Łukasiewicz Games: A Logic-Based Approach to Quantitative Strategic Interactions

ENRICO MARCHIONI and MICHAEL WOOLDRIDGE, Department of Computer Science, University of Oxford, UK

Boolean games provide a simple, compact, and theoretically attractive abstract model for studying multiagent interactions in settings where players will act strategically in an attempt to achieve individual goals. A standard critique of Boolean games, however, is that the strictly dichotomous nature of the preference relations induced by Boolean goals inevitably trivialises the nature of such strategic interactions: a player is assumed to be indifferent between all outcomes that satisfy her goal, and indifferent between all outcomes that do not satisfy her goal. While various proposals have been made to overcome this limitation, many of these proposals require the inclusion of nonlogical structures into games to capture nondichotomous preferences. In this article, we introduce Łukasiewicz games, which overcome this limitation by allowing goals to be specified using *Łukasiewicz logics*. By expressing goals as formulae of Łukasiewicz logics, we can express a much richer class of utility functions for players than is possible using classical Boolean logic: we can express every continuous piecewise linear polynomial function with rational coefficients over $[0, 1]^n$ as well as their finite-valued restrictions over $\{0, 1/k, \ldots, (k-1)/k, 1\}^n$. We thus obtain a representation of nondichotomous preference structures within a purely logical framework. After introducing the formal framework of Łukasiewicz games, we present a number of detailed worked examples to illustrate the framework, and then investigate some of their theoretical properties. In particular, we present a logical characterisation of the existence of Nash equilibria in finite and infinite Łukasiewicz games. We conclude by briefly discussing issues of computational complexity.

CCS Concepts: • **Theory of computation → Logic**; • **Computing methodologies → Artificial intelligence; Knowledge representation and reasoning; Multiagent systems**

Additional Key Words and Phrases: Logic, games, Łukasiewicz logics, knowledge representation, multiagent systems

ACM Reference Format:
Enrico Marchioni and Michael Wooldridge. 2015. Łukasiewicz games: A logic-based approach to quantitative strategic interactions. ACM Trans. Comput. Logic 16, 4, Article 33 (September 2015), 44 pages.
DOI: http://dx.doi.org/10.1145/2783436

33

1. INTRODUCTION

Boolean games provide a simple, compact, and elegant abstract mathematical model for studying multiagent interactions in settings where players will act strategically in an attempt to achieve individual goals [Harrenstein et al. 2001; Bonzon et al. 2006a;

This work greatly extends the results presented in Marchioni and Wooldridge [2014]. Marchioni acknowledges partial support of the Marie Curie Intra-European Fellowship "NAAMSI" (301625, FP7-PEOPLE-2011-IEF). Wooldridge was supported by the ERC under Advanced Investigator Grant "RACE" (291528). Both authors also acknowledge support from the EPSRC under grant EP/M009130/1 ("Combining Qualitative and Quantitative Reasoning for Logic-based Games").
Authors' addresses: E. Marchioni and M. Wooldridge, Department of Computer Science, University of Oxford, Oxford OX1 3QD, UK; emails: {enrico.marchioni, mjw}@cs.ox.ac.uk.

DOI: http://dx.doi.org/10.1145/2783436

Dunne et al. 2008; Grant et al. 2011]. In a Boolean game, each player i exercises unique control over a set of Boolean variables V_i, and will attempt to assign values for these variables in such a way as to satisfy an individual goal ϕ_i, expressed as a formula of propositional logic over the overall set of variables V. Strategic concerns in Boolean games arise from the fact that whether i's goal is in fact satisfied will depend in part on the choices made by other players, that is, the assignments that they make to the variables under their control. Players in Boolean games can be understood as representing nondeterministic computer programs, and the overall framework of Boolean games provides an elegant mathematical model through which to investigate issues of strategic interaction in multiagent systems.

A standard critique of Boolean games is that the binary nature of goals (satisfied or unsatisfied) inevitably trivialises the nature of strategic interactions. For example, players are assumed to be indifferent between all outcomes that satisfy their goal, and are indifferent between all outcomes that do not satisfy their goal. This assumption is clearly unrealistic for many situations, a concern that led researchers to extend the original Boolean games model with costs, leading to richer and more realistic preference structures for agents [Wooldridge et al. 2013]. While these refinements make it possible to model much richer types of interaction, the inherently dichotomous nature of preferences in Boolean games remains one of their most debated features, and the work in the present article is directly motivated by this limitation.

Specifically, we introduce *Łukasiewicz games*, which provide an alternative mechanism for going beyond dichotomous preferences in Boolean games. In *Łukasiewicz games*, we allow goals to be specified as formulae of *Łukasiewicz logics*, which form a family of both infinite- and finite-valued systems. The rationale for using Łukasiewicz logics is that these logics allow us to represent much richer utility functions than is possible using two-valued logic, while at the same time staying within the purely logical framework offered by Boolean games. In particular, if we use Łukasiewicz logics over n variables to represent player goals, then by the McNaughton Theorem (and its variants), we can express every continuous piecewise linear polynomial function with integer (and rational) coefficients over $[0, 1]^n$ as well as their finite-valued restrictions over

$$\left\{0, \frac{1}{k}, \ldots, \frac{k-1}{k}, 1\right\}^n$$

(see, e.g., Cignoli et al. [2000] and Gerla [2001]). Thus, we argue, Łukasiewicz logics provide a natural, compact, formally well-defined and expressive logical representation language for payoff functions, allowing much richer utility functions, and hence preference structures, than is easily possible in standard Boolean games.

The remainder of this article is structured as follows:

—We begin, in the following section, by discussing the motivation for our work and related work in more detail.

—In Section 3, we introduce the key technical notions that are used throughout the article. In particular, since Łukasiewicz logics are not as widely known as Classical logic, we provide a complete self-contained introduction to the five variations of Łukasiewicz logic that will be our key formal systems in what follows: infinite-valued Łukasiewicz logic, Rational Pavelka logic, Rational Łukasiewicz logic, finite-valued Łukasiewicz logics, and finite-valued Łukasiewicz logics with constants. In addition, we present some technical results relating to these logics that are used to prove results presented in the main text.

—Section 4 then introduces the formal framework of Łukasiewicz games, and in Section 5, we present a number of detailed worked examples, which illustrate how a range of strategic scenarios can be formalised within this framework. In particular, we argue, these scenarios cannot naturally be formalised using conventional Boolean games.

—In Section 6 we establish some basic properties of Łukasiewicz games and show that any game can be translated into a normalised form, which preserves the Nash equilibria of the original game.

—Sections 7 and 8 study finite and infinite Łukasiewicz games, respectively. In both cases, our key results relate to the presentation of a logical characterisation for the existence of a Nash equilibrium in the respective game. In particular, we show that for every Łukasiewicz game $\mathcal{G}$ there exists a formula whose satisfiability set coincides with the set of equilibria of $\mathcal{G}$. In addition, Section 9 presents some results on the existence of equilibria that are specific to infinite Łukasiewicz games.

—Finally, in Section 10, we investigate the complexity of the key decision problems for Łukasiewicz games.

2. MOTIVATION AND BACKGROUND

The use of game theoretic concepts has a long history in computing, but the past decade has been witness to an unprecedented growth of interest in the subject (see, e.g., Nisan et al. [2007]). This explosion of interest has been largely spurred by the realisation that in order to understand the behaviour of systems such as internet-based auction sites, it is necessary to take into account the fact that the participants in such systems will act strategically in pursuit of their personal goals and preferences. Game theory provides the basic mathematical framework through which self-interested strategic behavior can be modelled and analysed.[1]

Perhaps the most widely applied and best-known model used in game theory is that of a *strategic-form noncooperative game* (hereafter just "strategic-form game"), and indeed this model underpins those that we introduce in the present article. We will therefore begin by briefly recalling this model and the key concepts used to analyse it. A strategic-form game is populated by a finite and nonempty set P of agents—the players of the game. The task of a player is simply to choose one from a set S_i of *strategies*. When every player has selected a strategy, then the outcome is a *strategy profile*, $\vec{s} = (s_1, \ldots, s_n)$. Players have preferences over outcomes, given by *utility functions*

$$u_i : \mathsf{S}_1 \times \cdots \times \mathsf{S}_n \to \mathbb{R},$$

which for every possible combination of choices $\vec{s} \in \mathsf{S}_1 \times \cdots \times \mathsf{S}_n$ give a real number $u_i(\vec{s})$ representing the *utility* or *payoff* that player i would receive if the players made choices resulting in the outcome $\vec{s}$. Utility functions represent the preferences of players in the following way: player i strictly prefers outcome $\vec{s}_1$ over outcome $\vec{s}_2$ if $u_i(\vec{s}_1) > u_i(\vec{s}_2)$.

Definition 2.1 (*Strategic-Form Game*). A *strategic-form game* is given by a structure:

$$\mathcal{G} = \langle \mathsf{P}, \{\mathsf{S}_i\}_{i\in\mathsf{P}}, \{u_i\}_{i\in\mathsf{P}} \rangle,$$

where

(1) $\mathsf{P} = \{P_1, \ldots, P_n\}$ is a finite and nonempty set of *players*;
(2) S_i is the nonempty set of strategies available to player i; and
(3) $u_i : \mathsf{S}_1 \times \cdots \times \mathsf{S}_n \to \mathbb{R}$ is the utility function for player i.

Players seek to maximize the utility they receive in an outcome. However, since the utility player i will receive from an outcome depends not just on the choice made by

[1]The game theory literature is large, and it is difficult to identify a definitive reference. One comprehensive and authoritative contemporary textbook is Maschler et al. [2013], which also provides extensive references to the original game theory research literature.

player i, but on the choices made by others, then when making her choice of which strategy to play, i must reason strategically, taking into account the preferences of other players and the fact that they too will reason strategically. Game theory proposes a number of *solution concepts* for strategic-form games, which characterise the possible rational outcomes of a game under the assumption that players reason strategically. For the purposes of this article, the most relevant solution concept is that of a (*pure strategy*) *Nash equilibrium*. To define this, we need a little more notation. Where $\vec{s} = (s_1, \ldots, s_i, \ldots, s_n)$ is a strategy profile and $s_i' \neq s_i \in \mathsf{S}_i$, then by $(s_i', \vec{s}_{-i})$ we mean the strategy profile that is the same as $\vec{s}$ except that player i chooses s_i':

$$(s_i', \vec{s}_{-i}) = (s_1, \ldots, s_i', \ldots, s_n).$$

We say $\vec{s}$ is a *pure strategy Nash equilibrium* (hereafter just "Nash equilibrium") if for all players $i \in \mathsf{P}$ and for all strategies $s_i' \neq s_i \in \mathsf{S}_i$ we have $u_i(\vec{s}) \geq u_i(s_i', \vec{s}_{-i})$. Thus, the fact that $\vec{s}$ is a Nash equilibrium means that no player can benefit by unilaterally changing their choice.

If we consider strategic-form games from a computational perspective, then many natural problems suggest themselves (see, e.g., Nisan et al. [2007] for a detailed survey). The problem to which we address ourselves in the present article is that of *representing* games. If we examine the definition of games given above, we see that utility functions take as input a strategy profile and return as output a real number. Assuming there are n players in total ($|\mathsf{P}| = n$), and each player has m strategies to choose between ($|\mathsf{S}_i| = m$), then the domain of a utility function $u_i(\cdots)$ will be of size m^n. This implies that representing such utility functions *explicitly*, by listing the value of the function for every possible input, will be utterly infeasible in general. This motivates the development of *succinct representation schemes* for games. Observe that we cannot hope for a representation scheme that will allow us to represent all games compactly; a reasonable goal, however, is to seek representation schemes that have large space requirements in the *worst* case, but which admit compact representations for cases of interest. Of course, we cannot divorce the development of succinct representation schemes for games from the complexity of computing solutions to these games: as a general rule, the more compact a representation scheme for games we devise, the harder will be the associated computational problems for these games. An important research theme in computational game theory is therefore the development of representation schemes for games that admit efficient algorithms for the relevant computational problems (in particular, computing solution concepts), and to map out the frontier between tractable and intractable cases.

