Classical Logic as a subclass of Neutrosophic Logic

[1]Angelo de Oliveira, [2]Marina Nogueira Carvalho de Oliveira
[1]Departamento Acadêmico de Ciência da Computação - Universidade Federal de Rondônia - Brasil
[2]Undergraduate Student in Civil Engineering - Universidade Tecnológica Federal do Paraná - Brasil
angelo@unir.br[1], marina.nogueira.co@gmail.com.br[2]

Abstract

It is customary in mathematics that almost all new developments maintain compatibility with what is already proved and accepted. Following this way, neutrosophic logic has the classical logic as subset. However, in mathematics, all the affirmations must be proved first to be accepted, so the claim that the neutrosophic logic encompass classical logic must be also proved. Thus, this paper show that the main properties of the classical logic hold when translated to neutrosophic form at propositional level.

Keywords: Neutrosophic Logic; Classical Logic.

1 Introduction

Neutrosophic Logic (NL) is a recent logic born in the middle of the nineties by Florentin Smarandache,[1] it is characterized by having another component added to the common (standard) components of any logic, namely, truth and falsity. This added component is called indeterminacy.

Formally, a propositional neutrosophic logic variable x is represented by the triple

$$x = (t, i, f)$$

where t is the degree of truth, i is the degree of indeterminacy and f is the degree of falsity.

In this way, neutrosophic logic clearly embodies indeterminacy as component that must be treated in its framework, while other logics this element is not explicit (though sometimes taken in account).

In a general way, the definition of the components representing the truth, indeterminacy and falsity are sets. Thus, the neutrosophic logic can be considered as (and indeed it is) a multivalued logic. Now it is time to present the formal definition.

Definition 1.1. Let T, I, F be sets (real standard or non-standard) in the non-standard unity interval $]^{-}0, 1^{+}[$, with

$$\begin{aligned} supT &= t\_sup, infT = t\_inf \\ supI &= i\_sup, infI = i\_inf \\ supF &= f\_sup, infF = f\_inf \end{aligned}$$

and

$$\begin{aligned} n\_sup &= t\_sup + i\_sup + f\_sup \\ n\_inf &= t\_inf + i\_inf + f\_inf \end{aligned}$$

Doi :10.5281/zenodo.3830094

Received: February 14, 2020 Revised: April15, 2020 Accepted: May 10, 2020

then the triplet $X = (T, I, F)$ represents the values of the truth (T), indeterminacy (I) and falsity (F) with respect of the variable X. The sets T, I and F are called neutrosophic components.

Additionally, the following restrictions must hold:

$$n\_sup = t\_sup + i\_sup + f\_sup \leq 3^+$$
$$n\_inf = t\_inf + i\_inf + f\_inf \geq^- 0$$

The use of real non-standard mathematics is just one of many topics that makes *neutrosophic logic* different from others types of logics. Here, should also be noted that the use of non-standard values allows the *neutrosophic logic* to distinguish between *absolute truth* (truth in all possible worlds, represented by 1^+) and *relative truth* (truth in one or just a few worlds, represented by 1). The same applies for *absolute falsity* (represented by $^-0$) and *relative falsity* (represented by 0).

The definition 1.1 is so general that the set (or subsets) does not necessarily have to be intervals, they can be any type of sets (discrete, continuous, open or closed or half-open/half-closed intervals, intersections or unions of the previous sets, etc.) in accordance with the given proposition. Also, a subset may have one element only in special cases of this logic.

The degree of generality is such that every time that someone wish to apply the neutrosophic logic to any specific matter, the operators and their meanings must be first defined. Besides, the most common case of neutrosophic logic use is real intervals or points in the real unitary interval.

Not being viable to list all possible nuances, it is time show how the neutrosophic logic encompass the classical logic.

2 Neutrosophic Logic and Classical Logic

Classical logic (or crisp logic) is nowadays a branch of mathematical logic which roots dated before Christ birth, being definitely developed from centuries XIX and XX from the works of Boole, Frege, Russell and Whitehead, just to name a few.

What constitutes a main characteristic of the classical logic is its bivalence, just two absolute values are considered, V (verum) for truth and F (falsum) for falsity, that can be conveniently substituted by 1 and 0 (or any two different signs). Thus, in the framework of the classical logic, indeterminacy cannot be considered (here is not being considered variants of the classical logic developed from the beginning of the XX century, such as modal logic, temporal logic and alike).

Now, let $V = (1, 0, 0)$ represent the truth value in classical logic, now represented in neutrosophic form. Also, let $F = (0, 0, 1)$ represent the falsity value. With these two values, it is easy to see how one can be the negation of the other, being enough to define the negation operator as:

$$\neg x = \neg(t, 0, f) = (f, 0, t)$$

thus, $\neg \underbrace{(1,0,0)}_{V} = \underbrace{(0,0,1)}_{F}$ and $\neg \underbrace{(0,0,1)}_{F} = \underbrace{(1,0,0)}_{V}$.

