Łukasiewicz' 3-valued logic can not be expressed in terms of SP3A logic

Jaime Díaz[1], José Luis Carballido[1], Mauricio Osorio[2], and Claudia Zepeda[1]

[1] Benemérita Universidad Autónoma de Puebla
jdiazalt27@gmail.com
[2] Universidad de las Américas Puebla,
osoriomauri@gmail.com

Abstract In this work we study some properties and characteristics of Béziau's SP3A logic; by this analysis, we determine that it is possible to express any well formed formula in SP3A in a Disjunctive Normal Form. We determine too that it is not always possible to transform any well formed formula into its Conjunctive Normal Form. Then, we demonstrate that there are only 10 different functions that can be obtained in SP3A from expressions of one variable. Finally, we prove that it is not possible to express the negation of Łukasiewicz' 3-valued logic in terms of SP3A, hence, this logic cannot be represented in SP3A.
Keywords: Paraconsistency, 3-Valued logic, Non-Classical Logic.

1 Introduction

Classical logic holds the Law of Non Contradiction as one of its main characteristics; Aristotle stated that it is so obvious that no contradiction can be proved true, that it is unacceptable to admit any contradiction as true [10]. However, presocratics thought in a different way. In 1910, Jan Łukasiewicz determined that it was not possible to formally prove this Law [9], and this fact started the resurgence of logics that do not hold the Law of Non Contradiction, in the western thought.

Łukasiewicz is considered one of the main forerunners of Paraconsistent Logics, understanding that Paraconsistent Logics are those logics in which there is a connective that does not obey the Law of Non Contradiction [2]; His work was formerly oriented to the justification of the Law of Non Contradiction, but found out that it was not possible to do it [9], and as a consequence, he developed a family of non-classical logics [5]; Łukasiewicz' $3-$valued logic ($Ł_3$) is part of this family. It is defined by 2 primitive connectives and its domain has 3 values: two of then behave just as classical logic values do, and the third is interpreted as "possible" [4]. Besides Béziau's SP3A logic, $Ł_3$ constitutes one of the objects of interest in this work.

SP3A and SP3B logics were proposed in 2016 by Jean-Yves Béziau [3]; both of them are 3-valued Paraconsistent logics. In this work we focus on SP3A logic, and study some of its characteristics, such as the behavior of the Law of Non Contradiction, Double Negation, de Morgan laws, among others. We determine that it is possible to transform any well formed formula in SP3A, into a Disjunctive Normal Formula, and that it is not always possible to perform an analogous process to transform any formula into a Conjunctive Normal Form. Knowing that it is possible to transform any formula into its Disjunctive Normal Form, we determine that there are only 10 different functions expressed in a single atom that can be obtained in SP3A. Finally, we demonstrate that it is not possible to express $Ł_3$ negation in terms of SP3A logic. Then, it follows that it is not possible to express $Ł_3$ in terms of SP3A.

2 Background

In this section we will define some preliminary concepts, such as functional equivalence, Well Formed Formula and negations from Łukasiewicz' 3-valued logic and from G_3 logic. These definitions will bee needed on further sections.

Definition 1. *Functional equivalence between two formulas occurs when both of them include exactly the same variables, and evaluate exactly to the same value for every interpretation of these variables. We will use the symbol $\equiv$ to represent functional equivalence between two formulas, so that given two well formed formulas A and B, $A \equiv B$ means that truth tables for A and B are identical to one another.*

Every logic is associated to a domain: a set of values that can be assumed by the variables; this domain constitutes the values to be considered when performing calculations; in classical logic, this set is $\{0, 1\}$, but in non-classical logics there can be domains with 3, 4 or more values. There are two kinds of values in every domain: designated and undesignated.

When evaluating a well formed formula in a logic, every variable of the formula can take any of the values of the domain, and a value comes out as a result. Each particular configuration of values of the variables is called an interpretation of the formula, and the result obtained from the expression is the evaluation.

