Available online at [www.sciencedirect.com](http://www.sciencedirect.com/)

[Electronic Notes in Theoretical Computer Science 315 (2015) 3–16](http://dx.doi.org/10.1016/j.entcs.2015.06.002)

[www.elsevier.com/locate/entcs](http://www.elsevier.com/locate/entcs)

A Single Proof of Classical Behaviour in da Costa’s *Cn* Systems

Mauricio Osorio,[1](#_bookmark0) Jos´e Abel Castellanos[2](#_bookmark1)

*Departamento de Actuar´ıa, F´ısica y Matem´aticas Universidad de las Am´ericas Puebla*

*Cholula, Puebla, Mexico*

**Abstract**

A strong negation in da Costa’s *Cn* systems can be naturally extended from the strong negation (*¬∗*) of *C*1. In [[6](#_bookmark23)] Newton da Costa proved the connectives *{→, ∧, ∨, ¬∗}* in *C*1 satisfy all schemas and inference rules of classical logic. In the following paper we present a proof that all logics in the *Cn* herarchy also behave classically as *C*1. This result tell us the existance of a common property among the paraconsistent family of logics created by da Costa.

*Keywords:* Paraconsistent logic, *Cn* systems, Strong negation

# Introduction

According to the authors in [[6](#_bookmark23)] a paraconsistent logic is the underlying logic for inconsistent but non-trivial theories. In fact, many authors [[2](#_bookmark19),[1](#_bookmark18)] have pointed out *paraconsistency* is mainly due to the construction of a negation operator which satisfies some properties about classical logic, but at the same time do not hold the so called law of explosion *α, ¬α ▶ β* for arbitrary formulas *α, β*, as well as others [[6](#_bookmark23)].

A common misconception related to paraconsistent logics is the confusion be- tween triviality and contradiction. A theory *T* is *trivial* when any of the sentences in the language of *T* can be proven. We say that a theory *T* is *contradictory* if exists a sentence *α* in the language of *T* such that *T* proves *α* and *¬α*. Finally, a theory *T* is *explosive* if and only if *T* is *trivial* in the presence of a *contradiction*. We can see that *contradictoriness* and *triviality* are equivalent if and only if for the underlying logic the law of explosion is valid[[4](#_bookmark21)]. One of the greatest achievements of

1 Email: [osoriomauri@gmail.com](mailto:%20osoriomauri@gmail.com)

2 Email: [jose.castellanosjo@gmail.com](mailto:%20jose.castellanosjo@gmail.com)

<http://dx.doi.org/10.1016/j.entcs.2015.06.002>

1571-0661/© 2015 The Authors. Published by Elsevier B.V.

This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

paraconsistent logic is to provide a general framework to the study of inconsistent theories based on the distinction of contradiction and triviality.

Paraconsistent logics were born in two different ways. In 1948, Jaskowski gave the following conditions that any paraconsistent logic should satisfy [[8](#_bookmark25)]:

J1. When applied to inconsistent systems it should not always entail their trivial- ization;

J2. It should be rich enough to enable practical inferences; J3. It should have an intuitive justification.

Also, in 1963, we can find a new approach given by da Costa, who independently de- fined a set of conditions that a paraconsistent logic should satisfy. These conditions are the following:

dC1. In these calculi the principle of non-contradiction, in the form *¬*(*α∧¬α*), should not be a valid schema;

dC2. From two contradictory formulae, *α* and *¬α*, it would not in general be possible to deduce any arbitrary formula *β*;

dC3. It should be simple to extend these calculi to corresponding predicate calculi; dC4. They should contain the most part of the schemata and rules of the classical

propositional calculus which do not interfere with the first conditions.

Nowadays we can find paraconsistent logics applications in many fields such as informatics, physics, medicine, etc. From Minsky’s comment we can see that para- consistent ideas are an approach in Artificial Intelligence [[10](#_bookmark27)]: *”But I do not believe that consistency is necessary or even desirable in a developing intelligent system. No one is ever completely consistent. What is important is how one handles para- dox or conflict, how one learns from mistakes, how one turns aside from suspected inconsistencies”*.

In physics the authors in [[11](#_bookmark28)] have established an approach to formalize concepts in quantum mechanics, the so called principle of superposition, via paraconsistent methods. In general most of scientific knowledge as theories can have inconsis- tencies. Most of the time scientist do not throw away these theories if they are successful in predicting results and describing phenomena [[4](#_bookmark21)].

In the literature we can find many proper paraconsistent logics [[3](#_bookmark20)] in the sense of da Costa. The most known paraconsistent logic is *C*1 which in [[6](#_bookmark23)] the author also introduces an increasingly weaker family/hierarchy of logics called *Cn*, for 1 *≤ n ≤ ω*. Also the authors mention that the strong negation defined in the da Costa’s *Cn* systems has all properties of the propositional classical negation.

Finding a strong negation in the *Cn* hierarchy is interesting because we can col- lapse a fragment of these logics into classical logic, that is, we can have a translation which provides an embedding of classical logic into any logic of this *Cn* system. This fact is mentioned in many papers [[6](#_bookmark23),[5](#_bookmark22)], on the other hand the proof does not explic- itly appears. In this paper we present an inductive proof about the relation between strong negation and classical behaviour in the *Cn* systems. The proof follows from three lemmas and two theorems. From this proof we can see that many properties in *C*1 can also hold in *Cn*, excluding the obvious ones.

The organization of this document is as follows: In Section 2 we present basic background in logic, including definitions of some basic properties (monotonicity, cut-elimination, deduction theorem) of the paraconsistent logic *Cω* that we are going to work with; In Section 3 we present a inductive proof about the classical behavior of the strong negation defined in the *Cn* systems; In Section 4 we study an extension

*∗*

of *Cω* called *Cω* where we show that we can find a same increasingly chain of weaker

logics as in *Cn* systems from the new extension; Finally, in Section 5, we present some conclusions about the proof presented.

# Background

We first introduce the syntax of logical formulas considered in this paper. Then we present a few basic definitions of how logics can be built to interpret the meaning of such formulas.

* 1. *Logic Systems*

We consider a formal (propositional) language built from: an enumerable set *L* of elements called *atoms* (denoted *a, b, c, ...*); the binary connectives *∧* (*conjun- tion*), *∨* (*disjunction*) and *→* (*implication*); and the unary connective *¬* (*negation*). Formulas (denoted *α, β, γ, ...*) are constructed as usual by combining these basic connectives together with the help of parentheses. We also use *α ↔ β* to abbreviate (*α → β*) *∧* (*β → α*). Finally, it is useful to agree on some conventions to avoid the use of many parenthesis when writing formulas in order to make easier the reading of complicated expressions. First, we may omit the outer pair of parenthesis of a formula. Second, the connectives are ordered as follows: *¬, ∧, ∨, →, ↔*, and paren- theses are eliminated according to the rule that, first, *¬* applies to the smallest formula following it, then *∧* is to connect the smallest formulas surrounding it, and so on.