Likewise, if the operators $\vee$ (disjunction) and $\wedge$ (conjunction) are defined as:

$$(t_1, 0, f_1) \vee (t_2, 0, f_2) = \{\max\{t_1, t_2\}, 0, \min\{f_1, f_2\})$$
$$(t_1, 0, f_1) \wedge (t_2, 0, f_2) = \{\min\{t_1, t_2\}, 0, \max\{f_1, f_2\})$$

then, those operators have the same behavior of their classical counterparts, being given by the following table:

Table 1: Disjunction and conjunction operators in NL with behavior of the classical ones.

p_1	p_2	$\vee$	$\wedge$
$(1,0,0)$	$(1,0,0)$	$(1,0,0)$	$(1,0,0)$
$(1,0,0)$	$(0,0,1)$	$(1,0,0)$	$(0,0,1)$
$(0,0,1)$	$(1,0,0)$	$(1,0,0)$	$(0,0,1)$
$(0,0,1)$	$(0,0,1)$	$(0,0,1)$	$(0,0,1)$

In classical logic, the set operators $CS = \{\neg, \vee, \wedge\}$ is said to be complete, that is, all the other operators in classical logic can be defined based in these operators. However, someone must be careful if he/she wishes that the operators and properties in classical logic holds in NL. The properties that remain true with the already given definitions of negation, conjunction and disjunction are next given.

Theorem 2.1 (Double Negation). *Let $x = (t, 0, f)$ be a neutrosophic logical variable representing a classical logic variable. The double negation propriety remains true, that is*

$$\neg\neg x = x$$

Proof.

$$\neg\neg x = \neg(\neg x) = \neg(\neg(t, 0, f)) = \neg(f, 0, t) = (t, 0, f)$$

□

Theorem 2.2 (Commutativity). *Let $x = (t_1, 0, f_1)$ and $y = (t_2, 0, f_2)$ be neutrosophic logic variables representing classical logic variables. The operators $\vee$ and $\wedge$ are commutative.*

Proof.

$$\begin{aligned} x \vee y = (t_1, 0, f_1) \vee (t_2, 0, f_2) = (\max\{t_1, t_2\}, 0, \min\{f_1, f_2\}) = \\ (\max\{t_2, t_1\}, 0, \min\{f_2, f_1\}) = (t_2, 0, f_2) \vee (t_1, 0, f_1) = y \vee x \end{aligned}$$

$$\begin{aligned} x \wedge y = (t_1, 0, f_1) \wedge (t_2, 0, f_2) = (\min\{t_1, t_2\}, 0, \max\{f_1, f_2\}) = \\ (\min\{t_2, t_1\}, 0, \max\{f_2, f_1\}) = (t_2, 0, f_2) \wedge (t_1, 0, f_1) = y \wedge x \end{aligned}$$

□

Theorem 2.3 (Associativity). *Let $x = (t_1, 0, f_1)$, $y = (t_2, 0, f_2)$ and $z = (t_3, 0, f_3)$ be neutrosophic logic variables representing classical logic variables. The operators $\vee$ and $\wedge$ are associative.*

Proof.

$$\begin{aligned} x \vee (y \vee z) = &(t_1, 0, f_1) \vee ((t_2, 0, f_2) \vee (t_3, 0, f_3)) = (t_1, 0, f_1) \vee (\max\{t_2, t_3\}, 0, \min\{f_2, f_3\})\}) = \\ &(\max\{t_1, \max\{t_2, t_3\}\}, 0, \min\{f_1, \min\{f_2, f_3\}\}) = (\{\max\{t_1, t_2, t_3\}, 0, \min\{f_1, f_2, f_3\}) = \\ &(\max\{\max\{t_1, t_2\}, t_3\}, 0, \min\{\min\{f_1, f_2\}, f_3\}) = (\max\{t_1, t_2\}, 0, \min\{f_1, f_2\}) \vee (t_3, 0, f_3) = \\ &((t_1, 0, f_1) \vee (t_2, 0, f_3)) \vee (t_3, 0, f_3) = (x \vee y) \vee z \end{aligned}$$

$$\begin{aligned} x \wedge (y \wedge z) = &(t_1, 0, f_1) \wedge ((t_2, 0, f_2) \wedge (t_3, 0, f_3)) = (t_1, 0, f_1) \wedge (\min\{t_2, t_3\}, 0, \max\{f_2, f_3\})\}) = \\ &(\min\{t_1, \min\{t_2, t_3\}\}, 0, \max\{f_1, \max\{f_2, f_3\}\}) = (\{\min\{t_1, t_2, t_3\}, 0, \max\{f_1, f_2, f_3\}) = \\ &(\min\{\min\{t_1, t_2\}, t_3\}, 0, \max\{\max\{f_1, f_2\}, f_3\}) = (\min\{t_1, t_2\}, 0, \max\{f_1, f_2\}) \wedge (t_3, 0, f_3) = \\ &((t_1, 0, f_1) \wedge (t_2, 0, f_3)) \wedge (t_3, 0, f_3) = (x \wedge y) \wedge z \end{aligned}$$

□

Theorem 2.4 (Distributivity). *Let $x = (t_1, 0, f_1)$, $y = (t_2, 0, f_2)$ and $z = (t_3, 0, f_3)$ be neutrosophic logic variables representing classical logic variables. The operators $\vee$ and $\wedge$ are distributive.*

Proof. At first, this property does not seem to be valid owing to the fact that when it is translated into the definitions of the operators in NL, the left and right side yields different formulas.

