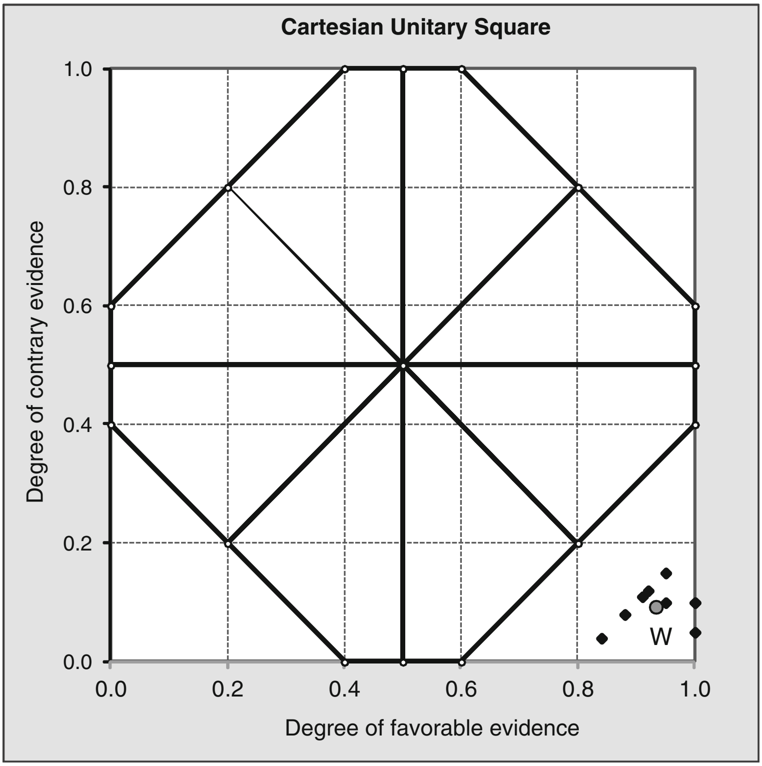


Fig. 6 All factors are highly favorable

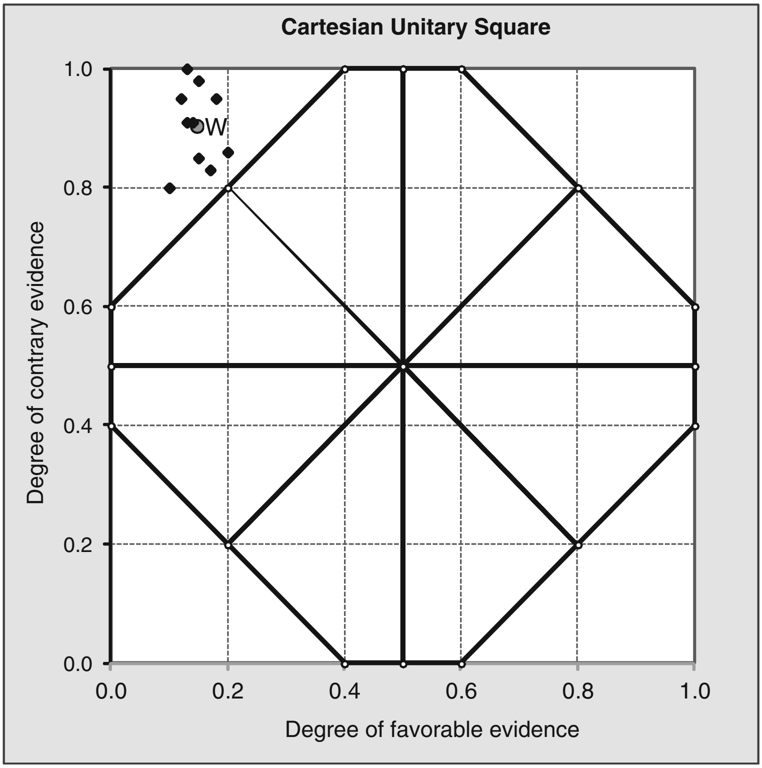


Fig. 7 All factors are highly unfavorable

3 Decision Method (PDM) and Statistical Decision Method (SDM): A Comparison [25, 26]

3.1 An Application of the Rule of Decision to Enable the Comparison

In order to do the comparison, the rule of decision (or para-analyzer algorithm) will be applied in a hypothetical case. To make it simple, we have picked enterprise Ω in which only ten factors (F_{01} to F_{10}) have significant influence [20]. We will assume that the opinions of four specialists (E_k) have been collected and that in order to apply the rules of maximization (**MAX**) and minimization (**MIN**), they have been grouped as: Group A: ($E_1 + E_2$) and Group B: ($E_3 + E_4$).

Therefore, the application scheme of Operators **MAX** and **MIN** is [27]:

$$[(E_1)\ \mathbf{MAX}\ (E_2)]\ \mathbf{MIN}\ [(E_3)\ \mathbf{MAX}\ (E_4)] \text{ or}$$
$$\mathbf{MIN}\ \{\mathbf{MAX}\ [(E_1), (E_2)], \mathbf{MAX}\ [(E_3), (E_4)]\}$$

For decision making, was choose a level of requirement equal to 0.70. So, the rule of decision is:

$$H \geq 0.70 \Rightarrow \text{favorable decision (viable enterprise);}$$
$$H \leq -0.70 \Rightarrow \text{unfavorable decision (unviable enterprise);}$$
$$-0.70 < H < 0.70 \Rightarrow \text{inconclusive analysis.}$$

Table 6 shows in columns 2–9 the degrees of favorable and contrary evidence that the specialists attributed to the factors; in columns 10–13, the results of the application of the rule of maximization (**MAX**) intra-groups; in columns 14 and 15, the degrees of favorable evidence ($\boldsymbol{a}_R$) and contrary evidence ($\boldsymbol{b}_R$) resulting from the application of the minimization rule (**MIN**) within the groups; and in columns 16–18, the analysis of results.

3.2 Analysis of Results

There are eight factors in the region of truth and two in the region of quasi-truth, as you can see in Table 6 and in Fig. 8.

Since ($H_W = \boldsymbol{a}_W - \boldsymbol{b}_W = 0.775$) and ($0.775 \geq 0.70$), the result is favorable, that is, it is possible to affirm the viability of the enterprise.

3.3 A Short Revision of the Statistical Decision Method (SDM)

Statistical decisions are decisions made concerning a specific population based on data gathered from its sample(s). For example, you are interested in determining the fairness of a coin, or comparing the efficiency of one drug over another in curing an illness, etc.

Statistical hypotheses about the population in question are formulated in an attempt to arrive at a decision. They constitute affirmations about the probability distributions of the population. Usually, a statistical hypothesis is formulated with the intention to be rejected [42].

So, to discover if a coin is faulty, one must formulate the hypothesis that it is not faulty, i.e., that the probability to obtain one of the faces (heads, for example) is $p = 0.5$. This is called the null hypothesis (H_0: the coin is fair). Any other

Table 6 PDM calculations table

	Group A				Group B				A	
	E_1		E_2		E_3		E_4		E_1 MAX E_2	
F_i	$a_{i,1}$	$b_{i,1}$	$a_{i,2}$	$b_{i,2}$	$a_{i,3}$	$b_{i,3}$	$a_{i,4}$	$b_{i,4}$	$a_{i,gA}$	$b_{i,gA}$
F_{01}	0.88	0.04	0.94	0.14	0.84	0.08	0.78	0.03	0.94	0.04
F_{02}	1.00	0.05	0.95	0.15	1.00	0.10	0.85	0.00	1.00	0.05
F_{03}	0.92	0.08	0.98	0.18	0.88	0.12	0.82	0.07	0.98	0.08
F_{04}	0.95	0.11	1.00	0.21	0.91	0.15	0.85	0.10	1.00	0.11
F_{05}	0.70	0.20	0.80	0.30	0.80	0.20	0.70	0.30	0.80	0.20
F_{06}	0.80	0.30	0.80	0.20	0.70	0.30	0.70	0.20	0.80	0.20
F_{07}	0.95	0.15	1.00	0.10	0.85	0.00	1.00	0.05	1.00	0.10
F_{08}	0.98	0.18	0.88	0.12	0.82	0.07	0.92	0.08	0.98	0.12
F_{09}	1.00	0.21	0.91	0.15	0.85	0.10	0.95	0.11	1.00	0.15
F_{10}	0.94	0.14	0.84	0.08	0.78	0.03	0.88	0.04	0.94	0.08

	A		B		A MIN B		Level of requirement = 0.600		
	E_1 MAX E_2		E_3 MAX E_4				Conclusions		
F_i	$a_{i,gA}$	$b_{i,gA}$	$a_{i,gB}$	$b_{i,gB}$	$a_{i,R}$	$b_{i,R}$	H	G	Decision
F_{01}	0.94	0.04	0.84	0.03	0.84	0.04	0.80	−0.12	Viable
F_{02}	1.00	0.05	1.00	0.00	1.00	0.05	0.95	0.05	Viable
F_{03}	0.98	0.08	0.88	0.07	0.88	0.08	0.80	−0.04	Viable
F_{04}	1.00	0.11	0.91	0.10	0.91	0.11	0.80	0.02	Viable
F_{05}	0.80	0.20	0.80	0.20	0.80	0.20	0.60	0.00	Inconclusive
F_{06}	0.80	0.20	0.70	0.20	0.70	0.20	0.50	−0.10	Inconclusive
F_{07}	1.00	0.10	1.00	0.00	1.00	0.10	0.90	0.10	Viable
F_{08}	0.98	0.12	0.92	0.07	0.92	0.12	0.80	0.04	Viable
F_{09}	1.00	0.15	0.95	0.10	0.95	0.15	0.80	0.01	Viable
F_{10}	0.94	0.08	0.88	0.03	0.88	0.08	0.80	−0.04	Viable
Baricenter W: weighted averages of the resultant degrees					0.888	0.113	0.775	0.001	Viable

hypothesis different from the null is called the alternative hypothesis (H_1: p ≠ 0.5, the coin is not fair) [7].

In practice, H_0 is accepted, and based on a random sample together with probability theory, one shall determine if the sampled results are very different from the expected, that is, if the observed difference is significant enough to reject H_0 and thereby accepting H_1.

For instance, in tossing a coin approximately 50 times, 25 heads are expected to be obtained; however, if 40 heads are attained, then there's an inclination to reject the hypothesis H_0 that the coin is fair (and accept the alternative hypothesis H_1).The process that allows us to decide upon rejecting a hypothesis by determining if the

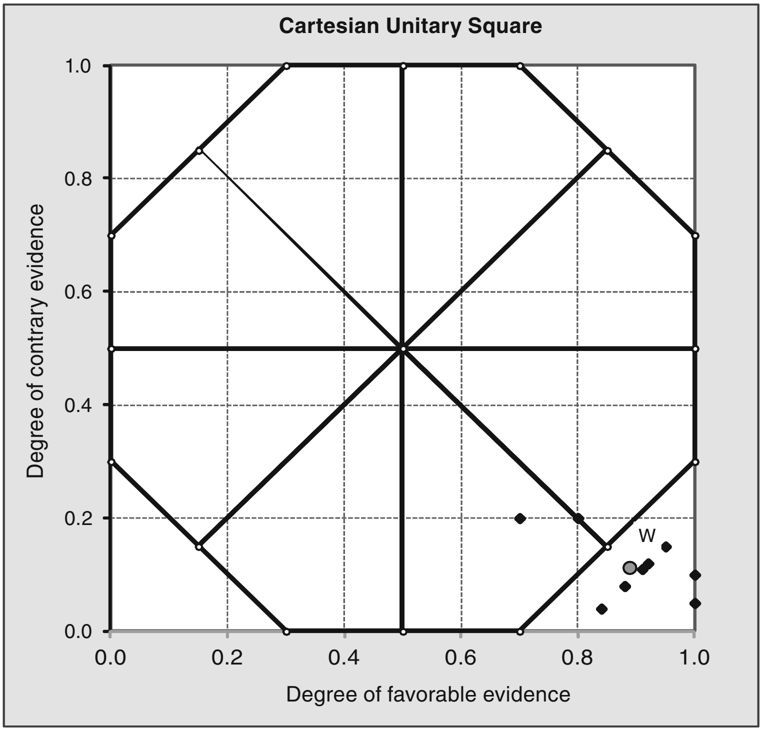


Fig. 8 Analysis of result by the para-analyzer algorithm

sampled data is significantly different from the expected, is called hypothesis testing or test of significance [5].

If H_0 is rejected when it should be accepted, one may say that a type I error has occurred; but, if it is accepted when it should have been rejected, the error is type II [42]. In both situations there is an error of decision. The use of larger samples, which is not often possible, can help reduce the chance of these errors from occurring.

In testing an established hypothesis, H_0, the maximum probability to commit a type I error is called the level of significance, often represented by α, for which the most common values are 0.05 (or 5 %) and 0.01 (or 1 %).

So, if α is set at 5 % in planning the hypothesis test, then there is a 5 in 100 chance that H_0 will be rejected when in fact it should be accepted, that is, there is a 95 % confidence of making the right decision and so one can say that H_0 is rejected at the 0.05 (or 5 %) level of significance. In the example of the coin, one should say that there are evidences that the coin is not fair, in the level of significance 0.05 (or 5 %).

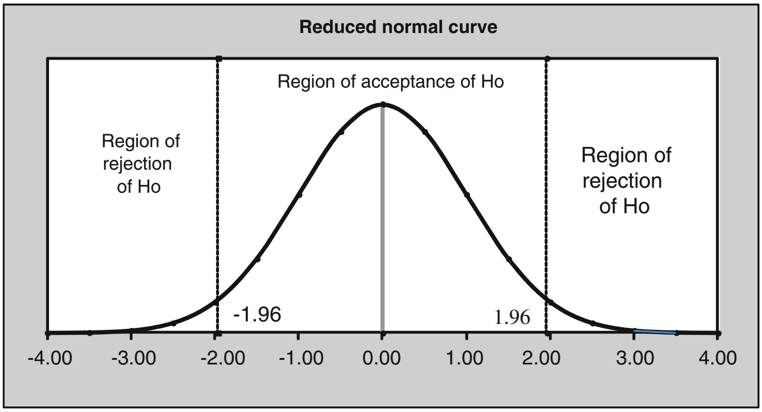


Fig. 9 Regions of acceptance and rejection in a normal curve

If a variable X has a normal distribution with mean μ_X and standard deviation σ_X, then the reduced variable distribution (or standard score) [$z = (X - \mu_X)/\sigma_X$] is normal with mean 0 and standard deviation 1 [31, 33, 42].