We consider a *logic* simply as a set of formulas that (i) is closed under Modus Ponens (i.e. if *α* and *α → β* are in the logic, then so is *β*) and (ii) is closed under substitution (i.e. if a formula *α* is in the logic, then any other formula obtained by replacing all occurrences of an atom b in *α* with another formula *β* is also in the logic). The elements of a logic are called *theorems* and the notation *▶X α* is used to state that the formula *α* is a theorem of *X* (i.e. *α ∈ X*). We say that a logic *X* is *weaker than or equal to a logic Y* if *X ⊆ Y* , similarly we say that *X* is *stronger than or equal to Y* if *Y ⊆ X*.

* + 1. *Hilbert proof systems*

There are many different approaches that have been used to specify the meaning of logic formulas or, in other words, to define logics. In Hilbert style proof systems, also known as axiomatic systems, a logic is specified by giving a set of axioms (which is usually assumed to be closed under substitution). This set of axioms specifies, so to speak, the ”kernel” of the logic. The actual logic is obtained when this ”kernel” is closed with respect to some given inference rules which include Modus Ponens.

The notation *▶X α* for provability of a logic formula *α* in the logic *X* is usually extended within Hilbert style systems; given a theory Γ, we use Γ *▶X α* to denote the fact that the formula *α* can be derived from the axioms of the logic and the formulas contained in Γ by a sequence of applications of the inference rules.

As a example of a Hilbert style system we present next a logic that is relevant for our work.

*Cω* [[6](#_bookmark23)] is defined by the following set of axiom schemata: Pos1: *α →* (*β → α*)

Pos2: (*α → β*) *→* ((*α →* (*β → γ*)) *→* (*α → γ*))

Pos3: *α ∧ β → α*

Pos4: *α ∧ β → β*

Pos5: *α →* (*β → α ∧ β*) Pos6: *α →* (*α ∨ β*) Pos7: *β →* (*α ∨ β*)

Pos8: (*α → γ*) *→* ((*β → γ*) *→* (*α ∨ β → γ*)) *Cω*1: *α ∨ ¬α*

*Cω*2: *¬¬α → α*

Note that the first 8 axiom schemata somewhat constrain the meaning of the

*→, ∧* and *∨* connectives to match our usual intuitions. It is a well known result that in any logic satisfying Pos1 and Pos2, and with Modus Ponens as its unique inference rule, the *deduction theorem* holds [[9](#_bookmark26)].

**Theorem 2.1** *[*[*12*](#_bookmark29)*] Let* Γ *and* Δ *be two sets of formulas. Let θ, θ*1*, θ*2*, α and ψ be arbitrary formulas. Let ▶ be the deductive inference operator of Cω. Then the following basic properties hold.*

1. Γ *▶ α → α (identity theorem)*
2. Γ *▶ α implies* Γ *∪* Δ *▶ α (monotonicity)*
3. Γ *▶ α and* Δ*,α ▶ ψ then* Γ *∪* Δ *▶ ψ (cut)*
4. Γ*,θ ▶ α if and only if* Γ *▶ θ → α (deduction theorem)*
5. Γ *▶ θ*1 *∧ θ*2 *if and only if* Γ *▶ θ*1 *and* Γ *▶ θ*2 *(∧ - rules)*
6. Γ*,θ ▶ α and* Γ*, ¬θ ▶ α if and only if* Γ *▶ α (strong proof by cases)*

# Strong negation in *Cn* systems

We will start giving some basic definitions in order to understand concepts needed in the *Cn* hierarchy.

**Definition 3.1** ([[6](#_bookmark23)]) *αo* =*def ¬*(*α ∧ ¬α*). We will refer to (*o*) as the consistency operator.

In fact *αo* can be seen as a modal operator to the formula *α* that captures the idea of consistency/well - behavior in *C*1.

**Definition 3.2** ([[5](#_bookmark22)]) We recursively define *αn*, 0 *≤ n < ω* as follows:

1. *α*0 =*def α*
2. *αn*+1 =*def* (*αn*)*o*

**Definition 3.3** ([[5](#_bookmark22)]) We recursively define *α*(*n*), 1 *≤ n < ω* as follows:

1. *α*(1) =*def α*1
2. *α*(*n*+1) =*def α*(*n*) *∧ αn*+1

For the careful reader should not confuse *α*0 with *αo*. Basically *αn* represents *n* applications of the consistency operator (*o*) to the formula *α*, and *α*(*n*) represents a conjunction of *α*1*,..., αn*.

**Definition 3.4** ([[6](#_bookmark23)]) We define *Cn* as an extension of *Cω*, which includes the fol- lowing axiom schemas:

*Cn*1: *β*(*n*) *→* ((*α → β*) *→* ((*α → ¬β*) *→ ¬α*)

*Cn*2: (*α*(*n*) *∧ β*(*n*)) *→* ((*α → β*)(*n*) *∧* (*α ∨ β*)(*n*) *∧* (*α ∧ β*)(*n*))

Also, we can see that in *Cn*, the axiom *Cn*1 can be replaced by the axiom schema (*β ∧ ¬β ∧ β*(*n*)) *→ α*. Intuitively from *Cn*2 we see that *α*(*n*) propagates what we call *n-consisteny* in *Cn*. Finally we define a strong negation in both *C*1 and *Cn*.

**Definition 3.5** ([[6](#_bookmark23)]) The strong negations for *C*1 and *Cn* are defined as:

* 1. For *C*1: *¬∗α* =*def ¬α ∧ αo*
  2. For *Cn*: *¬*(*n*)*α* =*def ¬α ∧ α*(*n*)

**Lemma 3.6** *For all n ∈* N *we have that ¬*(*αn*) *▶C α*

*ω*

**Proof.** By induction on *n*.

1 *o*

Base case (n = 1). By Definition [3.2](#_bookmark4) we have that *▶Cω ¬*(*α* ) *↔ ¬*(*α* ). Also

*o*

by Definition [3.1](#_bookmark3), *▶Cω ¬*(*α* ) *↔ ¬*(*¬*(*α ∧ ¬α*)), we can expand the last formula

1

to *▶Cω ¬*(*α* ) *↔ ¬*(*¬*(*α ∧ ¬α*)). We can use axiom schema *¬¬α → α* to prove

*▶Cω*

*ω*

*¬*(*α*1) *→ α ∧ ¬α*, which is by axiom schema Pos3 we have *▶C*

*¬*(*α*1) *→ α*.

