The Complexity of 3-Valued Łukasiewicz Rules ⋆

Miquel Bofill[1], Felip Manyà[2], Amanda Vidal[2], and Mateu Villaret[1]

[1] Universitat de Girona, Girona, Spain
[2] Artificial Intelligence Research Institute (IIIA, CSIC), Bellaterra, Spain

Abstract. It is know that determining the satisfiability of n-valued Łukasiewicz rules is NP-complete for $n \geq 4$, as well as that it can be solved in time linear in the length of the formula in the Boolean case (when $n = 2$). However, the complexity for $n = 3$ is an open problem. In this paper we formally prove that the satisfiability problem for 3-valued Łukasiewicz rules is NP-complete. Moreover, we also prove that when the consequent of the rule has at most one element, the problem is polynomially solvable.

1 Introduction

The proof theory of many-valued logics has been deeply studied for a wide variety of logics [1, 8, 9]. Nevertheless, the development of satisfiability solvers has received less attention despite of the fact that, without competitive solvers, it is extremely difficult to apply many-valued logics to solve real-world problems.

Given the recent development of Satisfiability Modulo Theory-based (SMT-based) solvers for many-valued logics [2, 4, 3, 10, 11], there is the need to empirically evaluate and compare them with other existing approaches. Because of that we are interested in developing instance generators that produce instances of varying difficulty, as well as in analyzing the complexity of relevant fragments of many-valued logics. It is extremely difficult to advance in the development of fast satisfiability solvers without the availability of challenging benchmarks.

Before describing the related work and contributions of the present paper, let us recall that developing satisfiability solvers for Łukasiewicz logics is particularly interesting because $SAT^{Ł} \subsetneq SAT^{Bool}$, where $SAT^{Ł}$ is the set of formulas in Łukasiewicz logic that evaluate to 1 for some interpretation, and SAT^{Bool} is the set of satisfiable Boolean formulas [8]. This implies that some propositional formulas are satisfiable in Łukasiewicz logic whereas they are unsatisfiable in Boolean logic. However, in other relevant many-valued logics such as Gödel (G) and Product (Π) we have that $SAT^{G} = SAT^{\Pi} = SAT^{Bool}$ and, therefore, satisfiability testing in these logics can be proved directly with a Boolean SAT solver.

We have recently investigated, in [5], how the Conjunctive Normal Forms (CNFs) used by Boolean SAT solvers can be extended to Łukasiewicz logics. In a first attempt,

⋆ Research partially supported by the Generalitat de Catalunya grant AGAUR 2014-SGR-118, and the Ministerio de Economía y Competividad projects AT CONSOLIDER CSD2007-0022, INGENIO 2010, CO-PRIVACY TIN2011-27076-C03-03, EDETRI TIN2012-39348-C02-01 and HeLo TIN2012-33042. The second author was supported by Mobility Grant PRX14/00195 of the Ministerio de Educación, Cultura y Deporte.

we replaced the classical disjunction in Boolean CNFs with Łukasiewicz strong disjunction, and interpreted negation using Łukasiewicz negation. Interestingly, we proved that the satisfiability problem of these clausal forms has linear-time complexity,[1] regardless of the size of the clauses and the cardinality of the truth value set (assuming it is greater than two). This result is surprising because deciding the satisfiability of Boolean CNFs is NP-complete when there are clauses with at least three literals [7]. So, we identified a problem that is NP-complete in the Boolean case but has linear-time complexity in Łukasiewicz logic.

With the aim of producing computationally difficult instances, we defined a new class of clausal forms, called Łukasiewicz (Ł-)clausal forms, that are CNFs in which, besides replacing classical disjunction with Łukasiewicz strong disjunction, we allow negations above the literal level; i.e., clauses are strong disjunctions of terms, and terms are either literals or negated strong disjunctions of literals. We proved that, in this case, 3-SAT is NP-complete whereas 2-SAT has linear-time complexity.[2] Hence, we defined problems in Łukasiewicz logic that have the same complexity as their Boolean counterparts.

