N'_5 as an extension of G'_3

Mauricio Osorio[1] and José Luis Carballido[2]

[1] Universidad de las Américas - Puebla
CENTIA, Sta. Catarina Mártir, Cholula, Puebla, 72820 México
osoriomauri@gmail.com
[2] Benemérita Universidad Autóma de Puebla,
Facultad de Ciencias de la Computación, Puebla, México
jlcarballido7@gmail.com

(Paper received on November 28, 2010, accepted on January 28, 2011)

Abstract. We present some results related to the substitution theorem and the standard form of formulas in the 5-valued paraconsistent logic N'_5 introduced in [1]. This logic has two negation connectives and extends G'_3, a paraconsistent logic recently introduced [15]. We also show that N'_5 is not a maximal paraconsistent logic.

Keywords: *paraconsistent, strong negation, substitution theorem,* N'_5 *logic.*

1 Introduction

The present work can be placed in the context of paraconsistent logics. Briefly speaking, following Béziau [4], a logic is paraconsistent if it has a negation $\neg$, which is paraconsistent in the sense that $a, \neg a \nvdash b$, and at the same time has enough strong properties to be called a negation. Nevertheless, there is no paraconsistent logic that is unanimously recognised as a good paraconsistent logic [4].

In spite of this lack of agreement, paraconsistent logics have important applications, specifically [7] mention three applications in different fields: Mathematics, Artificial Intelligence and Philosophy. In relation to the second one, the authors mention that in certain domains, such as the construction of expert systems, the presence of inconsistencies is almost unavoidable (see for example [8]). An application that has not been fully recognized is the use of paraconsistent logics in non-monotonic reasoning. In this sense [14,16,17,18] illustrate such novel applications. Thus, the research on paraconsistent logics is far from being over.

A paraconsistent logic of particular interest to us is G'_3, which has been studied in [15,16,18]. In this paper we study a logic closely related to G'_3. G'_3 is a three-valued paraconsistent logic that can express both, the Lukasiewicz L_3 and the Gödel G_3 logics. In particular it is worth mentioning that it can also express classical logic. G'_3 can also be expressed in terms of the Lukasiewicz L_3 logic [15]. G'_3 can be defined in terms of an axiomatic system, in fact in [15] the authors present an axiomatization of G'_3 together with a completeness and robustness theorem: the tautologies of the multivaled version of G'_3 are the same as the theorems in the proof system version of it. The family of axioms presented in [15] to define G'_3 include all the axioms of C_ω, thus G'_3 can be considered an extension of C_ω. An important property of G'_3 and shared by the logic Z [4], is that it satisfies the substitution theorem. G'_3 also satisfies the deduction theorem and the De Morgans laws. Another important feature about G'_3 is the fact that the formula $(a \wedge \neg a) \rightarrow b$ is not a theorem, that is, G'_3 is a paraconsistent logic. This property allows G'_3, as well as some other paraconsistent logics $(C_\omega, Pac, P - FOUR)$, to be the formalism to define the p-stable semantics, a semantics adequate to represent knowledge [18].

The p-stable semantics has been introduced recently as a tool to represent knowledge and has found applications in areas such as argumentation theory [10], updates [19] and preferences [20]. An implementation of the p-stable semantics using open sources is described in [23]. For more information about applications of the p-stable semantics see [21,22].

We study some properties of a five-valued paraconsistent logic strongly related to G'_3, we call it N'_5. N'_5 is a logic with strong negation that has the property of being a conservative extension of G'_3, i.e. a formula is a tautology in G'_3 if and only if, it is a tautology in N'_5. N'_5 satisfies the substitution theorem and besides, with N'_5 one can easily express the 5-valued Nelson's logic $N5$ [11].

N'_5 has one more connective not present in G'_3, its strong negation, which makes the logic richer from the theoretical point of view and also gives to it more possibilities of applications in the area of knowledge representation. Specifically, we have empirical evidence that it may be possible to extend the p-stable semantics to a more expressive one by using N'_5 in the same way G'_3 is used to formalize the p-stable semantics. At this point it is worth to mention the fact that the strong negation connective can be used in knowledge representation to express the notion of "usually", so that semantics defined in terms of a logic having a strong negation are more suitable in certain applications that those based on logics without it. For more information about strong negation in this context see[13,24].

The structure of our paper is as follows. In section 2 we present some of the related work that has been done. Section 3 describes the general background needed for reading the paper, including the definition of C_ω logic, a paraconsistent logic whose axioms are also axioms of G'_3. In the same section we present the original definition of the three-valued logic G'_3. In Section 4 we present N'_5 logic, a five-valued logic with two negations, and show that each of these negations can not be expressed by a formula in terms of the other four connectives in the logic. In the same section we present Theorem 6, one of our main results, which establishes that our logic N'_5 is a conservative extension of G'_3. We also present a substitution theorem for N'_5 and introduce the concept of standard form. On Section 5 we present our conclusions and we address future work.