From this we apply *deduction theorem* to obtain *¬*(*α*1) *▶C*

*ω*

*α* as desired.

Inductive step. We assume by induction hypothesis that *¬*(*αn*) *▶C*

*ω*

*α* holds.

*n*+1 *n o*

Accordingly to Definition [3.2](#_bookmark4) we have that *▶Cω ¬*(*α* ) *↔ ¬*(*α* ) , which in fact

*n n n o*

is *▶Cω ¬¬*(*α ∧ ¬*(*α* )) *↔ ¬*(*α* ) . From the latter and using *Cω*2 axiom and

transitivity property we can prove that *¬*(*αn*+1) *▶C ¬*(*αn*), and with the inductive

*ω*

hypothesis we have that *¬*(*αn*+1) *▶C*

*ω*

*α*. *2* *2*

**Lemma 3.7** *For all n ∈* N *we have that ▶C α ∨ αn*

*ω*

**Proof.** We can see that *αn ▶C α ∨ αn*. On the other hand, due to Lemma [3.6](#_bookmark6)

*ω*

we have that *¬*(*αn*) *▶C*

*ω*

*α*, therefore *¬*(*αn*) *▶C*

*α ∨ αn*. Applying strong proof by

*n*

*ω*

cases (Theorem [2.1](#_bookmark2)) we have that *▶Cω α ∨ α* . *2* *2*

**Lemma 3.8** *For all n ∈* N *we have that ▶Cω α ∨ α*(*n*)

**Proof.** By induction on *n*.

Base case (n = 1). From Lemma [3.6](#_bookmark6) we have that *▶Cω α∨αo* holds when n = 1.

(*n*)

Inductive step. We assume by induction hypothesis that *▶Cω α ∨ α* holds.

We know from Lemma [3.7](#_bookmark7) that *▶Cω*

*ω*

*α ∨ αn*+1. Thus *▶C*

(*α ∨ α*(*n*)) *∧* (*α ∨ αn*+1).

*n*+1 (*n*)

Applying the *distributive law* to the last formula we have that *▶Cω α∨*(*α ∧α* ),

(*n*+1)

which in fact it is by definition *▶Cω α ∨ α* . *2* *2*

**Theorem 3.9 (Excluded Middle)** *In Cω, we have that ▶C α ∨ ¬*(*n*)*α*

*ω*

**Proof.** In *Cω* we have the following:

*▶C* (*α ∨ ¬*(*n*)*α*) *↔* (*α ∨* (*α ∧ α*(*n*)))

*ω*

*▶C* (*α ∨ ¬*(*n*)*α*) *↔* (*α ∨ ¬α*) *∧* (*α ∨ α*(*n*))

*ω*

*▶C* (*α ∨ ¬*(*n*)*α*) *↔ α ∨ α*(*n*)

*ω*

Therefore it is only necessary to check that *α ∨ α*(*n*) holds, but accordingly to the

*Lemma* [*3.8*](#_bookmark8)this is true. *2* *2*

The next two theorems follows from a similar proof in [[6](#_bookmark23)] where the author proved the same theorems in *C*1.

**Theorem 3.10 (Reductio Ad Absurdum)** *In Cn we have that:*

(Γ *∪ {α} ▶C β*)*,* (Γ *∪ {α} ▶C ¬β*)*,* (Γ *∪ {α} ▶C β*(*n*)) *⇒* Γ *▶C ¬α*

*n n n n*

**Proof.** Using *Deduction Theorem* we can prove the following from the hypothesis

*n*

given: Γ *▶Cn*

*α → β*(*n*), Γ *▶C*

*α → β* and Γ *▶Cn*

*α → ¬β*. By the transitive

(*n*)

rule and the axiom schema *▶Cn β →* ((*α → β*) *→* ((*α → ¬β*) *→ ¬α*)) we

have that Γ *▶Cn α →* ((*α → β*) *→* ((*α → ¬β*) *→ ¬α*)). By the application of *Modus Ponens* twice we have that Γ *▶Cn α → ¬α*. From this, using theorem

*▶Cn ¬α → ¬α* (as an instance of Identity theorem), and axiom schemas *▶Cn α ∨¬α* and *▶Cn* (*α → ¬α*) *→* ((*¬α → ¬α*) *→* ((*α ∨ ¬α*) *→ ¬α*)) we can conclude that Γ *▶Cn ¬α*. *2* *2*

**Theorem 3.11 (Explosive Principle)** *In Cn we have that:*

*▶C α →* (*¬*(*n*)*α → β*)

*n*

**Proof.** According to the strong negation definition [3.5](#_bookmark5), we have that:

*n*

*α, ¬*(*n*)*α, ¬β ▶C*

*n*

*¬α ∧ α*(*n*), therefore *α, ¬*(*n*)*α, ¬β ▶C*

*¬α* and *α, ¬*(*n*)*α, ¬β ▶C*

*α*(*n*). Also we have that *α, ¬*(*n*)*α, ¬β ▶C*

*n*

*n*

*α*. By the theorem [3.10](#_bookmark10) is easy to prove

that *α, ¬*(*n*)*α ▶C*

*n*

*¬¬β*. *Cn* contains the axiom schemata *¬¬α → α*, which it let us

prove that *α, ¬*(*n*)*α ▶C*

*n*

*β*. Finally, applying two times *deduction theorem* to the

(*n*)

last formula we have that *▶Cn α →* (*¬ α → β*). *2* *2*

**Theorem 3.12** *The connectives {→, ∧, ∨, ¬*(*n*)*} in Cn satisfy all the axiom schemata and inference rules in classical propositional calculus.*

**Proof.** Any logic in *Cn* extends the positive logic axioms from *Cω*. Then, it is only necessary observe that the following axiom (*¬*(*n*)*α → ¬*(*n*)*β*) *→* (*β → α*) holds in *Cn*