For $x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$:

$$\begin{aligned} x \vee (y \wedge v) = x \vee ((t_2, 0, f_2) \wedge (t_3, 0, f_3)) = x \vee (\min\{t_2, t_3\}, 0, \max\{f_2, f_3\}) = \\ (t_1, 0, f_1) \vee (\min\{t_2, t_3\}, 0, \max\{f_2, f_3\}) = \\ (\max\{t_1, \min\{t_2, t_3\}\}, 0, \min\{f_1, \max\{f_2, f_3\}\}) \end{aligned}$$

and

$$\begin{aligned} (x \vee y) \wedge (x \vee z) = ((t_1, 0, f_1) \vee (t_2, 0, f_2)) \wedge ((t_1, 0, f_1) \vee (t_3, 0, f_3)) = \\ (\max\{t_1, t_2\}, 0, \min\{f_1, f_2\}) \wedge (\max\{t_1, t_3\}, 0, \min\{f_1, f_3\}) = \\ (\min\{\max\{t_1, t_2\}, \max\{t_1, t_3\}\}, 0, \max\{\min\{f_1, f_2\}, \min\{f_1, f_3\}\}) \end{aligned}$$

For $x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$:

$$x \wedge (y \vee z) = x \wedge ((t_2, 0, f_2) \vee (t_3, 0, f_3)) = x \wedge (\max\{t_2, t_3\}, 0, \min\{f_2, f_3\}) = (t_1, 0, f_1) \wedge (\max\{t_2, t_3\}, 0, \min\{f_2, f_3\}) = (\min\{t_1, \max\{t_2, t_3\}\}, 0, \max\{f_1, \min\{f_2, f_3\}\})$$

and

$$(x \wedge y) \vee (x \wedge z) = ((t_1, 0, f_1) \wedge (t_2, 0, f_2)) \vee ((t_1, 0, f_1) \wedge (t_3, 0, f_3)) = (\min\{t_1, t_2\}, 0, \max\{f_1, f_2\}) \vee (\min\{t_1, t_3\}, 0, \max\{f_1, f_3\}) = (\max\{\min\{t_1, t_2\}, \min\{t_1, t_3\}\}, 0, \min\{\max\{f_1, f_2\}, \max\{f_1, f_3\}\})$$

But now, checking the truth tables for $x \vee (y \wedge z)$ (Table 2) and $(x \vee y) \wedge (x \vee z)$ (Table 3), can be seen that both of the sides yields the same result.

Table 2: Truth table for $x \vee (y \wedge z)$.

x	y	z	$\max\{t_1, \min\{t_2, t_3\}\}$	$\min\{f_1, \max\{f_2, f_3\}\}$	Result
(1, 0, 0)	(1, 0, 0)	(1, 0, 0)	1	0	(1, 0, 0)
(1, 0, 0)	(1, 0, 0)	(0, 0, 1)	1	0	(1, 0, 0)
(1, 0, 0)	(0, 0, 1)	(1, 0, 0)	1	0	(1, 0, 0)
(1, 0, 0)	(0, 0, 1)	(0, 0, 1)	1	0	(1, 0, 0)
(0, 0, 1)	(1, 0, 0)	(1, 0, 0)	1	0	(1, 0, 0)
(0, 0, 1)	(1, 0, 0)	(0, 0, 1)	0	1	(0, 0, 1)
(0, 0, 1)	(0, 0, 1)	(1, 0, 0)	0	1	(0, 0, 1)
(0, 0, 1)	(0, 0, 1)	(0, 0, 1)	0	1	(0, 0, 1)

Table 3: Truth table for $(x \vee y) \wedge (x \vee z)$.

x	y	z	$\min\{\max\{t_1, t_2\}, \max\{t_1, t_3\}\}$	$\max\{\min\{f1, f2\}, \min\{f_1, f_3\}\}$	Result
(1, 0, 0)	(1, 0, 0)	(1, 0, 0)	1	0	(1, 0, 0)
(1, 0, 0)	(1, 0, 0)	(0, 0, 1)	1	0	(1, 0, 0)
(1, 0, 0)	(0, 0, 1)	(1, 0, 0)	1	0	(1, 0, 0)
(1, 0, 0)	(0, 0, 1)	(0, 0, 1)	1	0	(1, 0, 0)
(0, 0, 1)	(1, 0, 0)	(1, 0, 0)	1	0	(1, 0, 0)
(0, 0, 1)	(1, 0, 0)	(0, 0, 1)	0	1	(0, 0, 1)
(0, 0, 1)	(0, 0, 1)	(1, 0, 0)	0	1	(0, 0, 1)
(0, 0, 1)	(0, 0, 1)	(0, 0, 1)	0	1	(0, 0, 1)

Similarly, the same occurs for $x \wedge (y \vee z)$ (Table 4) and $(x \wedge y) \vee (x \wedge z)$ (Table 5).

