Journal of Automated Reasoning
https://doi.org/10.1007/s10817-020-09558-9

# Probably Partially True: Satisfiability for Łukasiewicz Infinitely-Valued Probabilistic Logic and Related Topics

**Marcelo Finger**[1] · **Sandro Preto**[1]

Received: 20 April 2020 / Accepted: 24 April 2020


**Abstract**
We study probabilistic-logic reasoning in a context that allows for "partial truths", focusing on computational and algorithmic properties of non-classical Łukasiewicz Infinitely-valued Probabilistic Logic. In particular, we study the satisfiability of joint probabilistic assignments, which we call ŁIPSAT. Although the search space is initially infinite, we provide linear algebraic methods that guarantee polynomial size witnesses, placing ŁIPSAT complexity in the NP-complete class. An exact satisfiability decision algorithm is presented which employs, as a subroutine, the decision problem for Łukasiewicz Infinitely-valued (non probabilistic) logic, that is also an NP-complete problem. We investigate efficient representation of rational McNaughton functions in Łukasiewicz Infinitely-valued Logic modulo satisfiability.

**Keywords** Fuzzy logics · Probabilistic fuzzy logics · Multivalued logics · Probabilistic multivalued logics · Łukasiewicz Infinitely-valued Logic

## 1 Introduction

This paper deals with the problem of determining the consistency of probabilistic assertions allowing for "partial truths"[1] considerations. This means that we depart from the classical probabilistic setting and instead employ a many-valued underlying logic. In this way we enlarge our capacity to model situations in which a gradation of truth may be closer to the perceptions of agents involved. We employ Łukasiewicz Infinitely-valued logic as it is one of the best studied many-valued logics, having interesting properties which lead to amenable computational treatment. Notably, it has been shown that foundational properties

[1] By the term "partial truth" we refer to the concept usually referred in the literature as "degree of truth", not to be confused with partial valuations or models.

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior—Brasil (CAPES)— Finance Code 001; M. Finger—Partially supported by Fapesp Projects 2015/21880-4 and 2014/12236-1 and CNPq Grant PQ 303609/2018-4.

✉ Marcelo Finger
mfinger@ime.usp.br

[1] Department of Computer Science, University of São Paulo, São Paulo, Brazil

Published online: 06 June 2020

of probabilistic theory such as de Finetti coherence criteria also applies to Łukasiewicz Infinitely-valued probabilistic theories [25].

We provide theoretical presentation leading to algorithms that decide the satisfiability of probabilistic assertions in which the underlying logic is Łukasiewicz logic with infinite truth values in the interval [0, 1]. For that, we employ techniques from linear programming and many-valued logics. In the latter case we need to solve several instances of the satisfiability problem in Łukasiewicz Infinitely-valued logic.[2] We show that the evaluation of formulas modulo satisfiability as used in the algorithm presented increases the expressivity of $Ł_\infty$, allowing it to represent rational McNaughton functions.

To understand the kind of situation in which our techniques can be applicable consider the following example.

***Example 1*** Three friends have the habit of going to a bar to watch their soccer team's matches. Staff at the bar claims that at every such match at least two of the friends come to the premises, but if you ask them, they will say that each of them comes to watch at most 60% of the games.

In classical terms, the claims of the staff and of the three friends are in contradiction. In fact, if there are always two of the three friends present at matches, someone must attend to least two-thirds of the team's matches.

However, one may allow someone to arrive for the second half of the match, and consider his attendance only "partially true", say, a truth value of 0.5 in that case. Then it may well be the case that staff and customers are both telling the truth, that is, their claims are jointly satisfiable. □

It turns out that the example above is unsatisfiable in classical probabilistic logic, but it is satisfiable in Łukasiewicz Infinitely-valued Probabilistic logic. In this work we are going to formalize such problems and present techniques and algorithms to solve them.

### 1.1 Classical and Non-classical Probabilistic Logic

Classical probabilistic logic combines classical propositional inference with classical (discrete) probability theory. The original formulation of such a mix of logic and probability is due to George Boole who, in his seminal work introducing what is now known as Boolean Algebras, already discussed the problem [4]. Among the foundational works on classical probabilistic theory we highlight that provided by de Finetti's notion of coherent probabilities [9,10].

The decision problem over classical probabilistic logic is called Probabilistic Satisfiability (PSAT). PSAT has been extensively discussed in the literature [16,20,27], and has recently received a lot of attention due to the improvements in SAT solving and linear programming techniques, having generated a variety of algorithms, for which the empirical phenomenon of phase-transition is by now established [13,14].

Łukasiewicz Infinitely-valued Logic is widely used in the literature to model situations that require the notion of "partial truth", seen as a many-valued logic and algebra [7]. A probability theory over such a many-valued context, including a notion of coherent probabilities in line with de Finetti's original work, was developed as a sound basis for non-classical probability theory [25]. The problem of deciding whether a set of probabilistic assignments over Łukasiewicz Infinitely-valued Logic is coherent was shown to be NP-complete by [6].

[2] Satisfiability problem in Łukasiewicz Infinitely-valued logic has been shown to be NP-complete [23] and there are some implementations discussed in the literature [3], but there are many implementation options with considerable efficiency differences which are analyzed in [15].

It is the goal of this paper to explore equivalent formulations and algorithmic ways to solve this problem and also to explore some representational implications that follow from those techniques.[3]

The rest of this paper is organized as follows. In Sect. 2 we describe the notions pertaining Łukasiewicz Infinitely-valued Logic and Łukasiewicz Infinitely-valued Probabilistic Logic and the notion of coherent probability over such logic. In Sect. 3 we study the theoretical relationship between linear algebraic methods and the solution of the ŁIPSAT problem. In Sect. 4 we develop a column generation algorithm for ŁIPSAT solving and show its correctness. Finally, in Sect. 5 we study how the expressivity of $Ł_\infty$ can be increased by setting a semantics modulo satisfiability (MODSAT) based on the evaluation used in the column generation algorithm.

Source code of the solvers developed are publicly available under license GPLv3 at http://lipsat.sourceforge.net.

## 2 Preliminaries

Łukasiewicz Infinitely-valued Logic ($Ł_\infty$) is arguably one of the best studied many-valued logics [7]. It has several interesting properties, such as a truth-functional semantics that is continuous, having classical logic as a limit case and possessing well developed proof-theoretical and algebraic presentations. The semantics of $Ł_\infty$-formulas represent all piecewise linear functions with integer coefficients—i.e. the McNaughton functions—and only those [22,24].

The basic $Ł_\infty$-language is built from a countable set of propositional symbols $\mathbb{P}$, and disjunction ($\oplus$) and negation ($\neg$) operators. For the semantics, define a $Ł_\infty$-valuation $v : \mathbb{P} \rightarrow [0, 1]$, which maps propositional symbols to a value in the rational interval [0, 1]. Then $v$ is extended to all $Ł_\infty$-formulas as follows

$$v(\varphi \oplus \psi) = \min(1, v(\varphi) + v(\psi))$$
$$v(\neg\varphi) = 1 - v(\varphi)$$

From those operations one usually derives the following:

| | |
|---|---|
| Conjunction: $\varphi \odot \psi =_{\text{def}} \neg(\neg\varphi \oplus \neg\psi)$ | $v(\varphi \odot \psi) = \max(0, v(\varphi) + v(\psi) - 1)$ |
| Implication: $\varphi \rightarrow \psi =_{\text{def}} \neg\varphi \oplus \psi$ | $v(\varphi \rightarrow \psi) = \min(1, 1 - v(\varphi) + v(\psi))$ |
| Maximum: $\varphi \vee \psi =_{\text{def}} \neg(\neg\varphi \oplus \psi) \oplus \psi$ | $v(\varphi \vee \psi) = \max(v(\varphi), v(\psi))$ |
| Minimum: $\varphi \wedge \psi =_{\text{def}} \neg(\neg\varphi \vee \neg\psi)$ | $v(\varphi \wedge \psi) = \min(v(\varphi), v(\psi))$ |
| Bi-implication: $\varphi \leftrightarrow \psi =_{\text{def}} (\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi)$ | $v(\varphi \leftrightarrow \psi) = 1 - \lvert v(\varphi) - v(\psi)\rvert$ |

A formula $\varphi$ is $Ł_\infty$-*valid* if $v(\varphi) = 1$ for every valuation $v$. A formula $\varphi$ is $Ł_\infty$-*satisfiable* if there exists a $v$ such that $v(\varphi) = 1$; otherwise it is $Ł_\infty$-*unsatisfiable*. A set of formulas $\Phi$ is satisfiable if there exists a $v$ such that $v(\varphi) = 1$ for all $\varphi \in \Phi$. Note that $v(\varphi \rightarrow \psi) = 1$ iff $v(\varphi) \leq v(\psi)$; similarly, $v(\varphi \leftrightarrow \psi) = 1$ iff $v(\varphi) = v(\psi)$.