* + 1. *¬*(*n*)*α → ¬*(*n*)*β* Hypothesis
    2. *β* Hypothesis
    3. *β →* (*¬*(*n*)*β → α*) From *Theorem* [*3.11*](#_bookmark11)
    4. *¬*(*n*)*β → α* Modus Ponens (2, 3)
    5. *¬*(*n*)*α → α* Transitivity (1, 4)
    6. *α → α* Identity theorem
    7. (*α → α*) *→* ((*¬*(*n*)*α → α*) *→* ((*α ∨ ¬*(*n*)*α*) *→ α*)) Axiom Pos8
    8. (*¬*(*n*)*α → α*) *→* ((*α ∨ ¬*(*n*)*α*) *→ α*) Modus Ponens (6, 7)
    9. (*α ∨ ¬*(*n*)*α*) *→ α* Modus Ponens (5, 8)
    10. *α ∨ ¬*(*n*)*α* From Theorem [3.9](#_bookmark9)
    11. *α* Modus Ponens (10, 9)
    12. (*¬*(*n*)*α → ¬*(*n*)*β*)*,β ▶C α* 1-11

*n*

* + 1. (*¬*(*n*)*α → ¬*(*n*)*β*) *▶C β → α* Deduction Theorem(12)

*n*

* + 1. *▶C* (*¬*(*n*)*α → ¬*(*n*)*β*) *→* (*β → α*) Deduction Theorem(13)

*n*

*2 2*

The author in [[6](#_bookmark23)] shows the strong negation in *C*1 has all the properties of the classical negation. With the above proof we extend this result to the hierarchy of logics in *Cn* in the sense that also the strong negation defined for each logic in the hi- erarchy behaves like the classical negation. This result is interesting because strong negations exhibit the possibility to develop theories in these paraconsistent logics to be equivalent to the classical counterpart. People with the desire to study para- consistent theories that distinguishes between triviallity and explosiveness should avoid strong negations in their theories.

# Some Additional Results

*∗*

In this section we studied a new hierarchy that we called *Cn*. This hierarchy includes

*α → ¬¬α* to each calculi of the hierarchy *Cn*. Since *¬¬α → α* is valid in each *Cn* due to *Cω*1, then the so called Double Negation Elimination [[12](#_bookmark29)] is valid in this new hierarchy. This property allows us to introduce or eliminate a negation from a proof. It is interesting to notice that the Double Negation Elimination is not valid in Intuitionistic Logic due the lack of constructivism of the proof; on the other hand only one side of the property (*α → ¬¬α*) is valid in Intuitionism.

We consider really important to study extensions of paraconsistent logics in order to develop richer and stronger paraconsistent systems that could be used for any purpose, from application in artificial intelligence to quantum physics; opening more possibilities to engineers and scientist respectively. In this section we proved

*∗*

in a similar way of [[7](#_bookmark24)] that *Cn* is indeed a hierarchy. In [[7](#_bookmark24)] the authors recursively

defined valuations *Tn* starting with the logic **P**1. For the purpose of our proof we used **P**2 [[8](#_bookmark25)], in which *α → ¬¬α* is a valid formula.

**Definition 4.1** ([[8](#_bookmark25)]) Let **P**2 be the logic defined by the following truth tables, where 1 and 2 are the designated values:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *∧* | 1 | 2 | 3 |  | *∨* | 1 | 2 | 3 |  | *→* | 1 | 2 | 3 |  |  | *¬* |
| 1 | 1 | 1 | 3 |  | 1 | 1 | 1 | 1 |  | 1 | 1 | 1 | 3 |  | 1 | 3 |
| 2 | 1 | 1 | 3 |  | 2 | 1 | 1 | 1 |  | 2 | 1 | 1 | 3 |  | 2 | 2 |
| 3 | 3 | 3 | 3 |  | 3 | 3 | 1 | 3 |  | 3 | 1 | 1 | 1 |  | 3 | 1 |
|  |  |  | *∗* |  |  |  |  |  |  |  |  |  |  |  |  |  |

**Definition 4.2** Let *Cω* be an extension of *Cω* where the following axiom is included:

*∗*

*Cω*2 : *α → ¬¬α*

*∗*

Also, we define the hierarchy *Cn* just adding *α → ¬¬α* to each calculi of *Cn*.

In the following definition we introduce a valuation of the form of truth tables called *Tn*. These tables are valuations of *n* +2 values (in N) being the only non

*∗*

designated the greatest (n + 2). We recursively define these tables beginning with

2 *∗* 2 *∗ ∗*

the logic **P** , so *T*1 is **P** . For *T*2 we keep the values from the previous table *T*1 and

we add one more value, in this case 4, which will be the only non - designated value,

*∗*

notice that the value 3 in *T*2 is no longer a non - designated value. The valuation

with formulas involving this new value are stated in the following definition. We

*∗ ∗*

repeated this process to generate the table *Tn* from the table *Tn—*1.

*∗*

**Definition 4.3** ([[7](#_bookmark24)]) We define *Tn* as truth tables in the following way:

*∗* 2

(Base Case) *T*1 = **P**

*∗ ∗*

(Inductive Step) *Tn* is obtained from *Tn—*1 adding a new value *n* + 2 in the current

table as the only non - designated value. The mapping of formulas involving this new value is defined as follows:

*∗*

1. The element in row *α* and column *β* of the conjunction table of *Tn* is *max*(*α, β*).

*∗*

1. The element in row *α* and column *β* of the disjunction table of *Tn* is *min*(*α, β*).

*∗*

1. The element in row *α* and column *β* of the implication table of *Tn* is 1 if *α* = *β*,

and *β* otherwise.

*∗*

1. Negation: *Tn* gives us the following table:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| *α* | 1 | 2 | 3 | ... | n | n + 1 | n + 2 |
| *¬α* | n + 2 | 2 | 2 | ... | n - 1 | n | 1 |

In the table above, *n* = 1*,* 2*,..., n,n* + 1 are the designated values and *n* +2 is the only non - designated one. We will use the notation *vn*(*α*) to indicate the valuation of the formula *α* in the *Tnj* valuation.

We remind the reader to notice that the definition of conjunction and dis- junction for the new value is the maximum and minimum element respectively because the greatest value in the valuation is the non - designated value.

The intuitive idea behind the above negation table is basically permute the

values 1 and *n* + 2 in the table and ”shift” one value for the middle values, except from 2 which remains the same.

*∗*

**Lemma 4.4** *In Tn we have that vn*(*α*)= *vn*(*α ∧ ¬α*) *if vn*(*α*) *∈ {*3*,...,n* + 1*} and*

*n ≥* 2*.*

**Proof.** By Induction on *n*:

(Base Case *n* = 2) It is just enough to ckeck *T*2*j* valuation.

(Inductive Step) Assume *vn*(*α*)= *vn*(*α ∧ ¬α*). We see two possible cases.