2 Related Work

Let us recall that there are many paraconsistent logics, however most of them are defined in terms of axiomatic systems. The family of paraconsistent logics

$C_n, 0 < n < \omega$ were introduced in [6], and have been very influential. Later they were generalized to stronger versions: $C_n^+, 0 < n < \omega$ [7]. All these logics are not many-valued, and are not maximal in the sense that they can be extended to other paraconsistent logics [5]. We are more interested in multivalued logics with a paraconsistent negation, a strong negation and a substitution theorem valid for a strong biconditional. Among some well studied paraconsistent logics we can mention Pac [2], however it does not have a strong negation and does not satisfy the standard substitution theorem, as the formula $\neg(a \to b) \leftrightarrow (a \wedge \neg b)$ shows: it is a tautology, but the formula $\neg\neg(a \to b) \leftrightarrow \neg(a \wedge \neg b)$ is not a tautology according to an interpretation that assigns values 1 and 0 to the atoms a and b respectively. $J3$ [9] is a paraconsistent logic that possesses a strong negation and extends Pac, but since it is a conservative extension of Pac it does not satisfy the substitution theorem either. In [5] the authors introduce new logical systems, $LFI1$ and $LFI2$ to handle inconsistent data. $LFI1$ is inter-definable with $J3$ and does not satisfy the standard substitution theorem [5]. We do not know whether any of these two logics satisfies a version of the substitution theorem for a strong biconditional. Both, $LFI1$ and $LFI2$, are maximal systems with a strong negation and are defined in terms of axioms.

3 Background

We assume that the reader has some familiarity with basic logic such as chapter one in [12].

We first introduce the syntax of logic formulas considered in this paper. Then we present a few basic definitions about how logics can be built to interpret the meaning of such formulas in order to, finally, give a brief introduction to several of the logics that are relevant for the results of our later sections.

3.1 Syntax of formulas

We consider a formal (propositional) language built from: an enumerable set $\mathcal{L}$ of elements called *atoms* (denoted a, b, c, …); the binary connectives $\wedge$ (*conjunction*), $\vee$ (*disjunction*) and $\to$ (*implication*); and the unary connective $\neg$ (*negation*). Formulas (denoted α, β, γ, …) are constructed as usual by combining these basic connectives together with the help of parentheses. We also use $\alpha \leftrightarrow \beta$ to abbreviate $(\alpha \to \beta) \wedge (\beta \to \alpha)$ and $\alpha \leftarrow \beta$ to abbreviate $\beta \to \alpha$. It is useful to agree on some conventions to avoid the use of so many parenthesis in writing formulas. This will make the reading of complicated expressions easier. First, we may omit the outer pair of parenthesis of a formula. Second, the connectives are ordered as follows: $\neg, \wedge, \vee, \to$, and $\leftrightarrow$, and parentheses are eliminated according to the rule that, first, $\neg$ applies to the smallest formula following it, then $\wedge$ is to connect the smallest formulas surrounding it, and so on.

3.2 Logic systems

We consider a *logic* simply as a set of formulas that, satisfies the following two properties: (i) is closed under modus ponens (i.e. if α and $\alpha \to \beta$ 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 still in the logic). The elements of a logic are called *theorems* and the notation $\vdash_X \alpha$ is used to state that the formula α is a theorem of X (i.e. $\alpha \in X$). We say that a logic X is *weaker than or equal to* a logic Y if $X \subseteq Y$, similarly we say that X is *stronger than or equal to* Y if $Y \subseteq X$.

Hilbert style proof systems There are many different approaches that have been used to specify the meaning of logic formulas, 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 the inference rule of modus ponens. In [15] the authors present an axiomatization of G'_3 , that includes all of the axioms of C_ω [6], (and in particular all of the axioms of positive logic). A slight variant of that axiomatization consists of all of the axioms of C_ω plus the following axioms:

E1 $(\neg\alpha \to \neg\beta) \leftrightarrow (\neg\neg\beta \to \neg\neg\alpha)$
E2 $\neg\neg(\alpha \to \beta) \leftrightarrow ((\alpha \to \beta) \wedge (\neg\neg\alpha \to \neg\neg\beta))$
E3 $\neg\neg(\alpha \wedge \beta) \leftrightarrow (\neg\neg\alpha \wedge \neg\neg\beta)$
E4 $(\beta \wedge \neg\beta) \to (- - \alpha \to \alpha)$
E5 $\neg\neg(\alpha \vee \beta) \leftrightarrow (\neg\neg\alpha \vee \neg\neg\beta)$

We observe that classical logic is obtained by adding to the set any of the formulas, $\alpha \to \neg\neg\alpha, \alpha \to (\neg\alpha \to \beta), (\neg\beta \to \neg\alpha) \to (\alpha \to \beta)$.