$Ł_\infty$ also serves as a basis for a well-founded non-classical probability theory [26]. Define a *convex combination* over a finite set of valuations $v_1, \dots, v_m$ as a function on formulas into [0, 1] such that

[3] An earlier version of this paper has appeared in [15]. In this work we present proofs of lemmas and theorems that were omitted. Section 5 on representation of rational McNaughton functions is totally new; on the other hand, due to space limitations, implementational and experimental issues have been omitted.

$$C(\varphi) = \lambda_1 v_1(\varphi) + \cdots + \lambda_m v_m(\varphi) \tag{1}$$

where $\lambda_i \geq 0$ and $\sum_{i=1}^{m} \lambda_i = 1$. So a $Ł_\infty$-probability distribution $\lambda = [\lambda_1, \ldots, \lambda_m]$ is a set of coefficients that form the convex combination of $Ł_\infty$-valuations. To distinguish $Ł_\infty$-probabilities from classical ones, we use the notation $C(\cdot)$, following [26]; it is important to note that $C$ is defined over *any* ***finite*** set of valuations.[4] Note that classical discrete probabilities are also convex combinations of $\{0, 1\}$-valuations.

This notion of probability associates non-zero values only to a finite number of $Ł_\infty$-valuations; thus the notion of $Ł_\infty$-probability is intrinsically discrete. As there are infinitely many possible $Ł_\infty$-valuations, the remaining ones are assumed to be zero. In this work we are interested in deciding the existence of convex combinations of the form (1) given a set of constraints. So, in theory, the search space is infinite.

It follows immediately from this definition that $C(\alpha) = 1$ if there is a convex combination over $v_1, \ldots, v_m$ where $v_i(\alpha) = 1$, $1 \leq i \leq m$.

**Lemma 1** *$C(\alpha \to \beta) = 1$ iff $C(\alpha) \leq C(\beta)$.* □

Lemma 1 is a direct consequence from the fact that $v(\varphi \to \psi) = 1$ iff $v(\varphi) \leq v(\psi)$.

We define a Łukasiewicz Infinitely-valued Probabilistic (ŁIP) assignment as an expression of the form

$$\Sigma = \left\{ C(\alpha_i) = q_i \mid q_i \in [0, 1], 1 \leq i \leq k \right\}.$$

As a foundational view of probabilities, it is possible to define a coherence criterion over ŁIP-assignments, in analogy to the de Finetti classical notion of coherent assignment of probabilities [8,10]. Thus, define the $Ł_\infty$-coherence of a ŁIP-assignment $\{C(\alpha_i) = q_i \mid 1 \leq i \leq k\}$ in terms of a bet between two players, Alice the bookmaker and Bob the bettor. The outcome on which the players bet is a $Ł_\infty$-valuation describing an actual "possible world". For each formula $\alpha_i$, Alice states her betting odd $C(\alpha_i) = q_i \in [0, 1]$ and Bob chooses a "stake" $\sigma_i \in \mathbb{Q}$; Bob pays Alice $\sum_{i=1}^{k} \sigma_i \cdot C(\alpha_i)$ with the promise that Alice will pay back $\sum_{i=1}^{k} \sigma_i \cdot v(\alpha_i)$ if the outcome is the possible world (or valuation) $v$. As in the classical case, the chosen stake $\sigma_i$ is allowed to be negative, in which case Alice pays Bob $|\sigma_i| \cdot C(\alpha_i)$ and gets back $|\sigma_i| \cdot v(\alpha_i)$ if the world turns out to be $v$. Alice's total balance in the bet is

$$\sum_{i=1}^{k} \sigma_i (C(\alpha_i) - v(\alpha_i)).$$

We say that there is a *ŁIP-Dutch Book* against Alice's ŁIP-assignment if there is a choice of stakes $\sigma_i$ such that, for every possible outcome $v$, Alice's total balance is always negative, indicating a bad choice of betting odds made by Alice.

**Definition 1** Given a probability assignment to propositional formulas $\{C(\alpha_i) = q_i \mid 1 \leq i \leq k\}$, the ŁIP-assignment is *coherent* if there are no Dutch Books against it.

While the coherence of an assignment provides a foundational view to deal with $Ł_\infty$-probabilities, a more computational view is possible, based on the satisfiability of assignments. Such a view will allow a more operational way of dealing with $Ł_\infty$-probabilistic assignments.

[4] Thus $C$ is more restrictive than the full class of states of an MV-algebra, in the sense of [26], which will not be discussed here.

**Definition 2** A ŁIP-assignment is *satisfiable* if there exists a convex combination $C$ and a set of valuations that jointly verifies all restrictions in it.

***Example 2*** Consider again Example 1, let $x_1, x_2, x_3$ be variables representing the presence at the bar of each of the three friends. An $Ł_\infty$-valuation assigns to each variable a value in [0, 1]. The probabilistic constraint expressing that each friend comes at most 60% of the games can be expressed as

$$C(x_1) = C(x_2) = C(x_3) \leq 0.6, \tag{$*$}$$

and the fact that at least two of them are present is expressed by the constraints

$$C(x_1 \oplus x_2) = C(x_1 \oplus x_3) = C(x_2 \oplus x_3) = 1 \tag{$**$}$$

which means that no two of them are simultaneously absent. There are infinitely many ways of obtaining a convex combination of $Ł_\infty$-valuations that satisfy all six conditions, the simplest of which is achieved with a single $Ł_\infty$-valuation $v$, $v(x_1) = v(x_2) = v(x_3) = 0.6$; in fact, $v(x_1 \oplus x_2) = v(x_1 \oplus x_3) = v(x_2 \oplus x_3) = \min(1, 0.6 + 0.6) = 1$, so we can attribute 100% of probability mass to $v$.

A similar result can be obtained with three "classical" valuations $v_i(x_i) = 0$, $v_i(x_j) = v_i(x_k) = 1$, for pair-wise distinct $i, j, k \in \{1, 2, 3\}$ and a fourth valuation $v_4(x_1) = v_4(x_2) = v_4(x_3) = 0.5$. Note all four valuations satisfy the formulas in ($**$). The convex valuation assigns probability 0.2 to $v_1, v_2, v_3$ and 0.4 to $v_4$, satisfying all constraints ($*$) and ($**$). □

The following result is the characterization of coherence for Łukasiewicz Infinitely-valued Probabilistic Logic.

**Proposition 1** (Mundici [25]) *Given a ŁIP-assignment $\Sigma = \{C(\alpha_i) = q_i \mid 1 \leq i \leq k\}$, the following are equivalent:*

(a) *$\Sigma$ is a coherent ŁIP-assignment.*
(b) *$\Sigma$ is a satisfiable ŁIP-assignment.*

□

Proposition 1 asserts that deciding ŁIP coherence is the same as determining ŁIP-assignment satisfiability, which we call *ŁIPSAT*. This result is the $Ł_\infty$ analogous to de Finetti's characterization of coherence of classical probabilistic assignment as equivalent to the *probabilistic satisfiability* (PSAT) of the assignment, which was shown to be an NP-complete problem that can be solved using linear algebraic methods [16,27]. It has also been shown by Bova and Flaminio [6] that deciding the coherence of a ŁIP-assignment is also an NP-complete problem.

Our goal here is to explore efficient ways to decide the coherence of ŁIP-assignments. In analogy to the algorithms used for deciding PSAT [13,14], we explore a linear algebraic formulation of the problem.

## 3 Algebraic Formulation of ŁIPSAT

We consider an *extended* version of ŁIP-assignments of the form

$$\Sigma = \left\{ C(\alpha_i) \bowtie_i q_i \mid q_i \in [0, 1], \bowtie_i \in \{=, \leq, \geq\}, 1 \leq i \leq k \right\}. \tag{2}$$

Extended ŁIP-assignments may have both inequalities and equalities. Such an assignment is satisfiable if there is a $Ł_\infty$-probability distribution $\lambda$ that verify all inequalities and equalities in it.

Given an extended ŁIP-assignment $\Sigma = \{C(\alpha_i) \bowtie_i q_i\}$, let $q = (q_1, \dots, q_k)'$ be the vector of probabilities in $\Sigma$, $\bowtie$ the "vector" of (in)equality symbols. Suppose we are given $Ł_\infty$-valuations $v_1, \dots, v_m$ and let $\lambda = (\lambda_1, \dots, \lambda_m)'$ be a vector of convex weights. Consider the $k \times m$ matrix $A = [a_{ij}]$ where $a_{ij} = v_j(\alpha_i)$. Then an extended ŁIP-assignment of the form (2) is satisfiable if there are $v_1, \dots, v_m$ and $\lambda$ such that the set of algebraic constrains (3) has a solution:

$$\begin{aligned} A \cdot \lambda &\bowtie q \\ \sum \lambda_j &= 1 \\ \lambda &\geq 0 \end{aligned} \tag{3}$$

The condition $\sum \lambda_j = 1$ can be incorporated as an all-1 row $k+1$ in matrix $A$, $q = (q_1, \dots, q_k, 1)'$ and $\bowtie_{k+1}$ is "=". Note that the number $m$ of columns in $A$ is in principle unbounded, but the following consequence of Carathéodory's Theorem [11] yields that if (3) has a solution, then it has a "small" solution.