Definition 2. *A well formed formula that delivers only undesignated values for every possible interpretation of its variables is called a contradiction; a well formed formula that evaluates only to designated values for every possible interpretation of its variables is called a tautology.*

Due to the action of inference rules over *wfs*, new *wfs* can be generated any time; if we can assure that a system will never generate two contradictory expressions x and $\neg x$, then we say that the system is consistent, and otherwise it is said to be Inconsistent (contradictory). Classical logic holds this property as one of its main characteristics. However, Presocratic philosophers thought it was legitimate to believe in contradictions since nature of many things is contradictory. Aristotle, on the other hand, thought that there was nothing more obvious than contradictions cannot be true. Since the middle ages Aristotle's stance about the Law of non-contradiction was taken for granted until 20th Century, with no further analysis [10].

Up to the current times, classical logic is the most well-known and the most used logic. Some of its main characteristics are listed below:

- Double negation: $\neg\neg x \equiv x$.
- Law of non-contradiction (LNC): $\neg(x \wedge \neg x)$.
- Explosive behavior: $x \wedge \neg x \vdash y$.

In order to be consistent, a logic must obey LNC. Not obeying it leads to an explosive behavior, where the acceptance of a contradiction trivializes the formal axiomatic system [6].

As a consequence of this orthodoxy, acceptance of Aristotle's authority makes consistency a mandatory matter when working with logic, and classical logic was the only one accepted in science until the beginning of 20th Century. But then, can a formal axiomatic system that accepts some contradictions be considered a logic? The question arose paralell to the revolution that quantum physics (non-classical physics) meant for that discipline. And just as quantum physics did with physics, paraconsistent logics revolutionized the world of logic.

Jan Łukasiewicz and Nikolai Vasili'év are considered the forerunners of paraconsistent ideas; in 1910 and 1911 respectively, and independently of each other, they re-marked the importance of the revision of some Aristotelian laws [5]; their work started a new era in the study of logics, and prepared the field for those logics that now we call paraconsistent. According to Béziau, a paraconsistent logic is a logic that has a connective which does not obey the LNC, and is usually called Paraconsistent Negation, and it represents a main problem to determine wether it is legitimate to call it *negation* [2].

Łukasiewicz developed a 3-valued logic, denoted as Ł$_3$. The values in its domain can be interpreted as *true*, *false* and a third value that in this case will be understood as *possible*[4]; we will consider Ł$_3$ domain to be $D = 0, 1, 2$, where 2 is the only designated value, and 1 is the value that represents *possible*. Primitive connectives in Ł$_3$ are $\perp$ (bottom) and $\rightarrow$ (implication), and they are defined as follows [4]:

$$\perp = 0$$

$$x \rightarrow y = min(2, (2 - x) + y)$$

It is in our interest to show explicitly Table 1 for Ł$_3$ negation. From now on, this negation will be denoted by $\neg_{\text{Ł}}$. G_3 logic, also called "Here and There logic" (HT) belongs to the Gödel's family of multivalued logics. It is a 3-valued, non-paraconsistent logic, but despite this fact, we will use G_3 negation as a connective that will be defined later in terms of SP3A logic in order to help in the development of this work. G_3 domain is $D = \{0, 1, 2\}$, where 2 is the only designated value [4]. We will denote G_3 negation as $\sim$, and it is defined by Table 1.

3 The paraconsistent logic SP3A

In 2016, Jean-Yves Béziau presents "Two Genuine 3-Valued Paraconsistent Logics". In this work, Béziau introduces SP3A and SP3B as two logics that hold

Table 1. Truth tables of connectives $\neg_{Ł}$ and $\sim$ in G_3.

x	$\neg_{Ł} x$
0	2
1	1
2	0

x	$\sim x$
0	2
1	0
2	0

characteristics of paraconsistency [3]. In this work we will focus on SP3A; our aim is to demonstrate that $Ł_3$ logic cannot be properly expressed by SP3A.

SP3A is a 3-valued logic, where each variable can take any value $x \in \{0, 1, 2\}$; in this logic, 1 and 2 are the designated values. Extreme values 0 and 2 behave just as *False* (0) and *True* (1) do in classical logic; intermediate value (1) must not be understood as the exact middle value between False and True; it must be understood to be some truth value placed somewhere between 0 and 2 [3]. There are only 3 primitive connectives defined in SP3A: negation, conjunction and disjunction. These connectives are defined by Table 2.