Multivalued logics An alternative way to define the semantics for a logic is by the use of truth values and interpretations. Multivalued logics generalize the idea of using truth tables that are used to determine the validity of formulas in classical logic. The core of a multivalued logic is its *domain* of values $\mathcal{D}$, where some of such values are special and identified as *designated*. Logic connectives (e.g. $\wedge$, $\vee$, $\to$, $\neg$) are then introduced as operators over $\mathcal{D}$ according to the particular definition of the logic.

An *interpretation* is a function $I : \mathcal{L} \to \mathcal{D}$ that maps atoms to elements in the domain. The application of I is then extended to arbitrary formulas by mapping first the atoms to values in $\mathcal{D}$, and then evaluating the resulting expression in terms of the connectives of the logic (which are defined over $\mathcal{D}$). A formula is said to be a *tautology* if, for every possible interpretation, the formula evaluates to a designated value. The most simple example of a multivalued logic is classical logic where: $\mathcal{D} = \{0, 1\}$, 1 is the unique designated value, and connectives are defined through the usual basic truth tables. If X is any logic, we write $\models_X \alpha$ to denote that α is a tautology in the logic X. We say that α is a logical consequence of a set of formulas $\Gamma = \{\varphi_1, \varphi_2, \ldots, \varphi_n\}$ (denoted by $\Gamma \models_X \alpha$) if $\bigwedge \Gamma \to \alpha$ is a tautology, where $\bigwedge \Gamma$ stands for $\varphi_1 \wedge \varphi_2 \wedge \ldots \wedge \varphi_n$.

Note that in a multivalued logic, so that it can truly be a *logic*, the implication connective has to satisfy the following property: for every value $x \in \mathcal{D}$, if there is a designated value $y \in \mathcal{D}$ such that $y \to x$ is designated, then x must also be a designated value. This restriction enforces the validity of modus ponens in the logic. The inference rule of substitution holds without further conditions because of the functional nature of interpretations and how they are evaluated.

In this paper we consider the *standard* substitution, here represented with the usual notation: $\varphi[\alpha/p]$ will denote the formula that results from substituting the formula α in place of the atom p, wherever it occurs in φ. Recall the recursive definition: if φ is atomic, then $\varphi[\alpha/p]$ is α when φ equals p, and φ otherwise. Inductively, if φ is a formula $\varphi_1 \# \varphi_2$, for any binary connective $\#$. Then $\varphi[\alpha/p]$ will be $\varphi_1[\alpha/p] \# \varphi_2[\alpha/p]$. Finally, if φ is a formula of the form $\neg\varphi_1$, then $\varphi[\alpha/p]$ is $\neg\varphi_1[\alpha/p]$.

3.3 The multivalued logic G'_3

As previously noted G'_3 can also be presented as a multivalued logic. Such presentation is given in [18]. In this form it is defined through a 3-valued logic with truth values in the domain $\mathcal{D} = \{0, 1, 2\}$ where 2 is the designated value. The evaluation functions of the logic connectives are then defined as follows: $x \wedge y = \min(x, y)$; $x \vee y = \max(x, y)$; and the $\neg$ and $\to$ connectives are defined according to the truth tables given in Table 1. We write $\models \alpha$ to denote that the formula α is a tautology, namely that α evaluates to 2 (the designated value) for every valuation. We say that α is a logical consequence of a set of formulas $\Gamma = \{\varphi_1, \varphi_2, \ldots, \varphi_n\}$ (denoted by $\Gamma \models \alpha$) if $\bigwedge \Gamma \to \alpha$ is a tautology, where $\bigwedge \Gamma$ stands for $\varphi_1 \wedge \varphi_2 \wedge \ldots \wedge \varphi_n$.

x	$\neg x$
0	2
1	2
2	0

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

Table 1. Truth tables of connectives in G'_3.

The next couple of results are facts we already know about the logic G'_3

Theorem 1. *[15] For every formula α, α is a tautology in G'_3 iff α is a theorem in G'_3.*

Theorem 2 (Substitution theorem for G'_3-logic). *[15] Let α, β and ψ be G'_3-formulas and let p be an atom. If $\alpha \leftrightarrow \beta$ is a tautology in G'_3 then $\psi[\alpha/p] \leftrightarrow \psi[\beta/p]$ is a tautology in G'_3.*

Corollary 1. *[15]Let α, β and ψ be G'_3-formulas and let p be an atom. If $\alpha \leftrightarrow \beta$ is a theorem in G'_3 then $\psi[\alpha/p] \leftrightarrow \psi[\beta/p]$ is a theorem in G'_3.*

Next, we present a new result, it gives an extension of G'_3, however the resulting logic does not satisfy the substitution theorem.

