MNiBLoS: A SMT-based solver for continuous t-norm based logics and some of their modal expansions

Amanda Vidal

Institute of Artificial Intelligence - CSIC
Campus UAB, E, 08193, Barcelona (Spain)
&
Institute of Computer Science, Czech Academy of Sciences
Pod vodárenskou věží 2, 182 07 Prague (Czech Republic)

Abstract

In the literature, little attention has been paid to the development of solvers for systems of mathematical fuzzy logic, and in particular, there are few works concerned with infinitely-valued logics. In this paper it is presented *mNiBLoS (a modal Nice BL-Logics Solver)*: a modular SMT-based solver complete with respect to a wide family of continuous t-norm based fuzzy modal logics (both with finite and infinite universes), restricting the modal structures to the finite ones. At the propositional level, the solver works with some of the best known infinitely-valued fuzzy logics (including BL, Łukasiewicz, Gödel and product logics), and with all the continuous t-norm based logics that can be finitely expressed in terms of the previous ones; concerning the modal expansion, mNiBLoS imposes no boundary on the cardinality of the modal structures considered. The solver allows to test 1-satisfiability of equations, tautologicity and logical consequence problems. The logical language supported extends the usual one of fuzzy modal logics with rational constants and the Monteiro-Baaz Δ operator. The code of mNiBLoS is of free distribution and can be found in the web page of the author.

Keywords: Fuzzy logics, modal logics, automated reasoning, continuous t-norms, SMT, infinitely valued logics

1. Introduction

Mathematical Fuzzy Logic (MFL) is a sub discipline of Mathematical Logic that studies a certain family of formal logical systems whose algebraic semantics involve some notion of truth degree.These truth degrees have motivations coming from different fields, like philosophy, fuzzy set theory and many-valued logics. Along the XXth century, and particularly from the 90s, MFL have been widely studied, and there is a large number of studies on the applications of this framework to model knowledge and reasoning with incomplete and uncertain information. However, in the literature, with a few exceptions mainly for finitely valued Łukasiewicz logics [29, 31, 30], little attention has been paid to the development of efficient solvers for systems of mathematical fuzzy logic, even though there is an important number of studies on complexity and proof theory for them [28, 22, 23, 3, 20]. This is a problem that limits the use of fuzzy logics in real applications, while the number of theoretical proposals relying on these logics has rapidly grown. Moreover, the relative poor knowledge of more complex t-norm based logics outside the fuzzy logic community, and in particular, in more applied areas, limits the use of these logics in situations where ad-hoc logics (i.e., logics where the behavior of the connectives is defined accordingly to the user's necessities) could be very interesting.

In [5] it is proposed a new approach for implementing a theorem prover (i.e., that checks validity of a given formula) for Łukasiewicz, Gödel and product fuzzy logics using Satisfiability Modulo Theories (SMT). The idea of using SMT solvers in order to solve this problem is new and allows to treat infinitely-valued logics without relying on reductions to finite universes, and so being able to solve the validity problem over the previous logics. The results are very interesting in particular for Łukasiewicz and Gödel logics because the execution times turn out to be optimistically efficient. In the case of product logic, while the importance of this new solver is clear since no previous automatic tool for product logic existed before, the results are however not very satisfactory: the execution times are quite long, mainly due to the fact that the implementation of this logic uses theories of the SMT-solver with non-linear arithmetic. Inspired by the previous work, in [35] it is presented a solver for the whole family of logics based on a (finitely representable) continuous t-norm and BL, and the particular case of the product logic is strongly enhanced relying on theoretical results from [19]. However, if truth constants are added to the language, their behavior (for what concerns the product logic) following this new approach is completely uncontrolled. Moreover, it is not studied how to include this new treatment of the product logic when considering arbitrary continuous t-norms that included some product t-norm in their representation as ordinal sums. Thus, the new approach is limited to product logic only.

Email address: amanda@iiia.csic.es (Amanda Vidal)
URL: http://www.iiia.csic.es/~amanda/ (Amanda Vidal)

On the other hand, modal expansions of MFL, which allow to reason over *qualified* concepts (like always, sometimes, etc.), have also been intensively studied in the latter years, partially motivated by the rich expressive power of these logics and their possible applications. Concerning the development of applications oriented to automatically reason over these kind of logics, the main works lie within the framework of the so-called Fuzzy Description Logics (FDL), multi-modal logics oriented to represent ontologies of different areas (see for instance [33, 8, 34]). In relation to the present paper, we can cite two works presenting a reasoner for some FDLs which, while developing several very involved tools related to FDL, are however quite restricted in terms of the logics and basic tasks implemented. In [9] it is presented a solver that checks satisfiability over Łukasiewicz and Zadeh logics. It considers finite sets of truth values, sufficient to deal with the satisfiability problem in a very efficient way over the implemented logics, but not enough to check validity or to work with other logics that do not allow a finitary reduction (like for instance Hájek's BL or product logic). On the other hand, some notes towards the implementation of a solver that works with the problem of positive satisfiability for product description logic can be found in [4]. The approach followed to solve reason at a propositional level over the product standard algebra is the same one followed in [35]. On the other hand, concerning the expansion to FDL, it relies in the possibility of reducing a problem of positive satisfiability to the satisfiability of a quantifier free boolean formula, which is, however, not likely to work in more general cases (like 1-satisfiability or validity). Moreover, the resulting formula has non-linear real arithmetic properties, making it quite challenging, as we already commented above, from an efficiency point of view.

Our objective within this paper is expanding the solver presented in [35] and solve several problems left open in the previous work and in [5]. We present a solver (mNiBLoS) with a quite simple interface, a rich expressive power and a reasonable level of efficiency, that hopefully can help a non-specialized community to work with several fuzzy modal logics through a quite short process of intuitive learning. Even if the software presented is command-line only, interested users needs not to be more than slightly familiar with continuous t-norms and logical languages in order to be able to use it: only the decomposition of the t-norm, the desired task and the formulas/equations associated to the desired task are asked to the user. We offer the possibility of working with a large set (countable infinitely many) of infinitely valued logics, including Hajek's Basic Logic and logics based on finite ordinal sums of Gödel, Łukasiewicz and product t-norms. mNiBLoS also supports a rich language, including rational truth constants, Δ and the usual modal operators $\Box$ and $\Diamond$. We have implemented the basic operations that seem to be of most interest concerning the use of a fuzzy modal logic: validity, logical deduction and 1-satisfiability (united to the generation of a solution when possible) of sets of formulas or equations. Focusing on the efficiency of the solver, we have proven that reasoning over a logic based on a continuous t-norm that includes a product t-norm can be equally done over the same structure but whose product components are moved into the negative cone of the real numbers, by means of a translation that allows to keep working with rational truth constants. Moreover, we give a constructive proof of the fact that the operations we are concerned with are decidable over over finite Kripke structures, in a similar fashion as it was done in [24] for Łukasiewicz FDL. For doing this, we present an algorithm that builds the minimum (finite) structure that is necessary to check in order to determine, for a finite set of formulas/equations, the answer for the theoremhood, logical deduction and satisfiability problems.

Structure of the paper. Section 2 introduces the propositional logics considered and some of their characteristics, the modal expansions we are going to implement and gives a brief introduction to SMT. Section 3 presents the theoretical results developed for the design of mNiBLoS: the approach to product logic based on linear arithmetic, the treatment of BL and the study of the expansions with truth constants and modal operators. Section 4 comments on the design decisions of the solver and explains some details of its implementation, and shows several empirical studies run on mNiBLoS. We conclude the paper with Section 5, by outlining the conclusions after this research and some possible future work.

2. Preliminaries

With the objective of being as self-contained as possible, we first describe in this section the logical background necessary to present the theoretical results of Section 4. With this in mind, we mainly focus on semantic aspects of the logics, which are the ones further used in the development of mNiBLoS. We refer the interested reader for details on the topics presented in this section to well-known monographs on fuzzy and many-valued logics like [23] and [20]. In the last part of this section we briefly present Satisfiability Modulo Theories and provide some details concerning the particular theories used in our proposal from Section 4.

2.1. Continuous t-norm based fuzzy logics

Fuzzy logic in its narrow/technical sense, as presented by Zadeh in his foundational paper, is a term that refers to logics with more than two truth vales that handle gradual properties (as opposed to fuzzy logic in its wide sense, which is a generic expression that mostly refers to that part of soft computing that uses fuzzy sets and rules). A formalism that has been commonly adopted in order to uniformly approach the study of fuzzy logics is that of taking as truth values subsets of the real unit interval, and generalizing the definition of the usual logical operations (conjunction, implication, negation, etc) to this framework. Already the first generalizations studied for the conjunction (for instance, the Łukasiewicz and Gödel strong conjunctions) are particular cases of *triangular norms* (t-norms): binary operations $*$ on $[0, 1]$ that are associative, commutative, monotonically increasing and have 1 as the neutral element. Concerning the implication, the operation that fits the best with the previous definition of strong conjunction $*$ is the one arising from its residuated operation (whenever it exists):

$$x \Rightarrow_* y = \max\{z : x * z \leq y\}$$

It is known that the necessary and sufficient condition for a t-norm to have a residuum is that it is left-continuous, but the more restricted class of continuous t-norms enjoys other important characteristics that made us restrict the current work to them. Namely, it is possible to easily build new continuous t-norms from others in the following way.

Lemma 2.1. *Let $\{*_i\}_{i\in I}$ for I countable be a set of continuous t-norms and $\{(b_i, t_i)\}_{i\in I}$ a family of pairwise disjoint open intervals of $[0,1]$ such that $\bigcup_{i\in I}[b_i, t_i] = [0,1]$. Then, the function $*\colon [0,1]\times[0,1]\to[0,1]$ defined as*

$$x * y = \begin{cases} b_i + (t_i - b_i)\cdot(\frac{x-b_i}{t_i-b_i} *_i \frac{y-b_i}{t_i-b_i}) & \text{if } x, y \in [b_i, t_i] \\ min\{x, y\} & \text{otherwise} \end{cases}$$

is a continuous t-norm.

The t-norm resulting from the previous construction is called the ***ordinal sum*** of $\{\langle *_i, (b_i, t_i)\rangle\}_{i\in I}$ and is denoted by

$$* = \bigoplus_{i\in I}\langle *_i, (b_i, t_i)\rangle$$

When I is finite with cardinal n, we also refer to it by

$$* = \langle *_1, (b_1, t_1)\rangle \oplus \ldots \oplus \langle *_n, (b_n, t_n)\rangle$$

Intuitively, the construction of an ordinal sum is just "piling" different t-norms and considering the ordered structure generated with this union.

In can be checked that the residuum of the previous construction also has a nice characterization in terms of $\{\langle *_i, (b_i, t_i)\rangle\}_{i\in I}$. If $* = \bigoplus_{i\in I}\langle *_i, (b_i, t_i)\rangle$, its residuum $\Rightarrow_*$ is given by:

$$x \Rightarrow_* y = \begin{cases} 1 & \text{if } x \le y \\ b_i + (t_i - b_i)\cdot(\frac{x-b_i}{t_i-b_i} \Rightarrow_{*_i} \frac{y-b_i}{t_i-b_i}) & \text{if } b_i \le y < x \le t_i \\ y & \text{otherwise} \end{cases}$$

The three best-known continuous t-norms are the Łukasiewicz Gödel and product t-norms, defined by:

Gödel t-norm

$$\begin{aligned} x *_G y &= \min(x, y) \\ x \Rightarrow_G y &= \begin{cases} 1, & \text{if } x \le y \\ y, & \text{otherwise.} \end{cases} \end{aligned}$$

Łukasiewicz t-norm

$$\begin{aligned} x *_Ł y &= \max(x + y - 1, 0) \\ x \Rightarrow_Ł y &= \begin{cases} 1, & \text{if } x \le y \\ 1 - x + y, & \text{otherwise.} \end{cases} \end{aligned}$$

Product t-norm

$$\begin{aligned} x *_\Pi y &= x \cdot y \quad \text{(product of reals)} \\ x \Rightarrow_\Pi y &= \begin{cases} 1, & \text{if } x \le y \\ y/x, & \text{otherwise.} \end{cases} \end{aligned}$$

The following is the well-known result by Mostert and Shields of decomposition of continuous t-norms in terms of the three previous ones.

Lemma 2.2 (c.f. [27])**.** *Any continuous t-norm can be expressed as an ordinal sum of countably many components $\langle *_i, (b_i, t_i)\rangle$ with $*_i \in \{*_\Pi, *_Ł, *_G\}$.*

This decomposition allows to work with any continuous t-norm in a modular way, paying attention only to the t-norms $\{*_\Pi, *_Ł, *_G\}$ and to the ordinal sum construction.