**Proposition 2** (Carathéodory's Theorem for ŁIP) *If a set of restrictions of the form* (3) *has a solution, then it has a solution in which at most* $k+1$ *elements of* $\lambda$ *are non-zero.* □

Given the algebraic formulation in (3), NP-completeness of ŁIP satisfiability, originally shown by Bova and Flaminio [6], can be seen as a direct corollary of Proposition 2. In fact, that ŁIPSAT is NP-hard comes from the fact that when all $q_i = 1$, the problem becomes $Ł_\infty$-satisfiability, which is NP-complete [23]; and Proposition 2 asserts the existence of a polynomial size witness for ŁIPSAT, hence is in NP; so ŁIPSAT is NP-complete. See Corollary 1.

However, to apply linear algebraic methods to efficiently solve ŁIPSAT, first we need to provide a normal form for it.

### 3.1 A Normal Form for ŁIP-Assignments

An extended assignment may seem more expressive than regular ŁIP-assignments, but we show that no expressivity is gained by this extension. In fact, we define a *normal form* ŁIP assignment as a pair $\langle \Gamma, \Theta \rangle$, where $\Gamma$ is a set of $Ł_\infty$-formulas and $\Theta$ is a set of ŁIP restrictions over propositional symbols of the form

$$\Theta = \Big\{ C(p_i) = q_i \mid q_i \in [0, 1], \, p_i \in \mathbb{P}, \, 1 \leq i \leq k \Big\}. \tag{4}$$

The formulas $\gamma \in \Gamma$ represent ŁIP-assignments of the form $C(\gamma) = 1$, that is, a set of hard constrains in the form of $Ł_\infty$-formulas which must be satisfied by all valuations in the convex combination that compose a $Ł_\infty$-probability distribution.

A normal form assignment $\langle \Gamma, \Theta \rangle$ is satisfiable if there are $Ł_\infty$-valuations $v_1, \dots, v_m$ such that $v_i(\gamma) = 1$ for every $\gamma \in \Gamma$ and there is a $Ł_\infty$-probability distribution $\lambda_1, \dots, \lambda_m$, such that for each assignment $C(p_i) = q_i \in \Theta$, $\sum_{j=1}^m \lambda_j \cdot v_j(p_i) = q_i$.

The satisfiability of extended ŁIP-assignments reduces to that of normal form ones, as follows.

**Theorem 1** *(Atomic Normal Form) For every extended ŁIP-assignment $\Sigma$ there exists a normal form ŁIP-assignment $\langle \Gamma, \Theta \rangle$ such that $\Sigma$ is satisfiable iff $\langle \Gamma, \Theta \rangle$ is; the normal form assignment can be built from $\Sigma$ in polynomial time.*

***Proof*** Start with $\Gamma = \Theta = \emptyset$. Given $\Sigma$, first transform it into $\Sigma'$ in which all assignments are of the form $C(\alpha) \leq p$; for that, if $\Sigma$ contains a constraint of the form $C(\alpha) \bowtie 1, \bowtie \in \{=, \geq\}$ (resp. $C(\alpha) = 0, C(\alpha) \leq 0$) we insert $\alpha$ (resp. $\neg\alpha$) in $\Gamma$ and do not insert the constraint in $\Sigma'$. If $C(\alpha) = q \in \Sigma$ we insert $C(\alpha) \leq q$ and $C(\alpha) \geq q$ in $\Sigma'$. Then all assignments of the latter form are transformed into $C(\neg\alpha) \leq 1 - q$. Also, insert constraints already in the form $C(\alpha) \leq q \in \Sigma$ into $\Sigma'$. All transformation steps preserve satisfiability and can be made in linear time, so $\Gamma \cup \Sigma'$ is satisfiable iff $\Sigma$ is.

For every $C(\alpha_i) \leq q_i \in \Sigma'$, $0 < q_i < 1$, consider a *new* symbol $y_i$; insert $\alpha_i \to y_i$ in $\Gamma$ and $C(y_i) = q_i$ in $\Theta$. Clearly $\langle \Gamma, \Theta \rangle$ is in normal form and is obtained in linear time. The fact that $\Sigma$ is satisfiable iff $\langle \Gamma, \Theta \rangle$ is follows from Lemma 1. □

***Example 3*** Note that the formalization presented in Example 2 is already in normal form, witnessing that this format is quite a natural one to formulate ŁIP-assignments. □

### 3.2 Algebraic Methods for Normal Form ŁIP-Assignments

For the rest of this paper we assume that ŁIP-assignments are in normal form. Here we explore their algebraic structure as it allows for the interaction between a ŁIP problem $\Theta$ and a $Ł_\infty$-SAT instance $\Gamma$, such that solutions satisfying the normal form assignment can be seen as probabilistic solutions to $\Theta$ constrained by the SAT instance $\Gamma$.

Furthermore, to construct a convex combination of the form (1) we will *only* consider $\Gamma$-satisfiable valuations. Given a ŁIP-assignment $\langle \Gamma, \Theta = \{C(p_i) = q_i\} \rangle$, a partial assignment $v$ over $p_i, \ldots, p_k$ is *$\Gamma$-satisfiable* if it can be extended to a full assignment that satisfies all formulas in $\Gamma$. Let $q$ be a $k+1$ dimensional vector $(q_1, \ldots, q_k, 1)'$. The following is a direct consequence of Theorem 1.

**Lemma 2** *A normal form instance $\langle \Gamma, \Theta \rangle$ is satisfiable iff there is a $(k+1) \times (k+1)$-matrix $A_\Theta$, such that all of its columns are $\Gamma$-satisfiable, $A_\Theta$ last row is all 1's, and $A_\Theta \lambda = q$ has a solution $\lambda \geq 0$.*

***Proof*** Let $m$ be the number of formulas in $\Gamma$ and let $l = m + k$. Suppose first that $\langle \Gamma, \Theta \rangle$ is satisfiable, thus the assignment admit a solution $A\lambda = q$, according to (3); the condition $\sum \lambda_j = 1$ is incorporated as the final row of $A$ containing only 1's. Each column $A_j$ in the $(l+1) \times 2^{(k+n)}$ matrix $A$ corresponds to a $Ł_\infty$-valuation $v_j$; and $\lambda$ is a convex combination over $A$'s columns. Clearly, $\lambda_j > 0$ implies that $v_j$ satisfies $\Gamma$. Let the $(k+1) \times l$ matrix $A'$ be obtained from $A$ by deleting each line corresponding to a formula $S \in \Gamma$, and deleting each column $A^j$ such that $\lambda_j = 0$. Moreover, let $\lambda'$ be obtained from $\lambda$ by deleting each entry $\lambda_j = 0$. Note that, by construction, $A'\lambda' = q, \lambda' \geq 0$, and the columns in $A'$ are $\Gamma$-satisfiable. Then, by Carathéodory's Theorem (Proposition 2) there exists a $(k+1) \times (k+1)$ matrix $A''$, built from $A'$ columns, and a $k+1$ dimensional vector $\lambda$ such that $A'' \cdot \lambda = q$ has a solution $\lambda \geq 0$.

Conversely, suppose that the desired matrix $A_\Theta$ exists, thus $A_\Theta \cdot \lambda = q$ for some $\lambda \geq 0$. Each column of $A_\Theta$, being $\Gamma$-satisfiable, can be transformed into a column of $A$ by extending it with $m$ 1's, corresponding to the formulas in $\Gamma$. It follows easily that restrictions (3) have a solution, and thus $\langle \Gamma, \Theta \rangle$ is satisfiable. □

Lemma 2 leads to a linear algebraic PSAT solving method as follows. Let $V$ be the set of partial valuations over the symbols in $\Theta$; consider a $|V|$-dimensional vector $c$ such that

$$c_j = \begin{cases} 0, & v_j \in V \text{ is } \Gamma\text{-satisfiable} \\ 1, & \text{otherwise} \end{cases} \tag{5}$$

The vector $c$ is a boolean "cost" associated to each partial valuation $v_j \in V$, such that the cost is 1 iff $v_j$ is $\Gamma$-unsatisfiable. Consider a matrix $A$ whose columns are the valuations in $V$. Now consider linear program (6) which aims at minimizing that cost, weighted by the corresponding probability value $\lambda_j$.

$$\begin{array}{ll} \min & c' \cdot \lambda \\ \text{subject to} & A \cdot \lambda = q \\ & \sum \lambda_i = 1 \\ & \lambda \geq 0 \\ & A'\text{s columns are partial valuations in } V \end{array} \tag{6}$$

**Theorem 2** *A normal form instance $\langle \Gamma, \Theta = \{C(p_i) = q_i \mid 1 \leq i \leq k\}\rangle$ is satisfiable iff linear program (6) reaches a minimal solution $c' \cdot \lambda = 0$. Furthermore, if there is a solution, then there is a solution in which at most $k + 1$ values of $\lambda$ are not null.*

***Proof*** If linear program (6) reaches 0, we obtain $v_1, \ldots, v_m$ by selecting only the $\Gamma$-satisfiable columns $A_j$ for which $\lambda_j > 0$, obtaining a convex combination satisfying $\Theta$. So $\langle \Gamma, \Theta\rangle$ is satisfiable. Conversely, if $\langle \Gamma, \Theta\rangle$ is satisfiable, by Lemma 2 there exists a matrix $A_\Theta$ such that all of its columns are $\Gamma$-satisfiable partial valuations and $A_\Theta \cdot \lambda = q$; clearly $A_\Theta$ is a submatrix of $A$; make $\lambda_j = 0$ when $A_j$ is a $A_\Theta$ column and thus $c' \cdot \lambda = 0$. Again by Lemma 2, $A_\Theta$ has at most $k + 1$ columns so at most $k + 1$ values of $\lambda$ are not null. □

The following consequence of Theorem 2 was originally proven by Bova and Flaminio [6] as the decision of ŁIP-assignment coherence, which is equivalent to ŁIP satisfiability by Proposition 1.