Table 2. Truth tables of connectives $\wedge$, $\vee$, and $\neg$ in $SP3A$.

$\wedge$	0	1	2
0	0	0	0
1	0	1	2
2	0	2	2

$\vee$	0	1	2
0	0	1	2
1	1	1	2
2	2	2	2

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

In SP3A only one of the distributive laws is satisfied; it is the distributivity of the conjunction over the disjunction, while distributive law of disjunction over conjunction is not satisfied functionally (Expresions 1, 2). The non-equivalence of $x \vee (y \wedge z)$ and $(x \vee y) \wedge (x \vee z)$ is only functional, and appears only when evaluating 2 specific interpretations (Table 3).

$$x \wedge (y \vee z) \equiv (x \wedge y) \vee (x \wedge y) \tag{1}$$

$$x \vee (y \wedge z) \not\equiv (x \vee y) \wedge (x \vee y) \tag{2}$$

Table 3. Interpretetions of variables that do not satisfy $x \vee (y \wedge z) \equiv (x \vee y) \wedge (x \vee y)$

x	y	z	$x \vee (y \wedge z)$	$(x \vee y) \wedge (x \vee y)$
1	0	2	1	2
1	2	0	1	2

When analyzing a paraconsistent logic, the properties that are related to negation use to present a weird behavior; Here, we present Table 4 that describes the behavior of some properties of SP3A; all of them include negation. One of the most well-known properties of classical logic is the double negation. This property stands for $\neg\neg x \equiv x$, so that when we get an even number of negations applied to any expression, they can be considered to cancel out pairwise and the expression will act as if it was free of these negations. This property does not hold in SP3A logic.

This behavior of negation is the main reason for this logic to be non-classical; this fact alone changes the behavior of the law of no contradiction ($\neg(x \wedge \neg x)$ is tautology). When evaluating the three possible interpretations of the law of no contradiction, we obtain that when $x = 1$, $\neg(x \wedge \neg x) = 0$, so the law of non-contradiction is not a tautology. This can be, of course interpreted as: "not every contradiction is false". There is so much to be said about this conclusion: in "Why it's irrational to believe in consistency", Graham Priest discusses the rationality of Aristotelian logic (classical logic), and in "Torn by Reason: Łukasiewicz on the Principle of Contradiction" Priest remarks that Łukasiewicz found it impossible to prove Aristotle statement about the unacceptability of contradictions. According to Łukasiewicz, Aristotle seems to limit the significance of this law to substantial beings and concludes that "it is not the case that all contradictions are true" [9,10].

Table 4. Some negation-related properties of SP3A

	Double negation	Excluded middle	Non Contradiction Law
x	$\neg\neg x$	$x \vee \neg x$	$\neg(x \wedge \neg x)$
0	0	2	2
1	0	2	0
2	2	2	2

In some of the sections to come, we will need G_3 negation as an auxiliary connective. For that reason, in the present section we will define this negation in terms of SP3A logic. Native negation in G_3 behaves as Table 1 defines and can be modeled in SP3A using Expression 3 Where $\vee$, $\wedge$ and $\neg$ are primitive connectives of SP3A. This equivalence can be easily verified.

$$\sim x := \neg(x \vee (x \wedge \neg x)) \tag{3}$$

Paraconsistent behavior of SP3A logic makes it necessary to verify if de Morgan laws hold; in this section we will explore de Morgan laws from the point of view of two negations: the native SP3A negation and G_3 negation, due to the need of an auxiliary negation for normalization of formulas. As Béziau states, SP3A logic complies with only one of the two de Morgan Laws: Equations 4,5 [3].

$$\neg(x \vee y) \equiv \neg x \wedge \neg y \tag{4}$$

$$\neg(x \wedge y) \not\equiv \neg x \vee \neg y \tag{5}$$

As we need an expression that helps us to change the negation of a disjunction into an expression where negation is acting directly over atomic formulas, we propose Expression 6, which can be easily verified. Note that this expression uses as a connective the G_3 negation.