When we talk about a logic based on a continuous t-norm $*$, what formally is done is considering a logical calculi with the real interval $[0,1]$ as set of truth values and 1 as only designated element, and connectives given by a strong conjunction $\&$, an implication $\to$ and the truth constant $\overline{0}$, interpreted respectively by the t-norm $*$, its residuum $\Rightarrow_*$, and number 0 (see for instance [23]). Further connectives in the language can be defined as follows:

$$\begin{aligned} \neg\varphi &\quad \text{is} \quad \varphi \to \overline{0}, \\ \varphi \wedge \psi &\quad \text{is} \quad \varphi\&(\varphi \to \psi), \\ \varphi \vee \psi &\quad \text{is} \quad ((\varphi \to \psi) \to \psi) \wedge ((\psi \to \varphi) \to \varphi), \\ \varphi \equiv \psi &\quad \text{is} \quad (\varphi \to \psi)\&(\psi \to \varphi). \end{aligned}$$

Given a continuous t-norm, ***$*$-evaluations*** of propositional variables are mappings e assigning to each propositional variable p a truth value $e(p) \in [0,1]$, which is uniquely extended to compound formulas as follows:

$$\begin{aligned} e(\overline{0}) &= 0 \\ e(\varphi\&\psi) &= e(\varphi) * e(\psi) \\ e(\varphi \to \psi) &= e(\varphi) \Rightarrow e(\psi) \end{aligned}$$

Each continuous t-norm defines an algebra

$$[\mathbf{0}, \mathbf{1}]_* = ([0,1], \min, \max, *, \Rightarrow, 0, 1)$$

called ***standard $*$-algebra***, and $*$-evaluations are simply the homomorphisms (i.e., mappings that preserve the connectives) from the formulas to $[\mathbf{0}, \mathbf{1}]_*$. More in general, ***$*$-algebras*** are those algebras that are defined using the same set of equations that the standard one, but over different universes, i.e., the algebras belonging to the variety generated by $[\mathbf{0}, \mathbf{1}]_*$.

Note that, from the above definitions, $e(\neg\varphi) = e(\varphi) \Rightarrow 0$, $e(\varphi \wedge \psi) = \min(e(\varphi), e(\psi))$, $e(\varphi \vee \psi) = \max(e(\varphi), e(\psi))$ and $e(\varphi \equiv \psi) = e(\varphi \to \psi) * e(\psi \to \varphi)$.

For a continuous t-norm $*$ and any finite[1] set of formulas $\Gamma \cup \{\varphi\}$ we say that φ is logical consequence of Γ in the logic of $*$, and write $\Gamma \models_{[\mathbf{0},\mathbf{1}]_*} \varphi$, whenever for any $*$-evaluation e such that $e(\gamma) = 1$ for all $\gamma \in \Gamma$, it holds that $e(\varphi) = 1$ too. In the particular case of Γ being the empty set, we say that φ is a theorem of the logic, or equivalently, that it is valid in $[\mathbf{0}, \mathbf{1}]_*$. A formula φ is *1-satisfiable* in L_* if $e(\varphi) = 1$ for some L_*-evaluation e.

[1] It is not in the scope of this paper, for the obvious motivation of applicability, to treat infinitary logics, i.e., those that consider also the deductions from infinite sets of premises.

If we consider instead a set C of continuous t-norms, we define the finitary logical deduction similarly, letting $\Gamma \models_{\{[\mathbf{0},\mathbf{1}]_*:*\in C\}} \varphi$ whenever for any $*$ in C, $\Gamma \models_{[\mathbf{0},\mathbf{1}]_*} \varphi$. The notions of theorems of the logic and 1-satisfiability are defined accordingly. More in general, a logic L is simply a logical deduction relation, and we write $\Gamma \vdash_{\mathrm{L}} \varphi$ whenever φ is consequence of Γ in L.

The most well known fuzzy logics, namely Łukasiewicz (Ł), Gödel (G) and product (Π) logics are defined as above using their corresponding t-norms. Moreover, the logic of the standard algebras of all continuous t-norms, BL, is introduced and studied by Hajek in [23] as the most basic logic that has the deductions shared by all continuous t-norm based logics. [2] It is known that any axiomatic extension of BL is the logic of a continuous t-norm, and so we can call BL-***logics*** the logics arising from a continuous t-norm. Analogously, we call BL-***algebras*** to the $*$-algebras for $*$ being a continuous t-norm.

An expansion of a logic consist on the addition of new connectives, and possibly, axioms and inference rules, to it. Usual expansions of BL-logics are the ones that include truth constants different from $\{0, 1\}$ and the Monteiro-Baaz Δ operator, that acts as a *de-fuzzyfier*. Semantically, the inclusion of truth constants assigned to the rational numbers in $[0, 1]$ consists simply in adding a new set of symbols $C = \{\overline{c} : c \in [0,1] \cap \mathbb{Q}\}$[3] to the language and fixing, for each continuous t-norm $*$ and $*$-interpretation e

$$e(\overline{c}) = c$$

The unary Δ connective is interpreted in the standard algebra[4] of a continuous t-norm is given by

$$\Delta a = \begin{cases} 1 & \text{if } a = 1 \\ 0 & \text{otherwise} \end{cases}$$

We denote the standard algebra of $*$ expanded with rational truth constants and Δ by $[\mathbf{0},\mathbf{1}]_*^{\mathbb{Q}}$, and sometimes refer to it as the canonical standard algebra of $*$.

The previous expansions seem of great interest if the logic is oriented to represent real-world behaviors, since they offer a more precise language that allows to explicitly refer to the truth values in the logic and to determine, in some sense, the strict truth of terms.

2.2. Fuzzy Modal Logics

Modal logics expand classical propositional logic with modal operators, that allow to *qualify* sentences and talk about concepts like *possibly* or *always*, using $\Diamond$ and $\Box$ as usual connectives to denote these concepts. The most intuitive semantics for these logics is based on (classical) Kripke models: structures of the form $\langle W, R, V\rangle$, where W is a non-empty set of so-called worlds, $R \subseteq W \times W$ is an accessibility relation between worlds and $V : Var \to \mathcal{P}(W)$, which determines the worlds where a variable is true. Propositional formulas are evaluated depending only on the mapping V, while the interpretation of modal operators depends also on R: for a formula φ, $\Box\varphi$ is true in a world u whenever φ is true in all the worlds accessible (trough R) from u (and similarly, $\Diamond\varphi$ is true whenever there is some accessible world where φ is true). It is usual to think in the previous semantics like a possible-world semantics, where $\langle v, u\rangle \in R$ means that u is a world (i.e., interpretation of variables) that can be reached from v by means of a certain action or change in the universe determined by R.

This semantics can be naturally generalized to a fuzzy setting by simply enriching the structures with fuzzy evaluations. Some previous works on modal expansions of continuous t-norm based fuzzy logics are [17, 16], [14][25] and [36]. Other works, mainly corresponding to FDL (see for instance [33, 8, 34]), expand this semantics with more than one accessibility relation (and so, more modal operators). The definition adapted to complete (i.e., where all arbitrary infima and suprema exist) linearly ordered BL-algebras, with Δ and truth constants from C, as presented in the previous section, is the following one.

Definition 2.3. Let $\mathbf{A} = \langle A, \odot, \Rightarrow, \Delta, \{\overline{c}^{\mathbf{A}}\}_{c\in C}\rangle$ be a complete linearly ordered BL-algebra with Δ and truth constants from C. An **A-*Kripke model*** is a triple $\mathfrak{M} = \langle W, R, e\rangle$ where

- W is a non-empty set (of so called worlds)
- R is a fuzzy accessibility relation in W, i.e., a mapping from $W \times W$ to A
- e is a mapping of the variables at each world to A, i.e., a mapping from $W \times \mathcal{V}ar \cup C$ to $[0, 1]$, uniquely extended to the whole set of formulas by letting:
  - $e(w, \overline{c}) = c$
  - $e(w, \Delta\varphi) = \Delta e(w, \varphi)$
  - $e(w, \varphi\&\psi) = e(w, \varphi) \odot e(w, \psi)$
  - $e(w, \varphi \to \psi) = e(w, \varphi) \Rightarrow e(w, \psi)$
  - $e(w, \Box\varphi) = \inf\{R(w, v) \Rightarrow e(v, \varphi) : v \in W\}$
  - $e(w, \Diamond\varphi) = \sup\{R(w, v) \odot e(v, \varphi) : v \in W\}$

If the set W is finite we say that the model is ***finite***.

We say that a formula φ is true in a world w from a **A**-Kripke model $\mathfrak{M}$, and write $\mathfrak{M}, w \Vdash \varphi$ when $e(w, \varphi) = 1$. From here, two notions of logical deduction arise (the local and global one), but we will focus on the first one for being more representative of the meaning behind the modal operators (since the other one only depends on the structure as a whole). We refer the interested reader to well-known manuals on modal logics like [18], [7],. We say that a formula φ is ***local consequence*** of a finite set of premises Γ in a model $\mathfrak{M}$ when for each world w of the model for which $e(w, \gamma) = 1$ for all $\gamma \in \Gamma$, then $e(w, \varphi) = 1$ too. Similarly, for a class of Kripke models C, we say that φ is local consequence of Γ in C whenever it happens for all models in C. In this case, we write $\Gamma \Vdash_{\mathrm{C}} \varphi$.

We are also interested in checking the satisfiability of equations in classes of models. In our framework, we let an equation to be a triple $\langle op, \varphi_1, \varphi_2\rangle$ with φ_1, φ_2 are formulas and

[2] For simplicity, we will write $\vdash_{\mathrm{BL}}$ instead of $\models_{\{[\mathbf{0},\mathbf{1}]_*:*\text{ continuous t-norm }\}}$.

[3] Clearly, constants $\overline{0}$ and $\overline{1}$ coincide with the ones already existing in BL.

[4] We denote the algebraic operation and the connective in the language by the same symbol Δ, since there is no risk of confusion.

$op \in \{<, \leq, =, \geq, >\}$. We say that an equation $\langle op, \varphi_1, \varphi_2 \rangle$ is ***locally satisfiable*** in a class of models C when there is a model $\mathfrak{M} \in$ C and a world $w \in W$ such that $e(w, \varphi_1)\ op\ e(w, \varphi_2)$.[5]

It is not known in general whether the local modal logic of an arbitrary BL-algebra as above (or even of an standard BL-algebra $[\mathbf{0}, \mathbf{1}]_*$) is decidable (that is to say, the problems of 1-satisfiability and logical deduction), so in what follows we will mainly consider classes of models with a finite number of worlds.

Let us conclude this brief introduction to fuzzy modal logics by remarking some deep differences between classical and fuzzy modal logics, that can give the reader an idea of the new behavior of the modalities. First, it is not hard to see that the well-known normality axiom $\mathsf{K} : \Box(x \to y) \to (\Box x \to \Box y)$, valid in all normal modal logics (including the minimum classical modal logic), is no longer necessarily true in the fuzzy context. Indeed, it keeps being valid whenever the accessibility relation is crisp (see for instance [14]), but in other cases, it is very simple to build a counter-model with only two worlds for it. Another important difference we have already commented above is that, while the minimum classical (local) modal logic K and many of its extensions, including the most well known modal logics (whose models' accessibility relations enjoy properties like reflexivity, transitivity, seriality, etc.) are complete with respect to finite models, and theoremhood and 1-satisfiability problems are decidable thanks to this fact, this is not so clear in the many-valued setting. It has been proved in the case of Gödel logic [15], and arises naturally in the particular case of Łukasiewicz modal logic linked to the concept of witnessed models, but in other t-norms this is not known. On some ongoing works we have proved that global modal

2.3. Satisfiability Modulo Theories (SMT)

The satisfiability problem, i.e., determining whether a formula expressing a constraint has a solution, is one of the main problems in theoretical computer science. If this constraint refers to Boolean variables, then we are facing a well-known problem, the propositional satisfiability problem (SAT).

On the other hand, some problems require to be described in more expressive logics (like first order logics or many valued logics), and so a formalism extending SAT has also been widely studied: Satisfiability Modulo Theories (SMT). A SMT instance is the generalization of a Boolean formula in which some propositional variables are replaced by predicates with predefined interpretations from background theories. These predicates can then be bi-valued functions over non-binary variables.

The most common approach [32**?**] for the existing SMT solvers is the integration of a T-solver, i.e. a decision procedure for a given theory T, and a SAT solver. In this model, the SAT solver is in charge of the Boolean formula, while the T-solver analyzes sets of atomic constraints in T. With this, the T-solver checks the possible assignments generated by the SAT solver and rejects them if there exist inconsistencies with the theory. In doing so, it gets the efficiency of the SAT solvers for Boolean reasoning, long time tested, and the capability of the more concrete T-oriented algorithms inside the respective theory T.

The current general-use library for SMT is SMT-LIB [6], and there are several implementations of SMT-solvers for it.

For our experiments, we are using an established SMT-solver developed by Microsoft Research, Z3 [37, 21], which implements the two theories we need for our purposes, and which has been proved quite efficient. The main theory we require is the linear real arithmetic theory (i.e., $\langle \mathbb{R}, +, - \rangle$), and we also make use of arrays, even though this last theory could be removed by manually unraveling the variables of type array of length n in n new variables.