Table 4: Truth table for $x \wedge (y \vee z)$.

x	y	z	$\min\{t_1, \max\{t_2, t_3\}\}$	$\max\{f_1, \min\{f_2, f_3\}\}$	Result
(1, 0, 0)	(1, 0, 0)	(1, 0, 0)	1	0	(1, 0, 0)
(1, 0, 0)	(1, 0, 0)	(0, 0, 1)	1	0	(1, 0, 0)
(1, 0, 0)	(0, 0, 1)	(1, 0, 0)	1	0	(1, 0, 0)
(1, 0, 0)	(0, 0, 1)	(0, 0, 1)	0	1	(0, 0, 1)
(0, 0, 1)	(1, 0, 0)	(1, 0, 0)	0	1	(0, 0, 1)
(0, 0, 1)	(1, 0, 0)	(0, 0, 1)	0	1	(0, 0, 1)
(0, 0, 1)	(0, 0, 1)	(1, 0, 0)	0	1	(0, 0, 1)
(0, 0, 1)	(0, 0, 1)	(0, 0, 1)	0	1	(0, 0, 1)

Table 5: Truth table for $(x \wedge y) \vee (x \wedge z)$.

x	y	z	$\min\{\max\{t_1, t_2\}, \max\{t_1, t_3\}\}$	$\max\{\min\{f_1, f_2\}, \min\{f_1, f_3\}\}$	Result
(1, 0, 0)	(1, 0, 0)	(1, 0, 0)	1	0	(1, 0, 0)
(1, 0, 0)	(1, 0, 0)	(0, 0, 1)	1	0	(1, 0, 0)
(1, 0, 0)	(0, 0, 1)	(1, 0, 0)	1	0	(1, 0, 0)
(1, 0, 0)	(0, 0, 1)	(0, 0, 1)	0	1	(0, 0, 1)
(0, 0, 1)	(1, 0, 0)	(1, 0, 0)	0	1	(0, 0, 1)
(0, 0, 1)	(1, 0, 0)	(0, 0, 1)	0	1	(0, 0, 1)
(0, 0, 1)	(0, 0, 1)	(1, 0, 0)	0	1	(0, 0, 1)
(0, 0, 1)	(0, 0, 1)	(0, 0, 1)	0	1	(0, 0, 1)

□

Theorem 2.5 (De Morgan's Laws). *Let $x = (t_1, 0, f_1)$ and $y = (t_2, 0, f_2)$ be neutrosophic logic variables representing classical logic variables. Then, it is true that $\neg(x \vee y) = \neg x \wedge \neg y$ and $\neg(x \wedge y) = \neg x \vee \neg y$, that is, the De Morgan's Laws remain valid.*

Proof.

$$\begin{aligned}\neg(x \vee y) &= \neg((t_1, 0, f_1) \vee (t_2, 0, f_2)) = \neg(\max\{t_1, t_2\}, 0, \min\{f_1, f_2\}) \\ &= (\min\{f_1, f_2\}, 0, \max\{t_1, t_2\}) = (f_1, 0, t_1) \wedge (f_2, 0, t_2) \\ &= \neg(t_1, 0, f_1) \wedge \neg(t_2, 0, f_2) = \neg x \wedge \neg y\end{aligned}$$

$$\begin{aligned}\neg(x \wedge y) &= \neg((t_1, 0, f_1) \wedge (t_2, 0, f_2)) = \neg(\min\{t_1, t_2\}, 0, \max\{f_1, f_2\}) \\ &= (\max\{f_1, f_2\}, 0, \min\{t_1, t_2\}) = (f_1, 0, t_1) \vee (f_2, 0, t_2) \\ &= \neg(t_1, 0, f_1) \vee \neg(t_2, 0, f_2) = \neg x \vee \neg y\end{aligned}$$

□

Definition 2.6 (Tautology). A tautology in NL representing a classical logic variable is an expression that is always equal to $(1, 0, 0)$, independent of the values of its components. A tautology may be represented by τ.

Definition 2.7 (Contradiction). A contradiction in NL representing a classical logic variable is an expression that is always equal to $(0, 0, 1)$, independent of the values of its components. A contradiction may be represented by γ.

Theorem 2.8 (Law of Noncontradiction). *Let $x = (t_1, 0, f_1)$ be a neutrosophic logic variable representing a classical logic variable. The Law of Noncontradicition, that is, $\neg(x \wedge \neg x)$, remains valid, being a tautology.*

Proof.

$$\begin{aligned}\neg(x \wedge \neg x) &= \neg((t_1, 0, f_1) \wedge \neg(t_1, 0, f_1)) = \neg((t_1, 0, f_1) \wedge (f_1, 0, t_1)) \\ &= \neg(min\{t_1, f_1\}, 0, \max\{f_1, t_1\}) = (\max\{f_1, t_1\}, 0, min\{t_1, f_1\})\end{aligned}$$

The result above is always equal to $(1, 0, 0)$. To comprehend this, consider that x is $(1, 0, 0)$ or $(0, 0, 1)$, exclusively. So, $t_1 = 1$ and $f_1 = 0$, or $t_1 = 0$ and $f_1 = 1$, such $\max\{f_1, t_1\}$ is always equal to 1 and $min\{t_1, f_1\}$ equal to 0. The following table indicates this as well.