One representation for games that has received considerable attention within the multiagent systems research community is that of *Boolean games* [Harrenstein et al. 2001; Bonzon et al. 2006a; Dunne et al. 2008; Grant et al. 2011]. The basic idea of a Boolean game is that each player in the game is associated with a set of Boolean variables V_i, and the strategies available to player i correspond to the set of all possible Boolean assignments that player i can make to her variables. Since a player i controlling variable set V_i will have $2^{|\mathsf{V}_i|}$ strategies available to choose from, Boolean games have a compact representation for the set of strategies available to an agent. When every player has chosen a valuation for their variables, the result will be a propositional valuation for the total set of variables $\mathsf{V} = \bigcup_{i \in \mathsf{P}} \mathsf{V}_i$. Preferences in standard Boolean games are *dichotomous*: every agent is associated with an individual goal ϕ_i, represented as a propositional logic formula over V, and will attempt to assign values for the variables V_i under her control so as to satisfy ϕ_i. Strategic concerns in Boolean games arise from the fact that whether i's goal is in fact satisfied will depend in part on the choices made by other players, that is, the valuations that they make to their variables. A player will be assumed to strictly prefer an outcome that satisfies her goal over an outcome that does not, but is indifferent between two outcomes that satisfy her goal or fail to satisfy her goal. While in the worst case we might need to specify a player's goal using a formula of size exponential in the number of Boolean variables V, the use of logic to specify goals frequently permits a *compact* representation.

We formally define Boolean games and the strategic-form games corresponding to them as follows:

Definition 2.2 (*Boolean Games*). A *Boolean game*, $\mathcal{B}$, is given by a structure

$$\mathcal{B} = \langle \mathsf{P}, \mathsf{V}, \{\mathsf{V}_i\}_{i\in\mathsf{P}}, \{\phi_i\}_{i\in\mathsf{P}}\rangle,$$

where

(1) $\mathsf{P} = \{P_1, \ldots, P_n\}$ is a finite set of *players*.
(2) $\mathsf{V} = \{p_1, \ldots, p_m\}$ is a finite set of propositional variables.
(3) $\mathsf{V}_i \subseteq \mathsf{V}$ is the set of propositional variables under control of player P_i, so that the sets V_i form a partition of V.
(4) ϕ_i is a propositional formula over variables V representing the *goal* of player i.

A strategy for player $i \in \mathsf{P}$ is a Boolean valuation for the variables under the control of i, that is, a function $s_i : \mathsf{V}_i \to \{0, 1\}$. A strategy profile is a tuple $\vec{s} = (s_1, \ldots, s_n)$, which thus gives a valuation for the overall set of variables V. We write $\vec{s} \models \phi$ to mean that propositional formula ϕ is satisfied under the valuation corresponding to $\vec{s}$.

The Boolean game $\mathcal{B} = \langle \mathsf{P}, \mathsf{V}, (\mathsf{V}_i)_{i\in\mathsf{P}}, (\phi_i)_{i\in\mathsf{P}}\rangle$ then induces a strategic-form game $\mathcal{G}_{\mathcal{B}} = \langle \mathsf{P}, (\mathsf{S}_i)_{i\in\mathsf{P}}, (u_i)_{i\in\mathsf{P}}\rangle$ as follows:

(1) The player set P is the same.
(2) For each player $i \in \mathsf{P}$ we have

$$\mathsf{S}_i = \{s_i \mid s_i : \mathsf{V}_i \to \{0, 1\}\}.$$

(3) For each player $i \in \mathsf{P}$ and for each outcome $\vec{s}$ we have

$$u_i(\vec{s}) = \begin{cases} 1 & \text{if } \vec{s} \models \phi_i \\ 0 & \text{otherwise.} \end{cases}$$

With this correspondence in place, we can apply the standard game theoretic solution concepts for strategic-form games to Boolean games. The two most important computational problems associated with Boolean games are Membership (the task of checking whether a given outcome is a Nash equilibrium of a given game), and Nonemptiness (the task of checking whether a given game has any Nash equilibrium). Membership is co-NP-complete, while Nonemptiness is Σ_2^p-complete [Bonzon et al. 2006a].

Boolean games represent an important model for multiagent systems research for at least two reasons. First, players in Boolean games can naturally be understood as an abstract model of nondeterministic computer programs. The choices available to a player correspond to the assignments of truth or falsity that the player can make to the variables under her control. The model does not specify *which* choice a player will make—hence the nondeterminism in the model. Given their logically specified preferences, the idea is that a player in a game should resolve their nondeterminism *strategically*, attempting to choose values for their variables so as to satisfy their goal. We can understand the players as generating a computation as they assign values to their variables. Thus, Boolean games provide a high-level model of multiagent systems in which players have logically specified goals. As an aside, we note that the model has something of the flavour of the Reactive Modules system modelling language that is used in several model checkers [Alur and Henzinger 1999]. This nondeterministic language specifies the choices available to agents through rule-like guarded commands. Boolean games can naturally be captured within this language; the key difference is that in Boolean games, a player can make an *arbitrary* assignment of values to variables, while in REACTIVE MODULES, a player can only make an assignment corresponding to an enabled guarded command.

Second, the use of logic to define the goals of players is consistent both with the standard model of planning in artificial intelligence, where goals for agents are normally specified as logical formulae [Ghallab et al. 2004], and also, of course, with the mainstream computer science approach to the specification of computer systems. Overall, the framework of Boolean games provides a simple, elegant, compact, and powerful mathematical model through which to investigate issues of strategic interaction in multiagent systems, which has a natural computational interpretation.

It is also worth noting that, although their primary interest is as an abstract theoretical model, Boolean games have proved to have an increasing number of applications. For example, [Levit et al. 2013a] uses Boolean games for modelling recharging schemes for electric vehicles, while [Levit et al. 2013b] uses Boolean games to model traffic signalling systems.

However, the basic model of Boolean games that we described above has an important limitation: it is restricted to scenarios in which player preferences are strictly dichotomous. Thus, players are either satisfied or unsatisfied with an outcome. Various extensions to the basic Boolean games model have been proposed in an attempt to overcome this limitation. For example, associating costs with assignments induces pseudodichotomous preference structures: agents always prefer to achieve their goals rather than otherwise, but secondarily would prefer to minimise costs (see, e.g., Wooldridge et al. [2013]).

Some proposals to enrich logical models by adding numerical values to represent payoffs and their maximisation have also been put forward (see Bulling and Goranko [2013]). The logical systems defined are essentially expansions of existing modal logics for multiplayer games, obtained by adding arithmetical constraints to the language. These logics make it possible to express the fact that certain (coalitions of) agents bring about an outcome with a guaranteed payoff, represented by a classical modal two-valued formula.

Other alternatives considered in the literature include the use of weighted logical goal formulae [Mavronicolas et al. 2007]. Also worth mentioning here is the work of Bonzon et al. [2006b], who investigate a number of possible mechanisms for extending Boolean games in order to capture nondichotomous preferences. For example, the authors consider representing preferences via ordered lists of formulae, via sets of formulae (leading to "distributed evaluation games," where a player prefers to satisfy larger sets of goals), and by augmenting Boolean games with a CP-net representation [Boutilier et al. 2004]. All of these approaches have advantages, but note that all of them can be understood as augmenting a logical representation with a nonlogical representation in order to obtain a richer preference structure. In short, they involve moving away from the attractive purely logical structure offered by Boolean games in which players simply have a logically specified goal.

Finally, there have also been many attempts to formulate preference logics, typically as variants of modal logic. However, it is generally acknowledged that existing proposals have limitations [van Benthem 2014].

The aim of the present article is to develop a representation scheme for games that retains the *purely logical structure* of standard Boolean games, but allows for the definition of much richer utility functions than is possible in those games. That is, we want to be able to represent nondichotomous preference relations in a purely logical framework, without tacking on additional nonlogical constructions. In other words, while it is certainly possible to augment classical logic representations with features for representing nondichotomous preferences, we believe it is useful and valuable instead to attempt to develop purely logical representations for nondichotomous preferences. Our proposal in this article represents one such approach. We do not argue that the approach is necessarily superior to the alternatives presented above but because it is based on a purely logical representation, it retains the attractive features of such a representation (precise logical semantics, etc.).

The basic idea underpinning our approach is to use Łukasiewicz logics as specification languages for player's goals. We begin our presentation, in the following section, with an overview of Łukasiewicz logics, introducing the variations of Łukasiewicz logic that we use, and presenting some key results relating to those systems that we make use of later in the article.

3. PRELIMINARY DEFINITIONS

Since Łukasiewicz logics are fundamental to our present work, but are not as widely known as the classical two-valued logic that underpins conventional Boolean games, we begin by introducing the concepts of Łukasiewicz logics, their related class of functions, and the theories of their related semantic structures that will be extensively used in the remainder of the article. Notice that we limit our presentation to the notions that are needed to understand our treatment of Łukasiewicz games. The interested reader can find an extensive treatment of Łukasiewicz logics and their semantics in Cignoli et al. [2000], Di Nola and Leustean [2011], and Mundici [2011].

When we talk about Łukasiewicz logics, we refer to a class of systems that includes both finite- and infinite-valued logics. The main system in this class is the infinite-valued Łukasiewicz logic Ł$_\infty$, which is the logic of the class of continuous piecewise linear polynomial functions with integer coefficients over $[0,1]^n$, called McNaughton functions. Finite-valued Łukasiewicz logics Ł$_k$ are the systems obtained by restricting McNaughton functions over the set

$$L_k = \left\{0, \frac{1}{k}, \dots, \frac{k-1}{k}, 1\right\},$$

for $k \geq 1$. Ł$_\infty$ and Ł$_k$ share the same language and are the systems historically defined as Łukasiewicz logics.

Here, we include in this class also expansions of Ł$_\infty$ and Ł$_k$ that make it possible to define constant functions for every rational number in $[0,1]$ and every element of L_k, respectively. These logics are

(1) the Rational Pavelka logic RPŁ$_\infty$ [Pavelka 1979; Hájek 1998], which is the logic of McNaughton functions with rational truth constants;
(2) the Rational Łukasiewicz logic RŁ$_\infty$ [Gerla 2001], which is the logic of rational McNaughton functions, that is, continuous piecewise linear polynomial functions with rational coefficients over $[0,1]^n$; and
(3) the finite-valued logics Ł$_k^c$, which are the logics of restrictions of McNaughton functions over $(L_k)^n$ with constants from L_k.

The reason for including also these logics rather than just the basic Łukasiewicz systems is that we want to take advantage of their greater expressive power to specify a wider class of games while retaining a common logical framework. We can then offer a general study of Łukasiewicz games as a whole by exploiting the common properties shared by the logics of this class, which can be seen as the logics of continuous piecewise linear function with rational coefficients and their finite-valued restrictions. Table I offers an overview of the logics we introduce, along with the related class of functions, semantic structures, and their first-order theory.