For the 5 % level of significance, the critical values z (z_c), which separate the region of acceptance of H_0 from the region of rejection of H_0, are –1.96 and +1.96 (Fig. 9). So, if the value of X_0 of the variable X observed in the sample leads to a score z_0 less than or equal to –1.96, or greater than or equal to +1.96, then H_0 will be rejected at the 5 % level of significance. In this case, one can say that z_0 is significantly different from 0 (mean of z) to allow rejection of H_0 at the 5 % level of significance. Therefore, for this level of significance, the statistical decision rule is:

To accept H_0: if $\mathbf{-1.96 < z_0 < +1.96}$ or, in a more generic way,

$$\text{if } -\mathbf{z_c} < \mathbf{z_0} < +\mathbf{z_c};$$

To reject H_0: if $\mathbf{z_0 \leq -1.96}$ **or** $\mathbf{z_0 \geq +1.96}$ or, in a more generic way,

$$\text{if } \mathbf{z_0} \leq -\mathbf{z_c} \ \mathbf{or} \ \mathbf{z_0} \geq +\mathbf{z_c}.$$

At the 1 % level of significance, the critical values of z are –2.58 and +2.58 (for two-tail tests).

3.4 The PDM and Normal Distributions

In order to compare the Paraconsistent Decision Method (PDM) with the Statistical Decision Method (SDM), a few considerations in relation to PDM have been made.

Table 7 Classes, observed (PDM) and expected (Normal) frequencies, χ^2 (chi-square) calculation and accumulated areas under the PDM and NAC curves, with standard deviation = 0.444

1	2	3	4	5	6	7	8	9
Level of requirement (K)	Class	Middle point (M)	A_M	$f_H = f_O$	$f_N = f_E$	$(f_O - f_E)^2/f_E$	A_{acum} PDM	A_{acum} NAC
0.9	−1.0 ⊣ −0.9	−0.95	0.005	0.05	0.091	0.01853	0.005	0.021
0.8	−0.9 ⊣ −0.8	−0.85	0.015	0.15	0.144	0.00027	0.020	0.035
0.7	−0.8 ⊣ −0.7	−0.75	0.025	0.25	0.216	0.00544	0.045	0.057
0.6	−0.7 ⊣ −0.6	−0.65	0.035	0.35	0.308	0.00581	0.080	0.087
0.5	−0.6 ⊣ −0.5	−0.55	0.045	0.45	0.417	0.00258	0.125	0.129
0.4	−0.5 ⊣ −0.4	−0.45	0.055	0.55	0.538	0.00029	0.180	0.183
0.3	−0.4 ⊣ −0.3	−0.35	0.065	0.65	0.659	0.00011	0.245	0.249
0.2	−0.3 ⊣ −0.2	−0.25	0.075	0.75	0.767	0.00037	0.320	0.325
0.1	−0.2 ⊣ −0.1	−0.15	0.085	0.85	0.849	0.00000	0.405	0.410
0	−0.1 ⊣ 0.0	−0.05	0.095	0.95	0.893	0.00366	0.500	0.499
0	0.0 ⊢ 0.1	0.05	0.095	0.95	0.893	0.00366	0.595	0.589
0.1	0.1 ⊢ 0.2	0.15	0.085	0.85	0.849	0.00000	0.680	0.674
0.2	0.2 ⊢ 0.3	0.25	0.075	0.75	0.767	0.00037	0.755	0.750
0.3	0.3 ⊢ 0.4	0.35	0.065	0.65	0.659	0.00011	0.820	0.816
0.4	0.4 ⊢ 0.5	0.45	0.055	0.55	0.538	0.00029	0.875	0.870
0.5	0.5 ⊢ 0.6	0.55	0.045	0.45	0.417	0.00258	0.920	0.912
0.6	0.6 ⊢ 0.7	0.65	0.035	0.35	0.308	0.00581	0.955	0.942
0.7	0.7 ⊢ 0.8	0.75	0.025	0.25	0.216	0.00544	0.980	0.964
0.8	0.8 ⊢ 0.9	0.85	0.015	0.15	0.144	0.00027	0.995	0.978
0.9	0.9 ⊢ 1.0	0.95	0.005	0.05	0.091	0.01853	1.000	0.987
					$\chi2 = 0.07412$			

(a) The variation interval of the degree of certainty ($-1 \le H \le 1$) has been divided into classes with amplitude a = 0.1, with extremes on whole decimal values of H (0.0×10^{-1}, $\pm 1.0 \times 10^{-1}$, $\pm 2.0 \times 10^{-1}$, …) (column 2, Table 7). Therefore, the midpoints of the classes are: $\pm 0.5 \times 10^{-1} = \pm 0.05$, $\pm 1.5 \times 10^{-1} = \pm 0.15$, $\pm 2.5 \times 10^{-1} = \pm 0.25$, …, $\pm 9.5 \times 10^{-1} = \pm 0.95$ (Column 3, Table 7). To each class a value of the level of requirement (K) is associated (column 1, Table 7).

(b) If H = M is the middle point of one class, then its extremes are M – 0.05 and M + 0.05 (Column 2, Table 7). So, this class will be defined by the interval $K = M - 0.05 \le H < M + 0.05$, for $H \ge 0$, or $M - 0.05 < H \le M + 0.05 = K$, for $H < 0$, where K is the corresponding level of requirement (Fig. 10).

(c) For each class, the area of the defined (demarcated) CUS region was calculated (Fig. 10). It was called the class area and its value $A_M = 0.1 \times (1 - |M|)$ was obtained.

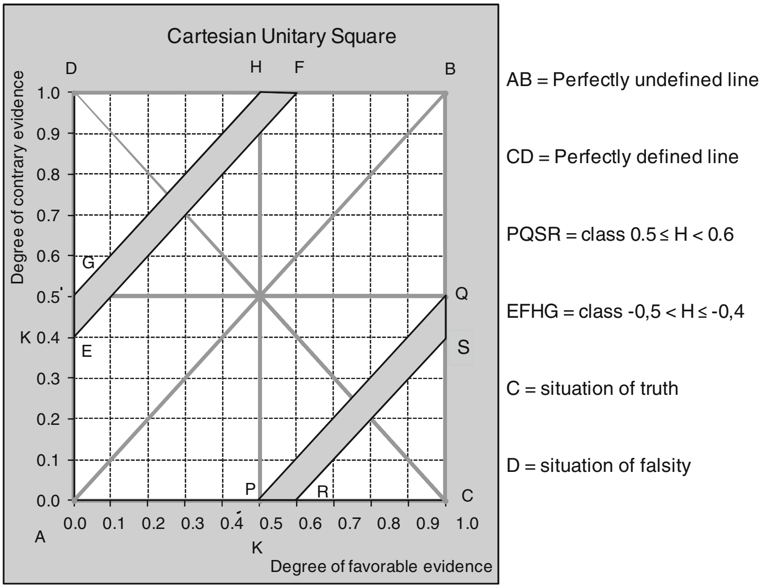


Fig. 10 Classes of the degree of certainty, emphasizing two ones: levels of requirement 0.5 (PQRS) and 0.4 (EFGH)

(d) Since the CUS area is equal to 1, the frequency of the class defined by the value H = M (center of class) is equal to the class area (A_M) divided by its amplitude (a).

Therefore: $f_{(H\,=\,M)} = A_M/a = 0.1 \times (1 - |M|)/0.1 = 1 - |M|$.

(e) So, it is possible to calculate areas (A_M) and frequencies (f_M) of all classes (columns 4 and 5, Table 7) and produce the corresponding frequencies diagram (Fig. 11).

(f) One level of requirement LR = K is adopted for decision making by PDM. This implies that the decision will be favorable if $H_W \geq K$ and unfavorable, if $H_W \leq -K$, H_W being the degree of certainty of the barycenter.

The decision will be favorable if the barycenter W belongs to the CUS region defined by condition $H \geq K$, that is, if it belongs to the tail-end of the curve formed by the classes of middle points M so that $M \geq K + 0.05$ or $|M| \geq K + 0.05$.

The decision will be unfavorable if the barycenter W belongs to the CUS region defined by condition $H \leq -K$, that is, if it belongs to the curve tail formed by the classes of middle points M so that $M \leq -K - 0.05$ or $|M| \geq K + 0.05$.

Therefore, if the barycenter W should belong to one of the tail-ends of the curve (right or left) of the distribution of H frequencies defined by the level of requirement LR = K, then it means that the degree of certainty of the barycenter is significantly different from zero so that one can make a decision (favorable or unfavorable).

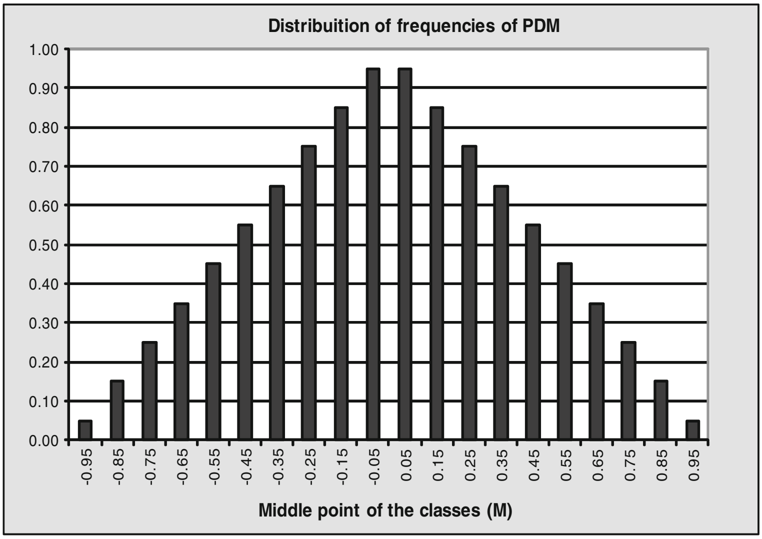


Fig. 11 Distribution of frequencies obtained by the PDM

In order to perform the comparison using the statistical method of decision (SMD), we have looked at the normal distribution of mean equal to zero (since distribution of H has mean equal to zero as well) that better adheres to the distribution of H frequency (of the PDM).

To measure this adherence, a χ^2 (chi-squared) test was applied. To do so, the frequency of each class of the degree of certainty ($f_O = f_H$) (column 5, Table 7, Fig. 11) was considered as observed frequency, while the frequency of the same class obtained by the normal curve ($f_E = f_N$) (column 6, Table 7, Fig. 12) was considered as the expected frequency. This frequency was obtained with the help of an Excel table by using the function DIST.NORM(X; SMDIA; DESVPAD; FALSE).

We have found out that the best adherence of the normal distribution of mean zero to the distribution of degree of certainty of the PDM occurs at a standard deviation equal to 0.444, for which the chi-squared is minimum and equal to $\chi^2 = 0.07412$ (Table 8, result from Column 7, Table 7, Figs. 13 and 14). This was called the normal adherent curve (NAC).

In these conditions, decision by PDM with a level of requirement equal to K (favorable if $H_W \geq K$, or unfavorable if $H_W \leq -K$) corresponds to a decision by SDM with a level of significance equal to the area under NAC, above K (favorable decision) or below –K (unfavorable decision) (See Table 8, Fig. 15).

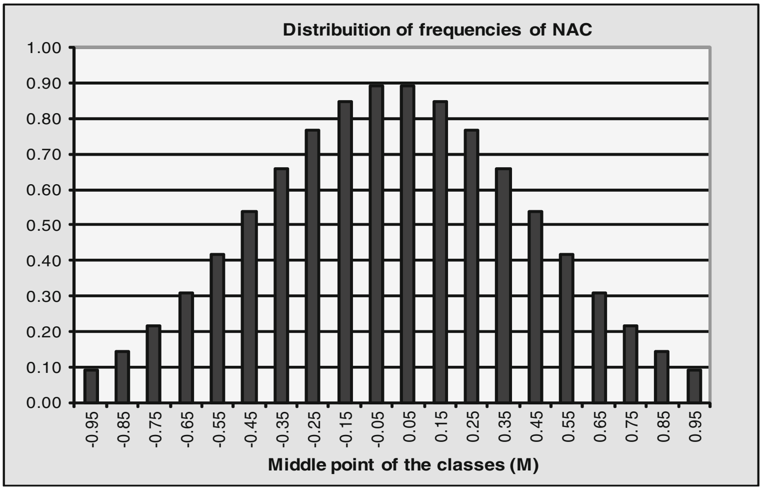


Fig. 12 Distribution of frequencies obtained by NAC curve (average zero e standard deviation 0.444)

Table 8 Comparison between the areas of tail distribution of PDM and NAC curves and variation of χ^2 value for some values of the standard deviation

Level of requirement	Level of uncertainty	Level of significance	Standard deviation	χ2 (chi-square)
Minimum value acceptable of the degree of certainty	Tail of PDM curve	Tail of NAC curve	0.437	0.07683
			0.438	0.07607
			0.439	0.07545
K	β (%)	λ (%)	0.44	0.07494
0	50.00	50.00	0.441	0.07456
0.1	40.50	41.07	0.442	0.07429
0.2	32.00	32.59	0.443	0.07415
0.3	24.50	24.92	0.444	0.07412
0.4	18.00	18.33	0.445	0.07420
0.5	12.50	12.96	0.446	0.07440
0.6	8.00	8.78	0.447	0.07470
0.7	4.50	5.71	0.448	0.07511
0.8	2.00	3.55	0.449	0.07563
0.9	0.50	2.11	0.450	0.07625

3.5 Comparison of PDM with SDM

For the normal curve, the area of each class has been calculated by the product of its frequency (column 6, Table 7) with the amplitude of classes (a = 0.1). The accumulated distribution areas of PDM and Normal curves (columns 8 and 9, Table 7)