Table 6: Table for the noncontradiction law.

x	$\neg x$	$\max\{f_1, t_1\}$	$min\{t_1, f_1\}$	Result
(1, 0, 0)	(0, 0, 1)	1	0	(1, 0, 0)
(0, 0, 1)	(1, 0, 0)	1	0	(1, 0, 0)

Thus, $\neg(x \wedge \neg x) = \tau$. □

Remark 2.9. The Law of Noncontradiction can be summarized in a few words: *it's false that one proposition can be true and false simultaneously.*

Theorem 2.10 (Law of Excluded Middle). *Let $x = (t_1, 0, f_1)$ be a neutrosophic logic variable representing a classical logic variable. The Law of Excluded Middle (*tertium non datur*), that is, $x \vee \neg x$, remains valid, being a tautology.*

Proof.

$$(x \vee \neg x) = (t_1, 0, f_1) \vee \neg(t_1, 0, f_1) = (t_1, 0, f_1) \vee (f_1, 0, t_1)$$
$$= (max\{t_1, f_1\}, 0, \min\{f_1, t_1\})$$

The result above is always equal to $(1, 0, 0)$. To comprehend this, consider that x is $(1, 0, 0)$ or $(0, 0, 1)$, exclusively. So, $t_1 = 1$ and $f_1 = 0$, or $t_1 = 0$ and $f_1 = 1$, such $\max\{t_1, f_1\}$ is always equal to 1 and $min\{f_1, t_1\}$ equal to 0. The following table reveals this as well.

Table 7: Table for the law excluded middle.

x	$\neg x$	$\max\{t_1, f_1\}$	$min\{f_1, t_1\}$	Result
(1, 0, 0)	(0, 0, 1)	1	0	(1, 0, 0)
(0, 0, 1)	(1, 0, 0)	1	0	(1, 0, 0)

Thereby, $x \wedge \neg x = \tau$. □

Remark 2.11. The Law of Excluded Middle can be summarized in a few words: *one proposition can be true or false, not being accepted another value*.

Theorem 2.12 (Identity Laws). *Let $x = (t_1, 0, f_1)$ be a neutrosophic logic variable representing a classical logic variable. It is true that: (i) $x \vee \gamma = x$, (ii) $x \wedge \gamma = \gamma$, (iii) $x \vee \tau = \tau$ and (iv) $x \wedge \tau = x$.*

Proof.
(i) $x \vee \gamma = (t_1, 0, f_1) \vee (0, 0, 1) = (\max\{t_1, 0\}, 0, \min\{f_1, 1\})$. As $t_1 \geq 0$, then $\max\{t_1, 0\} = t_1$ and as $f_1 \leq 1$, then $\min\{f_1, 1\} = f_1$. So, $(\max\{t_1, 0\}, 0, \min\{f_1, 1\}) = (t_1, 0, f_1) = x$.

(ii) $x \wedge \gamma = (t_1, 0, f_1) \wedge (0, 0, 1) = (\min\{t_1, 0\}, 0, \max\{f_1, 1\})$. As $t_1 \geq 0$, then $\min\{t_1, 0\} = 0$ and as $f_1 \leq 1$, then $\max\{f_1, 1\} = 1$. So, $(\min\{t_1, 0\}, 0, \max\{f_1, 1\}) = (0, 0, 1) = \gamma$.

(iii) $x \vee \tau = (t_1, 0, f_1) \vee (1, 0, 0) = (\max\{t_1, 1\}, 0, \min\{f_1, 0\})$. As $t_1 \leq 1$, then $\max\{t_1, 1\} = 1$ and as $f_1 \geq 0$, then $\min\{f_1, 0\} = 0$. So, $(\max\{t_1, 1\}, 0, \min\{f_1, 0\}) = (1, 0, 0) = \tau$.

(iv) $x \wedge \tau = (t_1, 0, f_1) \wedge (1, 0, 0) = (\min\{t_1, 1\}, 0, \max\{f_1, 0\})$. As $t_1 \leq 1$, then $\min\{t_1, 1\} = t_1$ and as $f_1 \geq 0$, then $\max\{f_1, 0\} = f_1$. So, $(\min\{t_1, 1\}, 0, \max\{f_1, 0\}) = (t_1, 0, f_1) = x$. □

Theorem 2.13 (Idempotent Laws). *Let $x = (t_1, 0, f_1)$ be a neutrosophic logic variable representing a classical logic variable. It is true that: (i) $x \wedge x = x$ and (ii) $x \vee x = x$.*

Proof.
(i) $x \wedge x = (t_1, 0, f_1) \wedge (t_1, 0, f_1) = (\min\{t_1, t_1\}, 0, \max\{f_1, f_1\}) = (t_1, 0, f_1) = x$.
(ii) $x \vee x = (t_1, 0, f_1) \vee (t_1, 0, f_1) = (\max\{t_1, t_1\}, 0, \min\{f_1, f_1\}) = (t_1, 0, f_1) = x$. □

Now, let's turn the attention to the operator $\rightarrow$ (conditional) and $\leftrightarrow$ (biconditional).