*∗ ∗*

1. Case *vn*+1(*α*) *∈ {*3*,...,n* + 2*}*. In this case, since *Tn*+1 is obtained from *Tn*, we

have that *vn*+1(*α*)= *vn*+1(*α ∧ ¬α*) from the inductive hypothesis.

1. Case *vn*+1(*α*)= *n* + 3. If *vn*+1(*α*)= *n* +3 then *vn*+1(*¬α*) = 1. From this applying the new rule for conjunction *vn*+1(*α ∧ ¬α*) = *n* + 3, which is *vn*+1(*α*), therefore *vn*+1(*α*)= *vn*+1(*α ∧ ¬α*).

*2*

*∗*

**Lemma 4.5** *In Tn we have that:*

⎧⎪⎨1 *if vn*(*α*)=1

*vn*(*αo*)=

*n* +2 *if vn*(*α*)=2

**Proof.** By cases:

⎪⎩*vn*(*¬α*) *otherwise*

Case *vn*(*α*)= 1. If *vn*(*α*)=1 then *vn*(*¬α*)= *n* + 2. The new value for conjunction is applied, so we get *vn*(*α ∧ ¬α*) = *n* + 2. From this *vn*(*¬*(*α ∧ ¬α*)) = 1, which is *vn*(*αo*)= 1.

Case *vn*(*α*)= 2. If *vn*(*α*)=2 then *vn*(*¬α*) = 2. Looking up **P**2 - valuation we can see that *vn*(*α ∧ ¬α*) = 1, from this we have that *vn*(*¬*(*α ∧ ¬α*)) = *n* + 2, which is *vn*(*αo*)= 1.

Case *vn*(*α*) = *n* + 2. If *vn*(*α*) = *n* +2 then *vn*(*¬α*) = 1. The new value for conjunction is applied, hence *vn*(*α ∧ ¬α*)= *n* + 2. From this *vn*(*αo*)=1 as in the case *vn*(*α*) = 1. But also *vn*(*¬α*)= 1, so *vn*(*αo*)= *vn*(*¬α*) as desired.

*∗*

Case *vn*(*α*) *∈ {*3*,...,n* + 1*}*. If *n* = 1 it is just enough to check *T*1. For *n ≥* 2 we

can see from lemma [4.4](#_bookmark12) that *vn*(*α*)= *vn*(*α ∧ ¬α*), hence *vn*(*¬*(*α ∧ ¬α*)) = *vn*(*¬α*), which is *vn*(*αo*)= *vn*(*¬α*) as desired. *2*

*∗*

**Lemma 4.6** *In Tn we have that:*

*v* (*α*(*n*))= 1 *if vn*(*α*)=1 *or vn*(*α*)= *n* +2

*n*

*n* +2 *otherwise*

**Proof.** By cases:

Case *vn*(*α*) = 1. Due to lemma [4.5](#_bookmark13) *vn*(*αo*) = 1 we observe that the consistency operator maps 1 to 1. Furthermore, the conjunction *vn*(*αo ∧· · · ∧ αn*) only involves

*∗*

the value 1. Using *T*1 we can see that the latter evaluates to 1.

Case *vn*(*α*)= *n* + 2. From lemma [4.5](#_bookmark13) *vn*(*αo*)= *vn*(*¬α*) = 1. Therefore *vn*(*α*(*n*))=1 by the same reason of last case.

Case *vn*(*α*) = 2. We can see that *vn*(*α*(*n*))= *n* + 2, because *vn*(*αo*) involves the new value *n* + 2.

Case *vn*(*α*) *∈ {*3*,...,n* + 1*}*. From lemma [4.5](#_bookmark13) *vn*(*αo*) = *vn*(*α*) *−* 1. Let *vn*(*α*) = *λ*; from this applying (*λ −* 2) *− times* the consistency operator to *α* we obtain *vn*(*αλ—*2) = 2. In the next application of the consistency operator we will obtain *vn*(*αλ—*1)= *n* + 2 due to lemma [4.5](#_bookmark13). Hence *n* + 2 is involved in one of the conjuncts of *αo ∧ ··· ∧ αn*. Therefore *vn*(*α*(*n*))= *n* + 2. *2*

**Lemma 4.7** *(∀n ∈* N*) (*b *∗*

*C*

*n*

*αn)*

**Proof.** By cases:

*∗*

Case *n* = 1. It is only necessary to check *T*1.

Case *n ∈* N *\ {*1*}*. We claim that if *vn*(*α*) = *n* +1 then *vn*(*αn*) = *n* + 2. If

*vn*(*α*) = *n* +1 then *vn*(*αo*) = *vn*(*¬α*) = *vn*(*α*) *−* 1 = *n*, due to lemma [4.5](#_bookmark13). To

evaluate to 2 we will need to apply (*n −* 1) *− times* more the consistency operator. In the next application of the consistency operator we evaluate *vn*(*αn*)= *n* + 2 due to lemma [4.5](#_bookmark13). Therefore *vn*(*αn*)= *n* + 2. *2*

**Theorem 4.8** *▶ ∗*

*C*

*n*+1

*x entails ▶ ∗ x*

*n*

*C*

**Proof.** It is only necessary to verify that (*A*9)*n*+1 and (*A*10)*n*+1 are provable in *C∗*

*n*

We wil show that *▶ ∗*

*C*

*n*

*β*(*n*+1) *→* ((*α → β*) *→* ((*¬α → β*) *→ β*))

1. *β*(*n*) *→* ((*α → β*) *→* ((*¬α → β*) *→ β*))

*∗*

(*Cn*1)

1. *β*(*n*+1) Hypothesis
2. *β*(*n*+1) *↔* (*βn*+1 *∧ β*(*n*))
3. (*βn*+1 *∧ β*(*n*)) *→ β*(*n*) (Pos4)
4. *β*(*n*) Transitivity (3, 4) and Modus Ponens with 2
5. ((*α → β*) *→* ((*¬α → β*) *→ β*)) Modus Ponens(5, 1)
6. *▶ ∗*