Independently of our work, Borgwardt et al. [6] investigated the complexity of finitely-valued Łukasiewicz rules (c.f. Section 2), and proved that the problem of deciding the satisfiability of such rules is NP-complete when the cardinality of the truth value set is at least four, but they left as an open problem the complexity of 3-valued Łukasiewicz rules. Analyzing the complexity of Łukasiewicz rules is appealing because this problem has linear-time complexity in the Boolean case whereas it is NP-complete for n-valued Łukasiewicz logics in which $n \geq 4$.

In this paper we prove that the satisfiability problem for 3-valued Łukasiewicz rules is NP-complete, solving this way an open problem. Moreover, we also prove that if the consequent of the rule has at most one element, the problem is polynomially solvable.

The paper is structured as follows. Section 2 defines basic concepts in Łukasiewicz logics, and n-valued Łukasiewicz rules. Section 3 shows that the satisfiability problem for 3-valued Łukasiewicz rules is NP-complete, but it is polynomially solvable in any finitely-valued Łukasiewicz logic if the consequent of the rule has at most one element. Section 4 concludes and points out future research directions.

2 Preliminaries

This section formally defines the finitely-valued and infinitely-valued logics of Łukasiewicz, as well as the language of Łukasiewicz rules.

Definition 1. *A **propositional language** is a pair $\mathbb{L} = \langle \Theta, \alpha \rangle$, where Θ is a set of logical connectives and $\alpha : \Theta \to \mathbb{N}$ defines the arity of each connective. Connectives with arity 0 are called constants. A language $\langle \Theta, \alpha \rangle$ with a finite set of connectives $\Theta = \{\theta_1, \ldots, \theta_r\}$ is denoted by $\langle \theta_1/\alpha(\theta_1), \ldots, \theta_r/\alpha(\theta_r) \rangle$.*

[1] In the following, when we say linear-time complexity we mean that the complexity is linear in the size of the formula.

[2] When the number of literals per clause is fixed to k, the corresponding SAT problem is called k-SAT.

Given a set of propositional variables $\mathcal{V}$, the set $L_{\mathcal{V}}$ of $\mathbb{L}$-formulas over $\mathcal{V}$ is inductively defined as the smallest set with the following properties: (i) $\mathcal{V} \subseteq L_{\mathcal{V}}$; (ii) if $\theta \in \Theta$ and $\alpha(\theta) = 0$, then $\theta \in L_{\mathcal{V}}$; and (iii) if $\phi_1, \ldots, \phi_m \in L_{\mathcal{V}}$, $\theta \in \Theta$ and $\alpha(\theta) = m$, then $\theta(\phi_1, \ldots, \phi_m) \in L_{\mathcal{V}}$.

Definition 2. *A **many-valued logic** $\mathcal{L}$ is a triplet $\langle \mathbb{L}, N, A \rangle$ where $\mathbb{L} =< \Theta, \alpha >$ is a propositional language, N is a truth value set, and A is an interpretation of the operation symbols that assigns to each $\theta \in \Theta$ a function $A_\theta : N^{\alpha(\theta)} \to N$.*

Many-valued logics are equipped with a non-empty subset D of N called the designated truth values, which are the truth values that are considered to affirm satisfiability.

Definition 3. *Let $\mathcal{L}$ be a many-valued logic. An **interpretation on** $\mathcal{L}$ is a function $I : \mathcal{V} \to N$. I is extended to arbitrary formulas ϕ in the usual way:*

1. *If ϕ is a logical constant, then $I(\phi) = A_\phi$.*
2. *If $\phi = \theta(\phi_1, \ldots, \phi_r)$, then $I(\theta(\phi_1, \ldots, \phi_r)) = A_\theta(I(\phi_1), \ldots, I(\phi_r))$.*

*A formula ϕ is **satisfiable** iff there is an interpretation such that $I(\phi) \in D$.*

Through this work, we focus on a particular family of many-valued logics: the Łukasiewicz logics. These were born from the generalization of a three valued logic proposed by J. Łukasiewicz in the early 20th century, and have been deeply studied both from theoretical and practical points of view. For a deeper study on these matters, see for instance [8].

The language of Łukasiewicz logic is given by

$$\mathbb{L}_{\text{Łuk}} = \langle \bot/0, \top/0, \neg/1, \to /2, \wedge/2, \vee/2, \odot/2, \oplus/2 \rangle.$$

We refer to $\bot$ as bottom, to $\top$ as top, to $\neg$ as negation, to $\to$ as implication, to $\wedge$ as weak conjunction, to $\vee$ as weak disjunction, to $\odot$ as (strong) conjunction, and to $\oplus$ as (strong) disjunction.