Table I. Łukasiewicz Logics, Their Associated Class of Functions, Their Semantic structures, and First-Order Theory

Logic	Class of Functions	Semantic Structure	Theory
Infinite-valued Łukasiewicz logic $Ł_\infty$	*McNaughton functions*: continuous piecewise linear functions with integer coefficients over $[0, 1]$	Standard Real MV-algebra: $\mathbb{MV}_\infty = \langle [0, 1], \oplus, \neg, 0 \rangle$	$\mathrm{Th}(\mathbb{MV}_\infty)$
Rational Pavelka logic $RPŁ_\infty$	McNaughton functions with rational constants	Standard Real MV-algebra with rational constants: $\mathbb{MV}^c_\infty = \langle [0, 1], \oplus, \neg, \{c\}_{c \in \mathbb{Q} \cap [0,1]} \rangle$	$\mathrm{Th}(\mathbb{MV}^c_\infty)$
Rational Łukasiewicz logic $RŁ_\infty$	*Rational McNaughton functions*: continuous piecewise linear functions with rational coefficients over $[0, 1]$	Standard Real DMV-algebra: $\mathbb{DMV}_\infty = \langle [0, 1], \oplus, \neg, \{\delta_n\}_{n>1}, 0 \rangle$	$\mathrm{Th}(\mathbb{DMV}_\infty)$
Finite-valued Łukasiewicz logics $Ł_k$	Restrictions of McNaughton functions over $L_k = \left\{0, \frac{1}{k}, \ldots, \frac{k-1}{k}, 1\right\}$	Standard finite MV-algebra: $\mathbb{MV}_k = \langle L_k, \oplus, \neg, 0 \rangle$	$\mathrm{Th}(\mathbb{MV}_k)$
Finite-valued Łukasiewicz logics with constants $Ł^c_k$	All functions $f : (L_k)^n \to L_k$	Standard finite MV-algebra with constants: $\mathbb{MV}^c_k = \left\langle L_k, \oplus, \neg, 0, \frac{1}{k}, \ldots, \frac{k-1}{k}, 1 \right\rangle$	$\mathrm{Th}(\mathbb{MV}^c_k)$

3.1. Łukasiewicz Logics

We begin by defining *infinite-valued Łukasiewicz logic* $Ł_\infty$. The language of $Ł_\infty$ is built from a countable set of variables $V = \{p_1, p_2, \ldots\}$, the binary connective "$\to$", and the truth constant $\overline{0}$ (for falsity). Further connectives are defined as follows:

$$
\begin{array}{rcl}
\neg\phi & \text{is} & \phi \to \bar{0}, \\
\overline{1} & \text{is} & \neg\overline{0}, \\
\phi \odot \psi & \text{is} & \neg(\phi \to \neg\psi), \\
\phi \oplus \psi & \text{is} & \neg(\neg\phi \odot \neg\psi), \\
\phi \ominus \psi & \text{is} & \phi \odot \neg\psi, \\
\phi \wedge \psi & \text{is} & \phi \odot (\phi \to \psi), \\
\phi \vee \psi & \text{is} & ((\phi \to \psi) \to \psi), \\
\phi \leftrightarrow \psi & \text{is} & (\phi \to \psi) \odot (\psi \to \phi), \\
d(\phi, \psi) & \text{is} & \neg(\phi \leftrightarrow \psi).
\end{array}
$$

We often write $n\phi$ as an abbreviation for $\underbrace{\phi \oplus \cdots \oplus \phi}_{n}$, with $n > 1$.

A *valuation*, e, is a mapping $e : \mathsf{V} \to [0,1]$, which assigns to all propositional variables a value from the real unit interval. The semantics of Łukasiewicz logic is then defined, with a small abuse of notation, by extending the valuation e to complex formulae. Although strictly speaking we only need state the rule for the "$\to$" connective (as we can define the remaining connectives in terms of this connective and $\overline{0}$), we present the complete ruleset in the interest of clarity:

$$
\begin{aligned}
e(\overline{0}) &= 0,\\
e(\phi \to \psi) &= \min(1 - e(\phi) + e(\psi), 1),\\
e(\neg\phi) &= 1 - e(\phi),\\
e(\overline{1}) &= 1,\\
e(\phi \odot \psi) &= \max(0, e(\phi) + e(\psi) - 1),\\
e(\phi \oplus \psi) &= \min(1, e(\phi) + e(\psi)),\\
e(\phi \ominus \psi) &= \max(0, e(\phi) - e(\psi))\\
e(\phi \wedge \psi) &= \min(e(\phi), e(\psi)),\\
e(\phi \vee \psi) &= \max(e(\phi), e(\psi)),\\
e(\phi \leftrightarrow \psi) &= 1 - |e(\phi) - e(\psi)|,\\
e(d(\phi, \psi)) &= |e(\phi) - e(\psi)|.
\end{aligned}
$$

We say that a formula ϕ is *satisfiable* if there exists a valuation e such that $e(\phi) = 1$. Given a formula $\phi(p_1, \ldots, p_n)$, $Sat(\phi(p_1, \ldots, p_n))$ denotes the *satisfiability set* of $\phi(p_1, \ldots, p_n)$, that is,

$$
\begin{aligned}
sat(\phi(p_1, \ldots, p_n)) = \big\{&(a_1, \ldots, a_n) \in [0,1]^n \,\big|\, e(p_1) = a_1, \ldots, e(p_n) = a_n,\\
&\text{and } e(\phi(p_1, \ldots, p_n)) = 1\big\}.
\end{aligned}
$$

A valuation e is a model for a theory Γ, that is, a set of formulae, if it satisfies every $\psi \in \Gamma$. We call a formula ϕ a *tautology*, if $e(\phi) = 1$ under every valuation e. Note that these notions of satisfiabilty, satisfiability set, and tautology will be used also for the rest of the logics introduced next with the obvious modifications.

Rational Pavelka logic RPŁ$_\infty$ is defined from Ł$_\infty$ by adding to the language a constant $\overline{c}$ for every rational in $[0,1]$. Each constant c is naturally interpreted as its corresponding rational number, that is, $e(\overline{c}) = c$, for all $c \in \mathbb{Q} \cap [0,1]$.

Rational Łukasiewicz logic RŁ$_\infty$ is obtained by expanding the language of Ł$_\infty$ with the unary connectives δ_n for each natural $n > 1$. Each connective δ_n functions as a *divisibility* operator. It has the following interpretation, for all valuations e into $[0,1]$:

$$e(\delta_n\phi) = \frac{e(\phi)}{n}.$$

In RŁ$_\infty$, it is possible to define new constants whose interpretation corresponds to each rational number in $[0,1]$. For example,

$$\tfrac{1}{n} \text{ is definable as } \delta_n(\neg\overline{0}), \qquad \text{while} \qquad \tfrac{m}{n} \text{ is definable as } m(\delta_n(\neg\overline{0})).$$

Consequently, while RPŁ$_\infty$ is an expansion of Ł$_\infty$, RŁ$_\infty$ is an expansion of RPŁ$_\infty$.

Finite-valued Łukasiewicz logics Ł$_k$, one for each $k \geq 1$, share the same language as infinite-valued Łukasiewicz logic Ł$_\infty$. In such logics, it is assumed that the domain of all valuations is a set of the following form:

$$L_k = \left\{0, \frac{1}{k}, \ldots, \frac{k-1}{k}, 1\right\}.$$

The interpretation of the connectives is the same as the one defined for Ł$_\infty$ but restricted to L_k, which is closed under all Ł$_\infty$-operations. Notice that when $k = 1$, Ł$_1$ simply corresponds to classical Boolean logic.

Finite-valued Łukasiewicz logics with constants $Ł_k^c$ are obtained from each $Ł_k$ by expanding the language with constants $\overline{c}$ for every value $c \in L_k$. We assume that valuation functions e interpret such constants in the natural way: $e(\overline{c}) = c$.

In all finite-valued Łukasiewicz logics (with or without constants) it is possible to define the unary connective Δ as follows:

$$\Delta\phi \quad \text{is} \quad \neg(k(\neg\phi)).$$

The semantic interpretation of Δ over L_k is

$$e(\Delta\phi) = \begin{cases} 1 & e(\phi) = 1 \\ 0 & \text{otherwise} \end{cases}.$$

Δ is a rather important connective since its application always outputs Boolean values, that is, 0 and 1. We will make explicit use of this fact when we provide a characterisation of the existence of equilibria in finite Łukasiewicz games in Section 7.

NOTATION 1. *We will simply refer to* $Ł_\infty$, $RPŁ_\infty$, *and* $RŁ_\infty$ *as* infinite Łukasiewicz logics, *while we will refer to* $Ł_k$ *and* $Ł_k^c$ *as* finite Łukasiewicz logics.

Let L be any of the logics introduced above. L is the logic of all tautologies in the L-language, that is, of all the L-formulae ϕ such that $e(\phi) = 1$ for every e in the related class of valuations.[2] So, as an example, $Ł_k$, for a fixed k, is the logic of all formulae in the language of Łukasiewicz logic that are given value 1 under all valuations into L_k.

As mentioned above each Łukasiewicz logic L can be shown to be the logic of a special class of functions. Given an L-formula $\phi(p_1, \ldots, p_n)$ we can define a real-valued function $f_\phi(x_1, \ldots, x_n)$ so that for each assignment e to the propositional variables $p_1, \ldots, p_n$,

$$f_\phi(e(p_1), \ldots, e(p_n)) = e(\phi(p_1, \ldots, p_n)).$$

The formula $\phi(p_1, \ldots, p_n)$ is said to *realise* $f_\phi(x_1, \ldots, x_n)$.[3]

The key notion in describing the functions associated with Łukasiewicz formulae is that of a McNaughton function.

Definition 3.1 (*McNaughton Function*). A function $f : [0, 1]^n \to [0, 1]$ is called a *McNaughton function* over $[0, 1]^n$ if and only if it satisfies the following conditions:

(1) f is continuous with respect to the natural topology of $[0, 1]^n$;
(2) there are linear polynomials $\mathsf{p}_1, \ldots, \mathsf{p}_k$ with integer coefficients,

$$\mathsf{p}_i(x_1, \ldots, x_n) = b_i + m_{i1}x_1 + \cdots + m_{in}x_n$$

$(b_i, m_{it} \in \mathbb{Z})$, such that for each point $\vec{y} = (y_1, \ldots, y_n) \in [0, 1]^n$ there is an index $j \in \{1, \ldots, k\}$ with $f(\vec{y}) = \mathsf{p}_j(\vec{y})$.

Since every function f realised by a $Ł_\infty$-formula is obtained as a composition of $\to$ and the constant 0, it is easy to see that f is a McNaughton function. The McNaughton Theorem shows that the converse is also true, that is, infinite-valued Łukasiewicz logic is the logic of continuous piecewise linear polynomial functions with integer coefficients over the unit cube $[0, 1]^n$.

[2]All the Łukasiewicz logics we have introduced have a specific axiomatisation, and their axiom systems are complete with respect to the semantics given here. Giving the precise axiomatisation is beyond the scope of this work. All the details can be found in Cignoli et al. [2000], Gerla [2001], Hájek [1998], and Esteva et al. [2011].

[3]Notice that whenever variables $p_1, \ldots, p_n$ are explicitly mentioned in a formula, that is, $\phi(p_1, \ldots, p_n)$, we assume they do actually all occur in ϕ. Similarly, for its associated function $f_\phi(x_1, \ldots, x_n)$ we assume that all $x_1, \ldots, x_n$ occur and so f_ϕ is defined over either $[0, 1]^n$ or $(L_k)^n$.