**Corollary 1** (ŁIPSAT Complexity) *The problem of deciding the satisfiability of a ŁIP-assignment is NP-complete.*

***Proof*** Suppose we have a ŁIP-assignment of the form $\{C(\alpha_i) = 1 \mid 1 \leq i \leq k\}$, then the problem is equivalent to deciding if the set $\{\alpha_1, \ldots, \alpha_k\}$ is $Ł_\infty$-satisfiable, which is NP-complete [23]. So ŁIPSAT is NP-hard.

Now suppose we have a ŁIP-assignment, which can be placed in normal form in polynomial time by Theorem 1. Then Theorem 2 shows that if the problem is satisfiable, it can be verified in polynomial time by guessing suitable valuations and "small distribution" $\lambda$, constructing matrix $A_\Theta$ and verifying in polynomial time that $A_\Theta \cdot \lambda = q$. So ŁIPSAT is in NP. □

Despite the fact that solvable linear programs of the form (6) always have polynomial size solutions, with respect to the size of the corresponding normal form ŁIP-assignment, the elements of linear program itself (6) may be exponentially large, rendering the explicit representation of matrix $A$ impractical. In the following, we present an algorithmic technique that avoids that exponential explosion.

## 4 A ŁIPSAT-Solving Algorithm

Based on the results of the previous section we are going to present an algorithm employing a linear programming technique called *column generation* [19,21], to obtain a decision procedure for Łukasiewicz Infinitely-valued Probabilistic Logic, which we call *ŁIPSAT solving*. This algorithm solves the potentially large linear program (6) without explicitly representing all columns and making use of an extended solver for $Ł_\infty$-satisfiability as an auxiliary procedure to generate columns.

To avoid the exponential blow of the size of matrix in (6), the algorithm basic idea is to employ the simplex algorithm [2,28] over a normal form ŁIP-assignment $\langle \Gamma, \Theta \rangle$, coupled with a strategy that generates cost decreasing columns without explicitly representing the full matrix $A$. In this process, we start with a *feasible solution*, which may contain several $Ł_\infty$ $\Gamma$-unsatisfiable columns. We minimize the cost function consisting of the sum of the probabilities associated to $\Gamma$-unsatisfiable columns, such that when it reaches zero, we know that the problem is satisfiable; if no column can be generated and the minimum achieved is bigger than zero, a negative decision is reached.

The general strategy employed here is similar to that employed to PSAT solving [13,14], but the column generation algorithm is considerably distinct and requires an extension of $Ł_\infty$ decision procedure.

From the input $\langle \Gamma, \Theta \rangle$, we implicitly obtain an unbounded matrix $A$ and explicit obtain the vector of probabilities $q$ mentioned in (6). The basic idea of the simplex algorithm is to move from one feasible solution to another one with a decreasing cost. The feasible solution consists of a square matrix $B$, called the basis, whose columns are extracted from the unbounded matrix $A$. The pair $\langle B, \lambda \rangle$ consisting of the basis $B$ and a ŁIP probability distribution $\lambda$ is a *feasible solution* if $B \cdot \lambda = q$ and $\lambda \geq 0$. We assume that $q_{k+1} = 1$ such that the last line of $B$ we will force $\sum_G \lambda_j = 1$, where $G$ is the set of $B$ columns that are $\Gamma$-satisfiable. Each step of the algorithm replaces one column of the feasible solution $\langle B^{(s-1)}, \lambda^{(s-1)} \rangle$ at step $s-1$ obtaining a new feasible solution $\langle B^{(s)}, \lambda^{(s)} \rangle$. The cost vector $c^{(s)}$ is a $\{0, 1\}$ vector such that $c_j^{(s)} = 1$ iff $B_j$ is $\Gamma$-unsatisfiable. The column generation and substitution is designed such that the total cost is never increasing, that is $c^{(s)\prime} \cdot \lambda^{(s)} \leq c^{(s-1)\prime} \cdot \lambda^{(s-1)}$.

Algorithm 4.1 presents the top level ŁIPSAT decision procedure. Lines 1–3 present the initialization of the algorithm. We assume the vector $q$ is in ascending order. Let the $D_{k+1}$ be a $k+1$ square matrix in which the elements on the diagonal and below are 1 and all the others are 0. At the initial step we make $B^{(0)} = D_{k+1}$, this forces $\lambda_1^{(0)} = q_1 \geq 0$, $\lambda_{j+1}^{(0)} = q_{j+1} - q_j \geq 0, 1 \leq j \leq k$; and $c^{(0)} = [c_1 \cdots c_{k+1}]'$, where $c_k = 0$ if column $j$ in $B^{(0)}$ is $\Gamma$-satisfiable; otherwise $c_j = 1$. Thus the initial state $s = 0$ is a feasible solution.

Algorithm 4.1 main loop covers lines 5–12 which contains the column generation strategy described above. Column generation occurs at beginning of the loop (line 5) which we are going to detail bellow. If column generation fails the process ends with failure in line 7. Otherwise a column is removed and the generated column is inserted in a process we called *merge* at line 9. The loop ends successfully when the objective function (total cost) $c^{(s)\prime} \cdot \lambda^{(s)}$ reaches zero and the algorithm outputs a probability distribution $\lambda$ and the set of $\Gamma$-satisfiable columns in $B$, at line 13.

The procedure *merge* is part of the simplex method which guarantees that given a $k+1$ column $y$ and a feasible solution $\langle B, \lambda \rangle$ there always exists a column $j$ in $B$ such that if $B[j{:=}y]$ is obtained from $B$ by replacing column $j$ with $y$, then there is $\lambda'$ such that $\langle B[j{:=}y], \lambda' \rangle$ is a feasible solution.

**Algorithm 4.1** ŁIPSAT-CG: a ŁIPSAT solver via Column Generation

**Input:** A normal form ŁIPSAT instance $\langle \Gamma, \Theta \rangle$. **Output:** No, if $\langle \Gamma, \Theta \rangle$ is unsatisfiable. Or a solution $\langle B, \lambda \rangle$ that minimizes (6).

$q$:=$[\{q_i \mid C(p_i) = q_i \in \Theta, 1 \leq i \leq k\} \cup \{1\}]$ in ascending order;
$B^{(0)}$:=$D_{k+1}$;
$s$:=0, $\lambda^{(s)} = (B^{(0)})^{-1} \cdot q$ and $c^{(s)} = [c_1 \cdots c_{k+1}]'$;
**while** $c^{(s)\prime} \cdot \lambda^{(s)} \neq 0$ **do**
  $y^{(s)} = \mathit{GenerateColumn}(B^{(s)}, \Gamma, c^{(s)})$;
  **if** $y^{(s)}$ column generation failed **then**
    **return** No; {ŁIPSAT instance is unsatisfiable}
  **else**
    $B^{(s+1)} = \mathit{merge}(B^{(s)}, b^{(s)})$;
    $s$++, recompute $\lambda^{(s)}$ and $c^{(s)}$;
  **end if**
**end while**
**return** $\langle B^{(s)}, \lambda(s) \rangle$; {ŁIPSAT instance is satisfiable}

**Lemma 3** *Let $\langle B, \lambda \rangle$ be a feasible solution of (6), such that $B$ is non-singular, and let $y$ be a column. Then there always exists a column $j$ such that $\langle B[j{:=}y], \lambda' \rangle$ is a non-singular feasible solution.*

***Proof*** As $\langle B, \lambda \rangle$ is a feasible solution,

$$\sum_{i=1}^{k+1} B_i \lambda_i = q. \tag{7}$$

Suppose we replace column $B_j$ by $y$. Due to the fact that $B$ is not singular, there are coefficients $\beta_1, \beta_2, \ldots, \beta_{k+1}$ such that $\sum_{i=1}^{k+1} \beta_i B_i = y$, and thus:

$$B_j = \frac{y}{\beta_j} - \frac{\beta_1}{\beta_j} B_1 - \cdots - \frac{\beta_{j-1}}{\beta_j} B_{j-1} - \frac{\beta_{j+1}}{\beta_j} B_{j+1} - \cdots - \frac{\beta_{k+1}}{\beta_j} B_{k+1}. \tag{8}$$

Substituting (8) for $B_j$ in (7) yields:

$$\frac{\lambda_j}{\beta_j} y + \sum_{i=1}^{k+1} (\lambda_i - \frac{\beta_i}{\beta_j} \lambda_j) B_i = q. \tag{9}$$

Note that the coefficient of $B_j$ in the sum is 0. We have now a new vector of coefficients $\lambda'$ such that $B[j{:=}y] \cdot \lambda' = q$. Properly choosing $j$ guarantees $\lambda' \geq 0$. As the elements of columns $B_i$ and $y$ are all non negative valuations, the set $\beta_{>0} = \{\beta_i \mid \beta_i > 0\}$ is not empty. Taking a $j$ from the set $\{j \mid \beta_j \in \beta_{>0}$ and $\forall i, \beta_i \lambda_j \leq \beta_j \lambda_i\}$ implies $\lambda_i - \frac{\beta_i}{\beta_j} \lambda_j \geq 0$, for all $i \neq j$, and $\lambda_j / \beta_j \geq 0$, so $\lambda' \geq 0$. Finally, as $\beta_j > 0$ and all columns in $B$ are linearly independent, $B[j{:=}y]$ is non-singular. □

Lemma 3 guarantees the existence of a column which may not be unique and further selection heuristic is necessary; in our implementation we give priority to remove columns which are associated to probability zero on a left-to-right order.