Definition 4. *The **infinitely-valued Łukasiewicz logic**, denoted by $[\mathbf{0}, \mathbf{1}]_{Ł}$, is the many-valued logic $\langle \mathbb{L}_{Łuk}, N, A \rangle$ equipped with the set of designated values $D = \{1\}$, where N is the real unit interval $[0, 1]$, and the interpretation of the operation symbols A is given by:*

$$\begin{aligned} A_\bot &= 0 & A_\wedge(x, y) &= \min\{x, y\} \\ A_\top &= 1 & A_\vee(x, y) &= \max\{x, y\} \\ A_\neg(x) &= 1 - x & A_\odot(x, y) &= \max\{0, x + y - 1\} \\ A_\to(x, y) &= \min\{1, 1 - x + y\} & A_\oplus(x, y) &= \min\{1, x + y\} \end{aligned}$$

The n**-valued Łukasiewicz logic**, denoted by $\mathbf{Ł_n}$, is the logic defined from the infinitely-valued Łukasiewicz logic by restricting the universe of evaluation to the set $N_n = \{0, \frac{1}{n-1}, ..., \frac{n-1}{n-1}\}$. That is to say, $\mathbf{Ł_n} = \langle \mathbb{L}_{\text{Łuk}}, N_n, A \rangle$ equipped with $D = \{1\}$. Note that the operations are well defined because N_n is a subalgebra of $[0, 1]$ with the interpretation of the operation symbols A (for any operation A_* and any value/pair of values of N_n, the result of A_* over this/these values also belongs to N_n).

The function interpreting negation is called Łukasiewicz negation, the function interpreting strong conjunction is called Łukasiewicz t-norm, the function interpreting implication is called its residuum, and the function interpreting strong disjunction is called Łukasiewicz t-conorm.

We say that a logic $\mathcal{L}$ is a Łukasiewicz logic if it is either $\mathbf{[0,1]_Ł}$ or $\mathbf{Ł_n}$ for some natural number n.

Given a Łukasiewicz logic $\mathcal{L}$, we denote by $SAT^{\mathcal{L}}$ the set of satisfiable formulas in $\mathcal{L}$; i.e.,

$$SAT^{\mathcal{L}} = \{\varphi : I(\varphi) = 1 \text{ for some interpretation } I \text{ on } \mathcal{L}\}.$$

The problem of deciding whether or not a formula belongs to the set $SAT^{\mathcal{L}}$ is called the **$\mathcal{L}$-satisfiability problem**.

Definition 5. *Given a finite truth value set N_n, an n-valued* **Łukasiewicz rule** *is an expression of one of the following two forms:*

- $x_1 \odot \cdots \odot x_k \to y_1 \odot \cdots \odot y_m \geq r$
- $x_1 \odot \cdots \odot x_k \to \bot \geq r'$

where $k \geq 0, m \geq 1, r, r' \in N_n$, and $x_1, \ldots, x_k,\ y_1, \ldots, y_m$ are propositional variables (if $k = 0$, $x_1 \odot ... \odot x_k$ stands for $\top$).

Definition 6. *An interpretation I satisfies a Łukasiewicz rule of the form*

$$x_1 \odot \cdots \odot x_k \to y_1 \odot \cdots \odot y_m \geq r$$

iff $I(x_1 \odot \cdots \odot x_k \to y_1 \odot \cdots \odot y_m) \geq r$, and a Łukasiewicz rule of the form

$$x_1 \odot \cdots \odot x_k \to \bot \geq r'$$

iff $1 - I(x_1 \odot \cdots \odot x_k) \geq r'$.

A set of n-valued Łukasiewicz rules is satisfiable iff there exists an interpretation that satisfies all the rules.