THEOREM 3.2 ([MCNAUGHTON 1951]). *A function $f : [0, 1]^n \to [0, 1]$ is a McNaughton function if and only if it is the function realised by some Ł$_\infty$-formula $\phi(p_1, \dots, p_n)$.*[4]

McNaughton functions are strongly connected with rational polyhedra. For $0 \le m \le n$, an m-simplex[5] in $\mathbb{R}^n$ is the convex hull $X = \mathrm{conv}(x_0, \dots, x_m)$ of $m+1$ affinely independent points in the n-dimensional Euclidean space $\mathbb{R}^n$. The vertices $x_0, \dots, x_m$ are uniquely determined by $\mathrm{conv}(x_0, \dots, x_m)$. An m-simplex X is called *rational* if its vertices are all rational points, that is, $x_0, \dots, x_m \in \mathbb{Q}$. A *polyhedron* P is the union of finitely many simplexes X_i in $\mathbb{R}^n$. If all the simplexes X_i are rational, P is called a *rational polyhedron*.

The next lemma shows that for every rational polyhedron $X \subseteq [0, 1]^n$ there always exists a McNaughton function $f : [0, 1]^n \to [0, 1]$ whose zero-set coincides with X.

LEMMA 3.3 ([MUNDICI 2011]). *Let $\emptyset \neq X \subseteq [0, 1]^n$. Then the following conditions are equivalent:*

(1) $X = \{(a_1, \dots, a_n) \mid f(a_1, \dots, a_n) = 0\}$ *for some McNaughton function* $f : [0, 1]^n \to [0, 1]$.
(2) $X = Sat(\phi(p_1, \dots, p_n))$ *for some Ł$_\infty$-formula* $\phi(p_1, \dots, p_n)$.
(3) *X is a rational polyhedron.*

The above lemma will be of fundamental importance in our study of infinite Łukasiewicz games in Section 8. We will make use of the fact that for every rational polyhedron X there exists a formula of infinite-valued Łukasiewicz logic whose satisfiability set corresponds to X. This is derived from the fact that X is the zero-set of some McNaughton function f, and consequently, it is the one-set of the function $1 - f$.

It is easy to see that RPŁ$_\infty$ is the logic of all functions obtained by composition of McNaughton functions and rational constant functions with the Łukasiewicz operations. In other words, every function $f : [0, 1]^n \to [0, 1]$ realised by a RPŁ$_\infty$-formula is such that, for all $(x_1, \dots, x_n) \in [0, 1]^n$,

$$f(x_1, \dots, x_n) = g(x_1, \dots, x_n, c_1, \dots, c_m),$$

where

$$g(x_1, \dots, x_n, x'_1, \dots, x'_m)$$

is a McNaughton function $g : [0, 1]^{n+m} \to [0, 1]$ and $c_1, \dots, c_m \in \mathbb{Q} \cap [0, 1]$.

RŁ$_\infty$ is the logic of rational McNaughton functions.

Definition 3.4 (*Rational McNaughton Function*). A continuous function $f : [0, 1]^n \to [0, 1]$ is called a *rational McNaughton function* if there are linear polynomials $\mathsf{p}_1, \dots, \mathsf{p}_k$ with rational coefficients,

$$\mathsf{p}_i(x_1, \dots, x_n) = b_i + m_{i1}x_1 + \cdots + m_{in}x_n$$

$(b_i, m_{it} \in \mathbb{Q})$, such that for each point $\vec{y} = (y_1, \dots, y_n) \in [0, 1]^n$ there is an index $j \in \{1, \dots, k\}$ with $f(\vec{y}) = \mathsf{p}_j(\vec{y})$.

THEOREM 3.5 ([GERLA 2001]). *A function $f : [0, 1]^n \to [0, 1]$ is a rational McNaughton function if and only if it is the function realised by some* RŁ$_\infty$*-formula $\phi(p_1, \dots, p_n)$.*[6]

Notice that Lemma 3.3 clearly holds also for McNaughton functions with constants and for rational McNaughton functions.

[4]See also Mundici [1994], Cignoli et al. [2000], and Aguzzoli et al. [2011].
[5]The definitions that appear in this paragraph are taken from Stallings [1967].
[6]See also Aguzzoli et al. [2011].

As for finite-valued Łukasiewicz logics $Ł_k$, it is easy to see that the functions associated with their formulae are just the restrictions of McNaughton functions over L_k. So, for instance, the function associated with a formula $\phi(p_1, \ldots, p_n)$ of $Ł_k$ is obtained by taking the function $f_{\phi(p_1,\ldots,p_n)}$ over $[0, 1]^n$ restricted to $(L_k)^n$.

In the case of finite-valued Łukasiewicz logics with constants $Ł^c_k$, the functions defined by a formula are combinations of the restrictions of McNaughton functions over $(L_k)^n$ and the constant functions for each $c \in L_k$. Notice that the class of functions definable by $Ł^c_k$ formulae coincides with the class of all functions $f : (L_k)^n \to L_k$, for every $n \geq 0$. In fact, for any function $f : (L_k)^n \to L_k$, we can define a formula realising f as follows:

$$\bigvee_{c_1,\ldots,c_n \in (L_k)^n} \left(\left(\bigwedge_{i=1}^{n} \Delta(p_i \leftrightarrow \overline{c}_i) \right) \wedge \overline{f(c_1, \ldots, c_n)} \right).$$

3.2. MV-Algebras and Their First-Order Theory

Boolean algebras provide the algebraic semantics for Classical logic; in Łukasiewicz logics, this role is played by MV-*algebras*. As Łukasiewicz logics generalise Classical logic, MV-algebras are more general structures than Boolean algebras. In this section, we introduce some basic notions about the real-valued structures that provide the standard semantics for Łukasiewicz logics. We simply refer to these structures as *standard* MV-*algebras* and, in the rest of the article, we will often use formulae of their first-order theory to express game-theoretic properties, such as the existence of equilibria.

Here a clarification is in order. Notice that in this work we do not deal with MV-algebras as the algebraic semantics of Łukasiewicz logics, but rather as the first-order structures of their truth values. This distinction is subtle, but it is the reason why we do not offer here a survey of the general theory of MV-algebras and simply provide some basic specific notions that will play a part in our study of Łukasiewicz games.[7] Once again, the interested reader can find an exhaustive treatment of the subject in Cignoli et al. [2000], Di Nola and Leustean [2011], and Mundici [2011].

The *standard real* MV-*algebra* $\mathbb{MV}_\infty$ is the structure

$$\mathbb{MV}_\infty = \langle [0, 1], \oplus, \neg, 0 \rangle,$$

where, for all $x, y \in [0, 1]$,

$$x \oplus y = \min(x + y, 1), \qquad \neg x = 1 - x.$$

The *standard real* MV-*algebra with constants* $\mathbb{MV}^c_\infty$ is the structure

$$\mathbb{MV}^c_\infty = \big\langle [0, 1], \oplus, \neg, \{c\}_{c \in \mathbb{Q} \cap [0,1]} \big\rangle,$$

where $\langle [0, 1], \oplus, \neg, 0 \rangle$ is the standard real MV-algebra.

The *standard real* DMV-*algebra* $\mathbb{DMV}_\infty$ is the structure[8]

$$\mathbb{DMV}_\infty = \big\langle [0, 1], \oplus, \neg, \{\delta_n\}_{n>1}, 0 \big\rangle,$$

where $\langle [0, 1], \oplus, \neg, 0 \rangle$ is the standard real MV-algebra, and for all $x \in [0, 1]$ and natural $n > 1$,

$$\delta_n x = \frac{1}{n} x.$$

[7]Notice, as an example, that we do not provide the general abstract definition of an MV-algebra, but only define the particular structures over the real numbers whose theory will be needed in the game-theoretic setting.

[8]The "D" in "DMV-algebra" stands for "Divisible." This is because the operators δ_n force divisibility for the set of elements of the domain over MV-algebras in general (see Gerla [2001]).

The *standard finite* MV-*algebra*

$$\mathbb{MV}_k = \langle L_k, \oplus, \neg, 0\rangle$$

is simply defined by taking the restriction of $\mathbb{MV}_\infty$ over the set

$$L_k = \left\{0, \frac{1}{k}, \ldots, \frac{k-1}{k}, 1\right\},$$

Finally, the *standard finite* MV-*algebra with constants*

$$\mathbb{MV}^c_k = \left\langle L_k, \oplus, \neg, 0, \frac{1}{k}, \ldots, \frac{k-1}{k}, 1\right\rangle$$

is the structure that extends the finite MV-algebra $\mathbb{MV}_k$ with the constants $\frac{1}{k}, \ldots, \frac{k-1}{k}, 1$.

NOTATION 2. *In the rest of the paper, we will talk about* standard MV-algebras *to refer all the above structures. We will simply refer to* $\mathbb{MV}_\infty$*,* $\mathbb{MV}^c_\infty$ *and* $\mathbb{DMV}_\infty$ *as* infinite standard MV-algebras. *In a similar fashion, we will refer to* $\mathbb{MV}_k$ *and* $\mathbb{MV}^c_k$ *as* finite standard MV-algebras. *Notice that if we want to specifically talk about* $\mathbb{MV}_k$*, we will use (if necessary) the expression* "the finite standard MV-algebra $\mathbb{MV}_k$"*. So, unless only* $\mathbb{MV}_k$ *is explicitly mentioned, the expression* "finite standard MV-algebras" *will refer to both* $\mathbb{MV}_k$ *and* $\mathbb{MV}^c_k$*. This should always be clear from the context.*

In all the structures defined above it is possible to introduce an order relation so that for all $x, y \in [0, 1]$ $x \leq y$ if and only if $\neg x \oplus y = 1$. Moreover, the following new operators can be defined, for all $x, y \in [0, 1]$:

$$\begin{array}{rcl@{\qquad}rcl}
x \to y & \text{is} & \neg x \oplus \neg y, & x \wedge y & \text{is} & x \odot (x \to y), \\
x \odot y & \text{is} & \neg(x \to \neg y), & x \vee y & \text{is} & ((x \to y) \to y), \\
x \leftrightarrow y & \text{is} & (x \to y) \odot (y \to x), & d(x, y) & \text{is} & \neg(x \leftrightarrow y), \\
x \ominus y & \text{is} & x \odot \neg y. & & &
\end{array}$$

All the operations defined above have an interpretation over $[0, 1]$ that corresponds to the one given in Section 3.1 for the related connectives. $\mathbb{MV}_\infty$, $\mathbb{MV}^c_\infty$, $\mathbb{DMV}_\infty$, $\mathbb{MV}_k$, and $\mathbb{MV}^c_k$ provide the standard semantics for $Ł_\infty$, $RPŁ_\infty$, $RŁ_\infty$, $Ł_k$, and $Ł^c_k$, respectively.[9]

Let $\mathbb{L}$ be any standard MV-algebra. We denote by $\mathcal{L}_\mathbb{L}$ the language of $\mathbb{L}$. We use $\mathsf{Th}(\mathbb{L})$ to refer to the first-order theory of $\mathbb{L}$ in the language $\mathcal{L}_\mathbb{L}$, that is, the set of sentences in $\mathcal{L}_\mathbb{L}$ that hold over $\mathbb{L}$. In particular,

(1) $\mathsf{Th}(\mathbb{MV}_\infty)$ is the first-order theory of $\mathbb{MV}_\infty$ in the language

$$\mathcal{L}_{\mathbb{MV}_\infty} = \langle \oplus, \neg, 0\rangle.$$

(2) $\mathsf{Th}(\mathbb{MV}^c_\infty)$ is the first-order theory of $\mathbb{MV}^c_\infty$ in the language

$$\mathcal{L}_{\mathbb{MV}^c_\infty} = \left\langle \oplus, \neg, \{c\}_{c \in \mathbb{Q}\cap[0,1]}\right\rangle.$$

(3) $\mathsf{Th}(\mathbb{DMV}_\infty)$ is the first-order theory of $\mathbb{DMV}_\infty$ in the language

$$\mathcal{L}_{\mathbb{DMV}_\infty} = \langle \oplus, \neg, \{\delta_n\}_{n>1}, 0\rangle.$$

(4) $\mathsf{Th}(\mathbb{MV}_k)$ is the first-order theory of $\mathbb{MV}_k$ in the language

$$\mathcal{L}_{\mathbb{MV}_k} = \langle \oplus, \neg, 0\rangle.$$

[9]The language we use for MV-algebras and Łukasiewicz logics is the same. Still, its meaning should be clear from the context.