We now describe the column generation method, which takes as input the current basis $B$, the current cost $c$, and the $Ł_\infty$ restrictions $\Gamma$; the output is a column $y$, if it exists, otherwise it signals **No**. The basic idea for column generation is the property of the simplex algorithm called the *reduced cost* of inserting a column $y$ with cost $c_y$ in the basis. The reduced cost is given by equation

$$r_y = c_y - c' B^{-1} y \tag{10}$$

and the simplex method guarantees that the objective function is non increasing if $r_y \leq 0$. Furthermore the generation method is such that the column $y$ is $\Gamma$-satisfiable so that $c_y = 0$. We thus obtain

$$c' B^{-1} y \geq 0 \tag{11}$$

which is an inequality on the elements of $y$. To force $\lambda$ to be a probability distribution, we make $y_{k+1} = 1$, the remaining elements $y_i$ are valuations of the variables in $\Theta$, so that we are searching for solution to (11) such that $0 \leq y_i \leq 1$, $1 \leq i \leq k$. To finally obtain column $y$ we must extend a $Ł_\infty$-solver that generates valuations satisfying $\Gamma$ so that it also respects the linear restriction (11). In fact this is not an expressive extension of $Ł_\infty$ as the McNaughton property guarantees that (11) is equivalent to some $Ł_\infty$-formula on variables $y_1, \ldots, y_k$ [7]. In practice, we tested two ways of obtaining a joint solver for $\Gamma$ and (11):

– Employ an SMT (SAT modulo theories) solver that can handle linear algebraic equations such as (11) and the linear inequalities generated by the $Ł_\infty$-semantics. $Ł_\infty$-solvers based on SMT can be found in the literature, see [3];
– Use a MIP (mixed integer programming) solver that encodes $Ł_\infty$-semantics. Equation (11) is simply a new linear restriction to be dealt by the MIP solver. $Ł_\infty$-solvers based on MIP solvers have been proposed by [18].

In both cases, the restrictions posed by $\Gamma$-formulas and (11) are jointly handled by the semantics of the underlying solver. Note that both MIP solving and SMT (linear algebra) are NP-complete problems. We have thus the following result.

**Lemma 4** *There are algorithmic solutions to the problem of jointly satisfying $Ł_\infty$-formulas and inequalities with common variables.* □

We now deal with the problem of termination. Column generation as above guarantees that the cost is never increasing. The simplex method ensures that a solvable problem always terminates if the costs always decrease, we are left with the problem of guaranteeing that the objective function does not become stationary. This is guaranteed in the implementation by a column selection strategy that respects *Bland's Rule* and also by plateau escaping strategies such as *Tabu search* [2,28].

**Lemma 5** *There are column selection strategies that guarantee that the Algorithm 4.1 always terminates.* □

We know that there are no column selection heuristics that guarantee that the simplex method terminates in a polynomial number of steps. However, the simplex method performs very well in most practical cases and its average complexity is known to be polynomial [5].

By placing all the results above together we can state the correction of Algorithm 4.1.

**Theorem 3** *Consider the output of Algorithm 4.1 with normal form input $\langle \Gamma, \Theta \rangle$. If the algorithm succeeds with solution $\langle B, \lambda \rangle$, then the input problem is satisfiable with distribution $\lambda$ over the valuations which are columns of $B$. If the program outputs no, then the input problem is unsatisfiable. Furthermore, there are column selection strategies that guarantee termination.*

***Proof*** Lemma 3 guarantees that all steps $\langle B^{(s)}, \lambda(s) \rangle$ is a feasible solution to the problem. If Algorithm 4.1 terminates with success, than cost zero has been reached, so by Theorem

2 the input problem is satisfiable. On the other hand, if column generation fails, this fails with a positive cost, this means there are no $\Gamma$-satisfiable columns that can reduce the cost. So, the problem in unsatisfiable. Finally, a suitable column selection strategy by Lemma 5 guarantees termination. □

***Example 4*** We show the steps for the solution of Example 2. Initially, we have

$$q = \begin{bmatrix} 0.6 \\ 0.6 \\ 0.6 \\ 1 \end{bmatrix}, B^{(0)} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}, \lambda^{(0)} = (B^{(0)})^{-1} \cdot q = \begin{bmatrix} 0.6 \\ 0 \\ 0 \\ 0.4 \end{bmatrix}, c^{(0)} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}.$$

$c^{(0)}$ expresses that the first two columns of $B^{(0)}$ are $\Gamma$-satisfiable. The total cost $\text{cost}^{(0)} = c^{(0)\prime} \cdot \lambda^{(0)} = 0.4$. At this point, column $y^{(1)}$ is generated substituting $B^{(0)}$'s column 3 in the *merge* procedure:

$$y^{(1)} = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \end{bmatrix}, B^{(1)} = \begin{bmatrix} 1 & 0 & \mathbf{1} & 0 \\ 1 & 1 & \mathbf{0} & 0 \\ 1 & 1 & \mathbf{1} & 0 \\ 1 & 1 & \mathbf{1} & 1 \end{bmatrix}, \lambda^{(1)} = \begin{bmatrix} 0.6 \\ 0 \\ 0 \\ 0.4 \end{bmatrix}, c^{(1)} = \begin{bmatrix} 0 \\ 0 \\ \mathbf{0} \\ 1 \end{bmatrix}.$$

$\text{cost}^{(1)} = 0.4$. Again, column generation provides $y^{(2)}$ in place of column 1:

$$y^{(2)} = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \end{bmatrix}, B^{(2)} = \begin{bmatrix} \mathbf{1} & 0 & \mathit{1} & 0 \\ \mathbf{1} & 1 & \mathit{0} & 0 \\ \mathbf{0} & 1 & \mathit{1} & 0 \\ \mathbf{1} & 1 & \mathit{1} & 1 \end{bmatrix}, \lambda^{(2)} = \begin{bmatrix} 0.3 \\ 0.3 \\ 0.3 \\ 0.1 \end{bmatrix}, c^{(2)} = \begin{bmatrix} \mathbf{0} \\ 0 \\ \mathit{0} \\ 1 \end{bmatrix}.$$

$\text{cost}^{(2)} = 0.1$. Finally, column generation provides $y^{(3)}$ in place of column 4:

$$y^{(3)} = \begin{bmatrix} 0.5 \\ 0.5 \\ 0.5 \\ 1 \end{bmatrix}, B^{(3)} = \begin{bmatrix} \mathit{1} & 0 & \mathit{1} & \mathbf{0.5} \\ \mathit{1} & 1 & \mathit{0} & \mathbf{0.5} \\ \mathit{0} & 1 & \mathit{1} & \mathbf{0.5} \\ \mathit{1} & 1 & \mathit{1} & \mathbf{1} \end{bmatrix}, \lambda^{(3)} = \begin{bmatrix} 0.2 \\ 0.2 \\ 0.2 \\ 0.4 \end{bmatrix}, c^{(3)} = \begin{bmatrix} \mathit{0} \\ 0 \\ \mathit{0} \\ \mathbf{0} \end{bmatrix}.$$

$\text{cost}^{(3)} = 0$, so that the problem is satisfiable with solution $\langle B^{(3)}, \lambda^{(3)} \rangle$. □

## 5 Ł$_\infty$-MODSAT Expressivity Over Rational McNaughton

Column Generation technique used in Algorithm 4.1 computes a valuation satisfying restriction (11) modulo the satisfiability of the set of formulas $\Gamma$. In this section we show that this technique of valuating a formula $\varphi$ modulo the 1-satisfiability of a set of formulas $\Psi$, that is, evaluate the truth value $v(\varphi)$ under the restriction $\{v(\psi) = 1 \mid \psi \in \Psi\}$, actually increases the expressivity of the logic Ł$_\infty$, the resulting system is called Ł$_\infty$-MODSAT, which employs pairs $\langle \varphi, \Psi \rangle$.

McNaughton's result guarantees that Ł$_\infty$-formulas corresponds to, and only to, McNaughton functions, that is, piecewise linear functions with integer coefficients. Several proposals in the literature tried to expand that expressivity to so called *rational McNaughton functions*, that is, piecewise linear functions with rational coefficients.