Remark 1. Łukasiewicz rules are called fuzzy Horn clauses by Borgwardt et al. [6], but we prefer not to refer to them as Horn clauses for the following reason: In Boolean propositional logic, a Horn clause is defined as a clause having at most one positive literal. Given a finite set of m Boolean Horn clauses of the form $x_1, \ldots, x_k \to y_i$, where $1 \leq i \leq m$ and all the clauses have the same antecedent, we have that such a set is equivalent to the clause $x_1, \ldots, x_k \to y_1, \ldots, y_m$, whose extension to Łukasiewicz logic corresponds to the first type of Łukasiewicz rules. However, in Lukasiewicz logic, a finite set of m rules of the form $x_1 \odot \cdots \odot x_k \to y_i \geq r$, where $1 \leq i \leq m$, is not equivalent to the rule $x_1 \odot \cdots \odot x_k \to y_1 \odot \cdots \odot y_m \geq r$.

In Section 3, we show that deciding the satisfiability of a set of 3-valued Łukasiewicz rules containing only rules of the form $x_1 \odot \cdots \odot x_k \to y_i \geq r$ or $x_1 \odot \cdots \odot x_l \to \bot \geq r'$ has linear-time complexity as in the Boolean case. However, deciding the satisfiability of the rules defined by Borgwardt et al. is polynomially solvable in the Boolean case, but is NP-complete in the Łukasiewicz case.

Remark 2. Observe that Łukasiewicz rules of the form $x_1 \odot \cdots \odot x_k \rightarrow y_1 \odot \cdots \odot y_m \geq r$ can be represented using strong disjunctions instead of strong conjuntions as $\neg x_1 \oplus \cdots \oplus \neg x_k \oplus \neg(\neg y_1 \oplus \cdots \oplus \neg y_m) \geq r$, and Łukasiewicz rules of the form $x_1 \odot \cdots \odot x_{k'} \rightarrow \bot \geq r'$ can be represented as $\neg x_1 \oplus \cdots \oplus \neg x_{k'} \geq r'$. So, Łukasiewicz rules are a fragment of the Łukasiewicz clausal forms defined in [5].

3 Complexity of the Satisfiability Problem of 3-Valued Łukasiewicz Rules

In this section we prove that the satisfiability problem of 3-valued Łukasiewicz rules is NP-complete, and give one subcase in which it can be solved in polynomial time.

Lemma 1. *The satisfiability problem of* 3*-valued Łukasiewicz rules is NP-complete.*

Proof. We will show that (i) this problem belongs to NP, and (ii) the Boolean 3-SAT problem is polynomially reducible to our problem.

The satisfiability problem of 3-valued Łukasiewicz rules clearly belongs to NP: given a set of rules, a nondeterministic algorithm can guess a satisfying interpretation and check that it satisfies the formula in polynomial time.

For what respects the second claim, let $\phi = \bigwedge_{i=1}^{n}(l_i^1 \vee l_i^2 \vee l_i^3)$ be a Boolean 3-SAT instance, where l_i^1, l_i^2, l_i^3 are literals over the set of Boolean variables $\{x_1, \ldots, x_m\}$. We construct the following set ϕ' of 3-valued Łukasiewicz rules from ϕ, over the set of three-valued variables $\{y_1, y_1', \ldots, y_m, y_m'\}$ [3] as follows:

1. For every Boolean variable x_k, $1 \leq k \leq m$, we add to ϕ' the following two rules:

   $\mathtt{A}_k)$ $\quad y_k \odot y_k' \rightarrow \bot \geq \frac{1}{2}$

   $\mathtt{B}_k)$ $\quad \rightarrow y_k \odot y_k' \geq \frac{1}{2}$

   Observe that any interpretation I that satisfies the previous two rules has a very determined behaviour. From $\mathtt{B}_k)$, it must hold that $I(y_k \odot y_k') \geq \frac{1}{2}$, and thus, by definition of the Łukasiewicz conjunction operation, either both y_k and y_k' are interpreted to 1 or one of them is interpreted to $\frac{1}{2}$, while the other is interpreted to 1. Moreover, if I must satisfy also rule $\mathtt{A}_k)$, it is not possible that y_k and y_k' are both interpreted to 1, so the only interpretations that meet all the requirements are those that send one of these variables to 1 and the other to $\frac{1}{2}$. In other words, exactly one of y_k and y_k' is evaluated to 1 in a satisfying interpretation, while the other is evaluated to $\frac{1}{2}$. The intuition behind this is that $I(y_k) = 1$ means that x_k is $true$, and $I(y_k') = 1$ means that x_k is $false$.
2. Let ρ be the function that maps (Boolean) literals to three-valued variables given by:

$$\rho(l_i^j) = \begin{cases} y_k \text{ if } l_i^j = x_k \\ y_k' \text{ if } l_i^j = \neg x_k \end{cases}$$