(5) $\mathsf{Th}(\mathbb{MV}_k^c)$ is the first-order theory of $\mathbb{MV}_k^c$ in the language

$$\mathcal{L}_{\mathbb{MV}_k^c} = \left\langle \oplus, \neg, 0, \frac{1}{k}, \ldots, \frac{k-1}{k}, 1 \right\rangle.$$

We call a formula of $\mathsf{Th}(\mathbb{L})$ a *quantified Łukasiewicz formula* (sentence if there are no free variables). We use the symbols $\sqcap, \sqcup, \Rightarrow, \sim$ to denote the Boolean metalanguage conjunction, disjunction, implication, and negation, respectively. Notice that a strict inequality relation is definable for all the previous structures as follows:

$$x < y \qquad \text{iff} \qquad \sim(\neg y \oplus x = 1).$$

Every quantified Łukasiewicz formula has the form

$$Q_1 x_1 \ldots Q_n x_n \ \Phi(x_1, \ldots, x_n, y_1, \ldots, y_m),$$

where $\Phi(x_1, \ldots, x_n, y_1, \ldots, y_m)$ is a Boolean combination of equalities and strict inequalities in $\mathcal{L}_{\mathbb{L}}$ and each Q_i is either an existential or universal quantifier whose associated variable x_i ranges over the related domain.

THEOREM 3.6. *Let $\mathbb{L}$ be any standard* MV*-algebra. The first-order theory* $\mathsf{Th}(\mathbb{L})$ *admits quantifier elimination in the language $\mathcal{L}_{\mathbb{L}}$.*

By the above theorem, every quantified Łukasiewicz formula in $\mathsf{Th}(\mathbb{L})$

$$Q_1 x_1 \ldots Q_n x_n \ \Phi(x_1, \ldots, x_n, y_1, \ldots, y_m)$$

is logically equivalent to a quantifier-free formula

$$\Psi(y_1, \ldots, y_m)$$

in the same language. As a consequence, both formulae define the same set over the domain L of the related structure $\mathbb{L}$, that is, for all $(a_1, \ldots, a_m) \in L^m$:

$$\mathbb{L} \models Q_1 x_1 \ldots Q_n x_n \ \Phi(x_1, \ldots, x_n, a_1, \ldots, a_m) \quad \text{iff} \quad \mathbb{L} \models \Psi(a_1, \ldots, a_m).$$

Proofs of quantifier elimination for $\mathsf{Th}(\mathbb{MV}_\infty)$, $\mathsf{Th}(\mathbb{DMV}_\infty)$, and $\mathsf{Th}(\mathbb{MV}_k)$ can be found in Baaz and Veith [1999], Caicedo [2007], and Lenzi and Marchioni [2014]. Quantifier elimination for $\mathsf{Th}(\mathbb{MV}_\infty^c)$ and $\mathsf{Th}(\mathbb{MV}_k^c)$ is a trivial consequence of the fact that the same result holds for $\mathsf{Th}(\mathbb{MV}_\infty)$ and $\mathsf{Th}(\mathbb{MV}_k)$, and that $\mathcal{L}_{\mathbb{MV}_\infty^c}$ and $\mathcal{L}_{\mathbb{MV}_k^c}$ are simply expansions of $\mathcal{L}_{\mathbb{MV}_\infty}$ and $\mathcal{L}_{\mathbb{MV}_k}$ (respectively) including a constant for each element of $\mathbb{Q} \cap [0, 1]$ and L_k (respectively).

4. ŁUKASIEWICZ GAMES

We now introduce the framework of Łukasiewicz games. First, let $\mathsf{V} = \{p_1, \ldots, p_m\}$ be a finite set of propositional variables, as above. Our games are populated by a finite, nonempty set P of players $\mathsf{P} = \{P_1, \ldots, P_n\}$ (also referred to as "agents"). Note that throughout this article, we assume that $|\mathsf{P}| = n$. Each player P_i controls a subset of propositional variables $\mathsf{V}_i \subseteq \mathsf{V}$, so that the sets V_i form a partition of V. The fact that player P_i is in control of the set V_i means that P_i has the unique ability within the game to choose values for the variables in V_i. It is assumed that variables take values from the set of truth values L of some Łukasiewicz logic L.

A *strategy* for an agent P_i is a function $s_i : \mathsf{V}_i \to L$, which corresponds to a valuation of the variables controlled by P_i. A strategy profile is a collection of strategies $(s_1, \ldots, s_n)$, one for each player. Every strategy profile directly corresponds to a valuation function $e : \mathsf{V} \to L$ and *vice versa*; we find it convenient to abuse notation a little by treating strategy profiles as valuations and valuations as strategy profiles.

We assume that each player is associated with an L-formula ϕ_i, with propositional variables from V, whose valuation is interpreted as the payoff for player P_i. That is, the player P_i seeks a valuation s_i that maximises the value of the corresponding function f_{ϕ_i}. Of course, not all the variables in ϕ_i will in general be under P_i's control and, consequently, the utility P_i obtains by playing a certain strategy (i.e., choosing a certain variable assignment) also potentially depends in part on the choices made by other players.

We now formally define Łukasiewicz games.

Definition 4.1 (*Łukasiewicz Games*). For a Łukasiewicz logic L, a *Łukasiewicz game* $\mathcal{G}$ is given by a structure

$$\mathcal{G} = \langle \mathsf{P}, \mathsf{V}, \{\mathsf{V}_i\}_{i\in\mathsf{P}}, \{\mathsf{S}_i\}_{i\in\mathsf{P}}, \{\phi_i\}_{i\in\mathsf{P}}\rangle,$$

where

(1) $\mathsf{P} = \{P_1, \dots, P_n\}$ is a finite set of *players*.
(2) $\mathsf{V} = \{p_1, \dots, p_m\}$ is a finite set of propositional variables taking values from L.
(3) $\mathsf{V}_i \subseteq \mathsf{V}$ is the set of propositional variables under control of player P_i, so that the sets V_i form a partition of V.
(4) S_i is the strategy set for player i that includes all valuations $s_i : \mathsf{V}_i \to L$ of the propositional variables in V_i, that is,

$$\mathsf{S}_i = \{s_i \mid s_i : \mathsf{V}_i \to L\}.$$

(5) ϕ_i is an L formula, built from variables in V, whose associated function

$$f_{\phi_i} : L^t \to L$$

corresponds to the *payoff function* (also called *utility function*) of P_i, and whose value will be determined by the valuations in $\{\mathsf{S}_1, \dots, \mathsf{S}_n\}$.

Notice that we often use both $p_1, \dots, p_m$ and $\vec{p}_1, \dots, \vec{p}_n$ to refer to the variables in a game. Both expressions refer to the same set of variables, but, while the former is often used to talk about the set of variables in general, the latter is used to refer to the tuples controlled by each player. This difference should be clear from the context.

A *strategy profile* $\vec{s}$ for $\mathcal{G}$ is a tuple $\vec{s} = (s_1, \dots, s_n)$, with each $s_i \in \mathsf{S}_i$ being the strategy selection for the corresponding player in $\mathcal{G}$. Given a strategy s_i for P_i, we denote by $\vec{s}_{-i}$ the collection of strategies $(s_1, \dots, s_{i-1}, s_{i+1}, \dots, s_n)$ not including s_i, and S_{-i} is the set of all $\vec{s}_{-i}$'s. With an abuse of notation, we use

$$f_{\phi_i}(s_1, \dots, s_n) \qquad \text{and} \qquad f_{\phi_i}(s_i, \vec{s}_{-i})$$

to denote P_i's payoff under the strategy profile $(s_1, \dots, s_n)$: recall that ϕ_i defines a payoff function f_{ϕ_i}, and a strategy profile $(s_1, \dots, s_n)$ corresponds to a valuation $e : \mathsf{V} \to L$.

NOTATION 3. *We call a Łukasiewicz game on a finite-valued Łukasiewicz logic a finite Łukasiewicz game. A Łukasiewicz game on an infinite Łukasiewicz logic is called an infinite Łukasiewicz game. Whenever we use the expression "Łukasiewicz game" without specifying whether the game is finite or infinite, we are referring to a game defined on an arbitrary Łukasiewicz logic. The use of this expression will happen in the most general cases when results and definitions hold for the whole class of Łukasiewicz games. When the choice of a specific logic* L *is relevant, we talk about a Łukasiewicz game on* L.

We now introduce the notion of a pure strategy Nash equilibrium for Łukasiewicz games.

Definition 4.2 (*Pure Strategy Nash Equilibrium*). Let $\mathcal{G}$ be a Łukasiewicz game. A strategy profile $(s_1^*, \dots, s_n^*)$ is called a *pure strategy Nash Equilibrium* (NE) for $\mathcal{G}$, if there exist no player P_i and no strategy s_i such that

$$f_{\phi_i}(s_1^*, \dots, s_i^*, \dots, s_n^*) < f_{\phi_i}(s_1^*, \dots, s_i, \dots, s_n^*).$$

Given a game $\mathcal{G}$, the set of its pure strategy Nash equilibria is denoted by $\mathsf{NE}(\mathcal{G})$. Notice that we have defined the elements of $\mathsf{NE}(\mathcal{G})$ as strategy profiles. However, in the rest of the article we will also often see the elements of the set of equilibria as the *values* assigned by a strategic choice, that is, by a valuation. So, $\mathsf{NE}(\mathcal{G})$ can be equivalently seen as a set whose elements are certain tuples of strategies $(s_1, \dots, s_n) \in \mathsf{S}$, or tuples of elements $(\vec{a}_1, \dots, \vec{a}_n) \in L^m$.

5. EXAMPLES

We now introduce a number of examples to illustrate Łukasiewicz games, and in particular, we choose examples that, we argue, cannot be expressed easily in the framework of conventional Boolean games (i.e., using classical logic to express goals).

5.1. (A Variant of the) Traveler's Dilemma

The Traveler's Dilemma was introduced in Basu [1994] in order to illustrate the tension between the rational solution suggested by the existence of a Nash equilibrium, and apparently reasonable behaviour based on intuition. We introduce (a slightly modified variant of) the Traveler's Dilemma and show how to formalise it as a Łukasiewicz game.