The work of Esteva, Godo and Montagna proposes logic Ł$\Pi\frac{1}{2}$ which extends Ł$_\infty$ logic with a product operator, its residuum, and a constant expressing the truth value $\frac{1}{2}$, not directly expressible in $Ł_\infty$ [12]. That logic not only allows for the expressivity of rational McNaughton functions but also expresses piecewise polynomials; as a consequence satisfiability over $ŁΠ\frac{1}{2}$ requires finding roots of polynomials of $n$-degree rendering its complexity extremely high. Aguzzoli and Mundici proposes logic ∃Ł which also expresses rational McNaughton functions and has complexity $\Sigma_2^p$ for the satisfiability problem [1]. Logic ∃Ł extends $Ł_\infty$ by providing restricted form of propositional quantification whose semantic counterpart is the maximization of a set of $Ł_\infty$-valuations of a formula.

Gerla introduces Rational Łukasiewicz Logic by extending $Ł_\infty$-language with unary operators $\delta_n$, for $n \in \mathbb{N}^*$, whose semantics is given by $v(\delta_n\varphi) = \frac{v(\varphi)}{n}$, for $\varphi$ a $Ł_\infty$-formula and $v$ a $Ł_\infty$-valuation [17]. Rational Łukasiewicz Logic expresses rational McNaughton functions and its associated tautology problem is coNP-complete, which is a very reasonable complexity for this task.

In this section we want to show that the expressivity of rational McNaughton functions can be obtained using $Ł_\infty$-MODSAT; in the end we compare our results with the ones about Rational Łukasiewicz Logic. Let the set of propositional symbols be given by $\mathbb{P} = \{x_1, x_2, \ldots\}$ and $\mathrm{Var}(\Psi)$ be the set of variables appearing in the formulas $\psi \in \Psi$; we write $\mathrm{Var}(\psi)$ instead of $\mathrm{Var}(\{\psi\})$. We call a *$\Psi$-sat valuation* any $Ł_\infty$-valuation $v$ that makes $v(\psi) = 1$ for all $\psi \in \Psi$.

According to McNaughton [22], given any $Ł_\infty$-formula $\varphi \in \mathcal{L}$ with $\mathrm{Var}(\varphi) \subset \{x_1, \ldots, x_n\}$, we inductively associate to $\varphi$ a function $f_\varphi : [0, 1]^n \to [0, 1]$ by:[5]

(i) $f_{x_i}(x_1, \ldots, x_n) = x_i$, for $i = 1, \ldots, n$;
(ii) $f_{\neg\varphi}(x_1, \ldots, x_n) = 1 - f_\varphi(x_1, \ldots, x_n)$;
(iii) $f_{\varphi_1 \oplus \varphi_2}(x_1, \ldots, x_n) = \min(1, f_{\varphi_1}(x_1, \ldots, x_n) + f_{\varphi_2}(x_1, \ldots, x_n))$.

Note that the definition of $f_\varphi$ depends on $n$; note also that given a $Ł_\infty$-formula $\varphi$, with $\mathrm{Var}(\varphi) = \{x_1, \ldots, x_n\}$, and a $Ł_\infty$-valuation $v$, $f_\varphi(v(x_1), \ldots, v(x_n)) = v(\varphi)$.

We extend this notion, given a pair $\langle\varphi, \Psi\rangle$, where $\varphi$ is a $Ł_\infty$-formula and $\Psi$ is a set of $Ł_\infty$-formulas, where $\mathrm{Var}(\varphi) \cup \mathrm{Var}(\Psi) \subset \{x_1, \ldots, x_n\}$, as follows. First, let function domain be

$$D_{\langle\varphi,\Psi\rangle} = \Big\{\langle r_1, \ldots, r_n\rangle \in [0, 1]^n \mid f_\psi(r_1, \ldots, r_n) = 1, \text{ for all } \psi \in \Psi\Big\}.$$

And thus we are able to inductively define the function $f_{\langle\varphi,\Psi\rangle} : D_{\langle\varphi,\Psi\rangle} \to [0, 1]$ by clauses in total analogy to (i)–(iii) above. The definitions of $D_{\langle\varphi,\Psi\rangle}$ and $f_{\langle\varphi,\Psi\rangle}$ also depend on $n$.

We say that a rational McNaughton function $f : [0, 1]^n \to [0, 1]$ is *representable in $Ł_\infty$-MODSAT* if there is a pair $\langle\varphi, \Psi\rangle$, with $\mathrm{Var}(\varphi) \cup \mathrm{Var}(\Psi) = \{x_1, \ldots, x_m\}$, $m \geq n$, and $m - n$ functions $z_i : [0, 1]^n \to [0, 1]$, $i = 1, \ldots, m - n$, such that:

- For any $\langle r_1, \ldots, r_m\rangle \in D_{\langle\varphi,\Psi\rangle}$, $r_{n+i} = z_i(r_1, \ldots, r_n)$, $i = 1, \ldots, m - n$;
- $f(x_1, \ldots, x_n) = f_{\langle\varphi,\Psi\rangle}(x_1, \ldots, x_n, z_1(x_1, \ldots, x_n), \ldots, z_{m-n}(x_1, \ldots, x_n))$.

The pair $\langle\varphi, \Psi\rangle$ is the *representation of $f$ in $Ł_\infty$-MODSAT*. We will write $\mathbf{x} = \langle x_1, \ldots, x_n\rangle$ and $\mathbf{z} = \langle x_{n+1}, \ldots, x_m\rangle$.

In order to establish our main result, that all rational McNaughton functions may be represented in $Ł_\infty$-MODSAT, we first show the possibility of defining constants in a $Ł_\infty$-formula $\varphi$ in the pair $\langle\varphi, \Psi\rangle$ within the $Ł_\infty$-MODSAT system. It is already possible to define 1 and 0 in $Ł_\infty$ by $x_1 \oplus \neg x_1$ and its negation, respectively. For $n$ a nonnegative integer and $x$ a propositional variable, we define $0x = 0$ and $nx = x \oplus (n - 1)x$.

[5] We abuse the notation by using the same symbols for the propositional variables and for the metavariables in the functions description.

**Lemma 6** *Given a rational number $c \in (0,1)$, there is a set $\Psi$ of $Ł_\infty$-formulas, with $z_c \in Var(\Psi)$, such that, for any $\Psi$-sat valuation $v$, we have $v(z_c) = c$.*

***Proof*** Let $d = \frac{1}{b}$ with $b \in \mathbb{Z}_+^*$ and $\psi_d = z_d \leftrightarrow \neg(b-1)z_d \in \Psi$. Any $\Psi$-sat valuation $v$, i.e. any valuation $v$ for which $v(\psi_d) = 1$, makes $v(z_d) = d$. Let $c = \frac{a}{b}$, with $a, b \in \mathbb{Z}$ and $0 < a < b$, and $\psi_c = z_c \leftrightarrow az_d \in \Psi$. Any $\Psi$-sat valuation $v$ makes $v(z_c) = c$. Note that, by the definition above, letting $\varphi = z_c$ and $\Psi = \{\psi_d, \psi_c\}$, the pair $\langle \varphi, \Psi \rangle$ represents the constant $c$ in $Ł_\infty$-MODSAT. □

Our next step is to show that linear functions may be represented in $Ł_\infty$-MODSAT. Let $g$ be a function, we write $g^\# = \min(\max(g, 0), 1)$.

**Lemma 7** *Let $g : [0,1]^n \to \mathbb{R}$ be a linear function with rational coefficients,*

$$g(\mathbf{x}) = \frac{a_1}{b_1}x_1 + \cdots + \frac{a_n}{b_n}x_n + c,$$

*where $a_i \in \mathbb{Z}$, $b_i \in \mathbb{Z}_+^*$, and $c \in \mathbb{Q}$. Then, $g^\#$ is representable in $Ł_\infty$-MODSAT.*

***Proof*** We proceed by induction on $a = |a_1| + \cdots + |a_n|$. If $a = 0$, the result follows by Lemma 6. For $a > 0$, assume the lemma holds for $a - 1$ and, with no loss of generality, that $|a_1| = \max(|a_1|, \ldots, |a_n|)$.