$$\neg(x \wedge y) \equiv (\sim x \vee \sim y) \vee ((x \wedge \neg x) \wedge (y \wedge \neg y)) \tag{6}$$

Both of de Morgan laws hold for G_3 negation (Expressions 7 and 8).

$$\sim (x \vee y) \equiv \sim x \wedge \sim y \tag{7}$$

$$\sim (x \wedge y) \equiv \sim x \vee \sim y \tag{8}$$

Now we are ready to analyze disjunctive normal forms within SP3A logic.

4 Disjunctive Normal Forms in SP3A Logic

As a consequence of Expression 2, we can say that there are cases of wfs that can not be transformed into a Conjunctive Normal Formula. In this section we prove that any wfs can be transformed into a Disjunctive Normal Formula if we allow the use of the G_3 negation connective. As an initial step, we evaluate seven cases that correspond to the formulas α , $c_1\alpha$, $c_1c_2\alpha$, where $c_1, c_2 \in \{\neg, \sim\}$ and α is an atomic formula. From this we obtain the 7 cases shown in Table 5. Note that $\sim \neg\alpha \equiv \neg\neg\alpha$ and $\neg \sim \alpha \equiv \sim\sim \alpha$, and this allows us to reduce the 7 original results to only 5

Table 5. Functions obtained by evaluating α, $C_1\alpha$ and $C_1C_2\alpha$, where $C_1, C_2 \in \{\neg, \sim\}$.

α	$\neg\alpha$	$\sim\alpha$	$\neg\neg\alpha$	$\sim\neg\alpha$	$\sim\sim\alpha$	$\neg\sim\alpha$
0	2	2	0	0	0	0
1	2	0	0	0	2	2
2	0	0	2	2	2	2

Definition 3. *A na-formula is any expression of the form $c_1c_2c_3...c_n\alpha$ where $c_i \in \{\neg, \sim\}$ and α is an atomic formula; An n-formula is a formula that holds any of the following structures: $a_1 := \alpha$, $a_2 := \neg\alpha$, $a_3 := \sim \alpha$, $a_4 := \neg\neg\alpha$, $a_5 := \sim\sim \alpha$, where α is an atomic formula.*

Definition 4. *Let be $A = \{a_1, a_2, a_3, a_4, a_5\}$. Then we say that A is the set of n-formulas*

Lemma 1. *Every na-formula is functionally equivalent to a n-formula.*
Proof: Let be $c \in \{\neg, \sim\}$. Then: $ca_1 \in \{a_2, a_3\}$, $ca_2 \equiv a_4$, $ca_3 \equiv a_5$, $ca_4 \equiv a_2$ and $ca_5 \equiv a_3$; It is not hard to verify that for any permutation of any number of negation connectives arranged before an atomic formula, an n-formula is obtained by equivalence.

Definition 5. *A cd-formula is a formula that uses only* $\wedge$ *and* $\vee$ *as connectives that operate over atomic formulas. An cdn-formula is a formula obtained from a cd-formula by replacing its atomic formulas by n-formulas.*

Lemma 2. *Every formula in SP3A is functionally equivalent to a cdn-formula. Proof: The only case to be treated is when there are negation connectives acting over a conjunction or a disjunction, counting on the fact that conjunction and disjunction are binary operators. In this case we apply the following equivalences:*

- $\neg(x \vee y) \equiv \neg x \wedge \neg y$
- $\neg(x \wedge y) \equiv (\sim x \vee \sim y) \vee ((x \wedge \neg x) \wedge (y \wedge \neg y))$
- $\sim (x \vee y) \equiv \sim x \wedge \sim y$
- $\sim (x \wedge y) \equiv \sim x \vee \sim y$

Definition 6. *A sdn-formula is a formula that only contains disjunctions of conjunctions, where the conjunctions operate exclusively over n-formulas. A sdn-formula will also be called a* ***Disjunctive Normal Formula****. The following two expressions are sdn-formulas:*