The game is as follows. Two travelers fly back home from a trip to a remote island where they bought exactly the same antiques. Unfortunately for them, their luggage gets damaged and all the items acquired are broken. Both travelers purchased the same travel insurance and, for that, they are potentially entitled to a refund of a sum between £2 and £100. The insurance agent of the airline promises a compensation for the inconvenience, but, not knowing the exact value of the objects, she puts forward the following proposal. As per the company's insurance policy, the airline will refund the travelers up to a total of £200, between the two of them. Both travelers must privately write down on paper a natural number corresponding to the cost of the antiques. This value must be within the amount they are entitled to receive under the insurance rules, that is, it must be any value between £2 and £100. If they both write the same number, the agent can assume that they are both telling the truth, so they will both receive exactly that amount. If the travelers write different numbers, the one who wrote the lower number, say x (assumed to be the honest one), will receive x plus a reward of two units. The other player, who is regarded by the agent as dishonest, will receive x with a penalty of two units.

The travelers are allowed to claim a positive sum below £2 and above £100 (up to £200), but this would break the insurance agreement. Anybody claiming more than they are entitled to would receive nothing, since they would be overestimating the value of their items. If both travelers write any amount between 0 and 2, then they are both considered as undervaluing the content of their luggage and will get £2 anyway.

Payoffs in the Traveler's Dilemma are defined by the following functions:

$$f_1, f_2 : \{0, \dots, 200\}^2 \to \{0, \dots, 200\},$$

	T2											
T1	0	1	2	3	4	···	98	99	100	101	···	200
0	**2, 2**	**2, 2**	**2, 2**	2, 0	2, 0	···	2, 0	2, 0	2, 0	2, 0	···	2, 0
1	**2, 2**	**2, 2**	**2, 2**	3, 0	3, 0	···	3, 0	3, 0	3, 0	3, 0	···	3, 0
2	**2, 2**	**2, 2**	**2, 2**	4, 0	4, 0	···	4, 0	4, 0	4, 0	4, 0	···	4, 0
3	0, 2	0, 3	0, 4	3, 3	5, 1	···	5, 1	5, 1	5, 1	5, 0	···	5, 0
4	0, 2	0, 3	0, 4	1, 5	4, 4	···	6, 2	6, 2	6, 2	6, 0	···	6, 0
⋮	⋮	⋮	⋮	⋮	⋮	⋱	⋮	⋮	⋮	⋮	⋱	⋮
98	0, 2	0, 3	0, 4	1, 5	2, 6	···	98, 98	100, 96	100, 96	100, 0	···	100, 0
99	0, 2	0, 3	0, 4	1, 5	2, 6	···	96, 100	99, 99	101, 97	101, 0	···	101, 0
100	0, 2	0, 3	0, 4	1, 5	2, 6	···	96, 100	97, 101	100, 100	102, 0	···	102, 0
101	0, 2	0, 3	0, 4	0, 5	0, 6	···	0, 100	0, 101	0, 102	0, 0	···	0, 0
⋮	⋮	⋮	⋮	⋮	⋮	⋱	⋮	⋮	⋮	⋮	⋱	⋮
200	0, 2	0, 3	0, 4	0, 5	0, 6	···	0, 100	0, 101	0, 102	0, 0	···	0, 0

Fig. 1. Traveler's Dilemma payoff matrix.

where

$$f_1(x, y) = \begin{cases} x & 2 \leq x = y \leq 100 \\ (x+2) & x < y, x \leq 100, 2 < y \\ \max(y-2, 0) & 100 \geq x > y, x > 2 \\ 0 & x > 100 \\ 2 & x, y \leq 2 \end{cases},$$

$$f_2(x, y) = \begin{cases} y & 2 \leq x = y \leq 100 \\ (y+2) & y < x, y \leq 100, 2 < x \\ \max(x-2, 0) & 100 \geq y > x, y > 2 \\ 0 & y > 100 \\ 2 & x, y \leq 2 \end{cases},$$

and whose payoff matrix is shown in Figure 1.

Given that each player wants to maximise her/his payoff, what choices should they make? First of all, no traveler has any incentive to claim an amount higher than 100, since that would result in ending up empty handed. Now, if both travelers choose 100, they both get 100. However, each traveler soon realises that if she deviates from the previous choice and writes 99, while the other player sticks to the original selection, she can increase her payoff to 101. Under the assumption of common knowledge and rationality, however, the other player is drawn to make the same decision, which leads to writing 99, yielding a mutual payoff of 99. Still, deviating from this selection is unilaterally beneficial for each individual, producing again a situation of coordination between the players' choices. This reasoning only ends when both players select 2, thus obtaining only the minimum refund. The same would happen if both travelers chose to undervalue the cost of their items. The strategy profiles in $\{0, 1, 2\}^2$ are then the unique pure strategy Nash equilibria of the game. This, however, clearly clashes with what intuition would suggest to be a rational choice. It seems implausible that two individuals would follow the previous line of reasoning and rationally come to the conclusion that the best solution is ending up either claiming the minimum amount or even underestimating the value of their possessions.

The Traveler's Dilemma can be formalised as a Łukasiewicz game over $Ł_{200}$ as follows. Define the following game

$$\mathcal{G} = \big\langle \{\mathrm{T}_1, \mathrm{T}_2\}, \{p_1, p_2\}, \{\{p_1\}, \{p_2\}\}, \{\mathsf{S}_1, \mathsf{S}_2\}, \{\phi_1(p_1, p_2), \phi_2(p_1, p_2)\} \big\rangle,$$

where T_1 and T_2 are the two travelers; $\{p_1, p_2\}$ is the set of propositional variables, with p_1 being controlled by T_1 and p_2 being controlled by T_2; $\mathsf{S}_i = \{s_i : \{p_i\} \to \{0, \dots, 200\}\}$, with $i \in \{1, 2\}$. The payoff formulae are defined as follows:

$$\begin{aligned}\phi_1(p_1, p_2) := \Big[&\Delta\left(p_1 \to \overline{\tfrac{100}{200}}\right) \wedge \Big[\Big(\Big(p_1 \wedge \Delta(p_1 \leftrightarrow p_2) \wedge \Delta\left(\overline{\tfrac{2}{200}} \to p_1\right)\Big)\\ &\vee \Big(\Big(p_1 \oplus \overline{\tfrac{2}{200}}\Big) \wedge \neg\Delta(p_2 \to p_1) \wedge \neg\Delta\left(p_2 \to \overline{\tfrac{2}{200}}\right)\Big)\\ &\vee \Big(\Big(p_2 \ominus \overline{\tfrac{2}{200}}\Big) \wedge \neg\Delta(p_1 \to p_2) \wedge \neg\Delta\left(p_1 \to \overline{\tfrac{2}{200}}\right)\Big)\Big]\Big]\\ &\vee \Big[\overline{\tfrac{2}{200}} \wedge \Delta\left(p_1 \vee p_2 \to \overline{\tfrac{2}{200}}\right)\Big],\end{aligned}$$

$$\begin{aligned}\phi_2(p_1, p_2) := \Big[&\Delta\left(p_1 \to \overline{\tfrac{100}{200}}\right) \wedge \Big[\Big(\Big(p_2 \wedge \Delta(p_2 \leftrightarrow p_1) \wedge \Delta\left(\overline{\tfrac{2}{200}} \to p_2\right)\Big)\\ &\vee \Big(\Big(p_2 \oplus \overline{\tfrac{2}{200}}\Big) \wedge \neg\Delta(p_1 \to p_2) \wedge \neg\Delta\left(p_1 \to \overline{\tfrac{2}{200}}\right)\Big)\\ &\vee \Big(\Big(p_1 \ominus \overline{\tfrac{2}{200}}\Big) \wedge \neg\Delta(p_2 \to p_1) \wedge \neg\Delta\left(p_2 \to \overline{\tfrac{2}{200}}\right)\Big)\Big]\Big]\\ &\vee \Big[\overline{\tfrac{2}{200}} \wedge \Delta\left(p_1 \vee p_2 \to \overline{\tfrac{2}{200}}\right)\Big].\end{aligned}$$

The above formulae define payoff functions that can be easily seen to be the linear transformation of f_1 and f_2 over L_{200}.

5.2. (A Variant of) Second-Price Sealed-Bid Auctions

A second-price sealed-bid auction with perfect information is an auction in which buyers independently assign a value, known to the others, to an item they want to purchase, and submit sealed bids. The buyer with the higher bid wins and pays an amount equal to the second highest bid (e.g., see Parsons et al. [2011]).

Let B_i, with $i \in \{1, \dots, n\}$, denote each buyer, and $\mathbf{v}_i$ denote the value B_i assigns to the item. B_i's payoff is given by the following function:

$$f_i(x_1, \dots, x_n) = \begin{cases} \mathbf{v}_i - \max_{j \neq i} x_j & x_i > \max_{j \neq i} x_j \\ 0 & \text{otherwise} \end{cases},$$

where the variables $x_1, \dots, x_n$ stand for the bid submitted by the players.

We can formalise this kind of auction as a finite Łukasiewicz game with certain restrictions. First, we assume that the bidders' valuation $\mathbf{v}_i$ is the same for every i. Second, we assume that buyers cannot submit a bid that is higher than the assigned value $\mathbf{v}_i$. Notice that whenever at least two players bid a price higher than $\mathbf{v}_i$, the payoff of the winning player is negative. This is clearly a problem in the context of Łukasiewicz logic, since the range of every function associated with a formula must be a subset of the set of values of each one of its variables. This restriction also makes it possible to avoid the winner's curse, that is, the situation where the player with the winning bid pays too much and loses money with respect to the item's valuation.

So, we define this variant of second-price auctions over L_k, with the following payoff functions $g_i : (L_k)^n \to L_k$, for each i:

$$g_i(x_1, \dots, x_n) = \begin{cases} 1 - \max_{j \neq i}(x_j) & x_i > \max_{j \neq i}(x_j) \\ 0 & \text{otherwise} \end{cases}.$$

Notice that, since buyers cannot submit bids that exceed the value of the item, we simply assume that $\mathbf{v}$ is 1, that is, the top of our scale.

Define then a game

$$\mathcal{G} = \langle \mathsf{P}, \mathsf{V}, \{\mathsf{V}_i\}_{i \in \mathsf{P}}, \{\mathsf{S}_i\}_{i \in \mathsf{P}}, \{\phi_i\}_{i \in \mathsf{P}} \rangle$$

over $Ł_k$ where

(1) $\mathsf{P} = \{B_1, \ldots, B_n\}$ is the set of bidders;
(2) $\mathsf{V} = \{p_1, \ldots, p_n\}$ is the set of propositional variables;
(3) $\mathsf{V}_i = \{p_i\}$ is the set of propositional variables under control of player B_i;
(4) S_i is the strategy set for player i that includes all valuations $s_i : \mathsf{V}_i \to L_k$ of the propositional variables in V_i; and
(5) each ϕ_i corresponds to the formula

$$\neg\Delta\left(p_i \to \left(\bigvee_{j\neq i} p_j\right)\right) \wedge \left(\overline{1} \ominus \left(\bigvee_{j\neq i} p_j\right)\right)$$

whose associated function is the function g_i defined above.

5.3. Generalised Matching Pennies

The following game is a generalisation of Matching Pennies,[10] the classic example of a zero-sum game without a pure strategy equilibrium. In the original game, two players P_1 and P_2 both have a penny and must secretly choose whether to turn it to head or tails, revealing their choices simultaneously. If their choices are the same, then P_1 takes both pennies; if they are different, P_2 takes both.