Let us consider first the case where $a_1 > 0$. Let $h = g - \frac{x_1}{b_1}$, such that

$$h(\mathbf{x}) = \frac{a_1 - 1}{b_1}x_1 + \cdots + \frac{a_n}{b_n}x_n + c.$$

By induction hypothesis, there are $\langle \varphi_h, \Psi_h \rangle$ and $\langle \varphi_{h+1}, \Psi_{h+1} \rangle$ such that $h^\# = f_{\langle \varphi_h, \Psi_h \rangle}$ and $(h+1)^\# = f_{\langle \varphi_{h+1}, \Psi_{h+1} \rangle}$. We define

$$\Psi = \Psi_h \cup \Psi_{h+1} \cup \left\{ z_{\frac{1}{b_1}} \leftrightarrow \neg(b-1)z_{\frac{1}{b_1}}, \quad b_1 z_1 \leftrightarrow x_1, \quad z_1 \to z_{\frac{1}{b_1}} \right\},$$

and claim that $\langle \varphi, \Psi \rangle$, with $\varphi =_{\text{def}} (\varphi_h \oplus z_1) \odot \varphi_{h+1}$, represents $g^\#$. Note that, with the three new $Ł_\infty$-formulas added to $\Psi$, the pair $\langle z_{\frac{1}{b_1}}, \Psi \rangle$ defines $\frac{1}{b_1}$ and the pair $\langle z_1, \Psi \rangle$ defines $\frac{x_1}{b_1}$, depending on the value of $x_1$. So, the new variables $z_{\frac{1}{b_1}}$ and $z_1$ added to $\text{Var}(\Psi)$ are associated to the part $\mathbf{z}$ of $\langle \mathbf{x}, \mathbf{z} \rangle \in D_{\langle \varphi, \Psi \rangle}$ and may only assume values that can be computed as a function of the values of $\mathbf{x}$. When $\mathbf{x}$ is such that $h(\mathbf{x}) \in [0,1]$,

$$g^\#(\mathbf{x}) = \left(h(\mathbf{x}) + \frac{x_1}{b_1}\right)^\# = f_{\langle \varphi_h \oplus z_1, \Psi \rangle}(\mathbf{x}, \mathbf{z}) = f_{\langle (\varphi_h \oplus z_1) \odot 1, \Psi \rangle}(\mathbf{x}, \mathbf{z}) = f_{\langle \varphi, \Psi \rangle}(\mathbf{x}, \mathbf{z}).$$

When $\mathbf{x}$ is such that $h(\mathbf{x}) \in [-1, 0]$,

$$g^\#(\mathbf{x}) = \left(h(\mathbf{x}) + \frac{x_1}{b_1}\right)^\# = \max\left(0, h(\mathbf{x}) + \frac{x_1}{b_1}\right) = \max\left(0, \frac{x_1}{b_1} + h(\mathbf{x}) + 1 - 1\right) =$$
$$= f_{\langle z_1 \odot \varphi_{h+1}, \Psi \rangle}(\mathbf{x}, \mathbf{z}) = f_{\langle \varphi, \Psi \rangle}(\mathbf{x}, \mathbf{z}).$$

The cases of $\mathbf{x}$ where $h(\mathbf{x}) > 1$ and $h(\mathbf{x}) < -1$ are trivial.

For the case where $a_1 < 0$, it is sufficient to apply the same reasoning to $1 - g$. As $1 - (1-g)^\# = g^\#$, the lemma follows. □

Finally, we show our version of McNaughton's Theorem for the $Ł_\infty$-MODSAT setting. First, a version for rational McNaughton functions with one variable.

**Theorem 4** *Let* $f : [0, 1] \to [0, 1]$ *be a one variable rational McNaughton function. Then,* $f$ *is representable in* $Ł_\infty$*-MODSAT.*

***Proof*** The domain $[0, 1]$ of $f$ may be partitioned into $[\alpha_i, \alpha_{i+1}]$, $i = 0, \ldots, n-1$, such that each part $f : [\alpha_i, \alpha_{i+1}] \to [0, 1]$ is a linear function; let $\beta_i = f(\alpha_i)$.

We define the *hat functions*[6] $\mathcal{H}_i : [0, 1] \to [0, 1]$, $i = 0, \ldots, n$, by:

- $\mathcal{H}_0$ has as graph the segments from $(\alpha_0, \beta_0)$ to $(\alpha_1, 0)$ and from $(\alpha_1, 0)$ to $(\alpha_n, 0)$;
- $\mathcal{H}_i$ has as graph the segments from $(\alpha_0, 0)$ to $(\alpha_{i-1}, 0)$, from $(\alpha_{i-1}, 0)$ to $(\alpha_i, \beta_i)$, from $(\alpha_i, \beta_i)$ to $(\alpha_{i+1}, 0)$, and from $(\alpha_{i+1}, 0)$ to $(\alpha_n, 0)$, $i = 1, \ldots, n-1$;
- $\mathcal{H}_n$ has as graph the segments from $(\alpha_0, 0)$ to $(\alpha_{n-1}, 0)$, from $(\alpha_{n-1}, 0)$ to $(\alpha_n, \beta_n)$.

By Lemma 7, hat functions $\mathcal{H}_0$ and $\mathcal{H}_n$ are easily representable in $Ł_\infty$-MODSAT using a function $g^\#$, where $g$ is linear. The other hat functions $\mathcal{H}_i, i = 1, \ldots, n-1$, may be represented in $Ł_\infty$-MODSAT by the pair $\langle \varphi_1 \wedge \varphi_2, \Psi_1 \cup \Psi_2 \rangle$, where $\langle \varphi_1, \Psi_1 \rangle$ and $\langle \varphi_2, \Psi_2 \rangle$ represent $g_1^\#$ and $g_2^\#$, respectively, where $g_1$ and $g_2$ are linear. Note that the variables $z_1$ associated to the variable $x_1$, with intention to have value $\frac{x_1}{b_1}$ as in Lemma 7, must be different for each representation $\langle \varphi_1, \Psi_1 \rangle$ and $\langle \varphi_2, \Psi_2 \rangle$.

Let $\langle \varphi_{\mathcal{H}_i}, \Psi_{\mathcal{H}_i} \rangle$ be the representation of $\mathcal{H}_i$ in $Ł_\infty$-MODSAT. Then, $f$ is representable in $Ł_\infty$-MODSAT by the pair $\langle \varphi_{\mathcal{H}_1} \oplus \cdots \oplus \varphi_{\mathcal{H}_n}, \Psi_{\mathcal{H}_1} \cup \cdots \cup \Psi_{\mathcal{H}_n} \rangle$. The same note about variables $z_1$ in the former paragraph also apply here. □

The proof of Theorem 4 above highlights how hat functions empowers the MODSAT technique to increase the expressivity of $Ł_\infty$. In the following we prove the main result of this section which generalizes Theorem 4 to the multivariate case; its proof uses constructions from the literature which subsume the use of hat functions above.

**Theorem 5** *Let* $f : [0, 1]^n \to [0, 1]$ *be a (multivariable) rational McNaughton function. Then,* $f$ *is representable in* $Ł_\infty$*-MODSAT.*

***Proof*** According to [7], the domain of $f$ may be decomposed as follows. Let $p_1, \ldots, p_k$ be the linear constituents of $f$, each pair $p_i$ and $p_j$ of these constituents defines two closed half-spaces $H^+$ and $H^-$ such that $p_i(\mathbf{x}) \le p_j(\mathbf{x})$ for $\mathbf{x} \in H^+$ and $p_j(\mathbf{x}) \le p_i(\mathbf{x})$ for $\mathbf{x} \in H^-$. Thus, for any permutation $\rho$ of the set $\{1, \ldots, k\}$, we define

$$P_\rho = \Big\{ \mathbf{x} \in [0, 1]^n \mid p_{\rho(1)}(\mathbf{x}) \le \cdots \le p_{\rho(k)}(\mathbf{x}) \Big\},$$

which is a closed convex polyhedron, since it is an intersection of $[0, 1]^n$ and a finite set of closed half-spaces. As the $p_i$'s have rational coefficients, the vertices of $P_\rho$ have rational coordinates. Let $\mathcal{W}$ be the set of simplices (also with rational coordinates) arising from some triangulation of $n$-dimensional polyhedra $P_\rho$; the union of $\mathcal{W}$ is the cube $[0, 1]^n$, the intersection of a pair of elements in $\mathcal{W}$ is either a common face between them or empty, and, for each $S \in \mathcal{W}$, there is an index $u_S \in \{1, \ldots, k\}$ such that, restricted to $S$, $f = p_{u_S}$.

For each vertex $\mathbf{v}$ of some simplex in $\mathcal{W}$, we define the hat function $\mathcal{H}_\mathbf{v} : [0, 1]^n \to [0, 1]$ so that:

- $\mathcal{H}_\mathbf{v}(\mathbf{v}) = f(\mathbf{v})$;
- $\mathcal{H}_\mathbf{v}(\mathbf{u}) = 0$ for each vertex $\mathbf{u}$ of a simplex in $\mathcal{W}$ different from $\mathbf{v}$;
- $\mathcal{H}_\mathbf{v}$ is linear over each simplex in $\mathcal{W}$.

[6] These functions are only different from the *Schauder hats* in [24] on the values $\beta_i \in \mathbb{Q}$.

As in the one variable case, the hat functions may be represented in $Ł_\infty$-MODSAT by a pair $\langle \varphi, \Psi \rangle$ where $\varphi$ represents a $(\bigvee \bigwedge)$-combination (as in [7]) of the hat function linear pieces given by Lemma 7. Thus, $f$ may be represented in $Ł_\infty$-MODSAT by $\langle \bigoplus_{\mathbf{v}} \varphi_{\mathcal{H}_\mathbf{v}}, \bigcup_{\mathbf{v}} \Psi_{\mathcal{H}_\mathbf{v}} \rangle$. □

The representation theorems above are inspired by the results for representing McNaughton functions into $Ł_\infty$ in [7,24]; these representations are said to be in disjunctive normal forms since they are disjunctions ($\oplus$) of hat functions.