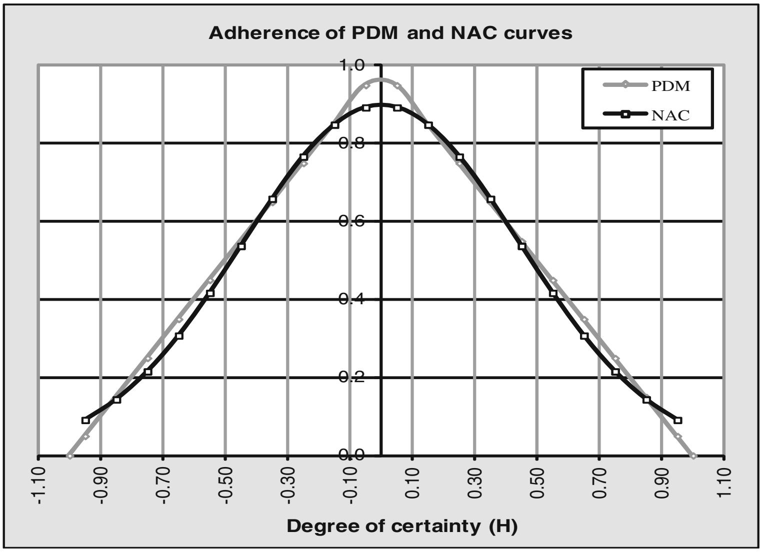


Fig. 13 PDM and NAC distribution curves

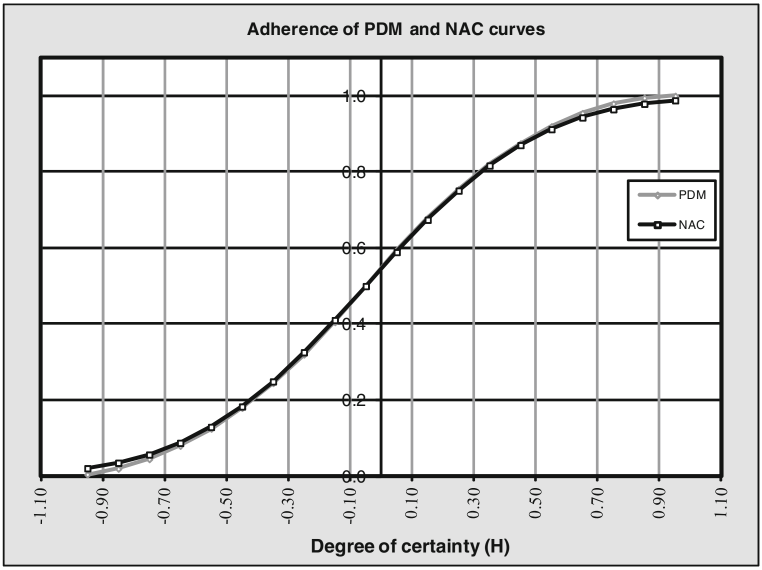


Fig. 14 Accumulated areas of PDM and NAC distribution curves

were obtained by the accumulated sum of the areas of the classes. In this calculation for the normal curve a correction was made corresponding to the area under the curve up to the value −1.0.

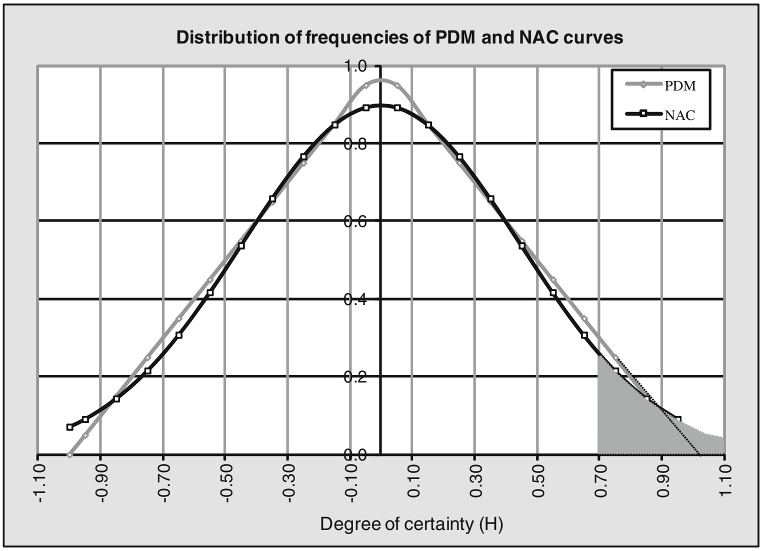


Fig. 15 Tail of normal curve with the best adherence to PDM curve (NAC)

The area in the tail-end (α) of the normal curve is called the level of significance and represents the uncertainty with which one can accept that the result obtained (H_W) is sufficiently different from zero (mean of H) in order to say that the enterprise is viable (favorable decision) or unviable (unfavorable decision).

Similarly, the PDM's tail-end of the curve that will be called the level of uncertainty (β), represents the area of the CUS region (here a triangle) to which $H \geq K$ or $H \leq -K$. So, when we state that a decision has been made by the PDM with a level of requirement, it means that the degree of certainty of the barycenter is, in module, greater than or equal to the level of requirement or that the decision displays a level of uncertainty β.

As seen before, in order to make a decision with PDM, it is necessary to calculate the degree of certainty of the barycenter (H_W) and compare it with the level of requirement. The example shows that $H_W = 0.775$ is compared with the level of requirement LR = 0.70. Since $H_W \geq$ LR, the decision is favorable (the enterprise is viable) to the level of requirement 0.70, that is, it is possible to say that the enterprise is viable with a maximum level of uncertainty β = 4.50 % (See Table 8).

In order to make a decision using the statistical process, it is necessary to calculate:

(a) the critical value of the standard variable of the normal adherent curve NAC ($*z_c$) that corresponds to the chosen level of requirement (0.70, in the example). To do so, it is necessary to check how many standard deviations of the NAC (0.444) the level of requirement is above the mean (zero), as follows:

$$^{*}z_c = (0.70 - 0)/0.444 = 1.58;$$

(b) the observed value of the standard variable of NAC ($*z_o$) that corresponds to the degree of certainty of the barycenter (0.775, in the example). To do so, it is necessary to check how many standard deviations of the NAC (0.444) the degree of certainty of the barycenter is above the average (zero), as follows:

$$^{*}z_o = (0.775 - 0)/0.444 = 1.75;$$

(c) Since $*z_o \geq *z_c$, it is clear that the value H_w is significantly larger than the mean zero, leading to the conclusion of the analysis as being favorable (the enterprise is viable) at a level of significance 5.71 % (See Table 8).

Note: In Table 8 we noticed that if the level of requirement adopted by the PDM is 0.60, then the degree of uncertainty of the PDM will be 8.00 % and the level of significance of the SMD will be 8.78 %; similarly, if it's 0.80, then these values will be 2.00 and 3.55 %, respectively.

3.6 Conclusions

During the development of the PDM and its comparison with the SDM, it was observed that they are similar in many aspects. For some aspects, the PDM seems to be more advantageous while for others, the SDM is the more advantageous.

Since it uses techniques of paraconsistent annotated evidential logic Eτ, the PDM represents a valuable and original tool in the process of decision making capable of dealing with uncertain and contradictory data—without being trivial—and without collapsing. Usually this feature is not present in classical decision processes such as the SDM, which are based on classical logic.

The PDM offers results that equally indicate if the survey displays viability (truth) or unviability (falsity) of the analyzed enterprise or even if the result is not conclusive, thus recommending a further and more accurate analysis. The SDM serves this purpose with the same efficiency.

Furthermore, judging the position of the representative points of the factors of influence and the barycenter in the lattice (τ), the PDM indicates the level of contradiction displayed by the data in relation to each factor—or all of them—put altogether.

Therefore, going further from the SDM, the PDM can state, for example, if there is any contradiction between the data used and if such contraction is emphasized or not. It also indicates if the contradiction shown constitutes inconsistency or para-completeness (lack of data). Therefore, not only does it accepts contradictory data, but it also points out the degree of contradiction of this data and, more importantly, it's possible for such data to be manipulated and utilized despite being contradictory.

The PDM offers the important possibility of qualitative analyses of balance sheets, investments, etc. to be transformed into quantitative analyses, which are more accurate and useful for professionals of those areas and are also easier to be handled in computational processes.

This is achievable because the PDM deals with degrees of evidence, which despite being objective, numbers translate subjective features—experts' opinions resulting from experience, knowledge and sensitivity accumulated throughout the years. Such subjectivity—although offering the PDM more opportunities, can be understood as a sore point in relation to the SDM that uses purely objective data.

In PDM the opinions of the experts are collected once, then stored in a database that may be used in many decision making instances. With this, and without any additional costs, it is possible to use high level experts and make their wise opinions last forever.

Another advantage common to the PDM and SDM is versatility. It is possible to make PDM more accurate and reliable in many ways such as using a larger number of factors of influence, or establishing more than three sections for each factor, increasing the requirement level and collecting the opinions of a larger number of experts to build the database, etc.

One of the greatest advantages of the PDM over SDM is that the first can only compare levels of evidence without having to operate on them. This is crucial considering that the evidence degrees are variables that get only to the ordinal level and therefore, they could not be applied to the SDM, which requires variables at reasoning levels. However, as it has been shown in item 13, application of the SDM with the PDM database can lead to significant coherent results. There are a fuzzy method of decision and its comparison with the statiscal method [26, 38, 43,44, 46]

References

1. Abe, J.M.: Fundamentos da Lógica Anotada. Doctor Thesis. In Portuguese. Tese apresentada para a obtenção do título de Doutor em Filosofia na Faculdade de Filosofia, Letras e Ciências Humanas (FFLCH) da Universidade de São Paulo (USP), 98pp. São Paulo (1992)
2. Abe, J.M.: Some Aspects of paraconsistent systems and applications. Logique et Anal. **157**, 83–96 (1997)
3. Abe, J.M., Da Silva Filho, J.I., Scalzitti A.: Introdução à Lógica para Ciência da Computação, 247pp. In Portuguese. Editora Arte e Ciência, São Paulo (2001)
4. Ávila, B.C.: Uma Abordagem Paraconsistente Baseada em Lógica Evidencial para Tratar Exceções em Sistemas de Frames com Múltipla Herança. Doctor Thesis. In Portuguese. Tese apresentada para a obtenção do título de Doutor em Engenharia, no Departamento de Engenharia de Computação e Sistemas Digitais da Escola Politécnica da Universidade de São Paulo (EPUSP), 120pp. São Paulo, Brasil, (1996)
5. Berger, J.O.: Statistical Decision Theory and Bayesian Analysis. In: Springer Series in Statistics, 2nd edn, 617pp. Springer, New York (1983)
6. Chalos, P.: Managing Cost in Today's Manufacturing Environment, 283pp. Prentice Hall Inc., Englewood Cliffs (1992)
7. Costa Neto, P.L.O.: Estatística. 2ª Edição, 266pp. In Portuguese. Editora Edgard Blücher Ltda, São Paulo (2002)

8. Costa Neto, P.L.O., Bekman, O.R.: Análise Estatística da Decisão. 2ª. Ed, 148pp. In Portuguese. Editora Blucher, São Paulo (2009)
9. Da Costa, N.C.A., Vago, C., Subrahmanian, V.S.: The paraconsistent Logics $P\tau$, in Zeitschr. f. math. Logik und Grundlagen d. Math, Bd. **37**, 139–148 (1991)
10. Da Costa, N.C.A., Abe, J.M., Subrahmanian, V.S.: Remarks on annotated logic, Zeitschrift f. math. Logik und Grundlagen d. Math. **37**, 561–570 (1991)
11. Da Costa, N.C.A.: Ensaio Sobre os Fundamentos da Lógica. In Portuguese. Hucitec-Edusp, São Paulo (1980)
12. Da Costa, N.C.A.: Sistemas Formais inconsistentes, 66pp. In Portuguese. Editora da Universidade Federal do Paraná (UFPR), Curitiba (1993)
13. Da Costa, N.C.A., Marconi, D.: An overview of paraconsistent logic in the 80's. J. Non-Class. Log. **6**(1), 5–32 (1989)
14. Da Costa, N.C.A., Abe, J.M., Murolo, A.C., Da Silva Filho, J.I., Leite, C.F.S.: Lógica Paraconsistente Aplicada, 214pp. In Portuguese. Editora Atlas S.A., São Paulo (1999)
15. Da Silva Filho, J.I.:: Métodos de Aplicações da Lógica Paraconsistente Anotada de Anotação com dois Valores—LPA2v com Construção de Algoritmo e Implementação de Circuitos Eletrônicos. In Portuguese. Tese apresentada para a obtenção do título de Doutor em Engenharia, no Departamento de Engenharia de Computação e Sistemas Digitais da Escola Politécnica da Universidade de São Paulo (EPUSP), 185pp. São Paulo, Brasil (1998)
16. Da Silva Filho, J.I., Abe, J.M.: Paraconsistent analyzer module. Int. J. Comput. Anticip. Syst. **9**, 346–352. ISSN 1373–5411, ISBN 2-9600262-1-7 (2011)
17. De Carvalho, F.R.: Lógica Paraconsistente Aplicada em Tomadas de Decisão: uma abordagem para a administração de universidades (Paraconsistent Logic Applied in Decision Making: an approach to Universities management), 120p. In Portuguese. Editora Aleph, São Paulo, Brasil (2002)
18. De Carvalho, F.R., Brunstein, I., Abe, J.M.: Tomadas de Decisão com Ferramentas da Lógica Paraconsistente Anotada. In Portuguese. In: José Luís D.R., Nivalda, L.C., de Souza, L.G.M., Silva, G.P. (eds.) Encontro Nacional de Engenharia de Produção, 23º, Proceedings, pp. 1–8. Ouro Preto, MG, Brasil, 21 a 24 de outubro de 2003
19. De Carvalho, F.R., Brunstein, I., Abe, J.M.: Um Estudo de Tomada de Decisão Baseado em Lógica Paraconsistente Anotada: Avaliação do Projeto de uma Fábrica. In Portuguese. In: Revista Pesquisa & Desenvolvimento Engenharia de Produção, da Universidade Federal de Itajubá, Edição nº 1, dez, pp. 47–62 (2003)
20. De Carvalho, F.R., Brunstein, I., Abe, J.M.: Paraconsistent annotated logic in analysis of viability: an approach to product launching. In: Daniel, M.D. (ed.) Sixth International Conference on Computing Anticipatory Systems (CASYS-2003). Published by American Institute of Physics, AIP Conference Proceedings, vol. 718, pp. 282–291. ISBN 0-7354-0198-5, ISSN 0094-243X
21. De Carvalho, F.R., Brunstein, I., Abe, J.M.: Decision making based on paraconsistent annotated logic. In: Nakamatsu, K., Abe, J.M. (eds.) Congress of Logic Applied to Technology (LAPTEC 2005), 5th. Himeji, Japan. Advances in Logic Based Intelligent Systems: Frontiers in Artificial Intelligence and Applications (Selected papers), pp. 55–62. IOS Press, Amsterdam, ISBN 1-58603-568-1, ISSN 0922-6389, 2–4 April 2005
22. De Carvalho, F.R., Brunstein, I., Abe, J.M.: Prevision of medical diagnosis based on paraconsistent annotated logic. In: Daniel, M.D. (ed.) Seventh International Conference on Computing Anticipatory Systems (CASYS-2005). International Journal of Computing Anticipatory Systems, vol .18, pp. 288–297. ISBN 2-930396-04-0, ISSN: 1373-5411. Liège, Belgium, 8–13 Aug 2005
23. De Carvalho, F.R.: Aplicação de Lógica Paraconsistente Anotada em Tomadas de Decisão na Engenharia de Produção. (Annotated Paraconsistent Logic Applied in Decision Making in Production Engineering). Doctor Thesis. In Portuguese. Thesis submitted to Escola Politécnica da Universidade de São Paulo, for PhD in Engineering, São Paulo, Brasil, 349pp (2006)
24. De Carvalho, F.R., Brunstein, I.: Aplicação de Lógica Paraconsistente Anotada em Tomadas de Decisão na Engenharia de Produção. In Portuguese. Boletim Técnico da Escola Politécnica

da USP—Departamento de Engenharia de Produção, BT/PRO/0704, ISSN 1413-2168, 19pp. São Paulo (2007)