*C*

*n*

*β*(*n*+1) *→* ((*α → β*) *→* ((*¬α → β*) *→ β*)) 1 - 6

We wil show that *▶ ∗*

*C*

*n*

(*α*(*n*+1) *∧ β*(*n*+1)) *→* (*α* ② *β*)(*n*+1), where ② *∈ {∧, ∨, →}*

1. (*α*(*n*+1) *∧ β*(*n*+1)) Hypothesis
2. (*α*(*n*+1) *∧ β*(*n*+1)) *→ α*(*n*+1) (Pos3)
3. (*α*(*n*+1) *∧ β*(*n*+1)) *→ β*(*n*+1) (Pos4)
4. *α*(*n*+1) Modus Ponens(1, 2)
5. *β*(*n*+1) Modus Ponens(1, 3)
6. *α*(*n*+1) *↔* (*αn*+1 *∧ α*(*n*))
7. *β*(*n*+1) *↔* (*βn*+1 *∧ β*(*n*))
8. *αn*+1 *∧ α*(*n*) Modus Ponens(4, 6)
9. *βn*+1 *∧ β*(*n*) Modus Ponens(5, 7)
10. (*αn*+1 *∧ α*(*n*)) *→ α*(*n*) (Pos4)
11. (*βn*+1 *Λ β*(*n*)) *→ β*(*n*) (Pos4)
12. *α*(*n*) Modus Ponens(8, 10)
13. *β*(*n*) Modus Ponens(9, 11)
14. *α*(*n*) *→* (*β*(*n*) *→* (*α*(*n*) *Λ β*(*n*))) (Pos5)
15. *β*(*n*) *→* (*α*(*n*) *Λ β*(*n*)) Modus Ponens(12, 14)
16. *α*(*n*) *Λ β*(*n*) Modus Ponens(13, 15)
17. (*α*(*n*) *Λ β*(*n*)) *→* (*α* ② *β*)(*n*)

*∗*

(*Cn*2)

1. (*α* ② *β*)(*n*) Modus Ponens(16, 17)
2. (*α* ② *β*)*n*+1 Lemma [4.4](#_bookmark12)
3. (*α* ② *β*)(*n*) *→* ((*α* ② *β*)*n*+1 *→* ((*α* ② *β*)(*n*) *Λ* (*α* ② *β*)*n*+1)) (Pos5)
4. (*α* ② *β*)*n*+1 *→* ((*α* ② *β*)(*n*) *Λ* (*α* ② *β*)*n*+1) Modus Ponens(18, 20)
5. (*α* ② *β*)(*n*) *Λ* (*α* ② *β*)*n*+1 Modus Ponens(19, 21)
6. (*α* ② *β*)(*n*+1)
7. *▶ ∗*

*C*

*n*

(*α*(*n*+1) *Λ β*(*n*+1)) *→* (*α* ② *β*)(*n*+1) 1 - 23

*2*

*∗ ∗*

**Theorem 4.9** *Tn is sound w.r.t. Cn.*

*∗*

**Proof.** We will prove that inference rules and all axiom schema in *Cn* are tautologies

*∗*

in the *Tn* valuation. All proofs in this sections are done by contradiction.

*∗*

Modus Ponens: We will show that *Tn* preserves Modus Ponens. We assume *vn*(*α*)

and *vn*(*α → β*) to be designated values. Suppose *v*(*β*) evaluates to *n* + 2 (the only

*∗*

non-designated value in *Tn*). Since *vn*(*α*) */*= *vn*(*β*) then *vn*(*α → β*)= *n* + 2. But

*∗*

*vn*(*α → β*) evaluates a designated value. Contradiction, therefore *Tn* preserves

Modus Ponens.

Pos1: We claim that: (*6α, β*)(*vn*(*α →* (*β → α*)) */*= *n* + 2).

We assume (*Eα, β*)(*vn*(*α →* (*β → α*)) = *n* + 2). Hence *vn*(*α*) */*= *vn*(*β → α*) = *n* + 2. From the latter *vn*(*β*) */*= *vn*(*α*)= *n* + 2. But *vn*(*α*) */*= *n* + 2. Contradiction. Therefore (*6α, β*)(*vn*(*α →* (*β → α*)) */*= *n* + 2).

Pos2: We claim that: (*6α, β, γ*)(*vn*((*α →* (*β → γ*)) *→* ((*α → β*) *→* (*α → γ*))) */*=

*n* + 2).

We assume (*Eα, β, γ*)(*vn*((*α →* (*β → γ*)) *→* ((*α → β*) *→* (*α → γ*))) = *n* + 2). Hence *vn*(*α →* (*β → γ*)) */*= *vn*((*α → β*) *→* (*α → γ*)) = *n* + 2. Also, from the last formula *vn*(*α → β*) */*= *vn*(*α → γ*)= *n* + 2. From the latter *vn*(*α*) */*= *vn*(*γ*)= *n* + 2. Since *vn*(*α*) */*= *n* + 2 and *vn*(*α →* (*β → γ*)) */*= *n* +2 then *vn*(*β → γ*) */*= *n* +2 due to Modus Ponens. Also, since *vn*(*α*) */*= *n* + 2 and *vn*(*α → β*) */*= *n* +2 then *vn*(*β*) */*= *n* + 2. Finally, because *vn*(*β*) */*= *n* + 2 and *vn*(*β → γ*) */*= *n* +2 then *vn*(*γ*) */*= *n* + 2. But *vn*(*γ*) = *n* + 2. Contradiction, therefore (*6α, β, γ*)(*vn*((*α →* (*β → γ*)) *→* ((*α → β*) *→* (*α → γ*))) */*= *n* + 2).

Pos3 and Pos4: We claim that: (*6α, β*)(*vn*((*αΛβ*) *→ α*) */*= *n*+2 and *vn*((*αΛβ*) *→*

*β*) */*= *n* + 2)

Suppose (*Eα, β*)(*vn*((*α Λ β*) *→ α*) = *n* + 2). Hence *vn*(*α Λ β*) */*= *vn*(*α*) = *n* + 2. Since the conjunction of *α* and *β* involves the non-designated value then *vn*(*α Λ β*) = *n* + 2, but *vn*((*α Λ β*) *→ α*) */*= *n* + 2. Contradiction, therefore (*6α, β*)(*vn*((*α Λβ*) *→ α*)= *n* + 2). The proof for (*6α, β*)(*vn*((*α Λβ*) *→ β*)= *n* + 2) is similar to the above proof.

Pos5: We claim that: (*6α, β*)(*vn*(*α →* (*β →* (*α Λ β*))) */*= *n* + 2).

Suppse (*Eα, β*)(*vn*(*α →* (*β →* (*α Λ β*))) = *n* + 2). Then *vn*(*α*) */*= *vn*(*β →* (*α Λ β*)) = *n* + 2. From the latter *vn*(*β*) */*= *vn*(*α Λ β*)= *n* + 2. Since *vn*(*α*) */*= *n* +2 and *vn*(*β*) */*= *n*+2 then *vn*(*αΛβ*) */*= *n*+2. Contradiction, therefore (*6α, β*)(*vn*(*α →* (*β →* (*α Λ β*))) */*= *n* + 2).