Theorem 3. *The G'_3 logic is not maximal, there exists at least one paraconsistent logic that contains properly all of the tautologies of G'_3.*

Proof. Let CG'_3 be the logic that results from G'_3 when we allow the values 1 and 2 to be designated, then it is clear that any formula that is a tautology in G'_3 is also a tautology in CG'_3. On the other hand the formula $((a \to b) \to a) \to a$ which is not a tautology in G'_3 as shown by a valuation that assigns the values 1 and 0 to a and b respectively, becomes a tautology in CG'_3 as it is easy to check.

To see that CG'_3 is paraconsistent, we note that an interpretation that assigns the values 1 and 0 to the atoms a and b respectively shows that the formula $(a \wedge \neg a) \to b$ is not a tautology. □

Let us observe that the substitution theorem that holds in G'_3 is not valid in the new logic CG'_3. The formula $[((a \to b) \to a) \to a] \leftrightarrow [(a \vee \neg a)]$ is a tautology in CG'_3, but the formula $\neg[((a \to b) \to a) \to a] \leftrightarrow \neg[(a \vee \neg a)]$ is not.

4 The multi-valued logic N'_5

We present N'_5, a 5-valued logic. We will use the set of values $\{-2, -1, 0, 1, 2\}$. Valid formulas evaluate to 2, the chosen designated value. The connectives $\wedge$ and $\vee$ correspond to the *min* and *max* functions in the usual way. For the other connectives, the associated truth tables are as follows:

$\to$	-2	-1	0	1	2
-2	2	2	2	2	2
-1	2	2	2	2	2
0	2	2	2	2	2
1	-1	-1	0	2	2
2	-2	-1	0	1	2

$\neg$	
-2	2
-1	2
0	2
1	2
2	-2

$\sim$	
-2	2
-1	1
0	0
1	-1
2	-2

$\leftrightarrow$	-2	-1	0	1	2
-2	2	2	2	-1	-2
-1	2	2	2	-1	-1
0	2	2	2	0	0
1	-1	-1	0	2	1
2	-2	-1	0	1	2

Table 2. Truth tables of connectives in N'_5.

We have defined 5 logical connectives, namely $N_c := \{\to, \wedge, \vee, \neg, \sim\}$. Formulas in this logic will also be referred to as N-formulas, they are built from this set of connectives. As usual, if α always evaluates to the designated value, then it is called a tautology. For example the formula $(\alpha \wedge \sim \alpha) \to \beta$ is a tautology for any formulas α and β. The formula $\sim \alpha \to \neg\alpha$ is also a tautology, that is why we will call the connective $\sim$ strong negation. On the other hand, the formula $(\alpha \wedge \neg\alpha) \to \beta$ is not a tautology, a fact easy to verify.

Let us note that N'_5 logic is in some way similar to N_5 logic (see [11] for more information on N_5). The only difference is that in N_5, $\neg 1 = -1$, but in N'_5, $\neg 1 = 2$. Moreover, with N'_5 logic we can express N_5 logic.

Remark 1. N'_5 logic can express N_5 logic.

For there are at least two ways of expressing the N_5 formula $\neg\alpha$ in terms of the connectives of N_5'. These are given by the expressions $\alpha \to \sim\alpha$ and $\alpha \to (\neg\alpha \wedge \neg\neg\alpha)$. In particular $\neg\alpha \wedge \neg\neg\alpha$ expresses the N_5 constant $\bot$.

It is important to note that N_5' is in fact a paraconsistent extension of Nelson $N5$ logic, it is the slight difference in the definition of the connective $\neg$, which makes the difference between the two logics. In fact the next result holds.

Theorem 4. *N_5' is a conservative extension of N_5. A N_5-formula A is a tautology in N_5' if and only if it is a tautology in N_5.*

We will use the abbreviation $\alpha \leftrightarrow \beta := (\alpha \to \beta) \wedge (\beta \to \alpha)$.

The next proposition observes the fact that the connective $\neg$ can not be expressed in terms of the other 4 connectives:

Proposition 1. *There is no formula $\beta(a)$ in N_5' containing only the atom a and connectives in $\{\sim, \wedge, \vee, \to\}$, such that $\models \neg a \leftrightarrow \beta(a)$.*

Proof. First we notice that the connective $\neg$ gives different signs to the truth values 1 and 2. We show that any formula built with the other 4 connectives gives the same sign to the truth values 1 and 2. We apply induction on the number of connectives in the formula $\beta(a)$. Let v be a valuation such that $v(a) = 1$, then $v(\sim a) = -1$, $v(a \wedge a) = 1$, $v(a \vee a) = 1$ and $v(a \to a) = 2$. If v is a valuation such that $v(a) = 2$, then $v(\sim a) = -2$, $v(a \wedge a) = 2$, $v(a \vee a) = 2$ and $v(a \to a) = 2$.