Nevertheless, mNiBLoS generates a code in SMT 2.0 format, so any other SMT-solver that implements the necessary theories can be used instead.

3. Theory behind mNiBLoS

As we commented in the Introductory section, the software application presented in this paper is oriented to open the practical use of many-valued logics to a public not exclusively belonging to the logic research community. For this reason we consider it is important to allow the use of a large family of logics without overlooking the efficiency when reasoning over these systems. For this reason, mNiBLoS is focused on working with a wide class of continuous t-norm based logics, which will be called "Nice BL Logics" (because they allow a "nice" representation). The only continuous t-norms not treated by mNiBLoS are those whose representation in terms of the Mostert and Shields theorem need an infinite number of components, with the exception of the BL logic, which is treated in an alternative fashion.

The motivation behind the interest in these Nice BL Logics is mainly practical. For an arbitrary left-continuous t-norm there is no general form to simplify the reasoning and thus, a reasoner for a logic based on it would simply consist on coding the operation specified by the user (and also its residuum) into the SMT solver. However, for Nice BL logics, a certain preprocessing of the operations can be done, codifying the reasoning over them in such a way that a much faster behavior (that the general approach commented above) can be obtained. On the other hand, we think that the potential users of this software have, with this amount of logics, a large enough basis to start exploiting the versatility of the t-norm based fuzzy logics.

Concerning the expansion of these fuzzy logics with modalities, a similar reasoning has lead us to face the problem in what we think is the most practical way. On the one hand, while it seems natural to think of problems that can be modeled within a Kripke structure (for instance, those that resort to graphs, or those related with the field of temporal logics), for what concerns non-theoretical uses it does not seem clear which kind of problems would need of a structure with an infinite number of worlds. Moreover, 1-satisfiability and deduction problems over these arbitrary structures, as we commented before, are

[5] Observe this is well defined since we are considering linearly ordered algebras only.

not known to be decidable. Since the objectives of this paper do not aim to cope with these issues we think that a first practical and useful solution is consider the modal logics arising from models with a finite set of worlds.

We present now some theoretical results that allow to design mNiBLoS in such a way that a greater modularity and gain of efficiency (respect to the solvers presented in [5] and [35]) is obtained.

3.1. The propositional level

3.1.1. Nice B*L logics*

Recall that any continuous t-norm can be expressed as an ordinal sum of countable many t-norms of the form the Łukasiewicz, Gödel and product t-norms. From this result, we know that a reasoner for the axiomatic extensions of the BL logic can be designed focusing only over intervals of $[0, 1]$ with the three previous t-norms. On the other hand, a computer application treating logics arising from one of these t-norms needs to take as an argument the t-norm itself. Since this is a function in $[0, 1]$, it does not seem clear how can this value can be specified if not as a list of Łukasiewicz Gödel and product components (each one associated with an interval in $[0, 1]$ determining the universe of that component), or with a unique name for some particular cases. This naturally limits the possible t-norms, expressed as lists, to those that are a finite ordinal sums of the three basic t-norms. However, logics arising from ordinal sums of the $*_Ł$, $*_G$ and $*_\Pi$ t-norms with a finite number of components are the main part but not the totality of the logics that will be accepted by our software application.

First, thinking in the possible applications, we have also included an additional family of operations (not strictly speaking continuous t-norms) that, in the same way that above, have naturally an associated (semantically defined) logic. We are talking about ordinal sums (with a finite number of components!) whose components can be either the previously commented three basic ones, or range over a finite universe, with uniformly distributed points. While it is well known that there do not exist finite linearly ordered product algebras different from the Boolean one (see for instance [19]), and thus there is no way to reason with the product operations over a finite universe, the cases of the Gödel and Łukasiewicz logics can enjoy this behavior. It is then possible to consider as components of an ordinal sum restrictions of the Łukasiewicz and Gödel ones to a finite universe over which the operations are closed. In order to simplify the notation, in this case we denote the operations in the representation of the ordinal sum by $*_{Łn}$ and $*_{Gn}$, for n being the number of (equidistant) elements considered in the universe.

We can extend the definition of ordinal sum in order to include these new operations as possible components in such a way that the sum of just one of them respectively coincides with the usual algebraic definition of the n-valued Łukasiewicz and Gödel logics over the real interval $[0, 1]$:

- $\mathbf{Ł_n}$ is the subalgebra of $[\mathbf{0}, \mathbf{1}]_{*_Ł}$ with universe $\{0, \frac{1}{n-1}, ..., \frac{n-1}{n-1}\}$.
- $\mathbf{G_n}$ is the subalgebra of $[\mathbf{0}, \mathbf{1}]_{*_G}$ with universe $\{0, \frac{1}{n-1}, ..., \frac{n-1}{n-1}\}$.

From now on, we abuse notation regarding ordinal sums, and generalize that name to a wider family of operations. Given $\{*_i\}_{i\in I}$ a set of operations in $\{*_Ł, *_G, *_\Pi\} \cup \{*_{Łn}, *_{Gn}\}_{n\in\mathbb{Z}, n>1}$, and $\{(b_i, t_i)\}_{i\in I}$ a family of pairwise disjoint open intervals of $[0, 1]$, we call ordinal sum of $\{\langle *_i, (b_i, t_i)\rangle\}_{i\in I}$ and is denoted by

$$* = \bigoplus_{i\in I} \langle *_i, (b_i, t_i)\rangle$$

to the operation defined as in 2.1 but with restricted universe $U \subseteq [0, 1]$ given by

$$U = \bigcup_{i\in I} \begin{cases} [b_i, t_i] & \text{if } *_i \in \{*_Ł, *_G, *_\Pi\} \\ b_i + (t_i - b_i)\cdot\{0, \frac{1}{n-1}, ..., \frac{n-1}{n-1}\} & \text{if } *_i \text{ is either } *_{Łn} \text{ or } *_{Gn} \end{cases}$$

where for $X \cup \{y\} \subseteq [0, 1]$, we write $y \cdot X$ to denote the set $\{y \cdot x\}_{x\in X}$.

This new notion allows to specify the well known n-valued Łukasiewicz and Gödel logics (which seem likely to be useful from an applied point of view), and also combinations of these with infinitely valued components.

On the other hand, thanks to several theoretical results concerning Hajek's BL, we can also work with this logic too. It is proven (see for instance [1, 26]) that BL is complete with respect to the standard algebra $\bigoplus_{i\in\mathbb{N}} [\mathbf{0}, \mathbf{1}]_{*_Ł}$.[6]

With just this, it could seem that BL does not fall in the cases detailed above since it is associated to a t-norm with infinitely many (Łukasiewicz) components. However, when dealing with a particular deduction (with a finite amount of variables), whether it is valid or not on BL coincides with the answer to the same question on an ordinal sum of finitely many components, as proven in the following result.

Lemma 3.1. *(cf. [2]) For a finite set of formulas* $\Gamma \cup \{\varphi\}$*,*

$$\Gamma \vdash_{\mathrm{BL}} \varphi \text{ if and only if } \Gamma \models_{(n+1)[\mathbf{0},\mathbf{1}]_{*_Ł}} \varphi$$

where n *is the number of different variables appearing in* $\Gamma \cup \{\varphi\}$ *and by* $(n+1)[\mathbf{0}, \mathbf{1}]_{*_Ł}$ *we denote the* BL*-algebra* $\bigoplus_{i\in\{n,\cdots,1\}} \langle *_Ł, (\frac{n-i}{n}, \frac{n-i+1}{n})\rangle$.[7]

After all the previous details, we can finally provide a formal definition of the logics that are supported by the reasoner.

Definition 3.2. A logic L is a ***Nice*** B***L Logic*** when one of the following cases holds:

1. L is equivalent to BL. That is to say, for any set of formulas $\Gamma \cup \{\varphi\}$

$$\Gamma \vdash_{\mathrm{L}} \varphi \text{ if and only if } \Gamma \vdash_{\mathrm{BL}} \varphi$$

[6] We write $\bigoplus_{i\in\mathbb{N}} [\mathbf{0}, \mathbf{1}]_{*_Ł}$ to denote the algebras isomorphic to $\bigoplus_{i\in\mathbb{N}} \langle *_Ł, (\frac{i}{i+1}, \frac{i+1}{i+2})\rangle$.

[7] Simply a standard algebra isomorphic to $n + 1$ copies of the standard Łukasiewicz algebra.

2. L coincides with the logic associated to the ordinal sum $*$ of $\{\langle *_i, (b_i, t_i)\rangle\}_{i\in I}$ for I finite, $\{*_i\}_{i\in I}$ a set of operations in $\{*_{\text{Ł}}, *_G, *_\Pi\} \cup \{*_{\text{Ł}n}, *_{Gn}\}_{n\in\mathbb{Z}, n>1}$ and $\{(b_i, t_i)\}_{i\in I}$ a family of pairwise disjoint open intervals of $[0, 1]$. That is to say, for $*$ ordinal sum as above, and for any set of formulas $\Gamma \cup \{\varphi\}$

$$\Gamma \vdash_{\text{L}} \varphi \text{ if and only if } \Gamma \models_{[\mathbf{0},\mathbf{1}]_*} \varphi$$

In this case we say that $*$ is a ***composed Nice*** BL ***t-norm***.

3.1.2. Efficiency issues: the product components

Studying previous works towards the development of a solver for fuzzy logics, we found out that the reasoners that had implemented the product logic case (see [5]) showed much worse results, in terms of reasoning time, than the other cases (Łukasiewicz and Gödel). This is a problem intrinsic to the operations of the associated algebra: while reasoning with linear operations (sums, subtractions and minimum/maximum operations) is fast in general and also in the particular case of the SMT-solvers, finding solutions for equation systems with multiplication and division operations is a much harder problem (since it makes the solver face non-linear equations) .

Cignoli and Torrens presented in [19] important studies concerning the product logic from an algebraic perspective. In particular, they proved a categorical equivalence between the standard product algebra and a fragment of the so-called Presburger arithmetic, which entirely omits multiplication. This fact was exploited in [35], and product logic was implemented over $\mathbb{Z}^-_\bullet$, the negative cone of the integers with bottom element with addition and (bounded) subtraction. However, the treatment of constant symbols different from $\{0, 1\}$ was out of question following this approach (since they cannot be moved from $[0, 1]$ to $\mathbb{Z}^-_\bullet$ consistently). Moreover, it was not proven how this treatment could be applied if the t-norm considered is not $*_\Pi$ but an ordinal sum containing one component of this form.

In the current approach, we gain inspiration in some of the ideas in [19], but we resort to an alternative codification of the product logic and also, as shown below, of the product components of any Nice BL logic. Motivated by the possibility of the addition of rational constants to the language, we rather considered to use the extension over the negative real numbers, $\mathbb{R}^-_\bullet$, instead of $\mathbb{Z}^-_\bullet$ itself. Formally,

$$\mathbb{R}^-_\bullet = \langle \{x \in \mathbb{R} : x \leq 0\} \cup \{-\infty\}, +, -', 0, -\infty\rangle$$

where

$$x + y = \begin{cases} x + y & \text{if } x, y \neq -\infty \\ -\infty & \text{otherwise} \end{cases}$$

$$x -' y = \begin{cases} 0 & \text{if } x \leq y \\ -\infty & \text{if } x > y \text{ and } y = -\infty \\ y - x & \text{otherwise} \end{cases}$$

$\mathbb{R}^-_\bullet$ is also a product algebra, and all its operations are linear, which is what we need in order to gain practical efficiency. We will see that using real arithmetic instead of integer arithmetic (as in [35]) does not increase the execution times, probably due to the treatment of Z3 of these theories. Moreover, it is easy to see that the standard product algebra is in fact isomorphic to $\mathbb{R}^-_\bullet$ and so the logics arising from them coincide.

Lemma 3.3. *For any pair $\langle a, b\rangle \in (0, 1) \times \{z \in \mathbb{R} : z < 0\}$, the function $\sigma_{\langle a,b\rangle} : [0, 1] \to \{x \in \mathbb{R} : x \leq 0\} \cup \{-\infty\}$ defined by*

$$\sigma_{\langle a,b\rangle}(x) = \begin{cases} -\infty & \text{if } x = 0 \\ b \cdot log_a x & \text{otherwise} \end{cases}$$

is an isomorphism between $[\mathbf{0}, \mathbf{1}]_\mathbf{\Pi}$ and $\mathbb{R}^-_\bullet$ (sending $\cdot$ to $+$ and $\to$ to $-$).

Proof. It is first clear that σ is order preserving: being $a \in (0, 1)$, the function log_a is monotonically decreasing (in $(0, 1]$) and being b a negative number, $b \cdot log_a$ is monotonically increasing. With the same basic calculations we know that it is also a bijection and its extension by mapping 0 to $-\infty$ results in a bijective mapping between $[0, 1]$ and $\{x \in \mathbb{R} : x \leq 0\} \cup \{-\infty\}$.

In order to prove it is a homomorphism, it is first clear that the top and bottom elements are properly mapped between the two algebras. For what concerns the operations, the results follow naturally using basic properties of the logarithm function.