The conditional operator has, in the classical form (now represented in neutrosophic form), the following table:

Table 8: Table for the conditional operator.

x	y	$x \rightarrow y$
(1, 0, 0)	(1, 0, 0)	(1, 0, 0)
(1, 0, 0)	(0, 0, 1)	(0, 0, 1)
(0, 0, 1)	(1, 0, 0)	(1, 0, 0)
(0, 0, 1)	(0, 0, 1)	(1, 0, 0)

Theorem 2.14. *The conditional operator $\rightarrow$ is equivalent to $\neg(x \wedge \neg y)$, that in turn is equivalent to $\neg x \vee y$.*

Proof. Let $x = (t_1, 0, f_1)$ and $y = (t_2, 0, f_2)$ be neutrosophic logic variables representing classical logic variables, then:

$$\neg(p \wedge \neg q) = \neg((t_1, 0, f_1) \wedge (f_2, 0, t_2)) = \neg(\min\{t_1, f_2\}, 0, \max\{f_1, t_2\}) = (\max\{f_1, t_2\}, 0, \min\{t_1, f_2\})$$
$$\neg p \vee q = (f_1, 0, t_1) \vee (t_2, 0, f_2) = (\max\{f_1, t_2\}, 0, \min\{t_1, f_2\})$$

So, $\neg(p \wedge \neg q)$ and $\neg p \vee q$ are equals. Now, let's examine the truth table bellow:

Table 9: Table showing that $z \to y = \neg(p \wedge \neg q) = \neg p \vee q$.

x	y	$(\max\{f_1, t_2\}, 0, \min\{t_1, f_2\})$
(1, 0, 0)	(1, 0, 0)	(1, 0, 0)
(1, 0, 0)	(0, 0, 1)	(0, 0, 1)
(0, 0, 1)	(1, 0, 0)	(1, 0, 0)
(0, 0, 1)	(0, 0, 1)	(1, 0, 0)

From the truth table, it turns out that $\neg(p \wedge \neg q) = \neg p \vee q$ and equivalent to $x \to y$. □

The biconditional operator has, in the classical form (now represented in neutrosophic form), the following table:

Table 10: Table for the biconditional operator.

x	y	$x \leftrightarrow y$
(1, 0, 0)	(1, 0, 0)	(1, 0, 0)
(1, 0, 0)	(0, 0, 1)	(0, 0, 1)
(0, 0, 1)	(1, 0, 0)	(0, 0, 1)
(0, 0, 1)	(0, 0, 1)	(1, 0, 0)

From the table, it is notable that the result is true iff the two variables have the same value.

Theorem 2.15. *The biconditional operator $\leftrightarrow$ is equivalent to $(x \to y) \wedge (y \to x)$.*

Proof. Let $x = (t_1, 0, f_1)$ and $y = (t_2, 0, f_2)$ be neutrosophic logic variables representing classical logic variables, then:

$$x \to y = \neg x \vee y = (f_1, 0, t_1) \vee (t_2, 0, f_2) = (\max\{f_1, t_2\}, 0, \min\{t_1, f_2\})$$
$$y \to z = \neg y \vee x = (f_2, 0, t_2) \vee (t_1, 0, f_1) = (\max\{f_2, t_1\}, 0, \min\{t_2, f_1\})$$
$$(x \to y) \wedge (y \to x) = (\min\{\max\{f_1, t_2\}, \max\{f_2, t_1\}\}, 0, \max\{\min\{t_1, f_2\}, \min\{t_2, f_1\}\})$$

Let's examine the truth table:

Table 11: Table showing that $x \leftrightarrow y = (x \to y) \wedge y(\to x)$.

x	y	$x \to y$	$y \to x$	$(x \to y) \wedge (y \to x)$
1 0 0	1 0 0	1 0 0	1 0 0	1 0 0
1 0 0	0 0 1	0 0 1	1 0 0	0 0 1
0 0 1	1 0 0	1 0 0	0 0 1	0 0 1
0 0 1	0 0 1	1 0 0	1 0 0	1 0 1

From the truth table, it turns out that $x \leftrightarrow y$ is equivalent to $(x \to y) \wedge y(\to x)$. □

Furthermore, other operators could be defined, for example, the operators $\oplus$ (XOR), $\downarrow$ (NOR) and $\uparrow$ (NAND).

Definition 2.16 (Exclusive OR - XOR)**.** The operator $\oplus$ (Exclusive OR - XOR) has its behavior given by the following truth table:

Table 12: Table for the Exclusive OR operator.

x	y	$x \oplus y$
1 0 0	1 0 0	0 0 1
1 0 0	0 0 1	1 0 0
0 0 1	1 0 0	1 0 0
0 0 1	0 0 1	0 0 1

A rapid view on the table truth of the operator XOR is enough to perceive that XOR is the negation of biconditional operator, so $\neg(x \leftrightarrow y) = x \oplus y$. To observe this, consider the following.

Theorem 2.17. *The negation of $x \to y$, that is, $\neg(x \to y)$ is equivalent to $x \wedge \neg y$.*

Proof.

$$\neg(x \to y) = \neg(\neg x \vee y) = \neg(\neg x) \wedge \neg y = x \wedge \neg y$$

□

Now, given that $x \leftrightarrow x = (x \to y) \wedge (y \to x)$, then $\neg(x \leftrightarrow y) = \neg(x \to y) \vee \neg(y \to x) = (x \wedge \neg y) \vee (y \wedge \neg x)$. So, letting $x = (t_1, 0, f_1)$ and $y = (t_2, 0, f_2)$:

$$\begin{aligned}(x \wedge \neg y) \vee (y \wedge \neg x) &= ((t_1, 0, f_1) \wedge (f_2, 0, t_2)) \vee ((t_2, 0, f_2), \wedge (f_1, 0, t_1)) = \\ &= (\min\{t_1, f_2\}, 0, \max\{f_1, t_2\} \vee (\min\{t_2, f_1\}, 0, \max\{f_2, t_1\}) \\ &= (\max\{\min\{t_1, f_2\}, \min\{t_2, f_1\}\}, 0, \min\{\max\{f_1, t_2\}, \max\{f_2, t_1\}\})\end{aligned}$$

Now, examining the truth table, one can be sure that $x \oplus y$ is equal to $\neg(x \leftrightarrow y)$.