Let ϕ, η be two N-formulas of the atom a. Let $\phi(i)$, $\eta(i)$ the truth value of the formulas for a given valuation v for which $v(a) = i$. Let us assume that $\phi(1)$ and $\phi(2)$ are both positive or negative, and also $\eta(1)$ and $\eta(2)$ are both positive or negative. Then according to the truth tables of the connectives $\to, \wedge, \vee, \sim$, we have:

$\phi(i) \to \eta(i)$ is positive for both $i \in \{1,2\}$, or is negative for both $i \in \{1,2\}$
$\phi(i) \vee \eta(i)$ is positive for both $i \in \{1,2\}$, or is negative for both $i \in \{1,2\}$
$\phi(i) \wedge \eta(i)$ is positive for both $i \in \{1,2\}$, or is negative for both $i \in \{1,2\}$
$\sim\phi(1), \sim\phi(2)$ are both positive or both negative.

We see from this and the definition of the connective $\leftrightarrow$, that the formula $\neg a \leftrightarrow \beta(a)$ does not evaluate to 2 for every interpretation. □

A similar result holds for the connective $\sim$.

Proposition 2. *There is no formula $\beta(a)$ in N_5' containing only the atom a and connectives in $\{\neg, \wedge, \vee, \to\}$, such that $\models \sim a \leftrightarrow \beta(a)$.*

Proof. Notice that the connective $\sim$ assigns the value 1 to the truth value -1. We show that any given formula of one single atom a built only in terms of the other 4 connectives assigns always one of the values $-1, -2, 2$ to the value -1.

We apply induction on the number of connectives in the formula $\beta(a)$. Let v a valuation such that $v(a) = -1$, then $v(\neg a) = 2$, $v(a \wedge a) = -1$, $v(a \vee a) = -1$ and $v(a \to a) = 2$.

Assume the result valid for any formula with less than n connectives. Let ϕ, η two formulas satisfying the induction hypothesis. Then $\phi(-1), \eta(-1) \in$

$\{2,-2,-1\}$. It follows from the tables for the connectives of N_5' that $\neg\phi(-1) \in \{2,-2\}$, and $(\phi \wedge \eta)(-1), (\phi \vee \eta)(-1), (\phi \rightarrow \eta)(-1) \in \{2,-2,-1\}$.

Thus, according to the truth table for $\leftrightarrow$, the formula $\sim a \leftrightarrow \beta(a)$ does not evaluate to 2 for every interpretation. □

Remark 2. Observe the following formulas in N_5':

1.− $\models \sim (\alpha \rightarrow \beta) \leftrightarrow \alpha \wedge \sim \beta$.
2.− $\models \sim (\alpha \wedge \beta) \leftrightarrow \sim \alpha \vee \sim \beta$.
3.− $\models \sim (\alpha \vee \beta) \leftrightarrow \sim \alpha \wedge \sim \beta$.
4.− $\models \sim\sim \alpha \leftrightarrow \alpha$.
5.− $\models \sim \neg\alpha \leftrightarrow \neg\neg\alpha$.
6.− $\models \sim \alpha \rightarrow \neg\alpha$.

What is important about this remark, is that these formulas have the same structure as those theorems of Nelson introduced previously in [11] for the construction of extensions, with strong negation, of intuitionistic logic; In fact, formulas $1-6$ show the classical-like behavior of strong negation, in particular, formula 6 honors the adjective *strong*.

Next we present a couple of theorems relative to our logic N_5'. The proofs can be found in [1]

Theorem 5. *Given α and β two formulas, then:*
1. If α and $\alpha \rightarrow \beta$ are tautologies, then β is also a tautology.
2. If α is a tautology, then $\neg\neg\alpha$ is also a tautology.

A very important result is that N_5' logic is a conservative extension of G_3' logic, as the following theorem shows.

Theorem 6. *For every G_3'-formula α, α is a tautology in N_5' iff α is a tautology in G_3'.*

This result together with theorem 4 shows that N_5' extends two different logics N_5 and G_3'.

4.1 Substitution

A particular feature of our N_5' logic is that the symbol $\leftrightarrow$ does not define a *congruential relation* on formulas, note that it can be the case that $\alpha \leftrightarrow \beta$ is a tautology, but $\sim\alpha \leftrightarrow \sim\beta$ is not. A particular example is the following: Take α_1 to be $\sim (a \rightarrow b)$ and α_2 to be $a \wedge \sim b$. Clearly $\alpha_1 \leftrightarrow \alpha_2$ is a tautology, but $\sim \alpha_1 \leftrightarrow \sim \alpha_2$ is not (take $I(a) = I(b) = 1$). This property also holds in N_5.