[3] Observe that only literals with positive polarity appear in Łukasiewicz rules, but the 3-SAT instance ϕ can contain occurrences of both positive and negative literals. Thus, we introduce the variable y_k' to simulate the literal $\neg x_k$, whereas y_k will simulate the literal x_k.

Then, for every clause $\delta_i = l_i^1 \vee l_i^2 \vee l_i^3$ in ϕ, we add to ϕ' the rule

$\mathtt{C}_{\delta_i}\mathtt{)}\quad \rho(\neg l_i^1) \odot \rho(\neg l_i^2) \odot \rho(\neg l_i^3) \to \bot \geq \frac{1}{2}$

Observe that this rule is only falsified if $\rho(\neg l_i^1) = \rho(\neg l_i^2) = \rho(\neg l_i^3) = 1$. This is equivalent to say that any interpretation that satisfies $\mathtt{C}_{\delta_i}\mathtt{)}$ must send at least one $\rho(\neg l_i^1), \rho(\neg l_i^2)$ or $\rho(\neg l_i^3)$ to a value that is less than or equal to $\frac{1}{2}$ (and if this interpretation also satisfies the family or rules $\{\mathtt{A}_k\mathtt{)}\,,\,\mathtt{B}_k\mathtt{)}\}_{1\leq k\leq m}$, this value will be in fact equal to $\frac{1}{2}$). Notice that the clause $l_i^1 \vee l_i^2 \vee l_i^3$ is only falsified if a Boolean interpretation sets l_i^1, l_i^2, l_i^3 to *false* . Also notice that the clause $l_i^1 \vee l_i^2 \vee l_i^3$ is equivalent to the Boolean rule $\neg l_i^1 \wedge \neg l_i^2 \wedge \neg l_i^3 \to \bot$.

This reduction can clearly be performed in polynomial time, and the size of ϕ' is linear in the size of ϕ. We also prove that ϕ' is satisfiable if and only if ϕ is satisfiable.

First, let I' be an evaluation on Ł$_3$ such that I' satisfies ϕ'. By construction of ϕ', from rules $\mathtt{A}_k\mathtt{)}$ and $\mathtt{B}_k\mathtt{)}$ we have that either $I'(y_k) = 1$ and $I'(y'_k) = \frac{1}{2}$ or $I'(y'_k) = 1$ and $I'(y_k) = \frac{1}{2}$ (for each $1 \leq k \leq m$). Moreover, for each $\delta_i = l_i^1 \vee l_i^2 \vee l_i^3$ of ϕ, rule $\mathtt{C}_{\delta_i}\mathtt{)}$ implies that I' sets to $\frac{1}{2}$ at least one of the literals $\rho(\neg l_i^1), \rho(\neg l_i^2), \rho(\neg l_i^3)$. Let then I be the Boolean interpretation defined by

$$I(x_k) = \begin{cases} \mathit{true} \text{ if } I'(y_k) = 1 \\ \mathit{false} \text{ if } I'(y'_k) = 1 \end{cases}$$

I satisfies at least one literal from each clause of ϕ. Indeed, consider without loss of generality that $I'(\rho(\neg l_i^1)) = \frac{1}{2}$.

- If $l_i^1 = x_j$, for some $1 \leq j \leq m$, then $\rho(\neg l_i^1) = y'_j$, and so, it is mandatory that $I'(y_j) = 1$, making $I(x_j) = I(l_i^1) = \mathit{true}$.
- If $l_i^1 = \neg x_j$ then $\rho(\neg l_i^1) = y_j$, and similarly, $I'(y'_j) = 1$. This implies that $I(x_j) = \mathit{false}$, making $I(l_i^1) = \mathit{true}$.

Then, I is a Boolean evaluation satisfying ϕ.