25. De Carvalho, F.R., Brunstein, I., Abe, J.M.: Decision making based on paraconsistent annotated logic and statistical method: a comparison. In: Daniel M.D. (ed.) Eighth International Conference on Computing Anticipatory Systems (CASYS-2007). Published by American Institute of Physics, AIP Conference Proceedings, vol. 1.051, pp. 195–208. ISBN 978-0-7354-0579-0, ISSN: 0094-243X (2008)
26. De Carvalho, F.R., Abe, J.M.: A simplified version of the fuzzy decision method and its comparison with the paraconsistent decision method. In: Daniel M.D. (ed.) Ninth International Conference on Computing Anticipatory Systems (CASYS-2009), Published by American Institute of Physics, AIP Conference Proceedings, vol. 1.303, pp. 216–235. ISBN 978-0-7354-0858-6, ISSN: 0094-243X (2010)
27. De Carvalho, F.R., Abe, J.M.: *Tomadas* de Decisão com Ferramentas da Lógica Paraconsistente Anotada: Método Paraconsistente de Decisão (MPD), 213pp. In Portuguese. Editora Blucher, São Paulo (2011)
28. Gaither, N., Frazier, G.: Administração da Produção e Operações (Production and Operations Management), 8a. ed, 598pp. In Portuguese. Editora Pioneira Thomson Learning Ltda, São Paulo (2001)
29. Gurgel, F.A.: Administração do Produto. 2ª. Ed, 537pp. In Portuguese. Editora Atlas S.A, São Paulo (2001)
30. Hilbert, D., Ackermann, W.: Principles of Mathematical Logic, 2nd edn, 172pp. Chelsea Publishing Co. New York (1950)
31. Hoffmann, R.: Estatística para Economistas, 4ª. Ed, 432pp. In Portuguese. Editora Thomson, São Paulo (2006)
32. Kleene, S.C.: Introduction to Meta-mathematics, 550pp. North-Holland Publishing Co, Amsterdam, Netherlands (1952)
33. Lapponi, J.C.: Estatística usando Excel, 4ª. Ed, 476pp. In Portuguese. Editora Elsevier-Campos, Rio de Janeiro (2005)
34. Megginson, L.C., Mosley, D.C., Pietri, P.H. Jr.: Administração: Conceitos e Aplicações (Management: Concepts and Applications). In Portuguese. Translation of Maria Isabel Hopp, 4th edn, 614pp. Editora Harbra Ltda., São Paulo (1998)
35. Mendelson, E.: Introduction to Mathematical Logic, 4th edn, 440pp. Chapman & Hall, New York (1997)
36. Mortari, C.A.: Introdução à Lógica, 393pp. In Portuguese. Editora UNESP: Imprensa Oficial do Estado, São Paulo (2001)
37. Nakamatsu, K., Abe, J.M.: Railway signal and para-consistency, advances in logic based intelligent systems. In: Nakamatsu, K., Abe, J.M. (eds.) Congress of Logic Applied TO Technology (LAPTEC 2005), 5th. Himeji, Japan. Advances in Logic Based Intelligent Systems: Frontiers in Artificial Intelligence and Applications (Selected papers), pp. 220–224. IOS Press, Amsterdam, Netherlands, April 2–4 2005
38. Negoita, C.V., Ralescu, D.A.: Applications of Fuzzy Sets to Systems Analysis, 186pp. Wiley, New York (1975)
39. Porter, M.: Vantagem competitiva: Criando e sustentando um desempenho superior (Translated by Elizabeth Maria de Pinho Braga; technical review of Jorge A. Garcia Gomez), 512pp. In Portuguese. Editora Campus, Rio de Janeiro (1989)
40. Shimizu, T.: Decisões nas Organizações, 2ª Ed, 419pp. In Portuguese. Editora Atlas, São Paulo (2006)
41. Simon, H.A.: The New Science of Management Decision, 175pp. Englewood Cliffs, Prentice-Hall (1977)
42. Spiegel, M.R.: Estatística. 3ª Ed. Tradução: Pedro Consentino, 643pp. In Portuguese. Makron Books Editora Ltda, São Paulo (1993)
43. Witold, P (with foreword by Lotfi A. Zadeh).: Fuzzy Sets Engineering, 332pp. CRC Press Inc., Florida, ISBN 0-8493-9402-3 (1995)

44. Witold, P., Gomide, F.: An Introduction to Fuzzy Sets, 465pp. MIT Press, Massachusetts, ISBN 0-262-16171-0 (1998)
45. Woiler, S., Mathias, W.F.: Projetos: Planejamento, Elaboração e Análise (Projects: Planning, Ellaboration and Analysis), 294pp. Editora Atlas, São Paulo (1996)
46. Zadeh, L.A.: Outline of a new approach to the analysis of complex systems and decision processes. IEEE Trans. Syst. Mam Cybern. **SMC-3**(1), 28–44 (1973)

Paraconsistent Neurocomputing and Biological Signals Analysis

Jair Minoro Abe, Helder Frederico da Silva Lopes and Renato Anghinah

Abstract In this work, we show two applications of Paraconsistent Artificial Neural Network (PANN) for signal analysis working with signal data as a numeric vector and analyzing its morphology, comparing the signal data with a reference database and their application as support for electroencephalogram exams and HIV genotyping.

Keywords Artificial neural network · Paraconsistent logics · Annotated logics · Pattern recognition · HIV genotyping · Alzheimer disease

1 Introduction

Generally speaking, Artificial Neural Network (ANN) can be described as a complex computational system consisting of a set of highly interconnected processing elements called artificial neurons, which process information as a response to external stimuli. An artificial neuron is a simplistic representation that emulates biological neurons (although nowadays it has own directions) signal integration and threshold firing behavior by means of mathematical structures. ANNs have shown

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

H.F. da Silva Lopes
Faculty of Medicine—University of São Paulo, São Paulo, Brazil
e-mail: helderfredlopes@usp.com

R. Anghinah
Reference Center of Behavioral Disturbances and Dementia (CEREDIC) of Faculty of Medicine, University of São Paulo, São Paulo, Brazil
e-mail: anghinah@usp.br

J.M. Abe (ed.), *Paraconsistent Intelligent-Based Systems*,
Intelligent Systems Reference Library 94, DOI 10.1007/978-3-319-19722-7_11

useful to tackle problems human beings are good at solving, like prediction and pattern recognition. ANNs have been applied to several branches, among them, in the medical domain, for clinical diagnosis, image analysis and interpretation signal analysis and interpretation, and drug development.

In this work, we show two applications of ANN for signal analysis using the same technique. In short, this technique consists of work with signal data as a numeric vector and analyzing its morphology, comparing the signal data with a reference database.

Therefore, ANN constitutes an interesting tool for electroencephalogram (EEG) qualitative analysis. On the other hand, in EEG analysis, we are faced with imprecise, inconsistent and paracomplete data.

This is the case of HIV gene analysis, too. The analysis of HIV gene mutation require imprecise, inconsistent and paracomplete data treatment.

1.1 The Electroencephalogram

The electroencephalogram (EEG) is a brain electric signal activity register, resultant of the space-time representation of synchronic postsynaptic potentials. The graphic registration of the sign of EEG can be interpreted as voltage flotation with mixture of rhythms, being frequently sinusoidal, ranging from 1 to 70 Hz [21]. In the clinical-physiological practice, such frequencies are grouped in frequency bands as can be seen in Fig. 1.

To obtain the registration of EEG signals it is necessary to place the electrodes on the individual's scalp (Fp1, Fp2, Fz, F3, F4, F7, F8, C3, C4, Pz, P3, P4, T5, T6, O1 and O2) distributed as to cover the entire head, as shown in Fig. 2.

EEG analysis, as well as any other measurements devices, is limited and subjected to the inherent imprecision of the several sources involved: equipment, movement of the patient, electric registers, and individual variability of physician visual analysis. Such imprecision can often include conflicting information or paracomplete data. The majority of theories and techniques available are based on classical logic and so they cannot handle adequately such set of information, at least directly [2].

With the need to improve the EEG analysis, it had an evolution called quantitative electroencephalogram (EEGq). The quantitative EEG (EEGq) is a topographic survey—functional, distinct from computed tomography and magnetic

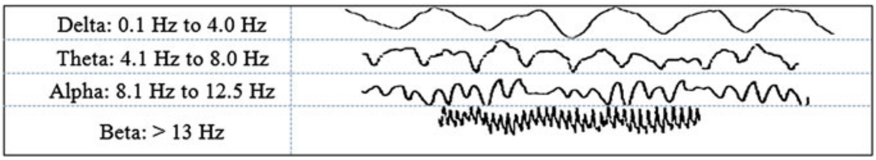


Fig. 1 Frequency bands clinically established and usually found in EEG

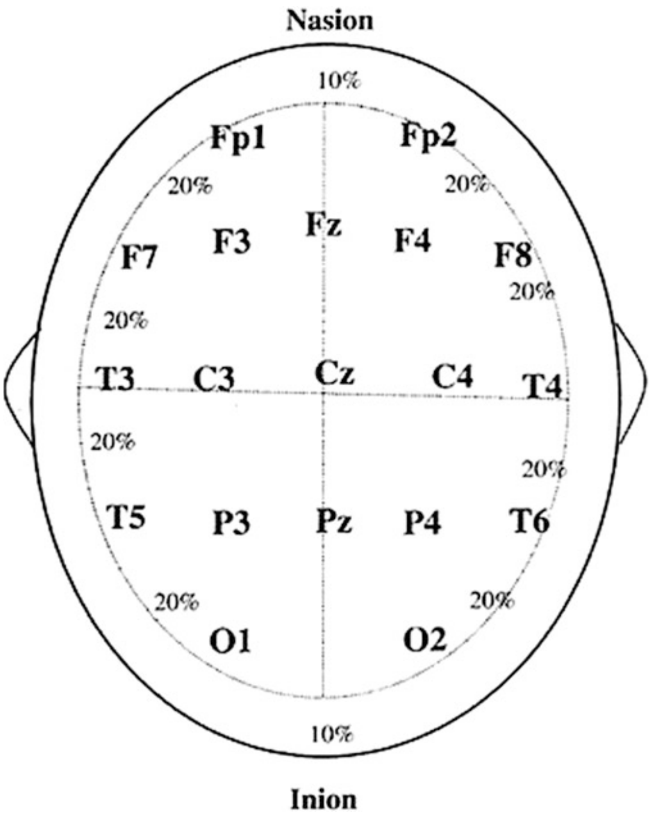


Fig. 2 System Diagram 10–20 for placing electrodes on the scalp. Odd numbers correspond to the electrodes of the left cerebral hemisphere. Even numbers correspond to the electrodes of the right cerebral hemisphere. The letter Z indicates the midline electrodes. The regions are defined by the letters: Phil—fronto-polar; F—front; C—Central; T—time; P—parietal; The—occipital

resonance imaging, which are tests of morphological or structural image, with its distinctive indication of these [24].

Two fundamental concepts in quantitative analysis of time series, particularly the EEG, are the concepts of analysis in the "time domain" and "frequency domain". When considering as independent variable time, as in a well characterized event that occurs at a particular point in time (for example, a paroxysm of spike-wave 2 min to record a EEG examination), the signal is considered in "time domain", that is, the very range at which the signal is represented by a graphic element or frequency versus the amplitude or power of this signal [3–5,14, 22].

The analysis in the frequency domain makes use of an important mathematical theorem, the Fourier theorem, which ensures that any periodic signal can be decomposed into a set of sine waves and cosine functions, called orthogonal bases in various multiple frequencies of the fundamental frequency, that in the same way, in reverse operation, adding to all your results components in the original signal [15].

For example, a path with alpha activity (the naked eye appears to be a sequence of waves in the alpha frequency) can be composed of other frequencies as the beta activities, theta or delta and even harmonics and sub-harmonics of the same smaller amplitude [30] and to be superimposed, resulting in electrical activity with the naked eye appearance of alpha activity. The fast Fourier transform (FFT) is an algorithmic implementation—very efficient computational employed to decompose the EEG signals at their different frequencies [15].

In the "frequency domain," you cannot conduct a study of an event in time, as a spike or a variant of normalcy, because the decomposition in the frequency domain is lost time information, or rather, the temporal relationship of events by decomposition of transient paroxystic EEG events of two components, frequency and phase [15]. One may perform the results analysis obtained by quantification of the EEG signal in various ways, such as histograms, line graph or bar graph, table or carto-graphic form [22]. A new kind of ANN based on paraconsistent annotated evidential logic Eτ, which is capable of manipulating imprecise, inconsistent and paracomplete data in order to do a first study of the recognition of EEG standards.

Several studies on behavioral and cognitive neurology have been conducted to characterize dementias through biological and functional markers, for instance, the EEG activity, aimed at understanding the evolution of AD, following its progression, as well as leading toward better diagnostic criteria for early detection of cognitive impairment [12, 16–20, 22]. Currently, there is no method available to determine a definitive diagnosis of dementia, where a combination of tests would be necessary to obtain a probable diagnosis [23].

In conducting a study for recognizing EEG standards, we can then apply the PANN method to obtain a tool for probable diagnosis AD.

1.2 The HIV Gene Mutation Analysis

A similar architecture using PANN was built in order to study the recognition of HIV genotyping. The approach of this study is justified by the fact that mutations in the HIV-1 directly interferes with the drug treatment, making their treatment ineffective if the patient does not have a proper monitoring of virus evolution [7, 13, 26, 27, 28, 31].