Table 13: Table showing that $x \oplus y$ has the same value as $\neg(x \leftrightarrow y)$.

x	$\neg y$	y	$\neg x$	$x \wedge \neg y$	$y \wedge \neg x$	$(x \wedge \neg y) \vee (y \wedge \neg x)$
(1, 0, 0)	(0, 0, 1)	(1, 0, 0)	(0, 0, 1)	(0, 0, 1)	(0, 0, 1)	(0, 0, 1)
(1, 0, 0)	(1, 0, 0)	(0, 0, 1)	(0, 0, 1)	(1, 0, 0)	(0, 0, 1)	(1, 0, 0)
(0, 0, 1)	(0, 0, 1)	(1, 0, 0)	(1, 0, 0)	(0, 0, 1)	(1, 0, 0)	(1, 0, 0)
(0, 0, 1)	(1, 0, 0)	(0, 0, 1)	(1, 0, 0)	(0, 0, 1)	(0, 0, 1)	(0, 0, 1)

Definition 2.18 (Operators NOR and NAND)**.** The operators $\downarrow$ (NOR) and $\uparrow$ (NAND) has behavior given by the following truth table:

Table 14: Table for the operators $\downarrow$ (NOR) and $\uparrow$ (NAND).

x	y	$\downarrow$	$\uparrow$
1 0 0	1 0 0	0 0 1	0 0 1
1 0 0	0 0 1	0 0 1	1 0 0
0 0 1	1 0 0	0 0 1	1 0 0
0 0 1	0 0 1	1 0 0	1 0 0

As the acronym suggests, NOR and NAND are the negations of OR and AND, that is, $\downarrow = \neg\vee$ and $\uparrow = \neg\wedge$, being enough to look at the truth table of OR and AND and negate the entries of the respective columns.

So far was showed that the main connectives and properties of the classical logic remains true when translated to neutrosophic form. However, for any logic be useful, it must supply rules of inference, and this is the scope of the following lines.

Theorem 2.19 (Modus Ponens)**.** *Let $x = (t_1, 0, f_1)$ and $y = (t_2, 0, f_2)$ be neutrosophic logic variables representing classical logic variables. If $x \to y$ is true and x is true, then could be inferred that y is true. This is represented by*

$$x \to y, x \vdash y$$

This rule is named as modus ponens *(mode that assert).*

Proof. Its enough to see the truth table of the conditional operator (Table 8). In the first line of the truth table, $x \to y$ is true iff x and y are true, so, given that $x \to y$ and x are true, one must infer that y is also true. This is the only possibility. □

Theorem 2.20 (Modus Tollens)**.** *Let $x = (t_1, 0, f_1)$ and $y = (t_2, 0, f_2)$ be neutrosophic logic variables representing classical logic variables. If $x \to y$ is true and $\neg y$ is true, then could be inferred that $\neg x$ is true. This is represented by*

$$x \to y, \neg y \vdash \neg x$$

This rule is named as modus tollens *(mode that denies).*

Proof. It is know that $x \to y = \neg x \vee y$ (Theorem 2.15), so:

$$\begin{aligned}\neg x \vee y &= \neg x \vee (\neg\neg y) \text{ (double negation of y - Theorem 2.1)}\\ &= \neg(\neg y) \vee \neg x \text{ (commutativity of the operator } \vee \text{ - Theorem 2.2)}\\ &= \neg y \to \neg x \text{ (backing to conditional form)}\end{aligned}$$

Another way to discern this is looking at the truth table of the conditional operator (Table 8). In the fourth line of the truth table, $x \to y$ is true iff x and y are false, so, given that $x \to y$ is true and $\neg x$ is true, can be inferred that $\neg y$ is also true. This is the only possibility. □

Theorem 2.21 (And Elimination). *Let $x = (t_1, 0, f_1)$ and $y = (t_2, 0, f_2)$ be neutrosophic variables representing classical logic variables. If $x \wedge y$ is true, then can be inferred that x (or y) is also true. This can be represented by:*

$$x \wedge y \vdash x \qquad \textit{or} \qquad x \wedge y \vdash y$$

Proof. It is enough to recall that the result given by the conjunction operator is true iff both of its operands are true (first line/last column of the Table 1). □

Theorem 2.22 (And Introduction). *Let $x = (t_1, 0, f_1)$ and $y = (t_2, 0, f_2)$ be neutrosophic variables representing classical logic variables. If x and y are both true, then can be inferred that $x \wedge y$ is also true. This can be represented by:*

$$x, y \vdash x \wedge y$$

Proof. Its enough to recall that the result given by the conjunction operator is true iff both of its operands are true (first line/last column of the Table 1). □