Let $x, y \in [0, 1]$. If $x = 0$, then clearly $\sigma_{\langle a,b\rangle}(x \cdot y) = \sigma_{\langle a,b\rangle}(0) = -\infty = \sigma_{\langle a,b\rangle}(0) + \sigma_{\langle a,b\rangle}(x)$. On the other hand, if both $x, y > 0$, then $\sigma_{\langle a,b\rangle}(x \cdot y) = b \cdot log_a(x \cdot y)$. By the properties of the logarithm, this is equal to $b \cdot (log_a x + log_b y)$ and so to $\sigma_{\langle a,b\rangle}(x) + \sigma_{\langle a,b\rangle}(y)$.

For what concerns the $\to$ operation, consider $x, y \in [0, 1]$. If $x \leq y$, then $\sigma_{\langle a,b\rangle}(x \Rightarrow_\Pi y) = \sigma_{\langle a,b\rangle}(1) = 0$. On the other hand, since $\sigma_{\langle a,b\rangle}$ is order preserving, we know that $\sigma_{\langle a,b\rangle}(x) \leq \sigma_{\langle a,b\rangle}(y)$ and so $\sigma_{\langle a,b\rangle}(x) -' \sigma_{\langle a,b\rangle}(y) = 1$ too. If $x > y$, then by definition we know that $\sigma_{\langle a,b\rangle}(x \Rightarrow_\Pi y) = \sigma_{\langle a,b\rangle}(y/x)$. If $y = 0$ then clearly $\sigma_{\langle a,b\rangle}(y/x) = \sigma_{\langle a,b\rangle}(0) = -\infty = \sigma_{\langle a,b\rangle}(y) - \sigma_{\langle a,b\rangle}(x)$. Otherwise, $\sigma_{\langle a,b\rangle}(y/x) = b \cdot log_a(y/x)$. Again, by simple properties of the logarithm function, this is equal to $b \cdot (log_a y - log_a x)$ which is the definition of $\sigma_{\langle a,b\rangle}(x) -' \sigma_{\langle a,b\rangle}(y)$. □

With the previous result in mind, it becomes clear how to translate each product component from an ordinal sum to a component formed by $\langle \mathbb{R}^-_\bullet, i\rangle$, where i denotes the index of the original product component. In fact and due to the behavior of the product algebras, it is enough to translate only the interior of the product components (to the interior of copies of $\mathbb{R}^-_\bullet$), and thus we avoid the problem of having the same element addressed in two ways.[8]

Let $* = \bigoplus_{i\in I}\langle *_i, (b_i, t_i)\rangle$ for I a finite indexes set, $\{*_i\}_{i\in I}$ a set of operations in $\{*_{\text{Ł}}, *_G, *_\Pi\} \cup \{*_{\text{Ł}n}, *_{Gn}\}_{n\in\mathbb{Z}, n>1}$, and $\{(b_i, t_i)\}_{i\in I}$ a family of pairwise disjoint open intervals of $[0, 1]$. Our objective is to substitute the product fragments (i.e., (b_i, t_i) such that $*_i = *_\Pi$) by copies of the $\mathbb{R}^-_\bullet$ properly positioned. First of all,

[8] Observe that, naturally, the elements b_i of component of these type could either be referred to by $b_i \in [0, 1]$, as belonging to the component below, or by $\langle -\infty, i\rangle$ and the same happens for t_i, which could lead to problems in the computation of the problem.

the universe of the new conjunction operation shall be no longer $[0, 1]$, but is given by

$$S = ([0, 1] \setminus \bigcup_{i\in I:*_i=*_\Pi} \{(b_i, t_i)\}) \cup \bigcup_{i\in I:*_i=*_\Pi} \{\langle x, i\rangle : x \in \mathbb{R}, x < 0\}$$

The order relation in S is the natural one, understanding that the $\langle x, i\rangle$ elements (for a fixed i) are placed (strictly) in between b_i and t_i Formally the definition is a follows.

Definition 3.4. Let S be the universe defined above and $x, y \in U$. Then x is smaller or equal to y in S ($x \leq_S y$) whenever one of the following cases holds:

- $x, y \in [0, 1]$ and $x \leq y$
- $x = \langle z, i\rangle$, $y \in [0, 1]$ and $y \geq t_i$
- $x \in [0, 1]$, $y = \langle z, i\rangle$ and $x \leq b_i$
- $x = \langle z_1, i_1\rangle$, $y = \langle z_2, i_2\rangle$ with $i_1 \neq i_2$ and $t_{i_1} \leq b_{i_2}$
- $x = \langle z_1, i\rangle$, $y = \langle z_2, i\rangle$ and $z_1 \leq z_2$.

We can define a new strong conjunction operation $*'$ over the universe S as follows:

$$x *' y = \begin{cases} x * y & \text{if } x, y \in [0, 1] \\ min\{x, y\} & \text{if } \begin{cases} x \in [0, 1] \text{ and } y = \langle z, i\rangle & \text{or} \\ x = \langle z, i\rangle \text{ and } y \in [0, 1] & \text{or} \\ x = \langle z_1, i_1\rangle, y = \langle z_2, i_2\rangle \text{ and } i_1 \neq i_2 \end{cases} \\ \langle z_1 + z_2, i\rangle & \text{if } x = \langle z_1, i\rangle \text{ and } y = \langle z_2, i\rangle \end{cases}$$

Clearly, the corresponding residuated operation is given by $x \Rightarrow_{*'} y =$

$$\begin{cases} 1 & \text{if } x \leq_S y \\ x \to y & \text{if } x >_S y \text{ and } x, y \in [0, 1] \\ y & \text{if } x >_S y \text{ and } \begin{cases} x \in [0, 1] \text{ and } y = \langle z, i\rangle & \text{or} \\ x = \langle z, i\rangle \text{ and } y \in [0, 1] & \text{or} \\ x = \langle z_1, i_1\rangle, y = \langle z_2, i_2\rangle \text{ and } i_1 \neq i_2 \end{cases} \\ \langle z_2 - z_1, i\rangle & \text{if } x >_S y \text{ and } x = \langle z_1, i\rangle \text{ and } y = \langle z_2, i\rangle \end{cases}$$

We denote $\mathbf{S}_{*'}$ the BL-algebra defined from these two operations over S, i.e.,

$$\mathbf{S}_{*'} = \langle S, *', \to', 0, 1\rangle$$

Theorem 3.5. *Let $*$ be a composed Nice* BL *t-norm. Then $[\mathbf{0}, \mathbf{1}]_*$ and $\mathbf{S}_{*'}$ are isomorphic* BL*-algebras.*

Proof. It is easy to see that we can define an embedding from $[\mathbf{0}, \mathbf{1}]_*$ into $\mathbf{S}_{*'}$ that moreover is surjective, by adjusting the proof of Lemma 3.3. It is just necessary to take into account a normalization of the values of the product components before applying the σ mapping defined before (and the identity function as the mapping from the elements outside the products components).

For each $i \in I$ with $*_i = *_\Pi$, define the normalization function $\mathbf{n_i}\colon (b_i, t_i) \to (0, 1)$ by $\mathbf{n_i}(x) = \frac{x-b_i}{t_i-b_i}$. Note that for $x, y \in (b_i, t_i)$, it holds that $\mathbf{n_i}(x * y) = \mathbf{n_i}(x) \cdot \mathbf{n_i}(y)$.

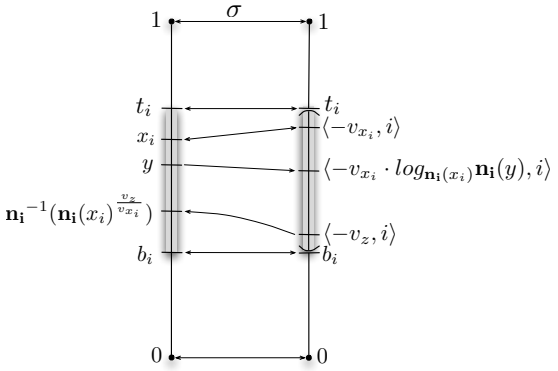

Figure 1: Diagram of the isomorphism σ_P over a product component

Given that in the case of Lemma 3.3 each pair of elements from $(0, 1)\times\{z \in \mathbb{R} : z < 0\}$ determines a different isomorphism, it is natural that now each set with one of this pairs for each product component determines different isomorphisms between the two algebras.

Let then P be a set of pairs of values from the product components at each side, i.e., for each $P = \{\langle x_i, \langle -v_{x_i}, i\rangle\rangle : i \in I$ with $*_i = *_\Pi, x_i \in (b_i, t_i), v_{x_i} \in \mathbb{R}, v_{x_i} > 0\}$.

We define the function $\sigma_P\colon [0, 1] \to S$ by

$$\sigma_P(x) = \begin{cases} \langle -v_{x_i} \cdot log_{\mathbf{n_i}(x_i)}\mathbf{n_i}(x), i\rangle & \text{if } x \in (b_i, t_i) \text{ for } i \in I \text{ s.t } *_i = *_\Pi \\ x & \text{otherwise} \end{cases}$$

Following the same reasoning that in the proof of Lemma 3.3, and taking in consideration that outside the product components we have the identity, it is easy to check that σ_P is injective and surjective because the identity function and the logarithm are so. As for proving that σ_P is a homomorphism, the methods coincide with those of Lemma 3.3, taking into consideration the possible types of pairs of elements depending on their position in $[0, 1]$ for what concerns the components. □

Observe that the inverse of σ_P is the identity function for all the elements outside a product component and for the rest, it is

$$\sigma_P^{-1}(\langle -v, i\rangle = \mathbf{n_i^{-1}}(\mathbf{n_i}(x_i^{\frac{v}{v_{x_i}}})).$$

Figure 1 represents the σ isomorphism in both directions, where (b_i, t_i) is a product component.

From the previous theorem, it is immediate that the logics arising from $[\mathbf{0}, \mathbf{1}]_*$ and those from $\mathbf{S}_{*'}$ coincide, which allows us to use the later algebra in order to equivalently compute the solutions of different questions on the logic.

3.1.3. Rational constants and Δ operator

Two different ways to expand at propositional level the previously presented logics are very interesting to point out: truth constants and the Monteiro-Baaz Δ projection operator. Both are natural extensions of the classical fuzzy language and have been widely studied from a theoretical point of view. On the other hand, the applicability, or we could even say necessity, of

this more expressive language from the point of view of applications is clear: it allows to address a particular variable of the system and fix its value to a previously known one (that is, use a constant), or being able to reason differently if a variable is equal to 1 or not (which is reachable using the Δ operator).

For what this work is concerned, it is clear how to work with the Δ operation, since it has a very determined semantic definition (over linearly ordered algebras, which is our case). By definition, in any linearly ordered BL-algebra

$$\Delta(x) = \begin{cases} 1 & \text{if } x = 1 \\ 0 & \text{otherwise} \end{cases}$$

In the case of Nice BL-logics, the two associated algebras we are concerned with (namely $[\mathbf{0},\mathbf{1}]_*$ and $\mathbf{S}_{*'}$) are linearly ordered and so the definition of the Δ operation coincides with the one above in both cases. Moreover, it is clear that the σ and σ' mappings used in the proof of Theorem 3.5 keep behaving as restricted embeddings when the Δ operation is also considered.

On the other hand, the inclusion of (rational) truth constants in the language was the main reason behind the use of the negative cone of the real numbers instead of that of the naturals when facing a t-norm containing some product component. The key fact in order to understand the treatment of constants in these components is that the function $f(x) = x^k$ is an endomorphism of the standard product algebra $[\mathbf{0},\mathbf{1}]_{\mathbf{\Pi}}$ for any $k \in \mathbb{R}$ with $k > 0$. This means that the deductions over $[\mathbf{0},\mathbf{1}]^{\mathbb{Q}}_{\mathbf{\Pi}}$ coincide with those over $[\mathbf{0},\mathbf{1}]_{\mathbf{\Pi}}$ with the constants "moved" in the interior of the component consistently among them (and left canonically interpreted in the non-product components). That is to say, for each product component, the interpretation of one arbitrary constant $\overline{c}$ can be set to any value (different from the top and bottom elements), and the other ones are evaluated depending on this value (and on the relation of their names).

In what follows, let $*$ be a composed Nice BL t-norm with $* = \bigoplus_{i\in I}\langle *_i, (b_i, t_i)\rangle$. Our objective is expanding with rational truth constants (and Δ, but this is immediate) the algebra $\mathbf{S}_{*'}$ in such a way that the logical deduction over this expansion coincides with that over $[\mathbf{0},\mathbf{1}]^{\mathbb{Q}}_*$. This means fixing an interpretation of the constants in this new algebra. For our purposes, we need not to address a unique rational expansion of $\mathbf{S}_*$ (that is to say, fix a unique interpretation of the rational constants). It suffices that, while the constants that fall out of any product component maintain their interpretations as rationals from $[0, 1]$, the other ones are interpreted consistently among them. Let $C_i = [0, 1]_{\mathbb{Q}} \cap (b_i, t_i)$ for $i \in I$,that is to say, the rationals from the component i.