Pos6 and Pos7: We claim that: (*6α, β*)(*vn*(*α →* (*α V β*)) */*= *n* + 2 and *vn*(*β →*

(*α V β*)) */*= *n* + 2).

Suppose (*Eα, β*)(*vn*(*α →* (*α V β*)) = *n* + 2). That is *vn*(*α*) */*= *vn*(*α V β*)= *n* + 2. From the latter we can see that both *vn*(*α*) and *vn*(*β*) evaluate to *n* + 2. But *vn*(*α*) */*= *n* + 2. Contradiction. Therefore (*6α, β*)(*vn*(*α →* (*α V β*)) */*= *n* + 2. Furthemore a similar reasoning we can prove that (*6α, β*)((*vn*(*β →* (*α V β*)) */*= *n* + 2)).

Pos8: We claim that: (*6α, β, γ*)(*vn*((*α → γ*) *→* ((*β → γ*) *→* ((*α V β*) *→ γ*))) */*=

*n* + 2).

We assume that (*Eα, β, γ*)(*vn*((*α → γ*) *→* ((*β → γ*) *→* ((*αVβ*) *→ γ*))) = *n*+2). Hence *vn*(*α → γ*) */*= *vn*((*β → γ*) *→* ((*α V β*) *→ γ*)) = *n* + 2. From the latter *vn*(*β → γ*) */*= *vn*((*α V β*) *→ γ*)= *n* + 2. Also, from this last formula we have that *vn*(*α Vβ*) */*= *vn*(*γ*)= *n* + 2. From *vn*(*α Vβ*) */*= *n* + 2 at least one disjunct evaluates to different to *n* + 2. Suppose *vn*(*α*) */*= *n* + 2, then because of *vn*(*α → γ*) */*= *n* +2 we have that *vn*(*γ*) */*= *n* + 2. Also if *vn*(*β*) */*= *n* +2 then *vn*(*γ*) */*= *n* + 2 due to *vn*(*β → γ*) */*= *n* + 2. Contradiction, so (*6α, β, γ*)(*vn*((*α → γ*) *→* ((*β → γ*) *→* ((*α V β*) *→ γ*))) */*= *n* + 2).

(*Cω*1): We claim that: (*6α*)(*vn*(*α V чα*) */*= *n* + 2).

We assume that (*Eα*)(*vn*(*α V чα*)= *n* + 2). We see two possible cases:

Case 1: We assume that *vn*(*α*)= *vn*(*чα*)= *n* + 2. But this is impossible from

*∗*

*Tn* definition.

Case 2: We assume that *vn*(*α*) */*= *vn*(*чα*). We can see two possible sub cases:

*∗*

Sub Case 1: We assume that *vn*(*α*)= *n*+2. From *Tn* we have that *vn*(*чα*)= 1.

Thefore the disjunction evaluates to *vn*(*α V чα*) = 1. Contradiction.

*∗*

Sub Case 2:]We assume that *vn*(*чα*)= *n* + 2. From *Tn* we have that *vn*(*α*)= 1.

Thefore the disjunction evaluates to *vn*(*α V чα*) = 1. Contradiction.

In all possible cases we reached a contradiction. Therefore (*6α*)(*vn*(*α V чα*) */*=

*n* + 2).

(*Cω*2): We claim that: (*6α*)(*vn*(*ччα → α*) */*= *n* + 2).

*∗*

We assume that (*Eα*)(*vn*(*ччα → α*)= *n*+2). From *Tn* we have that *vn*(*ччα*) */*=

*vn*(*α*) and *v*(*α*)= *n*+2. From the latter we have that *vn*(*чα*) = 1. From the latter

*∗*

and *Tn* we have that *vn*(*ччα*) = *n* + 2, but *vn*(*ччα*) */*= *vn*(*α*). Contradiction.

Therefore (*6α*)(*vn*(*ччα → α*) */*= *n* + 2).

*∗*

(*Cω* 2): We claim that: (*6α*)(*vn*(*α → ччα*) */*= *n* + 2).

*∗*

We assume that (*Eα*)(*vn*(*α → ччα*)= *n* + 2). From *Tn* we have that *vn*(*α*) */*=

*vn*(*ччα*) and *v*(*ччα*)= *n*+2. From the latter we have that *vn*(*чα*) = 1. From the

*∗*

latter and *Tn* we have that *vn*(*α*)= *n* + 2, but *vn*(*ччα*) */*= *vn*(*α*). Contradiction.

Therefore (*6α*)(*vn*(*α → ччα*) */*= *n* + 2).

*∗*

(*Cn*1): We claim that (*6α, β*)(*vn*(*β*

*n* + 2).

(*n*)

*→* ((*α → β*) *→* ((*α → чβ*) *→ чα*))) */*=

We assume (*Eα, β*)(*vn*(*β*(*n*) *→* ((*α → β*) *→* ((*α → чβ*) *→ чα*))) = *n* + 2).

*∗*

From *Tn*

we have that *vn*

(*β*(*n*)) */*= *vn*

((*α → β*) *→* ((*α → чα*) *→ чα*)) and

*∗*

*vn*((*α → β*) *→* ((*α → чα*) *→ чα*)) = *n* + 2. From the latter and *Tn* definition we

have that *vn*(*α → β*) */*= *vn*((*α → чβ*) *→ чα*) and *vn*((*α → чβ*) *→ чα*)= *n* + 2.

*∗*

Also by *Tn* definition and the latter we have that *vn*(*α → чβ*) */*= *vn*(*чα*) and that

*vn*(*чα*)= *n* + 2. From the latter *vn*(*α*) = 1. Due to *vn*(*β*(*n*)) */*= *n* + 2, then due to lemma [4.6](#_bookmark14) we have that *vn*(*β*(*n*)) = 1. The latter implies that *vn*(*β*) = 1 or *vn*(*β*)= *n* + 2. From this we distinguish two possible cases:

Case 1: *vn*(*β*) = 1. If *vn*(*β*) = 1 then *vn*(*чβ*) = *n* + 2. we can easily see that

*vn*(*α → чβ*)= *n* + 2. But *vn*(*α → чβ*) */*= *vn*(*чα*). Contradiction.

Case 2: *vn*(*β*) = *n* + 2. If *vn*(*β*) = *n* +2 then *vn*(*α → β*) = *n* + 2. But

*vn*(*α → β*) */*= *vn*((*α → чβ*) *→ чα*). Contradiction.