The genetic sequence of the HIV virus used in this study consists of 20 elements (genotyping area used for classify HIV Subtypes), where each element is classified with letters from A to Z. At any time, the genotyping process does not correctly recognize the gene, so, the gene position is filled with "–" character (Table 1).

Table 1 Sample of HIV genotyping

Samples
TTTTTTAGGGAAAATTTGGC
GCAGGAAGA-GGCCAGTAAA

Many HIV-1 subtypes have in its nature, with different therapeutic regiment, including protease inhibitors and reverse transcriptase, as follows [13, 26]:

- Subtype A: Common in West Africa.
- Subtype B: Dominant form in Europe, the Americas, Japan, Thailand, and Australia.
- Subtype C: Dominant form in Southern Africa, Eastern Africa, India, Nepal, and parts of China.
- Subtype D: Generally only seen in Eastern and Central Africa.
- Subtype E: Has never been identified as a non-recombinant, only recombinable with subtype A.
- Subtype F: Has been found in central Africa, South America and Eastern Europe.
- Subtype G: Has been found in Africa and Central Europe.
- Subtype H: Is limited to Central Africa.
- Subtype I: Was originally used to describe a complex recombination of several subtypes.
- Subtype J: Primarily found in North, Central and West Africa, and the Caribbean.
- Subtype K: Limited to the Democratic Republic of Congo and Cameroon.

Therefore, to determine a therapeutic regiment, it is necessary to know the HIV-1 subtype. However, the HIV virus can mutate generating tolerance to the therapeutic regimen. In this case, it is necessary to perform a new genotyping of the virus to find out what the new subtype of HIV is [27, 28].

2 Background

Paraconsistent Artificial Neural Network (PANN) is a new artificial neural network [11]. Its basis leans on paraconsistent annotated logic Eτ [1]. Let us present it briefly.

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

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

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

- Uncertainty degree:

$$G_{un}(\mu, \lambda) = \mu + \lambda - 1 \quad (1)$$

- Certainty degree:

$$G_{ce}(\mu, \lambda) = \mu - \lambda \quad (2)$$

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

An order relation is defined on [0, 1]: $(\mu_1, \lambda_1) \leq (\mu_2, \lambda_2) \Leftrightarrow \mu_1 \leq \mu_2$ and $\lambda_2 \leq \lambda_1$, constituting a lattice that will be symbolized by τ.

With the uncertainty and certainty degrees we can get the following 12 output states (Table 2): 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. All states are represented in the next figure (Fig. 3).

Table 2 Extreme and Non-extreme states

Extreme states	Symbol	Non-extreme states	Symbol
True	V	Quasi-true tending to Inconsistent	$QV \rightarrow T$
False	F	Quasi-true tending to Paracomplete	$QV \rightarrow \perp$
Inconsistent	T	Quasi-false tending to Inconsistent	$QF \rightarrow T$
Paracomplete	$\perp$	Quasi-false tending to Paracomplete	$Qf \rightarrow \perp$
		Quasi-inconsistent tending to True	$QT \rightarrow V$
		Quasi-inconsistent tending to False	$QT \rightarrow F$
		Quasi-paracomplete tending to True	$Q\perp \rightarrow V$
		Quasi-paracomplete tending to False	$Q\perp \rightarrow F$

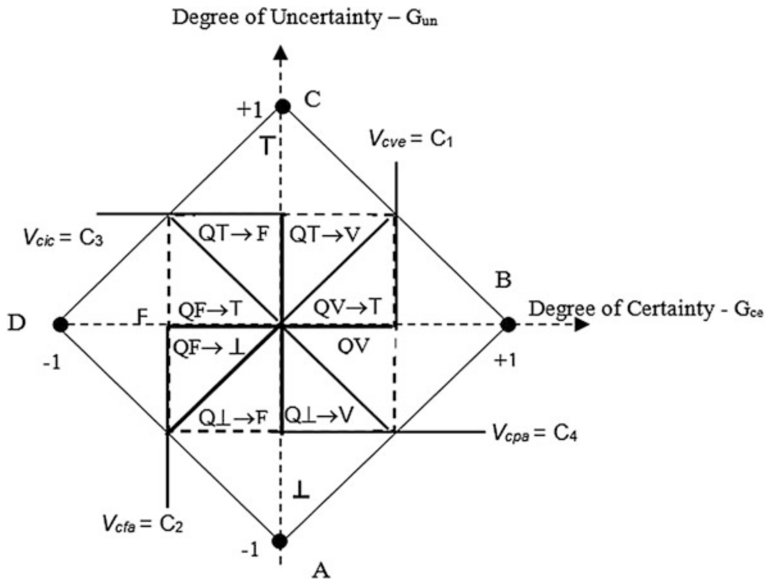


Fig. 3 Lattice with all theories logic states

3 The Main Artificial Neural Cells

In the PANN, the certainty degree G_{ce} indicates the 'measured' falsity or truth degree. The uncertainty degree G_{un} indicates the 'measure' of the inconsistency or paracompleteness. If the certainty degree in module is low, or the uncertainty degree in module is high, it generates a paracompleteness.

The resulting certainty degree G_{ce} is obtained as follows:

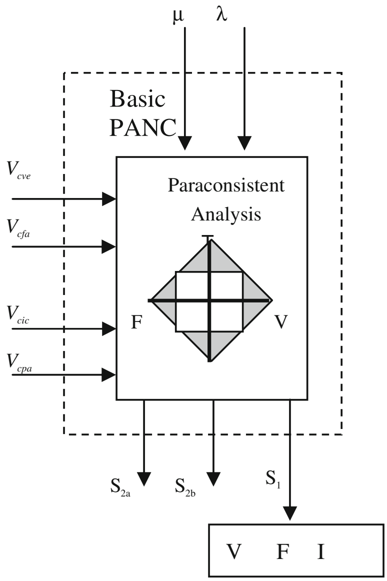
- If: $V_{cfa} = G_{ce} = V_{cve}$ or $-Ft_{ce} = G_{ce} = Ft_{ce} \Rightarrow G_{ce}$ = Indefiniteness
- For: $V_{cpa} = G_{un} = V_{cic}$ or $-Ft_{un} = G_{un} = Ft_{un}$
- If: $G_{ce} = V_{cfa} = -Ft_{ce} \Rightarrow G_{ce}$ = False with degree G_{un}
- If: $Ft_{ce} = V_{cve} = G_{ce} \Rightarrow G_{ce}$ = True with degree G_{un}

A Paraconsistent Artificial Neural Cell—PANC—is called basic PANC (Fig. 4) when, given a pair (μ, λ), it is used as input and results as the following output:

- $S_{2a} = G_{un}$ = resulting uncertainty degree
- $S_{2b} = G_{ce}$ = resulting certainty degree
- $S_1 = X$ = constant of Indefiniteness.

The uncertainty degree G_{un} indicates the 'measure' of the inconsistency or paracompleteness. If the certainty degree in module is low or the uncertainty degree in module is high, it generates an indefiniteness.

Fig. 4 Basic cell of PANN



The resulting certainty degree Gce is obtained as follows:

- If: $V_{cfa} = G_{ce} = V_{cve}$ or $-Ft_{ce} = G_{ce} = Ft_{ce} \Rightarrow G_{ce}$ = Indefiniteness
- For: $V_{cpa} = G_{un} = V_{cic}$ or $-Ft_{un} = G_{un} = Ft_{un}$
- If: $G_{ce} = V_{cfa} = -Ft_{ce} \Rightarrow G_{ce}$ = False with degree G_{un}
- If: $Ft_{ce} = V_{cve} = G_{ce} \Rightarrow G_{ce}$ = True with degree G_{un}

A Paraconsistent Artificial Neural Cell—PANC—is called basic PANC (Fig. 4) when given a pair (μ, λ) is used as input and resulting as output:

- $S_{2a} = G_{un}$ = resulting uncertainty degree
- $S_{2b} = G_{ce}$ = resulting certainty degree
- $S_1 = X$ = constant of Indefiniteness.

Using the concepts of basic Paraconsistent Artificial Neural Cell, we can obtain the family of PANC considered in this work: Analytic connection (PANCac), Maximization (PANCmax), and Minimization (PANCmin) as described in Table 3 below:

Making the understanding on the implementation of the algorithms of PANC easier, we use a programming language, Object Pascal, following logic of procedural programming in all samples.

Table 3 Paraconsistent artificial neural cells

PANC	Inputs	Calculations	Output
Analytic connection: PANCac	μ λ Ft_{un} Ft_{un}	$\lambda_c = 1 - \lambda$ G_{un} G_{ce}, $\mu_r = (G_{ce} + 1)/2$	If $\lvert G_{ce}\rvert > Ft_{ce}$ then $S_1 = \mu_r$ and $S_2 = 0$ If $\lvert G_{un}\rvert > Ft_{ct}$ and $\lvert G_{un}\rvert > \lvert G_{ce}\rvert$ then $S_1 = \mu_r$ and $S_2 = \lvert G_{un}\rvert$ if not $S_1 = ½$ and $S_2 = 0$
Maximization: PANCmax	μ λ	G_{ce} $\mu_r = (G_{ce} + 1)/2$	If $\mu_r > 0.5$, then $S_1 = \mu$ If not $S_1 = \lambda$
Minimization: PANCmin	μ λ	G_{ce} $\mu_r = (G_{ce} + 1)/2$	If $\mu_r < 0.5$, then $S_1 = \mu$ if not $S_1 = \lambda$

3.1 Paraconsistent Artificial Neural Cell of Analytic Connection—PANCac

The paraconsistent artificial neural cell of analytic connection cell (PANCac) is the principal cell of all PANN, in obtaining the certainty degree (G_{ce}) and the uncertainty degree (G_{un}) from the inputs, and the tolerance factors.

This cell is the link which allows different regions of PANN to perform signal processing in distributed and through many parallel connections [11].

The different tolerance factors certainty (or contradiction) acts as inhibitors of signals, controlling the passage of signals to other regions of the PANN, according to the characteristics of the architecture developed.

In Table 4, we have a sample of the implementation made in Object Pascal (Fig. 5).

3.2 Paraconsistent Artificial Neural Cell of Maximization—PANCmax

The paraconsistent artificial neural cell of maximization cell (PANCmax) allows selection of the maximum value among the entries.

Such cells operate as logical connectives OR between input signals. In order to do so, it is made a simple analysis, through the equation of the Degree of Evidence (Table 3), which thus will show us which of the two input signals is of greater value, thus establishing the output signal [11].

In Table 5, we have a sample of the implementation made in Object Pascal (Fig. 6).

Table 4 PANCac implementation

Sample of program code

```
Function TFaPANN.PANCAC(mi, lambda, Ftce, Ftct: real; output: integer): real;
var
  Gce: real;
  Gun: real;
  lambdacp: real;
  mir: real;
  S1, S2: real;
begin
  lambdacp := 1 - lambda;
  Gce := mi - lambdacp;
  Gun := mi + lambdacp - 1;
  mir := (Gce + 1) / 2;

  if (abs(Gce) > Ftce) then
  begin
    S1 := mir;
    S2 := 0;
  end
  else
  begin
    if (abs(Gun) > Ftct) and (abs(Gun) > abs(Gce)) then
    begin
      S1 := mir;
      S2 := abs(Gun);
    end
    else
    begin
      S1 := 0.5;
      S2 := 0;
    end;
  end;
  if output = 1 then result := S1 else result := S2;
end;
```

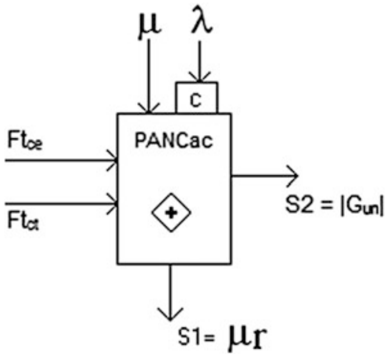


Fig. 5 Representation of PANCac

Table 5 PANCmax implementation

```
Sample of program code
Function TFaPANN.PANCMAX(mi, lambda: real): real;
var
  mir: real;
begin
  mir := ((mi - lambda) + 1) / 2;
  if (mir > 0.5) then
    result := mi
  else
    result := lambda;
end;
```

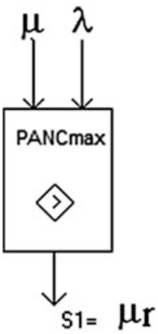


Fig. 6 Representation of PANCmax

3.3 Paraconsistent Artificial Neural Cell of Minimization—PANCmin

The paraconsistent artificial neural cell of maximization cell (PANCmin) allows selection of the minimum value among the entries.

Such cells operate as logical connectors AND between input signals. For this to happen, it is made a simple analysis, through the equation of the Degree of Evidence (Table 3), which thus will show us which of the two input signals is of lesser value, thus establishing the output signal [11].

In Table 6, we have a sample of the implementation made in Object Pascal (Fig. 7).

3.4 Paraconsistent Artificial Neural Unit

A Paraconsistent Artificial Neural Unit (PANU) is characterized by the association ordered PANC, targeting a goal, such as decision making, selection, learning, or some other type of processing.

When creating a PANU, one obtains a data processing component capable of simulating the operation of a biological neuron.

Table 6 PANCmin implementation

Sample of program code

```
Function TFaPANN.PANCMIN(mi, lambda: real): real;
var
  mir: real;
begin
  mir := ((mi - lambda) + 1) / 2;
  if (mir < 0.5) then
     result := mi
  else
     result := lambda;
end;
```

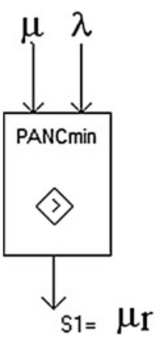


Fig. 7 Representation of PANCmin

3.5 Paraconsistent Artificial Neural System

Classical systems based on binary logic are inefficient to process data or information from uncertain knowledge. These data are captured or received information from multiple experts that usually come in the form of evidences.

Paraconsistent Artificial Neural Systems (PANS) modules are configured and built exclusively by PANU, whose function is to provide the signal processing ‘similar’ to processing that occurs in the human brain.

4 PANN for Morphological Analysis

The process of morphological analysis of a wave is performed by comparing it with a certain set of wave patterns (stored in the control database). A wave is associated with a vector (finite sequence of natural numbers) through digital sampling. This vector characterizes a wave pattern and is registered by PANN. Thus, new waves are compared, allowing their recognition or otherwise.