Definition 3.6. We say that an algebra $\mathbf{A}$ is a ***rational expansion*** of $\mathbf{S}_{*'}$ when $\mathbf{A} = \langle S, *', \Rightarrow_{*'}, 0, 1, \{\overline{c}^{\mathbf{A}}\}_{c\in[0,1]_{\mathbb{Q}}}\rangle$ and the following holds:

- For $i \in I$ such that $*_i = *_\Pi$ there is $d_i \in C_i$ such that $\overline{d_i}^{\mathbf{A}} = \langle -v_i, i\rangle$ for some $v_i \in \mathbb{R}$, $v_i > 0$, and for all $c \in C_i$ it holds that $\overline{c}^{\mathbf{A}} = \langle -v_i \cdot log_{\mathbf{n}(d_i)}\mathbf{n}(c), i\rangle$ (where $\mathbf{n}$ stands for the normalization function defined in the proof of Theorem 3.5).
- For each $c \in [0, 1]_{\mathbb{Q}}$ such that $c \notin C_i$ for any $i \in I$ such that $*_i = *_\Pi$, $\overline{c}^{\mathbf{A}} = c$.

Now, the addition of constants slightly changes the formulation of Theorem 3.5 and its proof.

Lemma 3.7. $[\mathbf{0},\mathbf{1}]^{\mathbb{Q}}_*$ *is isomorphic to any rational expansion of* $\mathbf{S}_{*'}$.

Proof. The proof is analogue to that of Theorem 3.5 in almost all aspects. The only different point is that of not having an embedding for each pair of values of the product components, but now the set of pairs is limited to pairs of constants from each side. For simplicity on the calculus, we consider these constants to be the d_i outlined at the definition of free rational expansion of $\mathbf{S}_{*'}$. Let $\mathbf{A}$ be a free rational expansion of $\mathbf{S}_{*'}$ and $P = \{\langle d_i, \overline{d_i}^{\mathbf{A}}\rangle : i \in I \text{ with } *_i = *_\Pi, d_i \in (b_i, t_i) \cap [0, 1]_{\mathbb{Q}}\}$ where $\overline{d_i}^{\mathbf{A}} = \langle -v_i, i\rangle$ for some $v_i \in \mathbb{R}, v_i > 0$.

The rest of the proof coincides with the one of Theorem 3.5 and the only point that needs to be checked is that σ_P is a homomorphism for the constants too. First, it is clear that σ_P behaves as the identity for all the constants whose name is a rational value outside the product components, and so these are sent to their interpretation in $\mathbf{A}$. Let $c \in (b_i, t_i)$ where $*_i = *_\Pi$. By definition of σ we know that $\sigma(\overline{c}) = \sigma(c) = \langle -v_i \cdot log_{\mathbf{n}_\mathbf{i}(d_i)}\mathbf{n}_\mathbf{i}(c), i\rangle$. This coincides with the definition of $\overline{c}^{\mathbf{A}}$, which concludes the proof. □

As before, this implies that the logic arising from $[\mathbf{0},\mathbf{1}]^{\mathbf{Q}}_*$ coincides with that of any free rational expansion of $\mathbf{S}_{*'}$. It is natural to see that using the inverse of σ, we can translate a particular evaluation over any rational expansion of $\mathbf{S}_{*'}$ $\mathbf{A}$, to $[\mathbf{0},\mathbf{1}]^{\mathbb{Q}}_*$, and thus, if we generate an evaluation in $\mathbf{A}$, we can translate it to the more intuitive framework of $[\mathbf{0},\mathbf{1}]^{\mathbb{Q}}_*$.

3.2. Some modal expansions

As commented in the preliminaries section, fuzzy modal logics do not enjoy in general the finite model property, and it is then unclear whether the process to decide if a formula is valid or satisfiable in them is decidable. It is not in the scope of this paper to study this problem, even though in ongoing works we have proven some undecidabiliy results over product modal logics that will be presented in future publications. Since our main objective was producing an application useful from a practical point of view, we also wondered whether a real-world problem would truly require modal structures with an infinite number of worlds. For this, we restrict the solver to reason over finite Kripke structures, even though the logic defined from these may not fully coincide with the intended modal logic of a continuous t-norm.

Definition 3.8. Let L be a Nice BL logic with strong conjunction $*$. We denote by M_L the class of finite $[\mathbf{0},\mathbf{1}]_*$-Kripke models.

By relying on the results shown in the previous section, the following characterization is immediate:

Lemma 3.9. *Let* L *be a Nice* BL *logic with strong conjunction* $*$*, and* $\widehat{\mathrm{M}_L}$ *the local modal logic of the class of finite* **A***-Kripke models, where* **A** *is an arbitrary rational expansion of* $\mathbf{S}_{*'}$*. Then*

- *For any finite set of formulas* $\Gamma \cup \{\varphi\}$

$$\Gamma \Vdash_{\mathrm{M}_L} \varphi \text{ if and only if } \Gamma \Vdash_{\widehat{\mathrm{M}_L}} \varphi$$

- *A formula* φ *is locally satisfiable in* M_L *if and only if it is locally satisfiable in* $\widehat{\mathrm{M}_L}$*.*

This allows us to equivalently work with the second class of models, in a much more efficient way over Z3.

To treat the problem of generating a finite model (either to prove that a certain formula is not a theorem, or to prove that a set of equations are satisfiable), we exploit the notion of witness of a modal formula in a world. This is, the existence of a particular world among the successors of the of the original one where the effective value of the modal formula on the original world is taken by the non-modal version of the formula. This clearly exists, since working over a finite model, the definitions of the $\Box$ and $\Diamond$ operations now become *min* and *max* of a set of values.

Formally and in order to get a clear design of the application and low computing times, for a given formula (or equivalently, a finite set of formulas) we generate a Kripke frame (i.e., a restriction $\langle W, R\rangle$ of a Kripke model) with some additional information attached to it. This allows to quickly generate a model that evaluates a non-theorem to a value less than 1. (and also for a set $\Gamma \cup \{\varphi\}$ such that $\Gamma \not\models_{\mathrm{K}_{\mathrm{L}}^{\omega}} \varphi$), and makes that mNiBLoS does not impose a maximum cardinality on the models checked. The construction of this frame is similar to the approach followed by Hájek in [24] in the context of fuzzy description logics. For our purposes, we define it here explicitly and in a purely modal way, since the algorithm is later implemented within mNiBLoS. It is based on the decomposition of the formulas up to modal level, which are the elements that end up determining the structure and complexity of this general frame.

Definition 3.10. Let φ be a formula. The set of ***propositional subformulas*** of φ, PS(φ) is inductively defined by:

$$\begin{aligned}
\mathrm{PS}(p) &= \{p\} \text{ for } p \text{ prop. variable or constant}\\
\mathrm{PS}(\mathtt{M}\psi) &= \{\mathtt{M}\psi\} \text{ for } \mathtt{M} \in \{\Box, \Diamond\}\\
\mathrm{PS}(\psi\&\chi) &= \mathrm{PS}(\psi) \cup \mathrm{PS}(\chi)\\
\mathrm{PS}(\psi \to \chi) &= \mathrm{PS}(\psi) \cup \mathrm{PS}(\chi)\\
\mathrm{PS}(\Delta\psi) &= \mathrm{PS}(\psi)
\end{aligned}$$

Using the previously defined set and taking into account that any formula has a finite number of subformulas (in the usual sense of the word), it is possible to generate recursively the structure commented above. It consists on a tree where all the worlds except for the root one are pairs of the form $\langle n, \mathtt{M}\psi\rangle$ with $n \in \mathbb{N}$, $\mathtt{M} \in \{\Box, \Diamond\}$ and $\varphi \in Fm$. The $\mathtt{M}\psi$ modal formula associated to each world indicates that the value of $\mathtt{M}\psi$ at the father of the world (it is only one, since it is a tree) coincides with the value of ψ in the current world.

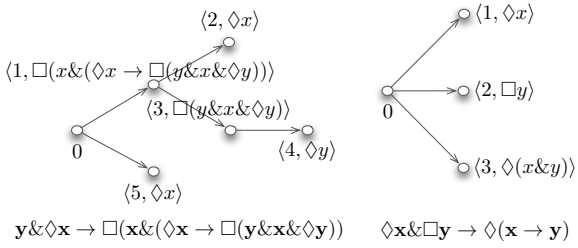

Figure 2: Examples of the Skeleton tree

The following algorithm shows how this structure can be constructed recursively, starting from a structure with only a root note $S = \{\langle 0, \emptyset\rangle\}$, the index of that world ($f = 0$), that indicates the father of successor worlds that will be added to the structure, and an empty set of accessibility relations $R = \emptyset$. We write mod(χ) to the denote the function that returns true if χ is of the form $\mathtt{M}\psi$ and false otherwise. Also, for a set of worlds as the above one, we denote by $last(S)$ the greatest index in the worlds from S.

```
Skeleton(Formula, S, R, f):
  NewS = S
  NewR = R
  MPS(Formula) = {g in PS(Formula) such that mod
      (g)}
  if MPS(Formula) is empty, return S, R

  otherwise do:
    for each g in MPS(Formula) do:
      newIndex = last(NewS)+1
      NewS = NewS U < newIndex, g >
      NewR = NewR U < f, newIndex >
      NewS, NewR = Skeleton(g, NewS, NewR,
          newIndex)
```

Figure 2 shows some examples of this construction.

The Skeleton algorithm generates a finite tree for a given formula φ, with maximum depth given by the maximum number of nested modalities in φ ($MMD(\varphi)$). Following the algorithm, it is easy to check that that the size of the skeleton tree (i.e., the number of worlds of the frame) is given by the amount of modal operators appearing in the formula. Indeed, a new world is only added for each formula beginning by $\Box$ or $\Diamond$. If several formulas are involved in the problem, the Skeleton frame for all of them is simply the one resulting from calling the algorithm with the conjunction of all the formulas involved (recall this is always a finite set, so this operation is well defined).

We can see that this skeleton is enough in order to "witness" the value of a formula in a particular world from an arbitrary model.

We say that $\langle W, R\rangle$ is a restriction of $\langle W', R'\rangle$ whenever $W = W'$ and for each $v, w \in W^2$ it holds that $R(v, w) \leq R'(v, w)$.

Lemma 3.11. *Let* $\mathfrak{M} = \langle W, R, e\rangle$ *be a finite* $*$*-kripke model,* $w_0 \in W$ *and* φ *a formula. Then, there is a model* $\mathfrak{U} = \langle W_U, R_U, e'\rangle$ *where* $\langle W_U, R_U\rangle$ *is a restriction of the* S*keleton*(φ) *frame and*

$$e(w_0, \varphi) = e'(0, \varphi)$$

(where 0 *is the root of the skeleton tree).*

Proof. Let us define the set of witnessing worlds of a modal formula in $\mathfrak{M}$ by

$$WS(\mathfrak{M}, w, \mathtt{M}\psi) = \begin{cases} \{v \in W : \ e(w, \Box\psi) = R(w, v) \Rightarrow_* e(v, \psi)\} & \text{if } \mathtt{M} \equiv \Box \\ \{v \in W : \ e(w, \Diamond\psi) = R(w, v) * e(v, \psi)\} & \text{if } \mathtt{M} \equiv \Diamond \end{cases}$$

Since W is finite, we know the model is witnessed, and thus, the previous set is always non-empty, and we can chose an arbitrary element from it and denote it by $wit(\mathfrak{M}, w, \mathtt{M}\psi)$.

We can consider a submodel $\mathfrak{M}'$of $\mathfrak{M}$ that behaves like $\mathfrak{M}$ in all the subformulas of φ. We endow each world (except for the initial one) with information related to whose is this a witnessing successor, and for which formula. Namely, let

$$\begin{array}{lcl} W_0 & = & \{\langle w_0, -1, \emptyset\rangle\} \\ W_1 & = & \{\langle wit(\mathfrak{M}, w_0, \mathtt{M}\psi), w_0, \mathtt{M}\psi\rangle : \mathtt{M}\psi \in PS(\varphi)\} \\ W_{i+2} & = & \{\langle wit(\mathfrak{M}, v, \mathtt{M}\psi), v, \mathtt{M}\psi\rangle : \langle v, \mathtt{M}_1\psi_1\rangle \in W_{i+1}, \mathtt{M}\psi \in PS(\psi_1)\} \end{array}$$

For some k it holds that $W_k = \emptyset$, since φ is a formula (and as such, it has finite modal depth). Let $W' = \bigcup_{1 \leq i} W_i$ and put

$$R'(\langle w_1, f_1, \delta_1\rangle, \langle w_2, f_2, \delta_2\rangle) = \begin{cases} R(w_1, w_2) & \text{if } f_2 = w_1 \\ 0 & \text{otherwise} \end{cases}$$

and

$$e'(\langle w, f, \delta\rangle, p) = e(w, p)$$

for each propositional variable p

First, observe that since $\mathfrak{M}'$ is a submodel of $\mathfrak{M}$ that always preserves some successor of each world, it holds that $e'(\langle v, f, \delta\rangle, \Box\psi) \geq e(v, \Box\psi)$ and $e'(\langle v, f, \delta\rangle, \Diamond\psi) \leq e(v, \Box\psi)$ for each $\langle v, f, \delta\rangle \in W'$ and any formula ψ. On the other hand, we can prove by induction on the formula complexity that $e'(\langle w, f, \Box\psi\rangle, \psi) = e(w, \psi)$.