- $((a_2 \wedge a_4) \wedge a_5) \vee (a_1 \wedge a_3) \vee (a_3 \wedge a_5)$
- $a_3 \vee (a_2 \wedge a_3) \vee (a_1 \wedge a_5) \vee (a_1 \wedge a_2 \wedge a_5) \vee (a_1 \wedge a_3) \vee a_5$

The following two expressions are not sdn-formulas:

- $((a_3 \vee a_4) \wedge a_5) \vee (a_4 \wedge a_5)$
- $(a_1 \wedge a_2) \vee \sim (a_2 \wedge a_4)$

Theorem 1. *Every formula in SP3A is equivalent to a sdn-formula.*

Proof: According to Lemma 2 every formula in SP3A can be transformed into a cdn-formula. Applying commutativity, associativity, distributivity of conjunction over disjunction, every cdn-formula can be transformed into a sdn-formula

Example: Let us have expression 9 in SP3A. It can be transformed into a sdn-formula as shown below:

$$\sim (x \wedge \neg y) \wedge \neg x \wedge (\sim x \vee \sim y) \tag{9}$$

- $(\sim x \vee \sim \neg y) \wedge (\neg x \wedge (\sim x \vee \sim y))$
- $(\sim x \vee \neg\neg y) \wedge (\neg x \wedge (\sim x \vee \sim y))$
- $(\sim x \vee \neg\neg y) \wedge ((\neg x \wedge \sim x) \vee (\neg x \wedge \sim y))$
- $((\sim x \wedge \neg\neg y) \wedge (\neg x \wedge \sim x)) \vee ((\sim x \vee \neg\neg y) \wedge (\neg x \wedge \sim y))$
- $(\neg x \wedge \sim x \wedge \sim x) \vee (\neg x \wedge \sim x \wedge \neg\neg y) \vee (\neg x \wedge \sim y \wedge \sim x) \vee (\neg x \wedge \sim y \wedge \neg\neg y)$

4.1 Functions of one variable that can be obtained from Disjunctive Normal Forms in SP3A

Up to this point, we have demonstrated that it is possible to write any formula in SP3A as a disjunctive normal formula. Now we will analyze these normal formulas in order to characterize their behavior. Łukasiewicz' negation is a function that requires only one variable, and if there was a SP3A formula able to behave just as this negation does, it would be an expression with only one variable. Our final goal is to demonstrate that it is impossible to express this negation in SP3A, so from now on all the analysis will be about expressions with only one variable.

Considering Lemma 1 and Definition 4, we can say that the structure of the disjunctive normal forms is that shown in Expression 10 where $a_x \in A$.

$$(a_f \wedge a_g \wedge ... \wedge a_h) \vee (a_i \wedge a_j \wedge ... \wedge a_k) \vee ... \vee (a_l \wedge a_m \wedge ... \wedge a_n) \tag{10}$$

When we have conjunction of n-formulas $(a_m \wedge a_n)$, there is a finite number of possible results; it is a characteristic behavior of SP3A that $x \wedge x \equiv x$, so that $a_n \wedge a_n \equiv a_n$, leaving only 10 conjunctions to be revised. Table 6 is obtained from the evaluation of conjunctions of n-formulas, where $b_1, b_2 \notin A$ are new results; these new results are operated under conjunctions too. Finally, the table specifies the truth tables of b_1 and b_2. Going further, we cannot obtain new results via conjunctions. For convenience, we define the set $\Gamma = A \bigcup \{b_1, b_2\}$.