For the other direction, assume that I is a Boolean interpretation satisfying ϕ. We construct a three-valued interpretation I' from I as follows:

$$I'(y_k) = \begin{cases} 1 \text{ if } I(x_k) = \mathit{true} \\ \frac{1}{2} \text{ if } I(x_k) = \mathit{false} \end{cases} \quad I'(y_k{}') = \begin{cases} \frac{1}{2} \text{ if } I(x_k) = \mathit{true} \\ 1 \text{ if } I(x_k) = \mathit{false} \end{cases}$$

I' is constructed in such a way that it naturally satisfies the family of rules $\{\mathtt{A}_k\mathtt{)},\mathtt{B}_k\mathtt{)}\}_{1\leq k\leq m}$. On the other hand, for every clause $\delta_i = l_i^1 \vee l_i^2 \vee l_i^3$ in ϕ, since I satisfies ϕ, at least one of the literals l_i^1, l_i^2, l_i^3 is set to true. Assume without loss of generality that the satisfied literal is l_i^1. There are two possibilities:

- If $l_i^1 = x_s$ for some $1 \leq s \leq m$, (and so, $I(x_s) = \mathit{true}$), I' satisfies rule $\mathtt{C}_{\delta_i}\mathtt{)}$ because $I'(y'_s) = \frac{1}{2}$ due to the fact that $I(x_s) = \mathit{true}$.
- If $l_i^1 = \neg x_s$ for some $1 \leq s \leq m$, (and so, $I(x_s) = \mathit{false}$), I' satisfies rule $\mathtt{C}_{\delta_i}\mathtt{)}$ because $I'(y_s) = \frac{1}{2}$ due to the fact that $I(x_s) = \mathit{false}$.

In all these cases, I' satisfies ϕ', which concludes the proof.

Remark 3. Observe that all the Łukasiewicz rules used in the reduction have either the consequent or the antecedent empty. So, the satisfiability problem of this fragment of 3-valued Łukasiewicz rules is NP-complete too. Also observe that this fragment corresponds to Łukasiewicz clausal forms only containing clauses of the form $(\neg x_1 \oplus \cdots \oplus \neg x_k) \geq r$ and of the form $(\neg x_1 \oplus \cdots \oplus \neg x_{k'}) \leq r'$.

Lemma 2. *The problem of deciding the satisfiability of a finite set ϕ of 3-valued Łukasiewicz rules containing only rules of the form $x_1 \odot \cdots \odot x_k \rightarrow y_i \geq r$ or $x_1 \odot \cdots \odot x_l \rightarrow \bot \geq r'$ is polynomially solvable.*

Proof. We can assume that there is no rule in which $r = 0$ or $r' = 0$ because such rules are tautologies and can be removed. Assume that r and r' are either $\frac{1}{2}$ or 1. If ϕ contains a rule of the form $x_j \rightarrow \bot \geq 1$, then x_j should be evaluated to 0 and, therefore, all the rules having x_j in the antecedent can be removed, and all the occurrences of x_j in the consequent of a rule can be substituted by $\bot$. This process is repeated until there are no more rules of the form $x_j \rightarrow \bot \geq 1$ or the empty rule is derived. If the empty rule is derived, it means that ϕ is unsatisfiable. Otherwise, we continue simplifying ϕ as follows: if ϕ contains a rule of the form $\rightarrow y_i \geq 1$, then y_i should be evaluated to 1 and, therefore, all the rules having y_i in the consequent can be removed, and all the occurrences of y_i in the antecedent of a rule can be removed too. This process is repeated until there are no more rules of the form $\rightarrow y_i \geq 1$ or the empty rule is derived. If the empty rule is derived, it means that ϕ is unsatisfiable.

Otherwise, ϕ is satisfied by the interpretation I that sets to $\frac{1}{2}$ all the variables. To see this, observe that the only remaining rules with exactly one literal are either of the form $x_j \rightarrow \bot \geq \frac{1}{2}$ or $\rightarrow y_i \geq \frac{1}{2}$ and, hence, they are satisfied by I. Besides, any rule containing more than one literal is also satisfied by I: when there are at least two literals in the antecedent, the rule evaluates to 1 under I because $\frac{1}{2} \odot \frac{1}{2} = 0$; and when there is a literal in the antecedent and one literal in the consequent, the rule evaluates also to 1 under I because $\min\{1, 1 - \frac{1}{2} + \frac{1}{2}\} = 1$.