Each wave of the survey examined, the EEG corresponds to a period of 1 s examination. Every second of the exam contains 256 positions.

The wave that has the highest favorable evidence and lowest contrary evidence is chosen as the more similar wave to the analyzed wave.

A control database is composed by waves presenting 256 positions with perfect sinusoidal morphology, with 0.5 Hz of variance, taking into account, Delta, Theta, Alpha and Beta (of 0.5–30.0 Hz) wave groups.

In other words, morphological analysis checks the similarity of the passage of the examination of EEG in a reference database that represents a wave pattern.

4.1 Data Preparation

The process of wave analysis by PANN consists previously of data capturing, adaptation of the values for screen examination, elimination of the negative cycle and normalization of the values for PANN analysis.

As the actual EEG examination values can vary highly, in module, something 10–1500 μV, we perform a normalization of the values between 100 μV and −100 μV by a simple linear conversion, to facilitate the manipulation the data:

$$x = \frac{100.a}{m} \tag{3}$$

where

m is the maximum value of the exam; a is the current value of the exam; x is the current normalized value.

The minimum value of the exam results is taken as zero value and the remaining values are translated proportionally.

It is valid to observe that the process above does not allow the loss of any wave essential characteristics for our analysis.

5 The PANN Architecture

The architecture of the PANN used in decision making is based on the architecture of Paraconsistent Artificial Neural System for Treatment of Contradictions.

Such a system promotes the treatment of contradictions, so continuous among information signals, which receives three input signals and presents as a result, a value that represents the consensus between the three information. The contradictions between the two values are added to the third value, so that the output, is the value proposed by the dominant majority. The analysis instantly carries all data processing in real time, similar to the functioning of biological neurons (Table 7).

This method is used primarily for PANN (Fig. 9) to balance the data received from expert systems. After this process, it uses a decision-making lattice to determine the soundness of the recognition (Fig. 8).

Table 7 Lattice for decision-making used in the morphological analysis (Fig. 8)

Limits of areas of lattice	
True	Fe > 0.61 Ce < 0.40 G_{ce} > 0.22
False	Fe < 0.61 Ce > 0.40 G_{ce} <= 0.23

Ce Contrary evidence, *Fe* Favorable evidence, G_{ce} Certainty degree

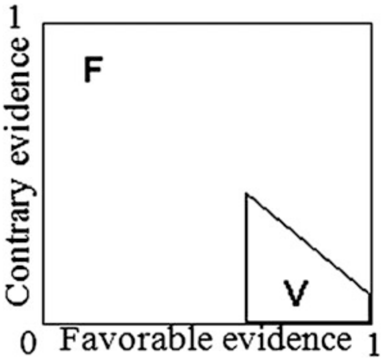


Fig. 8 Lattice for decision-making used in morphological analysis used after making PANN; F: logical state false (it is interpreted as wave not similar); V: logical state true (it is interpreted as wave similar)

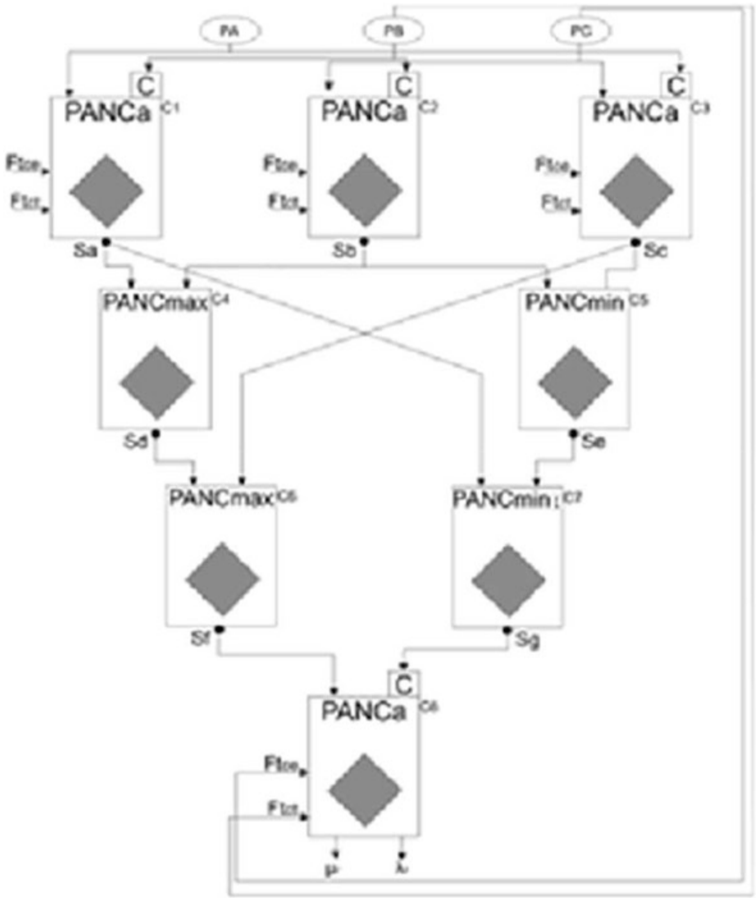


Fig. 9 The architecture for morphological analysis. Three expert systems operate: PA, for check the number of wave peaks; PB, for checking similar points, and PC, for checking different points: The 1st layer of the architecture: C1–PANC which processes input data of PA and PB; C2–PANC which processes input data of PB and PC; C3–PANC which processes input data of PC and PA. The 2nd layer of the architecture: C4–PANC which calculates the maximum evidence value between cells C1 and C2; C5–PANC which calculates the minimum evidence value between cells C2 and C3; The 3rd layer of the architecture: C6–PANC which calculates the maximum evidence value between cells C4 and C3; C7–PANC which calculates the minimum evidence value between cells C1 and C5. The 4th layer of the architecture: C8 analyzes the experts PA, PB, and PC and gives the resulting decision value. PANC A = Paraconsistent artificial neural cell of analytic connection. $PANCLs_{Max}$ = Paraconsistent artificial neural cell of simple logic connection of maximization. $PANCLs_{Min}$ = Paraconsistent artificial neural cell of simple logic connection of minimization. Ft_{ce} = Certainty tolerance factor; Ft_{un} = Uncertainty tolerance factor. S_a = Output of C1 cell; S_b = Output of C2 cell; S_c = Output of C3 cell; S_d = Output of C4 cell; Se = Output of C5 cell; Sf = Output of C6 cell; Sg = Output of C7 cell. C = Complemented value of input; μ_r = Value of output of PANN; λ_r = Value of output of PANN

Table 8 The architecture for morphological analysis implementation (Fig. 14)

Sample of program code
function Tf_pann.Morphological_analysis(PA, PB, PC: real; tipo: integer): real;

```
function Tf_pann.Morphological_analysis(PA, PB, PC: real; tipo: integer): real;
var
  C1, C2, C3, C4, C5, C6, C7: real;
begin
  C1 := FaPANN.PANCAC(PA, PB, 0, 0, 1);
  C2 := FaPANN.PANCAC(PC, PB, 0, 0, 1);
  C3 := FaPANN.PANCAC(PC, PA, 0, 0, 1);
  C4 := FaPANN.PANCMAX(C1, C2);
  C6 := FaPANN.PANCMAX(C4, C3);
  C5 := FaPANN.PANCMIN(C2, C3);
  C7 := FaPANN.PANCMIN(C1, C5);
  if tipo = 1 then
    result := FaPANN.CNAPCA(C6, C7, PC, PB, 1)
  else
    result := FaPANN.CNAPCA(C6, C7, PC, PB, 2);
end;
```

A sample of morphological analysis implementation using Object Pascal is showed in Table 8.

The definition of the regions of the lattice decision-making was done through double-blind trials, i.e., in each battery of tests, a validator checked the results and returned only the percentage of correct answers. After testing several different configurations, is adopted the configuration of the lattice regions whose decision-making had a better percentage of success.

For an adequate PANN wave analysis, it is necessary that each input of PANN be properly calculated. These input variables are called expert systems, as they are specific routines for information extraction.

In analyzing EEG signals, one important aspect to take into account is the morphological aspect. To perform this task, it is valuable to build a very simple Expert System, which allows analyses the signal behavior verifying which band it belongs to (delta, theta, alpha and beta).

The method of morphological analysis is comprised of three expert systems, responsible for feeding the inputs of PANN with information that are relevant to the wave being analyzed: number of peaks, similar points and different points.

5.1 Expert System 1—Checking the Number of Wave Peaks

The aim of the expert system 1 is to compare the waves and analyze their differences regarding the number of peaks.

In practical terms, one can say that when we analyze the wave peaks, we are analyzing the resulting frequency of the wave (as rudimentary as this).

It is worth remembering that, because of its biological signal, we should not work with absolute quantification due to the variability characteristic of this type of signal. Therefore one should always take into consideration a tolerance factor.

A sample checking of the number of wave peaks function implementation using Object Pascal is shown in Table 9.

$$Se_1 = 1 - \left(\frac{(|bd - vt|)}{(bd + vt)}\right) \tag{4}$$

where

vt is the number of peaks of the wave.

bd is the number of peaks of the wave stored in the database.

Se_1 is the value resulting from the calculation.

Table 9 Checking the number of wave peaks function implementation

Sample of program code

function Tf_pann.f_ EstimatedAveragePeak(vv: array of real; total_elements: integer; Ftr: real): real;
var
  last_larger_point, mean_peaks: real;
  peak_check: boolean;
  v_vector_aux_larger_value: real;
  a: integer;
  peaks: integer;
 	
begin
  last_larger_point := 0;
  peak_check := false;
  v_vector_aux_larger_value := 0;
  peaks := 0;
  mean_peaks := 0;
 	
  for a := 1 to total_elements do
  begin

(continued)

Table 9 (continued)

```
    if abs(vv[a - 1]) > v_vector_aux_larger_value then
      v_vector_aux_larger_value := vv[a - 1];

    if vv[a - 1] >= last_larger_point then
    begin
      last_larger_point := vv[a - 1];
      peak_check := false;
    end
    else
    begin
      if (peak_check = true) and
        (vv[a - 1] > vv[a - 2]) then
        last_larger_point := vv[a - 1];
      if abs(last_larger_point - vv[a - 1]) >= ((Ftr / 100) * last_larger_point) then
      begin
        if peak_check = false then
        begin
          peaks := peaks + 1;
          mean_peaks := mean_peaks + last_larger_point;
          peak_check := true;
        end;
      end;
    end;
  end;
 result := mean_peaks / picos;
end;
```

5.2 *Expert System 2—Checking Similar Points*

The aim of the expert system 2 is to compare the waves and analyze their differences regarding to similar points.

When we analyze the similar points, it means that we are analyzing how one approaches the other point.

It is worth remembering that, because of its biological signal, we should not work with absolute quantification due to the variability characteristic of this type of signal. Therefore, one should always take into consideration a tolerance factor.

A sample checking of similar points function implementation using Object Pascal is shown in Table 10.

$$Se_2 = \frac{\sum_{j=1}^{n} (x_j)}{n} \tag{5}$$

where

n is the total number of elements.
x is the element of the current position.
j is the current position.
Se_2 is the value resulting from the calculation.

Table 10 Checking similar points function implementation

Sample of program code

```
Function Tf_pann.f_SimilarPoints(vv, vb: array of real; total_elements: integer; Ftr: real;
max_value:real; lager_field_value:real): real;
var
  a: integer;
  fieldx_bd: real;
  q: real;
begin
    q:=0;
    for a := 1 to total_elements do
    begin
      fieldx_bd := vb[a - 1];
      fieldx_bd := ((max_value * fieldx_bd) / lager_field_value);
     if abs(fieldx_bd - vv[a - 1]) <= ((Ftr / 100) * max_value) then
      begin
        q := q + 1;
      end;
    end;
  result := 1 – (strtofloat(floattostrf(((q / total_elementos)), ffnumber, 18, 2)));
end;
```

5.3 Expert System 3—Checking Different Points

The aim of the expert system 3 is to compare the waves and analyze their differences regarding of different points.

When we analyze the different points, it means that we are analyzing how distant a point is from each other, so a tolerance factor should also be considered.

A sample checking of different points function implementation using Object Pascal is shown in Table 11.

$$Se_3 = 1 - \left(\frac{\sum_{j=1}^{n} \left(\frac{|x_j - y_j|}{a} \right)}{n} \right) \tag{6}$$

where

n is the total number of elements.
a is the maximum amount allowed.
j is the current position.

Table 11 Checking different points function implementation

Sample of program code

```
function Tf_pann.f_DifferentPoints(vv, vb: array of real; total_elements: integer; Ftr,
max_value, lager_field_value:real): real;
var
  a: integer;
  fieldx_bd, q: real;
begin
    q:=0;
    for a := 1 to total_elements do
    begin
      fieldx_bd := vb[a - 1];
      fieldx_bd := ((max_value * fieldx_bd) / lager_field_value);
      if abs(fieldx_bd - vv[a - 1]) > ((Ftr / 100) * max_value) then
      begin
        q := q + (abs(fieldx_bd - vv[a - 1]) / max_value);
      end;
    end;
  result := 1 – (strtofloat(floattostrf(((q / total_elementos)), ffnumber, 18, 2)));
end;
```

x is the value of wave 1.
y is the value of wave 2.
Se_3 is the value resulting from the calculation.

6 Experimental Procedures—Applying in Alzheimer Disease

It is known that the visual analysis of EEG patterns may be useful in aiding the diagnosis of Alzheimer disease (AD), and indicated in some clinical protocols for diagnosing the disease [8, 9]. The most common findings on visual analysis of EEG patterns are the slowing of brain electrical activity based on predominance of delta and theta rhythms and decrease or absence of alpha rhythm. However, these findings are more common and evident in patients in moderate or advanced stages of the disease [6, 12, 29].

In this study we have sixty-seven analyzed EEG records, thirty-four normal and thirty-three probable AD (p value = 0.8496) during the awakened state at rest (Table 12).

All tests were subjected to morphological analysis methodology to measure the concentration of the waves. This information is later submitted to a PANN unit responsible for assessing the data and arriving at a classification of the examination in Normal or probable AD (Figs. 10 and 11).

6.1 *Expert System 1—Detecting the Diminishing Average Frequency Level*

The aim of the expert system 1 is to verify the average frequency level of Alpha waves and compare them with a fixed external one (external parameter wave).