In all possible cases we reached a contradiction, therefore (*6α, β*)(*vn*(*β*(*n*) *→*

((*α → β*) *→* ((*α → чβ*) *→ чα*))) */*= *n* + 2).

*∗*

(*Cn*2) We claim that (*6α, β*)(*vn*((*α*

② *∈ {Λ, V, →}*.

(*n*)

*Λ β*(*n*)

) *→* (*α* ② *β*)

(*n*)

) */*= *n* + 2), where

We assume that (*Eα, β*)(*vn*

((*α*(*n*) *Λ β*(*n*)) *→* (*α* ② *β*)(*n*)) = *n* + 2). From *T ∗*

definition we have that *vn*(*α*(*n*) *Λβ*(*n*)) */*= *vn*((*α* ② *β*)(*n*)) and that *vn*((*α* ② *β*)(*n*))= *n* + 2. From the latter and lemma [4.6](#_bookmark14) we have that *vn*(*α* ② *β*) *∈ {*2*,...,n* + 1*}*. Since ② *∈ {Λ, V, →}* we distinguish three possible cases:

*n*

Case 1: ② =*→*. In this case we see that *vn*(*α*) */*= *vn*(*β*), otherwise the impli- cation would evaluate to 1. Furthermore we see that *vn*(*α*) *∈ {*1*,...,n* + 2*}*

and *vn*(*β*) *∈ {*2*,...,n* + 1*}*. From the latter and lemma [4.6](#_bookmark14) we have that *vn*(*β*(*n*))= *n* + 2. Hence the conjunction evaluates to *vn*(*α*(*n*) *Λ β*(*n*))= *n* + 2. But *vn*(*α*(*n*) *Λ β*(*n*)) */*= *vn*((*α → β*)(*n*)). Contradiction.

Case 2: ② = *Λ*. We can see that: *vn*(*α*) *∈ {*1*,...,n* +1*}* y *vn*(*β*) *∈ {*1*,...,n* +1*}* and that is not the case *vn*(*α*)= *vn*(*β*) = 1. From the latter at least one of the conjuncts of *vn*(*α*(*n*) *Λ β*(*n*)) evaluates something different of 1. Due to lemma

[4.6](#_bookmark14) at least one of the conjuncts of *vn*(*α*(*n*) *Λ β*(*n*)) evaluates to *n* + 2, hence

*vn*(*α*(*n*) *Λ β*(*n*))= *n* + 2. But *vn*(*α*(*n*) *Λ β*(*n*)) */*= *vn*((*α → β*)(*n*)). Contradiction.

Case 3: ② = *V*. We can see that: *vn*(*α*) *∈ {*2*,...,n* +2*}* y *vn*(*β*) *∈ {*2*,...,n* +2*}* and that is not the case *vn*(*α*) = *vn*(*β*) = *n* + 2. From the latter we see that at least one of the conjuncts of *vn*(*α*(*n*) *Λ β*(*n*)) evaluates something differents of *n* + 2. Due to lemma [4.6](#_bookmark14) at least one of the conjuncts *vn*(*α*(*n*) *Λ β*(*n*)) evaluates to *n* + 2, hence *vn*(*α*(*n*) *Λ β*(*n*))= *n* + 2. But *vn*(*α*(*n*) *Λ β*(*n*)) */*= *vn*((*α → β*)(*n*)). Contradiction.

*2*

*∗*

**Theorem 4.10** *Cn is a hierarchy of paraconsistent logics.*

**Proof.** The result holds from the following reasons:

1. For each formula *x* such that *▶ ∗*

*C*

*n*+1

*∗*

*x* then *▶ ∗*

*n*

*C*

*∗*

*x*. (Theorem [4.8](#_bookmark16)).

1. Exists a sound valuation for *Cn*, in this case *Tn* (Theorem [4.9](#_bookmark17)), where (*Ex*)(*▶C∗*

*n*

*x*) and b

*∗*

*n*+1

*C*

*x*. That formula is for instance *αn*+1 (Lemma [4.7](#_bookmark15)).

*2*

# Conclusions

The presented work gives a general idea how to extend a property in *C*1 to *Cn*, mainly using inductive proofs. We know that all logics in the *Cn* system are strictly weaker than *C*1 [[6](#_bookmark23)], perhaps many of them share many things in common as a strong negation. The section 4 introduce a new hierarchy of paraconsistent logics called

*∗*

*C**n* which is a stronger chaing than *Cn*. This new hierarchy could be useful for

theories and applications where the axiom *α → ччα* is crucial. In the future should be interesting to investigate how much these logics are related each other among relevant properties.

# References

1. Ofer Arieli, Arnon Avron, and Anna Zamansky. Ideal paraconsistent logics. *Studia Logica*, 99(1-3):31– 60, 2011.
2. Jean-Yves B´eziau. Adventures in the Paraconsistent Jungle, CLE e-Prints, Vol. 4(1), 2004 (Section Logic).
3. W. A. Carnielli and J. Marcos. A taxonomy of **C**-Systems. In *Paraconsistency: The Logical Way to the Inconsistent, Proceedings of the Second World Congress on Paraconsistency (WCP 2000)*, number 228 in Lecture Notes in Pure and Applied Mathematics, pages 1–94. Marcel Dekker, Inc., 2002.
4. Walter Carnielli and Rodrigues Abilio. On the philosophical motivations for the logics of formal consistency and inconsistency.
5. Walter Alexandre Carnielli and Joa˜o Marcos. Limits for paraconsistent calculi. *Notre Dame Journal* *of Formal Logic*, 40(3):375–390, 1999.
6. Newton C. A. da Costa. On the theory of inconsistent formal systems. *Notre Dame Journal of Formal Logic*, 15(4):497–510, 10 1974.
7. Newton CA Da Costa, D´ecio Krause, and Ot´avio Bueno. Paraconsistent logics and paraconsistency.
8. Joao Marcos. On a problem of da costa. *Essays on the Foundations of Mathematics and Logic*, 2:53–69, 2005.
9. Elliott Mendelson. *Introduction to Mathematical Logic*. Wadsworth, Belmont, CA, third edition, 1987.
10. M. Minsky. A framework for representing knowledge. In P. Winston, editor, *The Psychology of Computer Vision*, pages 211–277. Mcgraw-Hill, New York, 1975.
11. C. de Ronde N. da Costa. The paraconsistent logic of quantum superpositions. *Foundations of Physics*, 43:845–858, 2013.
12. Mauricio Osorio, Jos´e Luis Carballido, and Claudia Zepeda. Revisiting Z. *Notre Dame Journal of Formal Logic*, 55(1):129–155, 2014.