Thus, when we refer to *equivalence* of formulas, we will have to be more precise and make some particular considerations. The term *weak equivalence* will mean that $\alpha \leftrightarrow \beta$ is a tautology. There is a stronger notion of equivalence of N_5'-formulas, which we will call N_5'*-equivalence*, and it holds when both $\alpha \leftrightarrow \beta$ and $\sim\alpha \leftrightarrow \sim\beta$ are tautologies. For this purpose, we define a new connective $\Leftrightarrow$. We write $\alpha \Leftrightarrow \beta$ to denote the formula: $(\alpha \leftrightarrow \beta) \wedge (\sim\alpha \leftrightarrow \sim\beta)$. The reader can easily verify that $\alpha \leftrightarrow \beta$ is a tautology iff for every valuation v, $v(\alpha) > 0$ implies

$v(\alpha) = v(\beta)$ and by symmetry, $v(\beta) > 0$ implies $v(\alpha) = v(\beta)$, while $\alpha \Leftrightarrow \beta$ is a tautology iff for every valuation v, $v(\alpha) = v(\beta)$. This can be seen in the following truth tables:

$\leftrightarrow$	-2	-1	0	1	2
-2	2	2	2	-1	-2
-1	2	2	2	-1	-1
0	2	2	2	0	0
1	-1	-1	0	2	1
2	-2	-1	0	1	2

$\Leftrightarrow$	-2	-1	0	1	2
-2	2	1	0	-1	-2
-1	1	2	0	-1	-1
0	0	0	2	0	0
1	-1	-1	0	2	1
2	-2	-1	0	1	2

Table 3. Truth tables for the biconditionals.

The next two theorems are proved in [1].

Theorem 7 (Basic Substitution theorem). *Let α, β and ψ be N'_5-formulas and let p be an atom. If $\alpha \Leftrightarrow \beta$ is a tautology then $\psi[\alpha/p] \Leftrightarrow \psi[\beta/p]$ is a tautology.*

To be able to apply standard substitution we require N'_5-equivalence of formulas to hold. However, in certain cases this condition may be too strong. We are only interested in the particular cases where weak equivalence of formulas suffices for substituting. The first such a case is when substitution is not done inside the scope of a $\sim$ symbol.

Theorem 8. *Let α, β and ψ be N'_5-formulas and let p be an atom such that p does not occur in ψ within the scope of a $\sim$ symbol. If $\alpha \leftrightarrow \beta$ is a tautology then $\psi[\alpha/p] \leftrightarrow \psi[\beta/p]$ is a tautology.*

4.2 Standard form

We present the notion of a standard form of a formula.

Definition 1. *We define the function $S : N'_5 - formulas \to N'_5 - formulas$ as follows: If a is an atom and α, β are N'_5-formulas, then*

$$
\begin{array}{llll}
S(a) & = a, & S(\alpha \wedge \beta) & = S(\alpha) \wedge S(\beta), \\
S(\neg a) & = \neg a, & S(\alpha \vee \beta) & = S(\alpha) \vee S(\beta), \\
S(\sim a) & = \sim a, & S(\sim (\alpha \to \beta)) & = S(\alpha) \wedge S(\sim \beta), \\
S(\sim \neg \alpha) & = \neg\neg S(\alpha), & S(\sim (\alpha \wedge \beta)) & = S(\sim \alpha) \vee S(\sim \beta), \\
S(\neg \alpha) & = \neg S(\alpha), & S(\sim (\alpha \vee \beta)) & = S(\sim \alpha) \wedge S(\sim \beta), \\
S(\alpha \to \beta) & = S(\alpha) \to S(\beta). & S(\sim\sim \alpha) & = S(\alpha).
\end{array}
$$

Definition 2 (Standard Form). *An N'_5-formula φ is said to be in* standard form *if $S(\varphi) = \varphi$*

Intuitively a formula is in standard form if it has all occurrences of the $\sim$ connective just in front of an atom. Let us observe also that we did not define $S(\sim \neg\alpha)$ as $S(\alpha)$, we want the formula $\alpha \leftrightarrow S(\alpha)$ to be a tautology for any formula α and the formula $\sim \neg a \leftrightarrow a$ is not a tautology for an atom a.

Example 1. Take the formula $\varphi :=\sim (a \rightarrow \neg b) \wedge \sim c$. Then its standard form is $S(\varphi) := a \wedge b \wedge \sim c$.

The result we present in relation to standard forms affirms that $S(\phi)$ is a tautology if and only if ϕ is a tautology. In order to prove this, we point out the next properties about tautologies that are consequences of the definition of the $\leftrightarrow$ connective: If $A \leftrightarrow B$ and $B \leftrightarrow C$ are tautologies, then $A \leftrightarrow C$ is a tautology, if $A \leftrightarrow B$ and $C \leftrightarrow D$ are tautologies, then $A \wedge C \leftrightarrow B \wedge D$ and $A \vee C \leftrightarrow B \vee D$ are tautologies.

Theorem 9. *For any N_5'-formula ψ, $\psi \leftrightarrow S(\psi)$ is a tautology in N_5'.*

Proof. By structural induction:

Base case: One can easily check that the proposition is true for any of the twelve formulas that define the standard form if a, α, β are atoms.