Since the number of rules with exactly one literal that can be derived is bounded by the total number of rules, deciding the satisfiability of ϕ can be performed in polynomial time.

Lemma 2 proves that the satisfiability problem of the generalization of Horn clauses to our setting is polynomial solvable. We claim that a linear-time algorithm could be implemented by adapting the Boolean linear-time unit propagation algorithm for Boolean CNFs described in [12].

4 Conclusions

We proved that the satisfiability problem of 3-valued Łukasiewicz rules is NP-complete, but is polynomially solvable when the rules have at most one literal in the consequent. Actually, to get intractable instances is enough to require that the rules have an empty antecedent or an empty consequent. These results solve an open problem posed by Borgwardt et al. in [6], and the 3-valued Łukasiewicz rules become a challenging benchmark for Łukasiewicz satisfiability solvers.

As future work we propose to analyze the complexity of infinitely-valued Łukasiewicz rules, as well as identify —when testing the satisfiability of 3-valued Łukasiewicz rules with a fixed number of variables per rule, and generated uniformly at random— an easy-hard-easy pattern, and a phase transition phenomenon as the clause-to-variable ratio varies, similar to the ones identified for Łukasiewicz clausal forms in [5].

References

1. Stefano Aguzzoli, Brunella Gerla, and Zuzana Haniková. Complexity issues in Basic logic. *Soft Computing*, 9(12):919–934, 2005.
2. Carlos Ansótegui, Miquel Bofill, Felip Manyà, and Mateu Villaret. Building automated theorem provers for infinitely-valued logics with satisfiability modulo theory solvers. In *Proceedings, 42nd International Symposium on Multiple-Valued Logics (ISMVL), Victoria, BC, Canada*, pages 25–30. IEEE CS Press, 2012.
3. Carlos Ansótegui, Miquel Bofill, Felip Manyà, and Mateu Villaret. Automated theorem provers for multiple-valued logics with satisfiability modulo theory solvers. *Fuzzy Sets and Systems*, 2015. http://dx.doi.org/10.1016/j.fss.2015.04.011.
4. Carlos Ansótegui, Miquel Bofill, Felip Manyà, and Mateu Villaret. SAT and SMT technology for many-valued logics. *Multiple-Valued Logic and Soft Computing*, 24(1-4):151–172, 2015.
5. Miquel Bofill, Felip Manyà, Mateu Villaret, and Amanda Vidal. Finding hard instances of satisfiability in Łukasiewicz logics. In *Proceedings, 45th International Symposium on Multiple-Valued Logics (ISMVL), Waterloo, Canada*, page In press. IEEE CS Press, 2015.
6. Stefan Borgwardt, Marco Cerami, and Rafael Peñaloza. Many-valued Horn logic is hard. In *Proceedings of the First Workshop on Logics for Reasoning about Preferences, Uncertainty, and Vagueness, PRUV 2014, co-located with IJCAR 2014, Vienna, Austria*, pages 52–58, 2014.
7. M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-completeness*. Freeman, San Francisco, 1979.
8. Petr Hájek. *Metamathematics of Fuzzy Logic*. Kluwer, Dordrecht, 1998.
9. George Metcalfe, Nicola Olivetti, and Dov M. Gabbay. *Proof Theory of Fuzzy Logics*, volume 36 of *Applied Logic Series*. Springer, 2009.
10. Amanda Vidal. NiBLoS: a nice BL-logics solver. Master's thesis, Universitat de Barcelona, Barcelona, Spain, 2012.
11. Amanda Vidal, Félix Bou, and Lluis Godo. An SMT-based solver for continuous t-norm based logics. In *Proceedings of the 6th International Conference on Scalable Uncertainty Management, SUM 2012, Marburg, Germany*, pages 633–640. Springer LNCS 7520, 2012.
12. Hantao Zhang and Mark E. Stickel. An efficient algorithm for unit propagation. In *In Proceedings of the Fourth International Symposium on Artificial Intelligence and Mathematics (AI-MATH'96), Fort Lauderdale (Florida), USA*, pages 166–169, 1996.