Table 12 Lattice for decision-making (Fig. 11) used in diagnostic analysis used after making PANN analysis (Fig. 10)

Characterization of the lattice	
Area 1	$G_{ce} \leq 0.1999$ and $G_{ce} \geq 0.5600$ and $\lvert G_{un} \rvert < 0.3999$ and $\lvert G_{un} \rvert \geq 0.4501$
Area 2	$0.2799 < G_{ce} < 0.5600$ and $0.3099 \leq \lvert G_{un} \rvert < 0.3999$ and Fe < 0.5000
Area 3	$0.1999 < G_{ce} < 0.5600$ and $0.3999 \leq \lvert G_{un} \rvert < 0.4501$ and Fe > 0.5000
Area 4	$G_{ce} > 0.7999$ and $\lvert G_{un} \rvert < 0.2000$

Ce Contrary evidence; *Fe* Favorable evidence; G_{ce} Certainty degree; G_{un} uncertainty degree

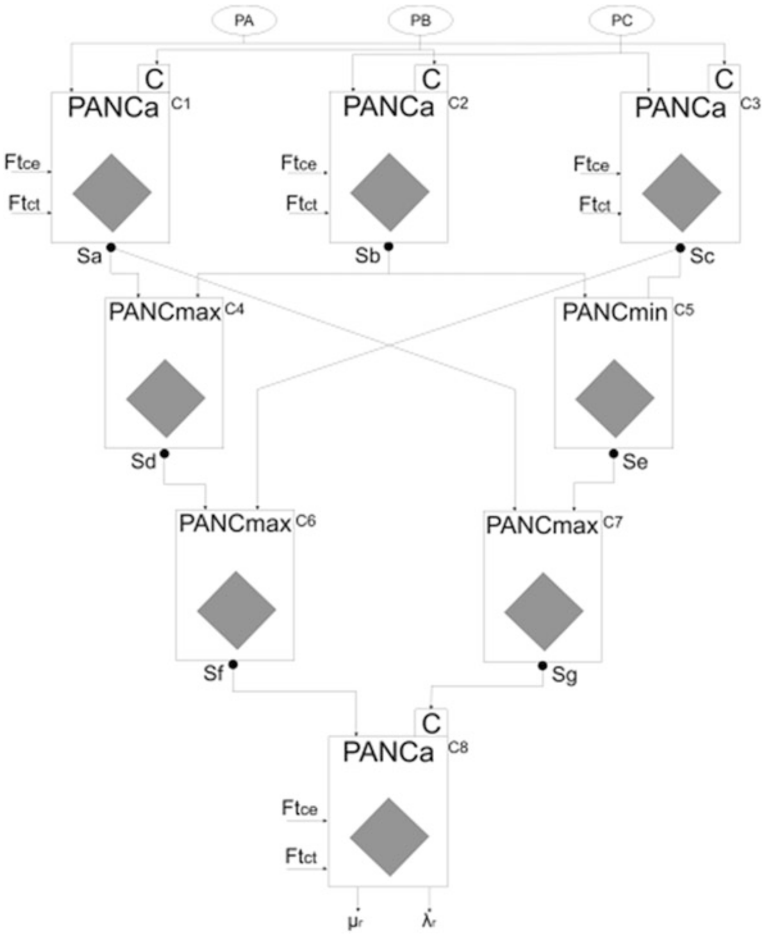


Fig. 10 The architecture for diagnosis analysis

Such external parameter can be, for instance, the average frequency of a population or the average frequency of the last exam of the patient. This system also generates two outputs: favorable evidence μ (normalized values ranging from 0 (corresponds to 100 %—or greater frequency loss) to 1 (which corresponds to 0 % of frequency loss) and contrary evidence λ (Eq. 7).

The average frequency of population pattern used in this work is 10 Hz.

$$\lambda = 1 - \mu \tag{7}$$

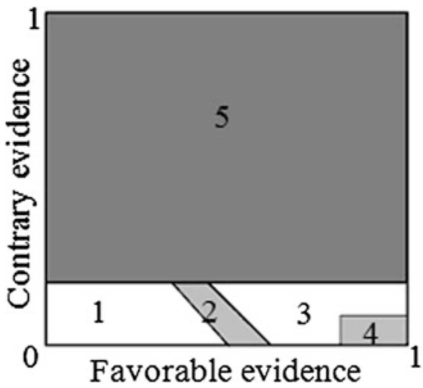


Fig. 11 Lattice for decision-making used in diagnostic analysis. Area *1*: State logical False (AD likely below average population), *2*: State logical Quasi-true (AD likely than average population); Area *3*: State logical Quasi-false (Normal below average population); Area *4*: State logical True (Normal above average population); Area *5*: logical state of uncertainty (not used in the study area)

6.2 Expert System 2—High Frequency Band Concentration

The aim of the expert system 2 is to utilize the alpha band concentration in the exam. For this, we consider the quotient of the sum of fast alpha and beta waves over slow delta and theta waves (Eq. 8) as the first output value. For the second output value (contrary evidence λ) Eq. 7 is used.

$$\mu = \left(\frac{(A + B)}{(D + T)} \right) \tag{8}$$

where

A is the alpha band concentration. B is the beta band concentration.

D is the delta band concentration. T is the theta band concentration.

μ is the value resulting from the calculation.

6.3 Expert System 3—Low Frequency Band Concentration

The aim of the expert system 3 is to utilize the theta band concentration in the exam. For this, we consider the quotient of the sum of slow delta and theta waves over fast alpha and beta waves (Eq. 9) as the first output value. For the second output value (contrary evidence λ) Eq. 7 is used.

Table 13 Diagnosis—Normal × Probable AD patients

Gold standard				
PANN		AD patient (%)	Normal patient (%)	Total (%)
	AD patient	35.82	14.93	50.75
	Normal patient	8.96	40.30	49.25
	Total	44.78	55.22	100.00

Sensitivity = 80 %; Specificity = 73 %; Index of coincidence (Kappa): 76 %

$$\mu = \left(\frac{(D+T)}{(A+B)}\right) \tag{9}$$

where

A is the alpha band concentration. B is the beta band concentration.
D is the delta band concentration. T is the theta band concentration.
μ is the value resulting from the calculation.

6.4 Results

Table 13

7 Experimental Procedures—Applying in Recognition of HIV Genotyping

The following is an example of the recognition process, which will consider three sequences (Fig. 12) of 20 elements, with maximum amplitude of 11 points (00–10) and hypothetical values (Table 14). This example is intended to explain in detail and didactic, the pattern recognition process using this methodology.

The following sample is the sequence that will be submitted for recognition of the PANN. The Reference 1 and Reference 2 are two sequences that were previously stored in the database control (normal range).

To be able to process the PANN sequence analysis, it is necessary that each entry of PANN be properly calculated. These input variables are called expert systems. They are related to specific routines for extracting information (Table 15).

The first expert system is responsible for quantifying the known mutations by comparing the sample and reference, according to the formula:

$$E1 = \frac{\sum |x-y|}{n} \tag{10}$$

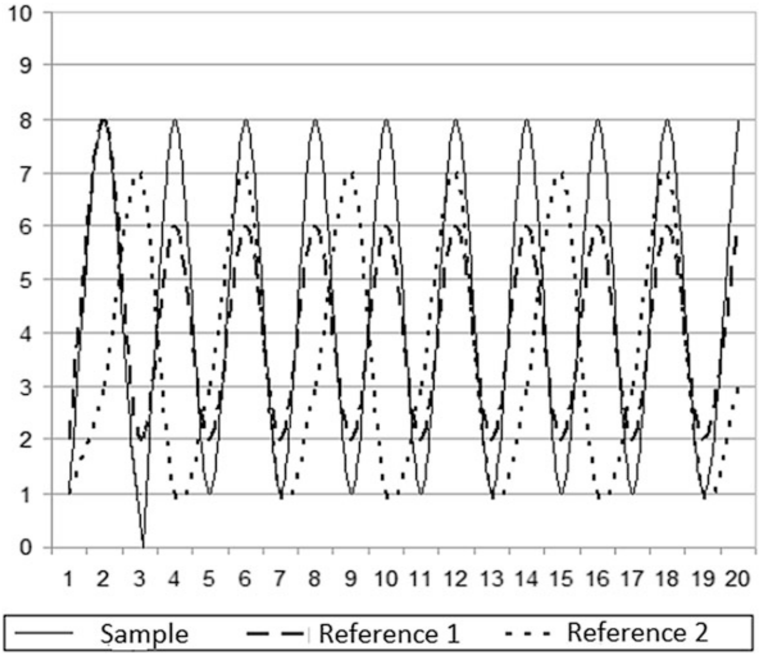


Fig. 12 Visual example of sequences used in morphological analysis

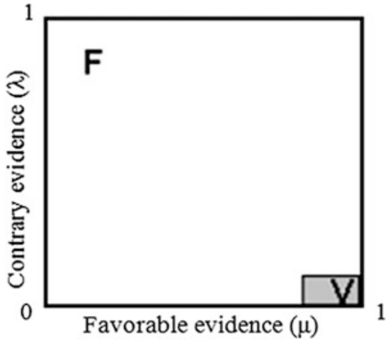


Fig. 13 Lattice for decision-making used in morphological analysis used after making PANN; F: logical state false (it is interpreted as reference not similar); V: logical state true (it is interpreted as reference similar)

where

$E1$ is the value of expert system 1.

x is the value of the position in the sequence of the sample.

y is the value of the position in the sequence of reference.

n total number of elements of the sequence.

Table 14 Values of the example of the process of recognition of sequences

Sample	Reference 1	Reference 2
1	2	1
8	8	3
0	2	7
8	6	1
1	2	3
8	6	7
1	2	1
8	6	3
1	2	7
8	6	1
1	2	3
8	6	7
1	2	1
8	6	3
1	2	7
8	6	1
1	2	3
8	6	7
1	2	1
8	6	3

Table 15 Limits of the areas of lattice for decision-make used in morphological analisys

Limits of the states of lattice	
True (V)	Fe ≥ 0.75
	Ce ≤ 0.15
False (F)	Fe < 0.75
	Ce > 0.15

Ce Contrary evidence (λ); *Fe* Favorable evidency (μ)

Performing the comparison between sequences, we have (Table 16):

The second expert system is responsible for quantifying the unknown mutations by comparing the sample and reference, according to the formula:

$$E2 = \frac{\sum_{c||x-y| > 0} c}{n} \quad (11)$$

where

$E2$ is the value of expert system 2.

c is the sum of positions equal 0 (equal of ? in genotyping notation) and $|x-y| > 0$.

Table 16 Verification the quantity of known mutations

Sequence	Sample 1 × Reference 1	Sample × Reference 2
Knows mutation	18	16
Number of mutations normalized by the number of elements of the sequence. Expert system 1 (Eq. 10)	0.90	0.8

Table 17 Verification the quantity of unknown mutations

Sequence	Sample 1 × Reference 1	Sample × Reference 2
Unknows mutation	1	1
Number of mutations normalized by the number of elements of the sequence. Expert system 2 (Eq. 11)	0.05	0.05

n is total element sequence.
x is the value of the position in the sequence of the sample. y is the value of the position in the sequence.

Performing a comparison between the sequences, we have (Table 17):

The third expert system is responsible for quantifying the size of the changes wrought by comparing the sample and reference, according to the formula:

$$E3 = \frac{\sum \frac{|x-y|}{a}}{n} \tag{12}$$

where
$E3$ is the value of expert system 3.
x is the value of the position in the sequence of the sample.
y is the value of the position in the sequence of reference.
a is the maximum amplitude of the samples (a = Z = 26).
n total number of elements of the sequence.

Performing the comparison between sequences, we have (Tables 18, 19 and 20):

The following are the values of each expert system that will be used as input values for the PANN (Fig. 14):

In practical terms, one can say that, by analyzing the sequences of these characteristics, we are forcing the PANN to “see” the profile of each sample sequence, combining such information, as the sequence is similar to one another.

This procedure is always performed by comparing a sample of all references in the database. It is voted the most similar to the reference sample which had the highest and lowest μ resulting λ resulting from the processing of PANN.

Table 18 Comparison between the sample and reference 1

Sample	Reference 1	Difference in module	Normalization of the maximum amplitude difference
1	2	0	0.04
8	8	1	0
0	2	2	0.08
8	6	2	0.08
1	2	1	0.04
8	6	2	0.08
1	2	1	0.04
8	6	2	0.08
1	2	1	0.04
8	6	2	0.08
1	2	1	0.04
8	6	2	0.08
1	2	1	0.04
8	6	2	0.08
1	2	1	0.04
8	6	2	0.08
1	2	1	0.04
8	6	2	0.08
1	2	1	0.04
8	6	2	0.08
Sum of normalized differences: 1.16			
Expert system 3 (Eq. 12): Normalized by the total of element: 0.06			

After the analysis of expert systems and PANN, the values of favorable evidences (the highest resultant μ) and contrary evidence (the smaller the resulting λ) are submitted to the lattice of logic states (Fig. 13) which will set its output logic state, i.e., if the similarity between the sequences is true or not.

Using real data, three hundred and eight samples from region sequences of the protease enzyme of the pol gene (polymerase) of HIV-1 subtype F, B and BF recombinants, with different therapeutic regimen, including protease inhibitors and reverse transcriptase, obtained from the database regarding HIV resistance to antiretroviral drugs and Stanford University, California (Stanford University HIV Drug Resistance Database). The reference sequences (consensus) used for analysis were obtained from the database of HIV sequences from the Los Alamos National Laboratory, USA.

In the preliminary test carried out with three hundred and eight protease sequences for subtypes F, B and BF of HIV-1 of the Stanford database, the

Table 19 Comparison between then sample and reference 2

Sample	Reference 2	Difference in module	Normalization of the maximum amplitude difference
1	1	0	0.00
8	3	5	0.19
0	7	7	0.27
8	1	7	0.27
1	3	2	0.08
8	7	1	0.04
1	1	0	0.00
8	3	5	0.19
1	7	6	0.23
8	1	7	0.27
1	3	2	0.08
8	7	1	0.04
1	1	0	0.00
8	3	5	0.19
1	7	6	0.23
8	1	7	0.27
1	3	2	0.08
8	7	1	0.04
1	1	0	0.00
8	3	5	0.19
Sum of normalized differences: 2.65			
Expert system 3 (Eq. 12): Normalized by the total of element: 0.13			

Table 20 Summary of results of expert systems

Case	Expert system 1 (E1)	Expert system 2 (E2)	Expert system 3 (E3)
Sample X Reference 1	0.90	0.05	0.06
Sample X Reference 2	0.80	0.05	0.13

methodology showed a high level of agreement (Coefficient Kappa 0.92) as can be seen on Table 21.