Case 1) $\psi = \phi \wedge \eta$.
Let us assume that $\phi \leftrightarrow S(\phi)$ and $\eta \leftrightarrow S(\eta)$ are tautologies. Applying Theorem 8 to the formula $p \wedge \eta$ and the first tautology above we obtain that $\phi \wedge \eta \leftrightarrow S(\phi) \wedge \eta$ is a tautology. Now we apply the same Theorem to the formula $S(\phi) \wedge p$ and the second tautology and obtain that $S(\phi) \wedge \eta \leftrightarrow s(\phi) \wedge S(\eta)$ is a tautology. From this we conclude that $\phi \wedge \eta \leftrightarrow S(\phi) \wedge S(\eta)$ is a tautology.
The cases $\psi = \phi \vee \eta$ and $\psi = \phi \rightarrow \eta$ are done the same way by just replacing the corresponding connective.

Case 2) $\psi = \neg\phi$.
We apply Theorem 8 to the formula $\neg p$ and the tautology $\phi \leftrightarrow S(\phi)$ according to the inductive hypothesis to obtain the tautology $\neg\phi \leftrightarrow \neg S(\phi)$. This is equivalent to $\neg\phi \leftrightarrow S(\neg\phi)$.

Case 3) $\psi = \sim\neg\phi$.
By hypothesis $\phi \leftrightarrow S(\phi)$ is a tautology. Since $\sim\neg\phi \leftrightarrow \neg\neg\phi$ is a tautology and $S(\sim\neg\phi) = S(\neg\neg\phi)$, it is enough to prove that $\neg\neg\phi \leftrightarrow S(\neg\neg\phi)$ is a tautology. But $S(\neg\neg\phi) = \neg\neg S(\phi)$, and the result follows by induction hypothesis.

Case 4) $\psi = \sim(\phi \wedge \eta)$.
To prove that $\psi \leftrightarrow S(\psi)$ is a tautology is equivalent to prove, according to remark 2, that $\sim\phi \vee \sim\eta \leftrightarrow S(\sim\phi) \vee S(\sim\eta)$ is a tautology. But by induction hypothesis $\sim\phi \leftrightarrow S(\sim\phi)$ and $\sim\eta \leftrightarrow S(\sim\eta)$ are tautologies, from which the result follows.
The cases $\psi = \sim(\phi \vee \eta)$ and $\psi = \sim(\phi \rightarrow \eta)$ are done the same way by just replacing the corresponding connective.

Case 5) $\psi = \sim\sim\phi$.
By hypothesis $\phi \leftrightarrow S(\phi)$ is a tautology. Since $S(\sim\sim\phi) = \sim\sim S(\phi)$, this case reduces to prove that $\sim\sim\phi \leftrightarrow \sim\sim S(\phi)$ is a tautology. The result follows from the fact that $\alpha \leftrightarrow \sim\sim\alpha$ is a tautology for any formula α. □

As an immediate consequence we have the next important result that has been already presented in [1]:

Corollary 2. *For any N_5'-formula φ, φ is a tautology in N_5' iff $S(\varphi)$ is a tautology in N_5'.* □

4.3 N'_5 is not a maximal paraconsistent logic

Before ending this section, we present one more result, similar to theorem 3, about the fact that N'_5 can be extended.

Theorem 10. *The N'_5 logic is not maximal, there exists at least one paraconsistent logic that contains properly all of the tautologies of N'_5*

Proof. Let CN'_5 be the logic that results from N'_5 when we allow the values 1 and 2 to be designated, then it is clear that any formula that is a tautology in N'_5 is also a tautology in CN'_5. On the other hand the formula $((a \to b) \to a) \to a$ which is not a tautology in N'_5 as shown by a valuation that assigns the values 1 and 0 to a and b respectively, becomes a tautology in CN'_5 as it is easy to check.

To see that CN'_5 is paraconsistent, we note that an interpretation that assigns the values 1 and 0 to the atoms a and b respectively shows that the formula $(a \wedge \neg a) \to b$ is not a tautology. □

As in the case of G'_3 and CG'_3 logics, the substitution theorem that holds in N'_5 is not valid in the new logic CN'_5. As the reader can easily check, the formula $[((a \to b) \to a) \to a] \leftrightarrow [(a \vee \neg a)]$ is a tautology in CN'_5, but the formula $\neg[((a \to b) \to a) \to a] \leftrightarrow \neg[(a \vee \neg a)]$ is not.