It is obvious for ψ being a propositional variable or a formula whose main connective is non-modal. For the case of modal operators, let us consider the case $\Box$. By definition,

$$\begin{array}{rl} e'(\langle w, f, \Box\Box\psi\rangle, \Box\psi) & = \\ min\{R'(\langle w, f, \Box\Box\psi\rangle, \langle u, f_2, \chi\rangle) \Rightarrow_* e'(\langle u, f_2, \chi\rangle, \psi)\} & \end{array}$$

Since by construction $\langle wit(\mathfrak{M}, w, \Box\psi), w, \Box\psi\rangle \in W'$, we have

$$\begin{array}{rl} min\{R'(\langle w, f, \Box\Box\psi\rangle, \langle u, f_2, \chi\rangle) \Rightarrow_* e'(\langle u, f_2, \chi\rangle, \psi)\} & \leq \\ R'(\langle w, f, \Box\Box\psi\rangle, \langle wit(\mathfrak{M}, w, \Box\psi), w, \Box\psi\rangle) & \Rightarrow_* \\ e'(\langle wit(\mathfrak{M}, w, \Box\psi), w, \Box\psi\rangle, \psi) & \end{array}$$

By definition of R' and applying induction hypothesis this is equal to $R(w, wit(\mathfrak{M}, w, \Box\psi)) \Rightarrow_* e(wit(\mathfrak{M}, w, \Box\psi), \psi)$ which equals to $e(w, \Box\psi)$. Since the "$\geq$" direction follows from the fact that $\mathfrak{M}'$ is a submodel of $\mathfrak{M}$, this proves the case. The proof of the $\Diamond$ operator is analogous.

On the other hand, in a very similar fashion and using that $e'(\langle w, f, \Box\psi\rangle, \psi) = e(w, \psi)$, it is not hard to check that $e'(\langle w_0, -1, \emptyset\rangle, \varphi) = e(w_0, \varphi)$. The propositional cases are again immediate. For any formula $\Box\psi$ in the propositional subformulas of φ, observe that the world $\langle wit(\mathfrak{M}, w_0, \Box\psi), w_0, \Box\psi\rangle \in W_1 \subset W'$, so we have

$$\begin{array}{rl} e'(\langle w_0, -1, \emptyset\rangle, \Box\psi) & \leq \\ R'(\langle w_0, -1, \emptyset\rangle, \langle wit(\mathfrak{M}, w_0, \Box\psi), w_0, \Box\psi\rangle) & \Rightarrow_* \\ e'(\langle wit(\mathfrak{M}, w_0, \Box\psi), w_0, \Box\psi\rangle, \psi) & \end{array}$$

By R' and applying induction hypothesis this is equal to $R(w_0, wit(\mathfrak{M}, w_0, \Box\psi)) \Rightarrow_* e(wit(\mathfrak{M}, w_0, \Box\psi), \psi)$ and so, to $e(w_0, \Box\psi)$. Again the "$\geq$" direction holds since $\mathfrak{M}'$ is a submodel of $\mathfrak{M}$, so this concludes the step. The $\Diamond$ case can be proven analogously.

At this point it only lacks to check that $\mathfrak{M}'$ is in fact a restriction of the frame $Skeleton(\varphi)$. This can be easily done by mapping each world from $Skeleton(\varphi)$ into W' with a function σ as follows. For simplicity, for $\langle w, f, \delta\rangle \in W'$ we let $\mathtt{N}(\langle w, f, \delta\rangle) = w$.

$$\begin{array}{rcl} \sigma(0) & = & \langle w_0, -1, \emptyset\rangle \\ \sigma(\langle w, \mathtt{M}\psi\rangle) & = & \langle wit(\mathfrak{M}, \mathtt{N}(\sigma(father(\langle w, \mathtt{M}\psi\rangle))), \mathtt{M}\psi) \\ & & , \mathtt{N}(\sigma(father(\langle w, \mathtt{M}\psi\rangle))), \mathtt{M}\psi)\rangle \end{array}$$

Observe that, by construction of $Skeleton(\varphi)$ and of $\mathfrak{M}'$, the image under σ of any word belongs indeed to W'. Moreover, if $\langle w_1, \mathtt{M}_1\psi_1\rangle$ is not related with $\langle w_2, \mathtt{M}_2\psi_2\rangle$, in the definition of R' we have also imposed that $R'(\sigma(\langle w_1, \mathtt{M}_1\psi_1\rangle), \langle w_2, \mathtt{M}_2\psi_2\rangle) = 0$, so $\mathfrak{M}'$ is indeed a restriction of $Skeleton(\varphi)$.

□

With the previous lemma we have shown that the problem of determining whether a formula (or a derivation) is is true in all the finite **A**-Kripke models and the problem of local 1-satisfiability over finite **A**-Kripke models is decidable, for **A** being any rational expansion of $\mathbf{S}_{*'}$ (and in particular, also for **A** being an standard BL-algebra). Moreover, the computational complexity of these problems is of the same order than in the propositional case. The is to say (see [23] or [20]), coNP-complete for what concerns the theoremhood and derivation problems and NP-complete in the case of 1-satisfiability.

4. mNiBLoS

4.1. Design issues

The software application presented here, mNiBLoS, performs, over a certain Nice BL logic L with strong conjunction $*$, three possible operations over M_L (see 3.8) (the modal expansion of L):

1. checking whether a formula is a theorem of M_L,
2. checking whether a pair formed by a sequence of premises and a formula hold in the local consequence of M_L,
3. checking whether a set of equations is locally satisfiable in M_L (i.e., hold at some world of some finite **A**-Kripke model for **A** being some rational expansion of $\mathbf{S}_{*'}$) and providing such model if the answer is affirmative.

It is implemented in python, and an outline of the application is first generating a .smt2 file based on the inputs of the user, which will be the argument of the SMT-solver (in our case, Z3) and then, processing the output message. The main design decisions are based on the theoretical results explained in the previous section, and concern the structure of the .smt2 file generated. We can remark two main points where the design has

played an crucial role in the latter development of the application.

The first consideration is that we encode the (logical) variables from the formulas as arrays of length n of pairs, where n is the number of worlds generated in the Skeleton structure from in the previous section associated to the original set of formulas. First, we encode each "local" variable (meaning each variable at each world) as a pair in order to be able to easily work with the product components: in the first element of the pair encoding each variable we store the index of the (product) component in the ordinal sum, or a fixed index that indicates that the component is not a product one; the second element of the pair is the value taken by the variable in $[0, 1]$ (if the component is not a product one) or in $\mathbb{R}^-_\bullet$. Second, we consider each variable as an array of length n since this is a very natural way to encode the value of the whole variable at each world in the model. This can be avoided (for instance, if we prefer to use a SMT-solver that does not implement this theory) by simply letting each variable split in n new ones, but the use of arrays strongly enhances the readability of the code. On the other hand, since conceptually this two things are the same, it is clear that Lemma3.1 must now be adapted, and it is necessary to take into account that now the number of Łukasiewicz components necessary to equivalently compute a task over BL is changed. If we are dealing with a set of formulas with k variables and a Skeleton structure generated from these formulas with cardinality n, the number of Łukasiewicz components necessary is be $k \cdot n + 1$.

On the other hand, the use of $\mathbb{R}^-_\bullet$ instead of the usual codification for the product components is not completely correct for what concerns the truth constants behavior. This is due to the fact that the exponential function, keystone in the construction of the isomorphism from Theorem 3.5, is not implemented in general in the real arithmetic, since it is not known whether the resulting system is decidable (this is known as Tarski's exponential function problem). Thus, it is not possible to exactly calculate the values of the constants in $\mathbb{R}^-_\bullet$ in such a way that the bijection is maintained. Our partial solution for this problem has been forcing a slight modification of the values given by the user in the product component (except of one, c_0, used as reference), in such a way that the logarithm in base c_0 of these new values are a rational numbers. The modification in this case depends on the sensibility of the programming language: the value given by the user and the one used in the reasoner are indiscernible in the used language. In our case we have a sensibility of 16 decimal numbers. It is also important to remark that the reasoning over BL does not make any treatment on the constant symbols. Observe that we are working with an algebra that has the same behavior as BL, but that does not include constants appart from $\{0, 1\}$ since we could have chosen any other algebra isomorphic to $(n + 1)$Ł. For this reason, the constants, interpreted canonically, might not behave as the user expects.

4.2. Experimental results

In order to check the efficiency of mNiBLoS, we have run several tests[9] to obtain consistent timings of different logics and operations, and compared, when possible, the execution times with other two solvers from the literature that share some of the problems treated by mNiBLoS. In this latter sense, we have executed some tests to compare our solver with *fuzzyDL* (see [9, 11]) concerning 1-satisfiability over Łukasiewicz logic (which is the only common logic between mNiBLoS and *fuzzyDL*), and we have also repeated the tests shown in [5] concerning validity of formulas over Łukasiewicz Gödel and product logics.

In the first case, we have executed 1-satisfiability over randomly generated formulas of increasing length (understanding this as number of connectives and variables) and number of different variables, and it is remarkable that, even if mNiBloS has a quite good behavior (solving times for formulas of length around 2000 and 400 different variables are around 35 seconds), *fuzzyDL* is extremely better (giving for the same formulas solving times of at most 2 seconds). This is possibly due to several factors that strongly differentiate the two solvers: on the one hand, *fuzzyDL* implements several optimization techniques, while mNiBLoS relies on Z3 solver on this matter. This is clearly less efficient, since being much more specific, *fuzzyDL* can implement tools more oriented to the particular problem of 1-satisfiability over this particular logic. On the other hand, while mNiBLoS does not rely on reducing the universe of truth values to a finite one (among other reasons, because otherwise theoremhood cannot be computed), *fuzzyDL* is based on this semantical simplification, which most probably results on very good results. It is an interesting open problem to study which ones of the optimization methods shown in *fuzzyDL* can be adapted to our framework, and to observe if, concerning the 1-satisfiability problem, it is possible to reduce the universes of the Łukasiewicz and Gödel components to finite ones, and to check if this reduction is well behaved in terms of efficiency under Z3 or other SMT solvers. [10]

On the other hand, the main objective of mNiBLoS was not solving 1-satisfiability over Łukasiewicz logic (which are the main fuzzy solvers existing already in the literature, as we remarked in the introduction of this paper), but rather on offering a large family of t-norm based logics to work with, and also solving not only 1-satisfiability but also validity and logical consequence problems. For this, even if the previous comparative did not show a n improvement with respect to *fuzzyDL*, we believe the efficiency of mNiBLoS is not well represented under the previous tests, but by a larger family of proves we proceed to detail.

Aiming towards a comparative with the results in [5], our first test-bench for theoremhood proving is the set of axioms of BL with a generalization based on a natural parameter. Namely,

[9] We have run two kind of intensive tests over mNiBLoS, over a machine with a 3.1 GHz I5-2400 processor and 8GB of RAM.

[10] We have partially implemented the previous reduction, and observed that the behavior of Z3 under this integer-based semantics does not entail a real enhancement. This can be due to the methods internally used by Z3 to treat the integers and reals theories.

the following family of formulas is checked, variying $n \in \mathbb{N}\setminus\{0\}$

$$
\begin{array}{ll}
\text{(A1)} & (p^n \to q^n) \to ((q^n \to r^n) \to (p^n \to r^n)) \\
\text{(A2)} & (p^n \& q^n) \to p^n \\
\text{(A3)} & (p^n \& q^n) \to (q^n \& p^n) \\
\text{(A4)} & (p^n \& (p^n \to q^n)) \to (q^n \& (q^n \to p^n)) \\
\text{(A5}a\text{)} & (p^n \to (q^n \to r^n)) \to ((p^n \& q^n) \to r^n) \\
\text{(A5}b\text{)} & ((p^n \& q^n) \to r^n) \to (p^n \to (q^n \to r^n)) \\
\text{(A6)} & ((p^n \to q^n) \to r^n) \to (((q^n \to p^n) \to r^n) \to r^n)
\end{array}
\tag{1}
$$

Evaluating the validity in Łukasiewicz and Gödel logics of the generalizations of the log *BL* axioms (1), ranging n from 0 to 500 with increments of 10, throws in general slightly better results than the ones obtained in [5], but since the new solver coincides, on these logics, with the one proposed by them, this can be assumed to be due to the use of different machines.
For product logic, however, much better timings were obtained. The difference of using linear arithmetic instead of non-linear one is clear: complex formulas are solved in a comparatively short time, whereas in [5] they could not even be processed in most of the cases. On the other hand, there are no remarkable differences between the results on the product logic case presented here and the ones given in [35], implemented over lineal integer arithmetic. It is remarkable that there is a high level of irregularity when proving theoremhood of certain axioms, while others behave more regularly, as in the Łukasiewicz or Gödel cases. We conjecture this is due to the intrinsic shape of the formulas, but why does this only happen over the product logic is not known. We show in Figure 3 the cases of Axiom 3 and Axiom 4, that illustrate the two kind of response we have obtained.