Theorem 2.23 (Or Introduction). *Let $x = (t_1, 0, f_1)$ and $y = (t_2, 0, f_2)$ be neutrosophic variables representing classical logic variables. If x is true, then $x \vee y$ is true, independent of the logical value of y. This can be represented by:*

$$x \vdash x \vee y$$

Proof. Its enough to recall that the result given by the disjunction operator is false iff both of its operands are false. However, as the first component is true, the result of the operation is true (first line/third column of the Table 1). Note that the theorem remains true if the roles of x and y are swapped, that is: $y \vdash x \vee y$. □

Theorem 2.24 (Unit Resolution). *Let $x = (t_1, 0, f_1)$ and $y = (t_2, 0, f_2)$ be neutrosophic variables representing classical logic variables. If $x \vee y$ is true and one of their elements is false, then the other element must be true. This can be represented by:*

$$x \vee y, \neg y \vdash x \qquad \textit{or} \qquad x \vee y, \neg x \vdash y$$

Proof. Its enough to recall the result given by the disjunction operator in the second and third line/third column of the Table 1. □

Theorem 2.25 (Resolution). *Let $x = (t_1, 0, f_1)$ and $y = (t_2, 0, f_2)$ and $z = (t_3, 0, f_3)$ be neutrosophic logic variables representing classical logic variables. The conditional operator is transitive, that is*

$$x \to y, y \to z \vdash x \to z$$

Proof. Firstly, take as hypothesis that $x \to y$ is true and x is true, so under the rule of *modus ponens* (Theorem 2.19), it is possible to infer that y is true, that is

$$x \to y, x \vdash y \tag{1}$$

Now, take as hypothesis that $y \to z$ is true. Noting that y is true by (1), it is possible to conclude that z is true, that is

$$y \to z, y \vdash z \tag{2}$$

Finally, given that x (by assumption) is true and z is true by (2), can be concluded that $x \to z$ is true (recall the first line from the Table 8). So,

$$x \to y, y \to z \vdash x \to z \tag{3}$$

□

The rules of inference given are not the only ones, there are several other. However, it must be sufficient to demonstrate that the main characters of the classical logical remain true when translated to neutrosophic form.

Is something missing? Yes. For example, the view of the classical logic as a formal system (*propositional calculi*), not to mention first, second and higher-order logics, have not been addressed.

3 Conclusion

This short paper has shown that neutrosophic logic, a recent type of logic, encompass the classical logic, at least at the propositional level. Then, it could be said that the classical logic is a type of neutrosophic logic, or that CL is a subset of NL.

Given the limited scope, it is not reasonable to show all aspects of the classical logic translated in neutrosophic form. However, this could be done in another time if the necessity arises.

Now, which future directions of research can be pursued? Given that the neutrosophic branch is in a wide scope, there are a bunch of paths, but of possible interest is to compare neutrosophic logic with other logics, mainly the class of the paraconsistent annotated Logics.

The paraconsistent annotated logics are a family of non-classical logics initially employed in logical programming by Subrahmanian.[2] Due to the obtained applications, a study of the foundations of the underlying logic of the investigated programming languages became convenient. It was verified that it was a paraconsistent logic, in some cases, also contained characteristics of paracomplete and non-alethic logic. To be concise and short, the class of paraconsistent annotated logics has been applied to several fields, from digital circuits[3],[4] decision making,[5] neural networks,[6] just to name a few.

Finally, it is worth follow this way? The answer to this question cannot be given here, but consider that comparing the strengths of neutrosophic logic and other logics, one or another can be enhanced, and as lateral effect, the general knowledgment grows with this.

Acknowledgments: The first author is in profound debit with Wandercy Savedra Gomes, a very understanding old friend, that kindly made the revision of the english language for this work.

References

1. Florentin Smarandache. *A Unifying Field in Logics, Neutrosophic Logics: Neutrosophic Logic. Neutrosophy, Neutrosophic Set, Neutrosophic Probability and Statistics*. InfoLearnQuest, 2007. Disponible at: http://fs.unm.edu/eBook-Neutrosophics6.pdf.

2. N. C. A da Costa, J. M. Abe, and V. S. Subrahmanian. Remarks on annotated logic. *Zeitschrift fur Mathematische Logik und Grundlagen der Mathematik*, (37):561–570, 1991.

3. Joao Inácio da Silva Filho. Implementação de circuitos lógicos fundamentados em uma classe de lógicas paraconsistentes anotadas. Master's thesis, Universidade de São Paulo, 1997.

4. Joao Inácio da Silva Filho. *Métodos de aplicações da lógica paraconsistente anotada de anotação com dois valores-LPA2v com construção de algoritmo e implementação de circuitos eletrônicos*. PhD thesis, Universidade de São Paulo, 1999.

5. Fábio Romeu de Carvalho and Jair Minoro Abe. *A Paraconsistent Decision-Making Method*, volume 87 of *Smart Innovation, Systems and Technologies*. Springer, 2018.

6. Sheila Souza and Jair Minoro Abe. *Paraconsistent Artificial Neural Networks and Aspects of Pattern Recognition*, volume 94 of *Intelligent Systems Reference Library*, chapter 9, pages 207–231. Springer, 2015.