Imagine that both players must perform an action with a certain cost and are in charge of the variables p_1 and p_2, respectively. P_1's overall strategy is to be as close as possible to P_2's choice. In contrast, P_2 wants to keep the greatest possible distance between the choices. The players' strategy spaces are given by the sets of functions

$$\mathsf{S}_1 = \{s_1 \mid s_1 : \{p_1\} \to [0, 1]\}, \qquad \mathsf{S}_2 = \{s_2 \mid s_2 : \{p_2\} \to [0, 1]\}.$$

Recall that the Łukasiewicz logic expression $d(p_1, p_2)$, defined in Section 3.1, realises the distance between the values assigned to the variables p_1 and p_2. Using this expression, we can define the payoff for P_1 as the formula $\neg d(p_1, p_2)$, whose associated function is $1 - |x_1 - x_2|$, while P_2's payoff is defined by the formula $d(p_1, p_2)$, with associated function $|x_1 - x_2|$. The game is formally defined as follows on $Ł_\infty$:

$$\mathcal{G} = \big\langle \{P_1, P_1\}, \{p_1, p_2\}, \{\{p_1\}, \{p_2\}\}, \{\mathsf{S}_1, \mathsf{S}_2\}, \{\neg d(p_1, p_2), d(p_1, p_2)\}\big\rangle.$$

Figure 2 shows the payoff functions for this generalised version of Matching Pennies on $Ł_\infty$.

5.4. Generalised Prisoner's Dilemma

Here, we offer a generalisation of the Prisoner's Dilemma over $[0, 1]$ in terms of Łukasiewicz games.

Suppose that two prisoners, both accused of committing the same crime, are asked by the police to provide evidence against each other. Each prisoner can either fully confess (i.e., defect), by testifying and offering the police the whole body of evidence supporting the incrimination of the other, or simply cooperate with their fellow criminal and remain silent. Alternatively, prisoners can choose to only partially confess by providing more or less evidence against each other. We use the real unit interval $[0, 1]$ as a scale to formally represent the degree of cooperation of each prisoner, so that 0 means full cooperation, and 1 full defection, and every other degree in between captures to which extent the prisoner is willing to defect and collaborate with the police.

We define a function to specify each prisoner's payoff by fixing the payoff at the extremes, so that at the points of full cooperation and full defection the outcome

[10]This example is taken from Kroupa and Majer [2014].

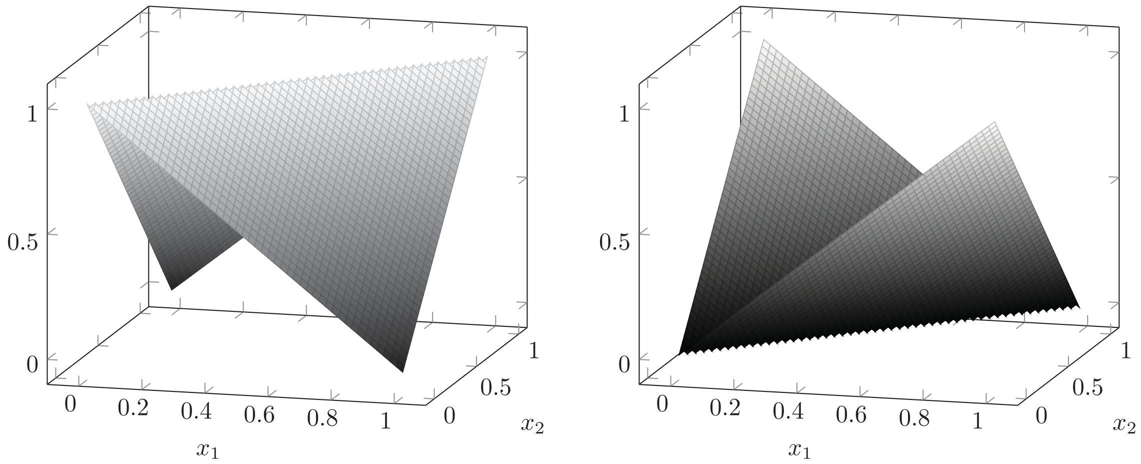

Fig. 2. Payoff functions $1 - |x_1 - x_2|$ and $|x_1 - x_2|$ for the generalised Matching Pennies.

		P_2: 0 (Cooperate)	P_2: 1 (Defect)
P_1	(Cooperate) 0	$\frac{2}{3}, \frac{2}{3}$	0, 1
	(Defect) 1	1, 0	$\frac{1}{3}, \frac{1}{3}$

Fig. 3. Payoff matrix at the extremes for the Prisoner's Dilemma.

is compatible with the payoff of the traditional version of the Prisoner's Dilemma (Figure 3) (see, for instance, Osborne and Rubinstein [1994] and Maschler et al. [2013]). We interpret the outcome as the utility for the prisoner depending on the sentence, so that 1 represents the best case scenario where the prisoner receives no penalty, while 0 means the prisoner is sentenced to the maximum punishment. Suppose the prisoners disagree in their choice, that is, we have either $(1, 0)$ or $(0, 1)$, so that one defects while the other cooperates. Then the one who defects maximises her/his outcome, while the other receives the full punishment. If, instead, the prisoners make the same choice they will both be punished, but will get a better outcome if they remain silent and cooperate.

In order to define proper payoff functions over $[0, 1]^2$, we treat these strategy profiles with their related payoff as coordinate points in $[0, 1]^3$ and use them to define the planes over which those points lie. A simple calculation shows that the planes are defined by the functions

$$f_1(x_1, x_2) = \frac{1}{3}x_1 - \frac{2}{3}x_2 + \frac{2}{3}, \qquad f_2(x_1, x_2) = \frac{1}{3}x_2 - \frac{2}{3}x_1 + \frac{2}{3},$$

which can be easily seen to be rational McNaughton functions whose associated formulae are

$$\phi_{f_1}(p_1, p_2) = (\delta_3 p_1 \ominus \delta_3 2p_2) \oplus \delta_3 2(\overline{1}), \qquad \phi_{f_2}(p_1, p_2) = (\delta_3 p_2 \ominus \delta_3 2p_2) \oplus \delta_3 2(\overline{1})$$

and whose graph is displayed in Figure 4.

This modified version of the Prisoner's Dilemma can be then formalised over RŁ$_\infty$ as the following game

$$\mathcal{G} = \big\langle \{P_1, P_2\}, \{p_1, p_2\}, \{\{p_1\}, \{p_2\}\}, \{\phi_{f_1}(p_1, p_2), \phi_{f_2}(p_1, p_2)\} \big\rangle,$$

where the payoff formulae for P_1 and P_2 are the ones defined above.

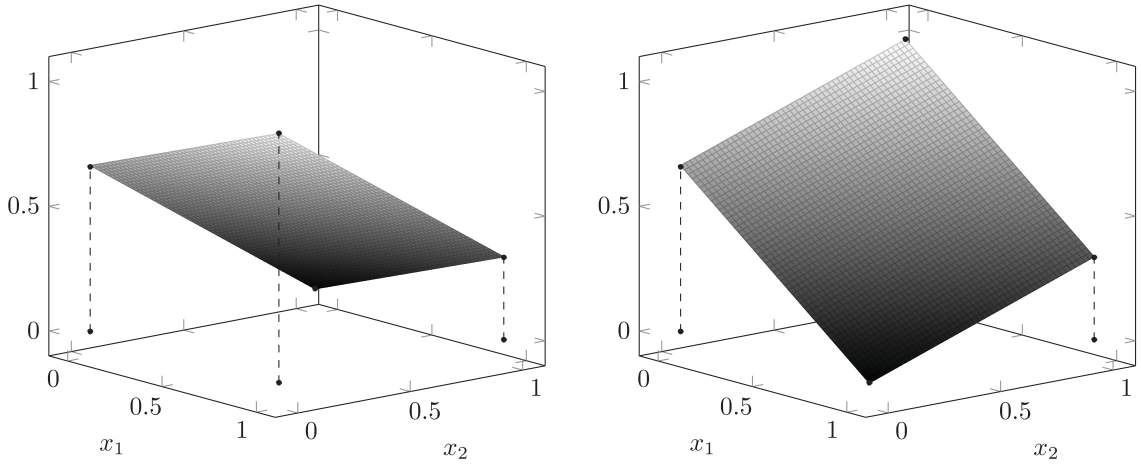

Fig. 4. Payoff functions for the generalised Prisoner's Dilemma.

It is now natural to ask whether this formalisation of the Prisoner's Dilemma over $[0, 1]$ is meaningful and actually respects our intuition when compared to the classical example. The answer is affirmative. In the classical case, full cooperation for both players is unstable, since one of them can always obtain a better outcome by deviating. The same happens in this generalisation, since, whenever the prisoners provide the same amount of evidence (but not the full amount), increasing the degree of defection always results in a better payoff for the deviating player. In addition, full defection is a strictly dominant strategy for each player, and the strategy profile $(1, 1)$ is a pure strategy Nash equilibrium.

5.5. Continuous Weak-Link Games

Weak-link games[11] are a class of coordination games, where the players benefit from mutually coordinating on the same strategy. The original version of the game (see Van Huyck et al. [1990]) consists of n players who simultaneously choose a number from a finite set $\{1, \ldots, m\}$. Each player i's payoff is defined by the following function:

$$u_i(x_1, \ldots, x_n) = a + a' \cdot \min(x_1, \ldots, x_n) - a'' \cdot (x_i - \min(x_1, \ldots, x_n)),$$

where x_i is the choice made by player i, and a, a', a'' are positive parameters. Intuitively, x_i is interpreted as the effort i is willing to make in her interaction with others. The payoff u_i is heavily influenced by the choice of the agent with the lower effort level. Therefore, each player's payoff depends on the weakest link in the strategic interaction. The game has m pure strategy Nash equilibria corresponding to the strategy profiles in which the players select the same values (see Van Huyck et al. [1990]).

We introduce here a continuous generalisation of weak-link games where each u_i is defined over $[0, 1]^n$, and $a, a', a'' \in (0, 1]$ (see also Anderson et al. [2001]). It is worth pointing out that not all instances of continuous weak-link games can be represented as a Łukasiewicz game, since the set of values assigned to each variable in u_i might be a strict subset of the function's range. Still, it is always possible to encode a continuous weak-link game as a Łukasiewicz game for a suitable choice of the parameters a, a', a''. In fact, for $a + a' \leq 1$ and $a'' \leq a$, the function u_i is always such that $u_i : [0, 1]^n \to [0, 1]$.

As an example, let $a = \frac{2}{3}, a' = \frac{1}{3}, a'' = \frac{1}{4}$. Each function

$$u_i(x_1, x_2) = \frac{2}{3} + \left(\frac{1}{3} \cdot \min(x_1, x_2)\right) - \left(\frac{1}{4} \cdot (x_i - \min(x_1, x_2))\right),$$

[11]Nothing to do with the popular TV game show "The Weakest Link."