Table 6. Conjunctions of n-formulas; conjunctions where b_1 and b_2 appear; specification of b_1 and b_2 corresponding to the 3 values 0, 1, 2 respectively

$\wedge$	a_1	a_2	a_3	a_4	a_5
a_1	a_1	b_1	b_2	a_4	a_5
a_2	b_1	a_2	a_3	b_2	b_1
a_3	b_2	a_3	a_3	b_2	b_2
a_4	a_4	b_2	b_2	a_4	a_4
a_5	a_5	b_1	b_2	a_4	a_5

$\wedge$	a_1	a_2	a_3	a_4	a_5	b_1	b_2
b_1	b_1	b_1	b_2	b_2	b_1	b_1	b_2
b_2	b_2	b_2	b_2	b_2	b_2	b_2	b_2

b_1	b_2
0	0
2	0
0	0

Definition 7. *A formula of the type $\gamma_1 \wedge \gamma_2 \wedge \gamma_3 \wedge \ldots \wedge \gamma_n$, with $\gamma_i \in \Gamma$ will be called an and-n-formula.*

Lemma 3. *Any and-n-formula will evaluate to γ_i,$\gamma_i \in \Gamma$.*
Proof: As shown in Table 6.

Note that Any formula consisting in disjunctions of and-n-formulas is what we call a sdn-formula (Definition 6).

When applying disjunctions over and-n-formulas we get the results shown in Table 11, where $c_1, c_2, c_3 \notin \Gamma$ are new results; if we evaluate disjunctions that include c_1, c_2 and c_3, there are no new results.We will define the set $\Delta = \Gamma \bigcup \{c_1, c_2, c_3\}$; elements of Δ are the 10 only possible results to be obtained from a Disjunctive Normal Formula defined in terms of one single variable α.

As a conclusion for this section, we have determined that any sdn-formula with one variable in SP3A will evaluate to $\delta \in \Delta$; therefore, there are only 10 functions that can be obtained from such a sdn-formula. These functions are shown in Table 8

Table 7. Disjunctions of and-n-formulas; conjunctions where c_1, c_2 and c_3 appear; specification of c_1, c_2 and c_3 corresponding to the 3 values 0, 1, 2 respectively

$\vee$	a_1	a_2	a_3	a_4	a_5	b_1	b_2
a_1	a_1	c_1	c_2	a_1	a_5	a_5	a_1
a_2	c_1	a_2	a_2	c_1	c_1	a_2	a_2
a_3	c_2	a_2	a_3	c_3	c_1	a_2	a_3
a_4	a_1	c_1	c_3	a_4	a_5	a_5	a_4
a_5	a_5	c_1	c_1	a_5	a_5	a_5	a_5
b_1	a_5	a_2	a_2	a_5	a_5	b_1	b_1
b_2	a_1	a_2	a_3	a_4	a_5	b_1	b_2

$\vee$	a_1	a_2	a_3	a_4	a_5	b_1	b_2	c_1	c_2	c_3
c_1	c_1	c_1	c_1	c_1	c_1	c_1	c_1	c_1	c_1	c_1
c_2	c_2	c_1	c_2	c_2	c_1	c_1	c_2	c_1	c_2	c_2
c_3	c_2	a_2	a_3	c_3	c_1	c_1	c_3	c_1	c_2	c_3

c_1	c_2	c_3
2	2	2
2	1	0
2	2	2

Table 8. Functions that can be obtained from a sdn-formula with one variable.

a_1	a_2	a_3	a_4	a_5	b_1	b_2	c_1	c_2	c_3
0	2	2	0	0	0	0	2	2	2
1	2	0	0	2	2	0	2	1	0
2	0	0	2	2	0	0	2	2	2

5 Łukasiewicz' logic cannot be expressed in terms of SP3A logic

Łukasiewicz' negation is defined by Table 1. Reviewing the analysis performed, up to this point, we know that:

- Every formula in SP3A can be expressed as a disjunctive normal form
- Any normal form evaluates to a $\delta \in \Delta$
- There is no $\delta \in \Delta$ that satisfies $\neg_{\text{Ł}} x$ definition

The final conclusion is stated in Theorems 2 and 3.

Theorem 2. *Łukasiewicz' 3-valued logic negation cannot be expressed in terms of SP3A logic.*

Proof: Development in section 4.

Theorem 3 follows from Theorem 2:

Theorem 3. *Łukasiewicz' 3-valued logic cannot be expressed in terms of SP3A logic.*

Proof: As Theorem 2 states, it is impossible to express $\neg_{\L}$ in terms of SP3A logic. Therefore, it follows that it is not possible to express Ł$_3$ in terms of SP3A logic.