In Table 22, we can see that the classification for subtypes F and non—F showed a high level of agreement (sensitivity 92 % and Specificity 100 %).

For subtypes B and non-B can also see a high rating (sensitivity 100 % and specificity 93 %) as can be seen on Table 23.

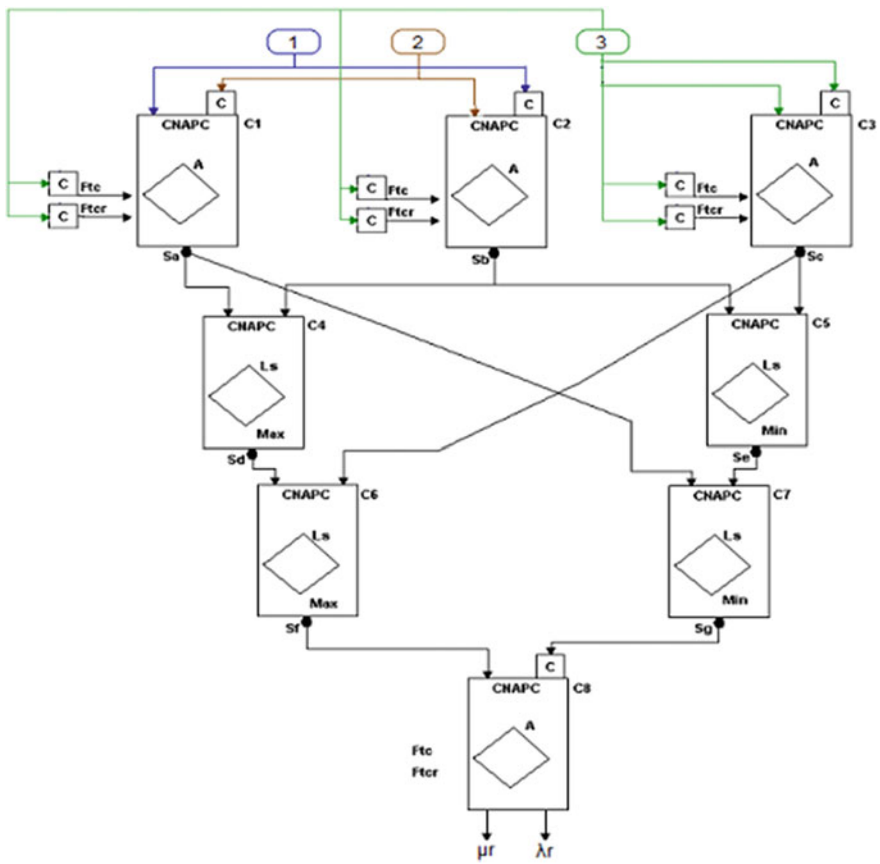


Fig. 14 The architecture for morphological analysis. Three expert systems operate: PA, for check the number of wave peaks; PB, for checking similar points, and PC, for checking different points: The 1st layer of the architecture: C1–PANC which processes input data of PA and PB; C2–PANC which processes input data of PB and PC; C3–PANC which processes input data of PC and PA. The 3rd layer of the architecture: C4–PANC which calculates the maximum evidence value between cells C1 and C2; C5–PANC which calculates the minimum evidence value between cells C2 and C3; C4 and C5 constitute the 2nd layer of the architecture; C6–PANC which calculates the maximum evidence value between cells C4 and C3; C7–PANC which calculates the minimum evidence value between cells C1 and C5. The 4th layer of the architecture: C8 analyzes the experts PA, PB, and PC and gives the resulting decision value. PANC A = Paraconsistent artificial neural cell of analytic connection. PANCLsMax = Paraconsistent artificial neural cell of simple logic connection of maximization. PANCLsMin = Paraconsistent artificial neural cell of simple logic connection of minimization. Ftce = Certainty tolerance factor; Ftct = Contradiction tolerance factor. Sa = Output of C1 cell; Sb = Output of C2 cell; Sc = Output of C3 cell; Sd = Output of C4 cell; Se = Output of C5 cell; Sf = Output of C6 cell; Sg = Output of C7 cell. C = Complemented value of input; μr = Value of output of PANN; λr = Value of output of PANN

Table 21 Summary of results of expert systems

Stanford database							
	Subtypes	A	B	C	D	F/BF	Total
	A	0	1	0	0	5	6
	B	0	29	0	0	18	47
PANN	C	0	0	0	0	0	0
	D	0	1	0	0	0	1
	F/BF	0	0	0	0	254	254
	Total	0	31	0	0	277	308

Table 22 Comparion of F and Non-F subtypes of HIV-1 Sequences between Stanford and PANN

	Stanford database			
PANN	Subtypes	F/BF	Non-F/BF	Total
	F/BF	254	0	254
	Non-F/BF	23	31	54
	Total	277	31	308

Table 23 Comparion of B and Non-B subtypes of HIV-1 sequences between stanford and PANN

	Stanford database			
PANN	Subtypes	B	Non-B	Total
	B	29	20	49
	Non-B	0	259	259
	Total	29	279	308

8 Conclusions

We believe that a process of the examination analysis using a PANN attached to EEG findings, such as relations between frequency bandwidth and inter hemispheric coherences, can create computational methodologies that allow the automation of analysis and diagnosis tendencies.

These methodologies could be employed as tools to aid in the diagnosis of diseases such as Alzheimer, provided they have defined electroencephalographic findings.

In the case of Alzheimer's disease, for example, in studies carried out previously shown satisfactory results [20] (but still far from being a tool to aid clinical) that demonstrated the computational efficiency of the methodology using a simple morphological analysis (only paraconsistent annotated logic Eτ). These results encouraged us to improve the morphological analysis of the waves and try to apply the method in other diseases besides Alzheimer's disease.

With the process of morphological analysis using the PANN, it becomes possible to quantify the average frequency of the individual without losing its temporal reference. This feature becomes a differential, compared to traditional analysis of quantification of frequencies, such as FFT, aiming at a future application in real-time analysis, i.e., at the time of acquisition of the EEG exams.

Regarding the specificity, the method showed more reliable results. Taking into account an overall assessment in the sense we take the arithmetic mean of sensitivity (75.50 %) and specificity (92.75 %), we find reasonable results that encourage us to seek improvements in this research.

Even finding a low sensitivity in the recognition of delta waves, the methodology of pattern recognition using morphological analysis showed to be effective, achieving recognizable waves patterns, similar to patterns stored in the database, allowing quantifications and qualifications of the examination of EEG data to be used by PANN in their process analysis of examination. PANN has been applied in other branches: MICR Automated Recognition [29], computer aided diagnosis (breast cancer) [6], and many other themes.

The tests apply in HIV [25] genotyping, Several configurations were tested for analysis until it has obtained the best configuration of the architecture of artificial neural network paraconsistent, prevailing until the moment of the configuration with the best sensitivity and specificity.

All analysis of the sequences were performed by means of double-blind trials (using control samples, not included in batteries of tests, i.e., the diagnostic validation was not released until the best configuration of PANN had been chosen, using as criterion the correlation between results and clinical diagnosis.

Comparing the clinical correlation obtained in this study with others in the literature, we can see a promising advantage over the levels of processing methods. While studies use artificial neural networks classic combined with other mathematical tools to arrive at a clinical correlation of 90 %, the methodology of this study has a clinical correlation value using only one type of analysis.

The methodology of pattern recognition using morphological analysis showed to be effective, achieving recognizable patterns of reference similar to patterns stored in the database, allowing quantifications and qualifications of the blood samples infected with HIV to be used by PANN in their process analysis of examination.

References

1. Abe, J.M., Foundations of annotated logics. Ph.D. thesis (in Portuguese) USP, Brazil (1992)
2. Abe, J.M.: Some aspects of paraconsistent systems and applications. Logique et Anal. **157**, 83–96 (1997)
3. Abe, J.M., Lopes, H.F.S., Anghinah, R.: Paraconsistent artificial neural network and alzheimer disease: a preliminary study. Dement. Neuropsychol. **3**, 241–247 (2007)
4. Anghinah, R.: Estudo da densidade espectral e da coerência do eletrencefalograma em indivíduos adultos normais e com doença de Alzheimer provável. Ph.D. thesis (in Portuguese), FMUSP, São Paulo (2003)
5. Ansari, D., Karmiloff-Smith, A.: Atypical trajectories of number development: a neuroconstructivist perspective. Trends Cogn. Sci. **12**, 511–516 (2002)

6. Amaral, F.V.: Paraconsistent mammography image attributes classifier in breast cancer diagnosis: based on Paraconsistent Artificial Neural Network. Ph.D. thesis (in Portuguese) UNIP, Brazil (2013)
7. Barre-Sinoussi, F., et al.: Isolation of a T-lymphotropic retrovirus from a patient at risk for Acquired Immune Deficiency Syndrome (AIDS). Science, vol. 220, pp. 868–871 (1983)
8. Claus, J.J., Strijers, R.L.M., Jonkman, E.J., Ongerboer De Visser, B.W., Jonker, C., Walstra, G.J.M., Scheltens, P., Gool, W.A.: The diagnostic value of EEG in mild senile Alzheimer´s disease. Clin. Neurophysiol. **18**, (15–23) (1999)
9. Crevel, H., Gool, W.A., Walstra, G.J.M.: Early diagnosis of dementia: which tests are indicated? What are their costs. J. Neurol. **246**, 73–78 (1999)
10. Da Silva Filho, J.I., Abe J.M., Torres, G.L.: Inteligência Artificial com as Redes de Análises Paraconsistentes, (in Portuguese) LTC-Livros Técnicos e Científicos Editora S.A., São Paulo, 313 pág. (2008)
11. Da Silva Filho, J.I., Torres, G.L., Abe, J.M.: Uncertainty treatment using paraconsistent logic—introducing paraconsistent artificial neural networks, vol. 211. IOS Press, Netherlands, 328 pp. (2010), ISBN 978-1-60750-557-0. doi:10.3233/978-1-60750-558-7-I
12. Duffy, F.H., Albert, M.S., Mcnulty, G., Garvey, A.J.: Age differences in brain electrical activity of healthy subjects. Ann. Neural. **16**, 430–438 (1984)
13. Grant, A.D., Cock, K.M.D.: ABC of AIDS—HIV infection and AIDS in the developing world. BMJ **322**, 1475–1478 (2001)
14. Gotman, J., Gloor, P., Ray, W.F.: A quantitative comparison of traditional reading of the EEG and interpretation of computer-extracted features in patients with supra tentorial brain lesions. Electroencephalogr. Clin. Neurophysiol. **38**, 623–639 (1975)
15. Gotman, J.: The use of computers in analysis and dysplasia of EEG and evoked potentials. In: Daly, D.D., Pedley, A.T. (eds.) Current Practice of Clinical Electroencephalography, 2nd edn, pp. 51–84. Raven Press, New York (1990)
16. Kwak, Y.T.: Quantitative EEG findings in different stages of Alzheimer's disease. J. Clin. Neurophysiol. **23**(5), 456–461 (2006)
17. Klimeshc, W.: EEG alpha and theta oscillations reflect cognitive and memory performance: a review and analysis. Brain Res. Ver. **29**, 169–195 (1999)
18. Kocyigit, Y., Alkan, A., Erol, H.: Classification of EEG recordings by using fast independent component analysis and artificial neural network. J. Med. Syst. Springer Neth. **32(**1), 17–20 (2008)
19. Lesser, R.P.: Guidelines committees of American electroencephalographic society. J. Clin. Neurophysiol. **11**, 09–11 (1994)
20. Lopes, H.F.S.: Aplicação de redes neurais artificiais paraconsistentes como método de auxílio no diagnóstico da doença de Alzheimer. M.Sc. Dissertation (in Portuguese), Faculdade de Medicina-USP, São Paulo, 473 pp. (2009)
21. Niedermeyer, E., da Silva, F.L.: Electroencephalography, 5th edn. Lippincott Williams & Wilkins, Philadelphia (2005)
22. Nuwer, M.R., Comi, G., Emerson, R., Fuglsang-Frederiksen, J., GuériT, M., Hinrichs, H., Ikeda, A., Luccas, F.J.C., Rappelsberger, P.: IFCN standards for digital recording of clinical EEG. Electroencephalogr. Clin. Neurophysiol. **106**, 259–261 (1998)
23. Nitrini, R., Caramelli, P., Bottino, C.M., Damasceno, B.P., Brucki, S.M., Anghinah, R.: Academia Brasileira de Neurologia. Diagnosis of Alzheimer's disease in Brazil: diagnostic criteria and auxiliary tests. Recommendations of the scientific department of cognitive neurology and aging of the brazilian academy of neurology. Arq Neuropsiquiatr. **63**(3A), 713–719 (2005)
24. Orrison, W.W., Lewine, J.D., Sanders, J.A., Hartshorne, M.F.: Functional brain imaging, pp. 327–368. Mosby, St Louis (1995)
25. Prusiner, S.R.: Discovering the cause of AIDS. Science **298**(5599), 1726–1727 (2002)
26. Santos, P.C.C.: Banco de Dados Inteligente e Ferramentas Associadas de Seqüências, Mutações e Resistências ao Antiretrovirais do Vírus HIV. Transl. (in Portuguese) Ph.D. thesis, Institute of Biomedical Sciences—USP, São Paulo, Brazil, 183 pp (2010)

27. Shafer, R.W., Schapiro, J.M.: HIV-1 drug resistance mutations: an update framework for the second decade of HAART. AIDS, vol. 10, pp. 67–84 (rev. 2008)
28. Shafer, R. W., Dupnik, K., Winters, M.A., Eshlemann, S.H.: HIV-1 reverse transcriptase and protease sequencing for drug resistance studies. In: Kuiken, C., et al. (ed.) HIV Sequence Compendium. Theoretical Biology and Biophysics, Los Alamos, pp. 83–133 (2001)
29. Souza, S., Abe, J.M., Nakamatsu, K.: MICR automated recognition based on paraconsistent artificial neural networks, vol. 22, pp. 170–178. Procedia Computer Science, Elsevier (2013)
30. Wong, P.K.H.: Introduction to brain topography, pp. 150–162. Plenum Press, New York (1991)
31. WIGG, M.D.: Virus da imunodeficiência humana. In: Santos, N.S.O., Romanos, M.T.V., Wigg, M.D. (eds.) Introdução à virologia humana. Guanabara Koogan, Rio de Janeiro, pp. 410–447 (2008)