Remarkably, the functions showed seem to follow a polynomial shape. This lead us to observe that the equations in (1) might not be a representative test bench for a BL-logics solver. In [5] the authors refer to [29] to justify why these formulas can be considered a good test bench for (at least) Łukasiewicz logic. In our opinion, these formulas have the problem of using only three variables. We consider this is a serious drawback because the known results on complexity of BL-logics state that Łukasiewicz validity is an coNP-complete problem when the number of variables in the input *is not fixed*. However, it seems likely that tautologicity for formulas with three variables can be solved in polynomial time.

With this in mind, to overcome the drawback of the bounded number of variables, we present an alternative family of BL-theorems to be used as a bench test.

For every $n \in \mathbb{N} \setminus \{0\}$,

$$
\bigwedge_{i=1}^{n} (\&_{j=1}^{n}\, p_{ij}) \;\to\; \bigvee_{j=1}^{n} (\&_{i=1}^{n}\, p_{ij}) \tag{2}
$$

is a BL-theorem which uses n^2 variables; the length of these formulas grows quadratically with n. As an example, we note that for $n = 2$ we get the BL-theorem $((p_{11}\&p_{12}) \wedge (p_{21}\&p_{22})) \to ((p_{11}\&p_{21}) \vee (p_{12}\&p_{22}))$. These formulas can be considered significantly harder than the ones from 1; indeed, the experimental results support this claim. It is important to notice that the natural way to compare this new formula with parameter n with

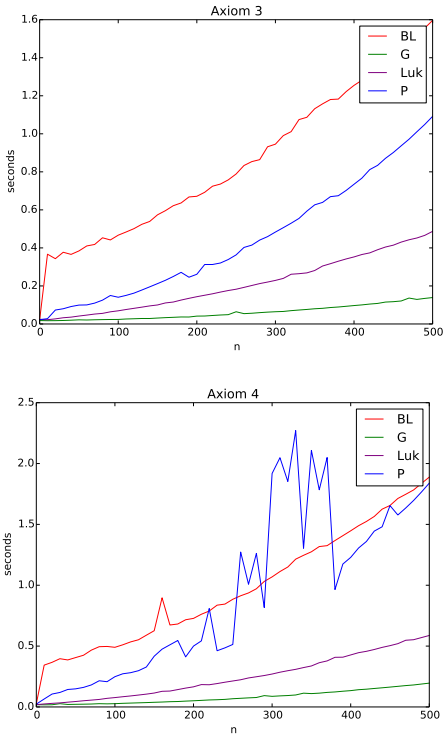

Figure 3: Generalizations of BL-axioms given in (1).

the previous set is to consider the formulas from [29] with the integer part of $\sqrt{n}$ as parameter.

The experiments done with (2) (see Figure 4 for the results) suggests that here the evaluation time is growing non-polynomially on the parameter n. In the graphs we give here, only those answers (for parameters $n \leq 70$) obtained in at most 3 hours of execution are shown (e.g. for the log *BL* case answers could be reached within this time only for the problems with $n \leq 4$). The high differences in time when evaluating the theorems were not surprising: Gödel and product logics answer quite fast thanks to the easy computation of the operations, Łukasiewicz logic is a bit harder, probably due to the fact that the connective operations require more effort, and BL is the more complex with a large difference since the method used for log *BL* (considering n^2+1 copies of Łukasiewicz, where n is the parameter of the formula) implies a harder internal reasoning.

To test the efficiency of a family of modal formulas of increasing modal complexity (in terms of the size and depth of the skeleton structure generated by them), we have run tests on validity over the family of formulas of the form:

$$
p_1 \& \Box(p_2 \& \Box(\ldots \Box(p_{n-1} \& \Box p_n)\ldots)) \tag{3}
$$

Figure 5 show the execution times obtained in the previous tests. It can be seen that execution times are quite high, even for an small parameter n, with an exponential behavior. It is likely that a recursive simplification of the formulas (as it is done, for instance, in *fuzzyDL*), and also optimization strategies oriented to cut down execution times in particularly simple examples as the above one might result in better results on this matter. How-

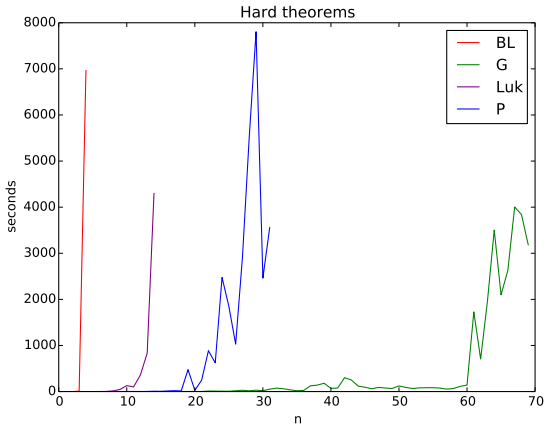

Figure 4: Our proposed BL-theorems given in (2).

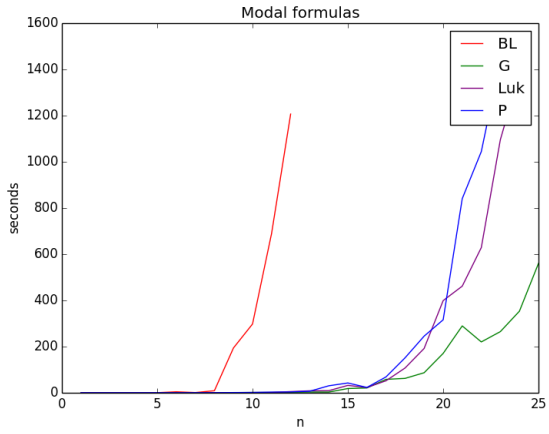

Figure 5: Theoremhood of the formulas from Equation 3

ever, if we compare this results with the ones from Figure 8, we can observe that including modalities does not necessarily imply such long times of execution, but rather the fact that they are all nested in the formulas from Equation 3.

With the aim of understanding the behavior of mNiBLoS when faced with a more irregular set of formulas, we designed and implemented a test that produces random formulas of varying length (understanding this as number of connectives and variables) and number of different variables. Using these, we have executed several kind of tests over the solver, in order to determine particular behaviors or patterns. In order to reduce irregularities due to marginal kinds of formulas we run each step of our tests (i.e., each coordinate ⟨length of the formula, number of variables⟩) with 10 different formulas (with the same parameters), and then obtain the average.

The following tests show, for different expansions of Łukasiewicz, Gödel, product and BL logics, the times of response of mNiBLoS trying to determine whether a formula with certain length and number of variables is a theorem of the logic or not.

In figure 6, we show the times considering the previous four logics alone, that is, without accepting constants in the language

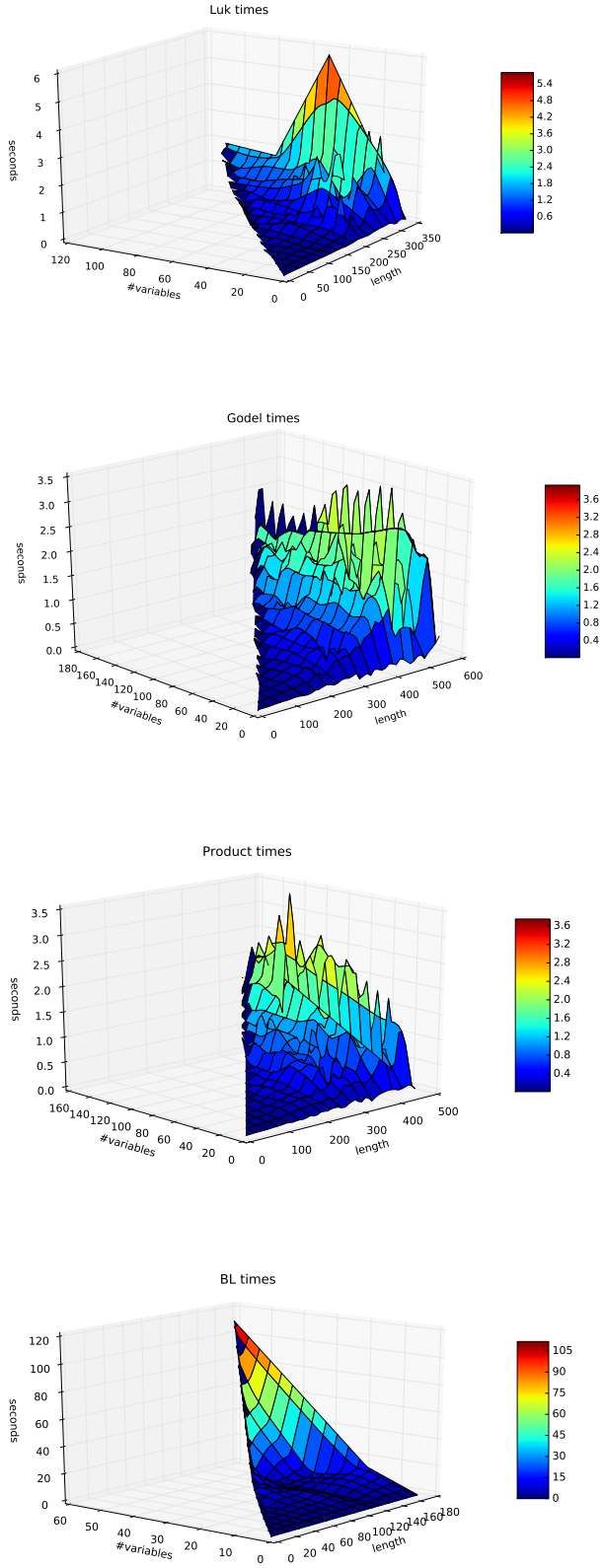

Figure 6: Random formulas, no constants nor modalities

and not using modalities.

Some characteristics of these graphs fit into our expect tions, but we can only guess concerning some other points the experiment. If we focus on the graph showing product log execution times, it seems clear that the alternative semanti of the product components plays a central role in the syste mNiBLoS treats product logic almost as fast as Gödel logic this randomized framework. We consider this to be a very inte esting result, since in the only previous works treating produ logic, this one was by far the slowest one, and its faster cor putation can open the door to more demanding applications r lying on the product t-norm. Moreover the relatively fast i crement in the reasoning times of BL was expected from t fact that the t-norm used internally in that case is much more complex than just one component of the basic ones (it directly depends on the number of variables).

The graphs obtained show a certain relation of the answering times and the relation between the length of the formula and the number of variables appearing in it. Similar in some sense to the face transition we can find in classical logic, we can see a peak on the execution times along the line where the relation length/# variables is between 6 and 8. This is a fact worth noticing, and in ongoing works this is being further developed in the particular case of the 1-satisfiability problem for generalized clausal forms over Łukasiewicz logic (see [12, 13]). In the other logics, this relation is not as remarkable, but nevertheless some interesting general behaviors can be seen. First, in the case of Gödel logic, the increasing in the number of variables over a fixed length of the formula does not add a very significant difference on the execution times. On the other hand, while the maximum values on the graph corresponding to the product logic are within the same range as the ones in Gödel, here the number of variables indeed modifies the execution times.

Finally, we guess that the BL case does not show a behavior similar to that of Łukasiewicz in this sense simply because the large increasing in the complexity due to the addition of variables (which make the logic much more complex, adding new Łukasiewicz components to the ordinal sum) makes unnoticeable any other variation.

Concerning the addition of constants to the language, the results are clearly fastened, see Figure 7. Using constants simplifies the calculus because, in a sense, they behave like variables that do not need to be tested by the solver. For this, the ratio length vs. number of variables of the formula gets much larger with the same execution times.

On the other hand, concerning the modal tests, the first result that we want to remark is that the time of preprocessing of the modal formulas (that is, the creation of the skeleton structure and of the code concerning the variables management) does not add any valuable time to the total reasoning time (<2ms in the worst cases). This happened also in the previous tests, but it is in the modal case where a more intensive preprocessing of the data is done .

The graphs in Figure 8 show some examples when adding modal operators to the formulas' language.