5 Conclusions and Future Work

We introduced a 5-valued logic called N'_5. This logic is a conservative extension of the 3-valued logic G'_3, which accepts an axiomatization. Our logic N'_5 possesses two negations, one of them, ($\sim$), is a strong negation that makes the logic more expressive when representing knowledge. Results presented in this paper include a substitution theorem for N'_5, the preservation of tautologies by the standard form in N'_5 and the fact that the N'_5 logic can express N_5 logic, a logic that is suitable to express ASP. Finally, as mentioned in the introduction, we are interested in exploring possible ways of extending the p-stable semantics to a more expressive semantics by means of the use of a logic with strong negation, N'_5 seems to be a suitable candidate for the formalization of such a semantics.

6 Funding

This work was supported by the Consejo Nacional de Ciencia y Tecnología [CB-2008-01 No.101581].

References

1. J. Arrazola and M. Osorio and E. Ariza. The N'5 logic. Sixt Latin Americn Workshop On New Mrthods of reasoning, 25-35, 2010.
2. A. Avron. *Natural 3-valued logics- characterization and proof theory.* The Journal of Symbolic Logic, 56(1): 276-294, 1991.

3. A. Avron. *5-valued Non-deterministic Semantics for the Basic Paraconsistent Logic mCi.* School of Computer Science, Tel Aviv University 2008.
4. J. Y. Béziau. The Paraconsistent Logic Z. A possible solution to Jaskowski's problem. In Logic and logical philosophy, 15: 99–111, 2006.
5. W. A. Carnielli, J. Marcos and R. de Amo. Formal Inconsistency and evolutionary databases. Logic and Logical Philosophy, Vol 8, 2000, pages 115–152.
6. N. da Costa. On the theory of inconsistent formal systems. *Notre Dame Journal of Formal Logic*, 15(4):497–510, 1974.
7. N. da Costa, J. I. Béziau and O. A. Bueno. Aspects of Paraconsistent Logic. Bull: of the IGPL, 3(4):597–614, 1995.
8. N. da Costa, V. S. Subrahmanian. Paraconsistent logics as a formalism for reasoning about inconsistent knowledge bases. *Artificial Intelligence in Medicine*, 1: 167–174, 1989.
9. I. L. D'Ottaviano and N. da Costa. Sur un problme de Jaskowsky. *Compted Rendus de l'Academie de Sciences de Paris* Sr.A-B 270: 1349-1353, 1970
10. P. M.Dung. On the acceptability of arguments and its fundamental role in non-monotonic reasoning, logic programming and n-person games. In *Artificial Intelligence*, 77(2): 321–358, 1995.
11. M. Kracht On extensions of intermediate logics by strong negation evaluation. *The Journal of Philosophical Logic*, 27(1):49–73, 1998.
12. E.Mendelson. *Introduction of Mathematical Logic.* Wadsworth, Belmont, CA, third edition, 1987.
13. M.Ortiz and M. Osorio. Strong Negation and Equivalence in the Safe Belief Semantics. In *Journal of Logic and Computation* , 499 - 515, April, 2007.
14. M. Osorio. GluckG logic and its applications to non-monotonic reasoning. *LANMR*,286, 2007.
15. M. Osorio and J. L. Carballido. Brief study of G'_3 logic. *Journal of Applied Non-Classical Logic*, 18(4):79–103, 2008.
16. M. Osorio, J.Arrazola, and J. L. Carballido. Logical weak completions of paraconsistent logics. *Journal of Logic and Computation*, 18(6):913–940, 2008.
17. M. Osorio, J. A. Navarro, J. Arrazola, and V. Borja. Ground nonmonotonic modal logic S5: New results. *Journal of Logic and Computation*, 15(5):787–813, 2005.
18. M. Osorio, J. A. Navarro, J. Arrazola, and V. Borja. Logics with common weak completions. *Journal of Logic and Computation*, 16(6):867–890, 2006.
19. M. Osorio and C. Zepeda. Update sequences based on minimal generalized pstable models. In *MICAI*, pages 283–293, 2007.
20. M. Osorio and C. Zepeda. Pstable theories and preferences. In *Electronic Proceedings of the 18th International Conference on Electronics, Communications, and Computers (CONIELECOMP 2008)*, March, 2008.
21. M. Osorio, C. Zepeda, and H. Castillo. A formal design model for mechatronic systems. In *Proceedings of 19th International Conference on Electrical, Communications, and Computers 2009, Cholula, Puebla, Mexico*, pages 125–129, 2009.
22. M. Osorio, C. Zepeda, J. C. Nieves, and J. L. Carballido. G'3-stable semantics and inconsistency. *Special Issue of Computacin y Sistemas on Innovative Applications of AI, Revista Iberoamericana de Computación. ISSN: 1405-5546*, 13(1):75–86, Ed. IPN. Centro de Investigación en Computación, 2009.
23. S. Pascucci and A. Lopez. Implementing p-stable with simplification capabilities. *Revista Iberoamericana de Inteligencia Artificial*, 13(41), Spain, 2008.
24. D. Pearce. From here to there: Stable Negation in Logic Programming. *What is the negation?*, Kluver academic publishers, 161-181, 1998.