Another possible path to obtain results in Theorems 4 and 5 would be the proof strategy of Gerla's McNaughton-like theorem, which states the 1–1 correspondence between equivalence classes modulo equi-provability of formulas of Rational Łukasiewicz Logic and rational McNaughton functions [17]. According to that result, a rational McNaughton function $f : [0,1]^n \to [0,1]$ is represented by a class of (equi-provable) formulas of Rational Łukasiewicz Logic which has among them the formula in special format

$$\varphi = \bigoplus_{i=0}^{s-1} \delta_s \varphi_i, \tag{12}$$

where $s$ is some integer for which the linear pieces of $s \cdot f$ have integer coefficients and $\varphi_i$ are representations in $Ł_\infty$ for the McNaughton functions $f_i : [0,1]^n \to [0,1]$ given, for $\mathbf{x} \in [0,1]^n$, by

$$f_i(\mathbf{x}) = \max(\min(s \cdot f(\mathbf{x}) - i, 1), 0).$$

We can adapt the arguments in Lemmas 6 and 7 and state the following result; in a sense it says that operators $\delta_n$ may be represented in $Ł_\infty$-MODSAT.

**Theorem 6** *Let $\varphi$ be a $Ł_\infty$-formula, with $Var(\varphi) \subset \{x_1, \ldots, x_n\}$, and $s \in \mathbb{N}^*$. Then, function $\frac{1}{s} \cdot f_\varphi : \mathbf{x} \in [0,1]^n \mapsto \frac{f_\varphi(\mathbf{x})}{s}$ is representable in $Ł_\infty$-MODSAT.*

***Proof*** With new variables $w$ and $w_{\frac{1}{s}}$, we define

$$\Psi = \left\{ w_{\frac{1}{s}} \leftrightarrow \neg(b-1) w_{\frac{1}{s}}, \quad sw \leftrightarrow \varphi, \quad w \to w_{\frac{1}{s}} \right\}$$

and claim that $\langle w, \Psi \rangle$ represents $\frac{1}{s} \cdot f_\varphi$. Indeed, $\langle w_{\frac{1}{s}}, \Psi \rangle$ defines $\frac{1}{s}$, and $\langle w, \Psi \rangle$ defines $\frac{f_\varphi(\mathbf{x})}{s}$, depending on the values $\mathbf{x} = \langle x_1, \ldots, x_n \rangle$. So, for $\langle \mathbf{x}, \mathbf{z} \rangle \in D_{\langle w, \Psi \rangle}$, the variables in $\mathbf{z} = \langle w_{\frac{1}{s}}, w \rangle$ may only assume values that can be computed as functions of $\mathbf{x}$ and

$$\frac{1}{s} \cdot f_\varphi(\mathbf{x}) \quad = \quad \frac{f_\varphi(\mathbf{x})}{s} \quad = \quad f_{\langle w, \Psi \rangle}(\mathbf{x}, \mathbf{z}).$$

□

By Theorem 6 and special format (12), we may say that any class of equi-provable formulas of Rational Łukasiewicz Logic is representable[7] in $Ł_\infty$-MODSAT: let $\varphi$ be the formula in such class as in (12), then the representation is given by the pair $\langle \psi, \Psi \rangle$, where

$$\psi = \bigoplus_{i=0}^{s-1} w_i$$

[7] Of course, since such classes are identified with rational McNaughton functions, that was already a consequence of Theorem 5.

and

$$\Psi = \left\{ w_{\frac{1}{s}} \leftrightarrow \neg(s-1)w_{\frac{1}{s}} \right\} \cup \bigcup_{i=0}^{s-1} \left\{ s w_i \leftrightarrow \varphi_i, \quad w_i \rightarrow w_{\frac{1}{s}} \right\}.$$

## 6 Conclusion and the Future

We provided the theoretical basis for the development and implementation of probabilistic reasoning over "partial truth" that respects Łukasiewicz Infinitely-valued Logic restrictions. The algorithm studied for the ŁIPSAT problem led to the formulation of a framework where rational McNaughton functions may be represented. For the future we hope to develop better solvers for the logics employed having the analysis of the phase transition as a qualitative guideline as well as develop solvers for the $Ł_\infty$-MODSAT system. We want to develop efficient algorithms that, given an $n$-dimensional piecewise linear function with rational coefficients, generates the pair $\langle \varphi, \Psi \rangle$ that represents it in $Ł_\infty$-MODSAT. We also hope to employ the mechanisms developed here to linearly approximate generic functions.

## References


1. Aguzzoli, S., Mundici, D.: Weierstrass Approximation Theorem and Łukasiewicz Formulas with One Quantified Variable, pp. 315–335. Physica-Verlag HD, Heidelberg (2003)
2. Bertsimas, D., Tsitsiklis, J.: Introduction to Linear Optimization. Athena Scientific Series in Optimization and Neural Computation. Athena Scientific, Belmont (1997)
3. Bofill, M., Manya, F., Vidal, A., Villaret, M.: Finding hard instances of satisfiability in Łukasiewicz logics. In: ISMVL. IEEE, pp. 30–35 (2015)
4. Boole, G.: An Investigation on the Laws of Thought. Macmillan, London (1854). http://www.gutenberg.org/etext/15114
5. Borgward, K.H.: The Simplex Method: A Probabilistic Analysis. Algorithms and Combinatorics, vol. 1. Springer, Berlin (1986)
6. Bova, S., Flaminio, T.: The coherence of Łukasiewicz assessments is NP-complete. Int. J. Approx. Reason. **51**(3), 294–304 (2010). https://doi.org/10.1016/j.ijar.2009.10.002
7. Cignoli, R., d'Ottaviano, I., Mundici, D.: Algebraic Foundations of Many-Valued Reasoning. Trends in Logic. Springer, Berlin (2000)
8. de Finetti, B.: La prévision: Ses lois logiques, ses sources subjectives (1937)
9. de Finetti, B.: Sul significato soggettivo della probabilità. Fundam. Math. **17**(1), 298–329 (1931)
10. de Finetti, B.: Theory of Probability: A Critical Introductory Treatment. Translated by Antonio Machí and Adrian Smith. Wiley, Hoboken (2017)
11. Eckhoff, J.: Helly, Radon, and Caratheodory type theorems. In: Gruber, P.M., Wills, J.M. (eds.) Handbook of Convex Geometry, pp. 389–448. Elsevier Science Publishers (1993)
12. Esteva, F., Godo, L., Montagna, F.: The Ł$\Pi$ and Ł$\Pi\frac{1}{2}$ logics: two complete fuzzy systems joining Łukasiewicz and product logics. Arch. Math. Logic **40**(1), 39–67 (2001)
13. Finger, M., Bona, G.D.: Probabilistic satisfiability: logic-based algorithms and phase transition. In: Walsh, T. (ed.) IJCAI, IJCAI/AAAI, pp. 528–533 (2011)
14. Finger, M., De Bona, G.: Probabilistic satisfiability: algorithms with the presence and absence of a phase transition. AMAI **75**(3), 351–379 (2015). https://doi.org/10.1007/s10472-015-9466-6
15. Finger, M., Preto, S.: Probably half true: probabilistic satisfiability over Łukasiewicz infinitely-valued logic. In: Galmiche, D., Schulz, S., Sebastiani, R. (eds.) Automated Reasoning, pp. 194–210. Springer, Cham (2018)
16. Georgakopoulos, G., Kavvadias, D., Papadimitriou, C.H.: Probabilistic satisfiability. J. Complex. **4**(1), 1–11 (1988). https://doi.org/10.1016/0885-064X(88)90006-4
17. Gerla, B.: Rational Łukasiewicz logic and DMV-algebras. Neural Netw. World **11**(6), 579–594 (2001)

18. Hähnle, R.: Towards an efficient tableau proof procedure for multiple-valued logics. In: Börger, E., Kleine Büning, H., Richter, M.M., Schönfeld, W. (eds.) Computer Science Logic, pp. 248–260. Springer, Heidelberg (1991)
19. Hansen, P., Jaumard, B.: Algorithms for the maximum satisfiability problem. Computing **44**, 279–303 (1990). https://doi.org/10.1007/BF02241270
20. Hansen, P., Jaumard, B.: Probabilistic satisfiability. In: Handbook of Defeasible Reasoning and Uncertainty Management Systems, vol. 5, p. 321. Springer (2000)
21. Kavvadias, D., Papadimitriou, C.H.: A linear programming approach to reasoning about probabilities. AMAI **1**, 189–205 (1990). https://doi.org/10.1007/BF01531078
22. McNaughton, R.: A theorem about infinite-valued sentential logic. J. Symb. Logic **16**, 1–13 (1951)
23. Mundici, D.: Satisfiability in many-valued sentential logic is NP-complete. Theor. Comput. Sci. **52**(1–2), 145–153 (1987)
24. Mundici, D.: A constructive proof of McNaughton's theorem in infinite-valued logics. J. Symb. Logic **59**(2), 596–602 (1994)
25. Mundici, D.: Bookmaking over infinite-valued events. Int. J. Approx. Reason. **43**(3), 223–240 (2006). https://doi.org/10.1016/j.ijar.2006.04.004
26. Mundici, D.: Advanced Łukasiewicz Calculus and MV-Algebras. Trends in Logic. Springer, Dordrecht (2011)
27. Nilsson, N.: Probabilistic logic. Artif. Intell. **28**(1), 71–87 (1986)
28. Papadimitriou, C., Steiglitz, K.: Combinatorial Optimization: Algorithms and Complexity. Dover, New York (1998)