The processing times of formulas with modalities are much higher than in the previous cases. This is reasonable, since the

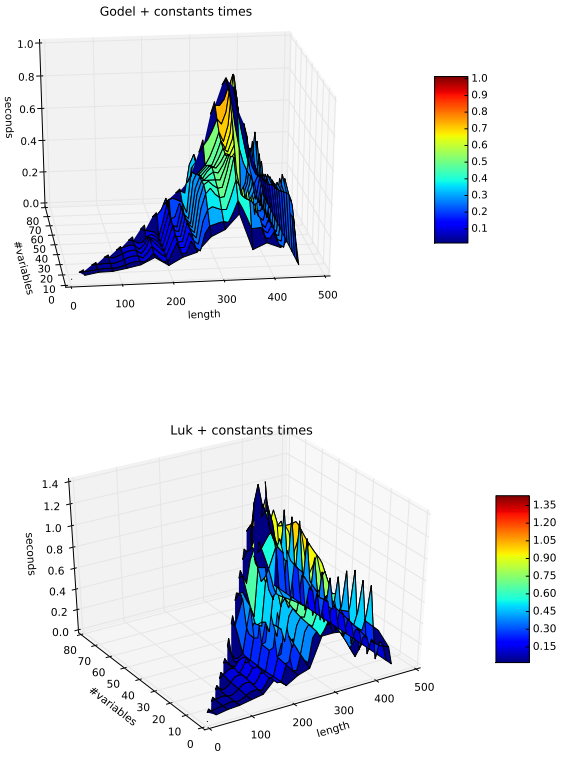

Figure 7: Random formulas, constants and no modalities

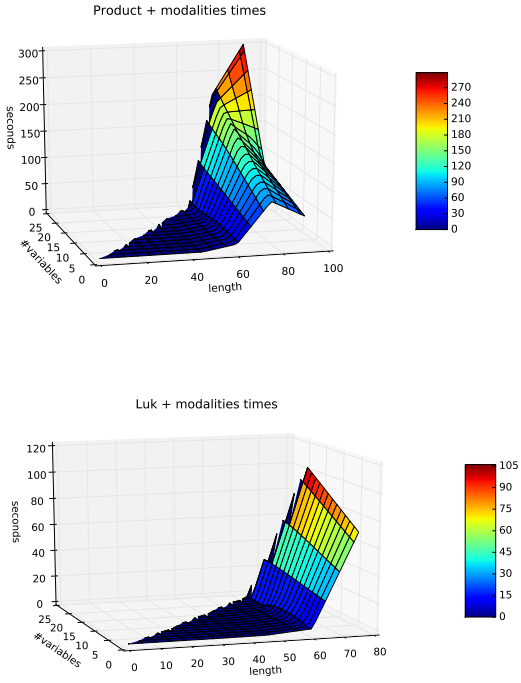

Figure 8: Random formulas, modalities and no constants

calculus are in some sense multiplied by the modal complexity of the formula (the skeleton it generates). Up to formulas of length 60, the solver is able to produce the answer in a quite short time (no more than 2 seconds in the worst cases), but when we consider more complex formulas, the reasoning times suffer an almost exponential increasing. We have observed, after repeatedly running the same tests, that the decreasing on the answering times on certain number of variables is not an isolated behavior (see the graph of product logic in Figure 8), but we do not know what can be the reason behind this. It is remarkable nevertheless that the solver works quite well up to formulas of length up to 80 elements and up to 25-30 variables, which we consider to be a good result for a first attempt on automated reasoning over infinitely-valued modal logics.

5. Conclusions and future work

We consider our objective of offering a tool that allows in a reasonably easy way to work with modal fuzzy logics has been reached with mNiBLoS. The theoretical results on these logics have allowed us to implement a quite efficient solver for a large family of fuzzy logics, and while the tasks originally implemented in mNiBLoS are the most basic ones, the modular codification allows to easily build add-ons if some other task is find of interest. With this in mind, the whole code of mNiBLoS can be freely downloaded from the web of the author.

The author is currently working on some problems related with the present paper, namely decidability problems for modal fuzzy logics, and topics related with the understanding and exploitation of generalized forms of conjunctive normal form formulas in a fuzzy setting. Several other interesting problems remain unsolved after this work, of which we remark the following ones.

- Improvement of the efficiency of mNiBLoS by developing and adapting optimizing algorithms used in FDL, studying which ones are applicable in an infinitely-valued setting without (in general) an involutive negation.
- Better understanding of fuzzy logics via intensive testings. For instance, we have not observed a relation, in the tests shown in Figure 6 and the following ones, between the answering time of mNiBLoS over a certain formula and whether it is or it is not a theorem of the logic. However, it is remarkable than theoremhood seemed to follow, in some cases, certain patterns concerning the length and number of variables of the formulas. Not only that: between our randomly generated formulas, it was much more common to get a non-theorem that a theorem, even though the cardinality of both sets is the same. All these are points worth noticing and an intensive study of these behaviors could offer a deeper understanding of logics based on continuous t-norms.
- Study of industrial applications that could benefit from using logics based on continuous t-norms. Interaction with a more applied community in order to determine other interesting features of a solver for fuzzy logics that might be of interest.
- Study of the complexity of the validity and 1-satisfiability problems over arbitrary modal expansion of continuous t-norm based logics (non limited to finite models). Implementation of some of these problems, relying in first order logics solvers and semi-decidable strategies when they are undecidable.

Acknowledgments This work is supported by the CSIC Intramural project 201450E045:"Logal: Lógica y algoritmos", the joint project of Austrian Science Fund (FWF) I1897-N25 and Czech Science Foundation (GACR) 15-34650L and the MINECO project EdeTRI:TIN2012-39348-C02-01. We want to thank the anonymous reviewers for their useful commentaries.

Bibliography

[1] P. Agliano and F. Montagna. Varieties of BL-algebras. I. General properties. *Journal of Pure and Applied Algebra*, 181(2-3):105–129, 2003.

[2] S. Aguzzoli and B. Gerla. On countermodels in Basic Logic. *Neural Network World*, 12(5):407–421, 2002.

[3] S. Aguzzoli, B. Gerla, and Z. Haniková. Complexity issues in basic logic. *Soft Computing*, 9(12):919–934, 2005.

[4] T. Alsinet, D. Barroso, R. Béjar, F. Bou, M. Cerami, and F. Esteva. On the implementation of a fuzzy dl solver over infinite-valued product logic with smt solvers. In W. Liu et al. editors, *Scalable Uncertainty Management*, volume 8078 of *Lecture Notes in Computer Science*, 325–330. Springer Berlin Heidelberg, 2013.

[5] C. Ansótegui, M. Bofill, F. Manyà, and M. Villaret. Building automated theorem provers for infinitely valued logics with satisfiability modulo theory solvers. In *Proceedings of the IEEE 42nd International Symposium on Multiple-Valued Logic (ISMVL 2012)*. p15–30, IEEE CS Press.

[6] C. Barrett, A. Stump, and C. Tinelli. The SMT-LIB standard: Version 2.0. Technical report, Department of Computer Science, The University of Iowa, 2010. Available at `www.SMT-LIB.org`.

[7] P. Blackburn, M. de Rijke and Y. Venema. Modal Logic *Cambridge University Press*, Number 53 of *Cambridge Tracts in Theoretical Computer Science* series. 2001

[8] F. Bobillo, M. Cerami, F. Esteva, A. García-Cerdaña, R. Peñaloza, and U. Straccia Fuzzy Description Logics in the framework of Mathematical Fuzzy Logic In P. Cintula et. al eds. Ch. 16 of *Handbook of Mathematical Fuzzy Logic, Volume 3*, volume 58 of *Studies in Logic, Mathematical Logic and Foundations*, p1105–1181, 2015.

[9] F. Bobillo and U. Straccia. *fuzzyDL*: An expressive fuzzy description logic reasoner In proceedings of the *IEEE World Congress on Computational Intelligence*, p923–930. 2008

[10] F. Bobillo and U. Straccia. Fuzzy description logics with general t-norms and datatypes. *Fuzzy Sets and Systems*, 160(23):p3382–3402, 2009.

[11] F. Bobillo and U. Straccia. The Fuzzy Ontology Reasoner *fuzzyDL Knowledge-Based Systems*, volume 95:p12–34, 2016

[12] M. Bofill, F. Manyà, A. Vidal, and M. Villaret. The complexity of 3-valued łukasiewicz rules. In Proceedings *Modeling Decisions for Artificial Intelligence - 12th International Conference, MDAI 2015*, p.221–229.

[13] M. Bofill, F. Manyà, A. Vidal, and M. Villaret. Finding hard instances of satisfiability in lukasiewicz logics. In *Proceedings of the IEEE 45nd International Symposium on Multiple-Valued Logic (ISMVL 2015)*. p30–35, IEEE CS Press.

[14] F. Bou, F. Esteva, L. Godo, and R. Rodríguez. On the minimum many-valued modal logic over a finite residuated lattice. *Journal of Logic and Computation*, 21(5):739–790, 2011.

[15] X. Caicedo, G. Metcalfe, R. Rodríguez, and J. Rogger. A finite model property for gödel modal logics. In L. Libkin et. al. editors, *Logic, Language, Information, and Computation*, volume 8071 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2013.

[16] X. Caicedo and R. O. Rodriguez. Bi-modal Gödel logic over [0, 1]-valued kripke frames. *Journal of Logic and Computation*, 25(1):37–55, 2015.

[17] X. Caicedo and R. Oscar Rodríguez. Standard Gödel modal logics. *Studia Logica*, 94(2):189–214, 2010.

[18] A. Chagrov and M. Zakharyaschev. Modal Logic. *Oxford University Press*, volume 35 of *Oxford Logic Guides* series. 1997.

[19] R. Cignoli and A. Torrens. An algebraic analysis of product logic. *Multiple-valued logic*, 5:45–65, 2000.

[20] P. Cintula, P. Hájek, and C. Noguera, editors. *Handbook of Mathematical Fuzzy Logic, 2 volumes*, volume 37 and 38 of *Studies in Logic. Mathematical Logic and Foundation*. College Publications, 2011.

[21] L. Mendonça de Moura and N. Bjørner. Z3: An efficient SMT solver. In C. R. Ramakrishnan et al. editors, Proceedings of *Tools and Algorithms for the Construction and Analysis of Systems, 14th International Conference, TACAS 2008, Held as Part of the ETAPS 2008*, p.337–340, 2008.

[22] R. Hähnle. *Automated deduction in multiple valued logics*, volume 10 of *International Series of Monographs on Computer Science*. The Clarendon Press Oxford University Press, New York, 1993. Oxford Science Publications.

[23] P. Hájek. *Metamathematics of fuzzy logic*, volume 4 of *Trends in Logic—Studia Logica Library*. Kluwer Academic Publishers, Dordrecht, 1998.

[24] P. Hájek. Making fuzzy description logic more general. *Fuzzy Sets and Systems*, 154(1):1–15, 2005.

[25] G. Hansoul and B. Teheux. Extending łukasiewicz logics with a modality: Algebraic approach to relational semantics. *Studia Logica*, 101(3):505–545, 2013.

[26] F. Montagna. Generating the variety of BL-algebras. *Soft Computing*, 9(12):869–874, 2005.

[27] P. S. Mostert and A. L. Shields. On the structure of semigroups on a compact manifold with boundary. *Annals of Mathematics. Second Series*, 65:117–143, 1957.

[28] D. Mundici. Satisfiability in many-valued sentential logic is NP-complete. *Theoretical Computer Science*, 52(1-2):145–153, 1987.

[29] R. Rothenberg. A class of theorems in Łukasiewicz logic for benchmarking automated theorem provers. In N. Olivetti et. al. editors, *TABLEAUX '07, Automated Reasoning with Analytic Tableaux and Related Methods, Position Papers*, LSIS.RR.2007.002, p.101–111, 2007.

[30] S. Schockaert, J. Janssen, and D. Vermeir. Satisfiability checking in Łukasiewicz logic as finite constraint satisfaction. *Journal of Automated Reasoning*, 2012. To appear.

[31] S. Schockaert, J. Janssen, D. Vermeir, and M. De Cock. Finite satisfiability in infinite-valued Łukasiewicz logic. In L. Godo et.al. eds., *Scalable Uncertainty Management (SUM 2009)*, volume 5785 of *Lecture Notes in Computer Science*, p.240–254. Springer, 2009.

[32] R. Sebastiani. Lazy satisability modulo theories. *Journal on Satisfiability, Boolean Modeling and Computation*, 3(3-4):141–224, 2007.

[33] U. Straccia, Foundations of Fuzzy Logic and Semantic Web Languages. In Chapman & Hall *CRC Studies in Informatics Series*, 2013

[34] U. Straccia All About Fuzzy Description Logics and Applications. In *Reasoning Web, 11th International Summer School, Tutorial Lectures.* volume 9203 of *Lecture Notes in Computer Science*, p.1–31. Springer, 2015

[35] A. Vidal, F. Bou, and L. Godo. An smt-based solver for continuous t-norm based logics. In E. Hüllermeier et. al. eds., *Scalable Uncertainty Management (SUM2012)*, volume 7520 of *Lecture Notes in Computer Science*, p.633–640. Springer Berlin Heidelberg, 2012.

[36] A. Vidal, F. Esteva, and L. Godo. On modal extensions of product fuzzy logic. *Journal of Logic and Computation*, (In press, 2015).

[37] Z3 solver https://z3.codeplex.com/