Theorems 1, 2 and 3 constitute the main contribution of this work.

6 Conclusions

We have performed an analysis of the structure of formulas of SP3A logic; we determined that any well formed formula in SP3A can be transformed to a Disjunctive Normal Form. If the formula is written in terms of only one atom, there are only 10 different functions that can be achieved. From here we could prove that Ł$_3$ negation cannot be expressed in terms of SP3A connectives. Hence Ł$_3$ logic cannot be expressed in terms of SP3A either.

In propositional classical logic, every formula is equivalent to a Conjunctive and to a Disjunctive normal forms [8]; normal forms are useful in the solution of satisfiability problems (SAT) which consists in determining wether there exits an interpretation for the variables of a logic formula that cause this formula to be evaluated as true [1]. SAT for Conjunctive Normal Forms is the first known NP-Complete problem, is usually difficult to solve; on the other hand, SAT for Disjunctive Normal Forms is not hard to solve. However, transforming a Conjunctive Normal Form into a Disjunctive Normal Form requires an exponential time [1]. As shown previously, every wfs in SP3A can be written as a formula in a Disjunctive Normal Form, which we think that can be useful in the satisfiability problem for SP3A logic. A drawback that follows from the fact that not every wfs in SP3A can be expressed in a Conjunctive Normal Form, is that we cannot use the Horn algorithm of satisfiability. This algorithm constitutes a "quick" way to prove satisfiability for Conjunctive Normal Formulas written as a Horn formula; Horn formulas are Conjunctive Normal Formulas where every disjunction contains at most one positive literal [1].

As for the fact that it is impossible to express Ł$_3$ logic in terms of SP3A, we must be concerned with the possible interpretations of the third truth-value of SP3A: in 3-valued logics, extreme values are interpreted classically (as true and false), but the interpretation of the third value depends on the behavior of the connectives when operating over it. In [7], the author reviews some of the main interpretations for this third value; Dubois identifies two main groups of interpretations: ontological (intrinsic to the definition of the propositions) and epistemic (Their state of truth or falsity has not been yet established but eventually can be so in a future moment) [7]. Some ontological interpretations could be: undefined, half-true, irrelevant, inconsistent. Possible and unknown are epistemic interpretations [7]. As Łukasiewicz 3-valued logic cannot be expressed in SP3A logic, we must understand that the interpretation of the third truth value proposed by Łukasiewicz (possible) cannot be adopted by SP3A. According to our research, Béziau has not proposed any interpretation for the third value of SP3A and therefore we still have to deal with the problem of giving an interpretation for it.

References

1. L. G. Amaru. Springer, 2017.
2. J.-Y. Béziau. What is paraconsistent logic? *Frontiers of paraconsistent logic*, pages 95–111, 2000.
3. J.-Y. Béziau. *Towards Paraconsistent Engineering. Intelligent Systems Reference Library, vol. 110*, chapter Two Genuine 3-Valued Paraconsistent Logics. Springer, Cham, 2016.
4. J. Carballido. *Fundamentos matemáticos de la semántica p-estable en programación lógica*. PhD thesis, Benemérita Universidad Autónoma de Puebla, 2009.
5. N. A. D. Costa, J.-Y. Béziau, and O. Bueno. Paraconsistent logic in a historical perspective. *Logique and analyse*, 38:111–25, 1995.
6. N. da Costa, J.-Y. Béziau, and O. A. S. Bueno. Aspects of paraconsistent logic. *Bulletin of the IGPL*, 3(4):597–614, 1995.
7. D.Ciucci and D. Dubois. A map of dependencies among three-valued logics. *Information Sciences*, (250):162–77, 2013.
8. S. Hedman. Oxford University Press, 2004.
9. G. P. Torn by reason: Łukasiewicz on the principle of contradiction. In *Early Analytic Philosophy-New Perspectives on the Tradition*, pages 429–44. Springer, 2016.
10. G. Priest. Why it's irrational to believe in consistency. In *Proceedings of the 23rd International Wittgenstein Symposium.*, pages 284–93, 2001.