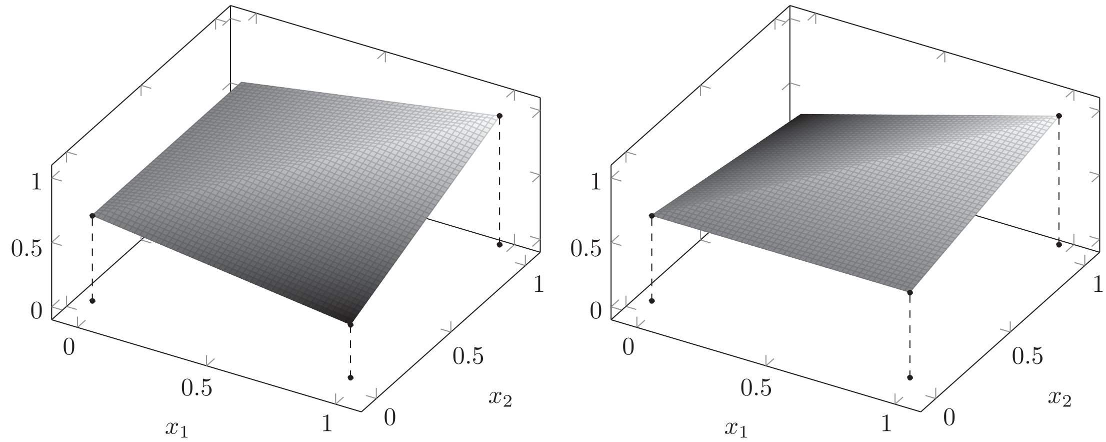

Fig. 5. Payoff functions for the continuous weak-link game.

with $i \in \{1, 2\}$, is a rational McNaughton function (see Figure 5). We can then define a two-player continuous weak-link game over RŁ$_\infty$ in the following form:

$$\mathcal{G} = \langle \mathsf{P}, \mathsf{V}, \{\mathsf{V}_i\}_{i\in\mathsf{P}}, \{\mathsf{S}_i\}_{i\in\mathsf{P}}, \{\phi_i\}_{i\in\mathsf{P}}\rangle,$$

where

(1) $\mathsf{P} = \{P_1, P_2\}$.
(2) $\mathsf{V} = \{p_1, p_2\}$.
(3) $\mathsf{V}_i = \{p_i\}$, with $i \in \{1, 2\}$.
(4) The strategy space is defined as follows, for each i:

$$\mathsf{S}_i = \{s_i \mid s_i : \{p_i\} \to [0, 1]\}.$$

(5) The players' payoff formulae are

$$\phi_1(p_1, p_2) := ((\delta_{12} 7 p_2 \ominus \delta_4 p_1) \oplus \delta_3 2\overline{1}) \wedge (\delta_3 p_1 \oplus \delta_3 2\overline{1}),$$

$$\phi_2(p_1, p_2) := ((\delta_{12} 7 p_1 \ominus \delta_4 p_2) \oplus \delta_3 2\overline{1}) \wedge (\delta_3 p_2 \oplus \delta_3 2\overline{1}),$$

and their associated functions correspond to u_i, with $i \in \{1, 2\}$, as defined above.

It is easy to see that

$$\left\{(b_1, b_2) \mid (b_1, b_2) \in [0, 1]^2, b_1 = b_2\right\}$$

is the set of equilibria of the game.

6. BASIC PROPERTIES OF ŁUKASIEWICZ GAMES

In this section, we study some general basic properties common to all Łukasiewicz games that will be used in the remaining part of this work.

In the normal form representation of noncooperative games, the existence of equilibria is equivalent to the nonemptiness of the intersection, for all i, of all the sets of strategy profiles $(s_i, \vec{s}_{-i})$ such that s_i maximises $u_i(x_i, \vec{s}_{-i})$. In other words, a strategic game in normal form admits an equilibrium if and only if

$$\bigcap_{i=1}^{n} \bigcup_{\vec{s}_{-i}\in\mathsf{S}_{-i}} \left\{(s_i, \vec{s}_{-i}) \mid \operatorname*{argmax}_{s'_i\in\mathsf{S}_i}(u_i(s'_i, \vec{s}_{-i})) = s_i\right\} \neq \emptyset.$$

A similar characterisation through (rational) McNaughton functions (and their finite-valued restrictions) for Łukasiewicz games is not possible since Łukasiewicz games are not defined in normal form. In fact, in a Łukasiewicz game, each payoff formula might contain a different subset of variables, and the corresponding McNaughton functions are not defined over the same domain. We show that this problem can be easily overcome by introducing the concept of a normalised game and showing that every game can be transformed into a normalised one preserving the equilibria. The concept of a normalised Łukasiewicz game can be interpreted as a representation in normal form for Łukasiewicz games. We begin introducing some preliminary notions and definitions.

A Łukasiewicz game $\mathcal{G}$ with a set of variables $\mathsf{V} = \{p_1, \dots, p_m\}$ is called *normalised* whenever each payoff formula ϕ_i is of the form $\phi_i(p_1, \dots, p_m)$, that is, all the variables from V occur in each ϕ_i.

Given a game $\mathcal{G}$, let

$$\eth : \mathsf{P} \to \{1, \dots, m\}$$

be a function assigning to each player P_i an integer from $\{1, \dots, m\}$ that corresponds to the number of variables in V_i: that is,

$$\eth(P_i) = m_i.$$

$\eth$ is called a *distribution function*.

Given a Łukasiewicz game $\mathcal{G}$, the *type* of $\mathcal{G}$ is the triple $\langle n, m, \eth\rangle$, where n is the number of players, m is the number of variables in V, and $\eth$ is the distribution function for $\mathcal{G}$. We say that two Łukasiewicz games $\mathcal{G}$ and $\mathcal{G}'$ belong to the *same class* if they are defined on the same Łukasiewicz logic L, they have type $\langle n, m, \delta\rangle$ and $\langle n, m, \delta'\rangle$, respectively, and there exists a permutation $\jmath$ of the indices $\{1, \dots, n\}$ such that, for all P_i,

$$\eth(P_{\jmath(i)}) = \eth'(P_i).$$

Notice that what matters in the definition of a class is not which players are assigned certain variables, but rather their distribution. In fact, up to a renaming of the variables and the players, two games in the same class have the same players, the same variables, and each player controls the same subset of variables.

For instance, take two games $\mathcal{G}$ and $\mathcal{G}'$ both having three players P_1, P_2, P_3 and the same variables $p_1, \dots, p_6$ so that the players control the variables as follows:

	$\mathcal{G}$	$\mathcal{G}'$
P_1	p_1	p_1, p_2, p_6
P_2	p_2, p_3	p_3
P_3	p_4, p_5, p_6	p_4, p_5

$\mathcal{G}$ and $\mathcal{G}'$ belong to the same class, since they have the same number of players, the same number of variables, and the permutation $\jmath$, where

$$1 \xmapsto{\jmath} 2 \qquad 2 \xmapsto{\jmath} 3 \qquad 3 \xmapsto{\jmath} 1,$$

is such that $\eth\left(P_{\jmath(i)}\right) = \eth'\left(P_i\right)$ for all P_i.

Now, let $\mathcal{G}$ and $\mathcal{G}'$ be two Łukasiewicz games of the same class. We say that $\mathcal{G}$ and $\mathcal{G}'$ are *equivalent* whenever

$$\mathsf{NE}(\mathcal{G}) = \mathsf{NE}(\mathcal{G}').$$

We are now going to see that every game can be transformed into a normalised one of the same class having the same set of equilibria. The main step is to show that any payoff formula can be rewritten in an equivalent form in the whole set V of variables.

We say that a formula $\phi(p_1, \dots, p_w)$ in any Łukasiewicz logic L has an *equivalent extension*

$$\phi^\sharp(p_1, \dots, p_w, q_1, \dots, q_v)$$

in $\{q_1, \ldots, q_v\}$, if, for every $(a_1, \ldots, a_w) \in L^w$

$$f_\phi(a_1, \ldots, a_w) = f_{\phi^\sharp}(a_1, \ldots, a_w, b_1, \ldots, b_v)$$

for all $(b_1, \ldots, b_v) \in L^v$. The next lemma shows that for any arbitrary L formula, we can always find an equivalent extension in any set of variables.

LEMMA 6.1. *Let $\phi(p_1, \ldots, p_w)$ be a formula in any Łukasiewicz logic* L. *For any set of variables $\{q_1, \ldots, q_v\}$, there exists an equivalent extension of $\phi(p_1, \ldots, p_w)$ in $\{q_1, \ldots, q_v\}$.*

PROOF. Take a formula $\phi(p_1, \ldots, p_w)$ and define

$$\psi(p_1, \ldots, p_w, q_1, \ldots, q_v) := \phi(p_1, \ldots, p_w) \oplus \bigoplus_{j=1}^{v} (q_j \odot \neg q_j).$$

For all $\{b_1, \ldots, b_v\} \in L$, the function

$$f_{\bigoplus_{j=1}^{v}(q_j \odot \neg q_j)}(b_1, \ldots, b_v)$$

is constantly equal to 0, since, for all $x \in L$

$$x \odot \neg x = \max(0, x + (1 - x) - 1) = 0.$$

Therefore, for every $(a_1, \ldots, a_w) \in L^w$

$$f_\phi(a_1, \ldots, a_w) = f_\psi(a_1, \ldots, a_w, b_1, \ldots, b_v)$$

for all $(b_1, \ldots, b_v) \in L^v$, and ψ is the equivalent extension of ϕ in $\{q_1, \ldots, q_v\}$.

Given Lemma 6.1, it is straightforward to prove that all Łukasiewicz games have a normalised counterpart.

PROPOSITION 6.2. *Every Łukasiewicz game $\mathcal{G}$ is equivalent to a normalised game.*

PROOF. Take any game

$$\mathcal{G} = \langle \mathsf{P}, \mathsf{V}, \{\mathsf{V}_i\}_{i\in\mathsf{P}}, \{\mathsf{S}_i\}_{i\in\mathsf{P}}, \{\phi_i\}_{i\in\mathsf{P}}\rangle$$

on an arbitrary L and let, for each i, $\{p_{1_i}, \ldots, p_{m_i}\}$ be the set of variables occurring in ϕ_i and $\{q_{1_i}, \ldots, q_{m'_i}\}$ be the set of variables not occurring in ϕ_i, that is, $\{q_{1_i}, \ldots, q_{m'_i}\} = \mathsf{V} \setminus \{p_{1_i}, \ldots, p_{m_i}\}$. Define a new game of the same type on L

$$\mathcal{G}' = \langle \mathsf{P}, \mathsf{V}, \{\mathsf{V}_i\}_{i\in\mathsf{P}}, \{\mathsf{S}_i\}_{i\in\mathsf{P}}, \{\psi_i\}_{i\in\mathsf{P}}\rangle,$$

where each ψ_i is the equivalent extension of $\phi_i(p_{1_i}, \ldots, p_{m_i})$ in the variables $\{q_{1_i}, \ldots, q_{m'_i}\}$.

Suppose that $(s_1, \ldots, s_n)$ is a NE for $\mathcal{G}$. This means that for each i, for all s'_i

$$f_{\phi_i}(s_i, \vec{s}_{-i}) \geq f_{\phi_i}(s'_i, \vec{s}_{-i}),$$

which, by Lemma 6.1 is equivalent to the fact that for each i, for all s'_i

$$f_{\psi_i}(s_i, \vec{s}_{-i}) \geq f_{\psi_i}(s'_i, \vec{s}_{-i}),$$

which, in turn, means that $(s_1, \ldots, s_n)$ is a NE for $\mathcal{G}'$. This proves that both games have the same set of Nash equilibria. □

As a consequence of the above discussion, we obtain that the existence of equilibria can be given a functional representation in terms of the (rational) McNaughton payoff functions (or their finite-valued restriction) of the game, that is, Proposition 6.3. *A Łukasiewicz game $\mathcal{G}$ admits a Nash equilibrium if and only if*

$$\bigcap_{i=1}^{n} \bigcup_{\vec{s}_{-i} \in \mathsf{S}_{-i}} \left\{ (s_i, \vec{s}_{-i}) \mid \underset{s'_i \in \mathsf{S}_i}{\operatorname{argmax}}(f_{\phi_i}(s'_i, \vec{s}_{-i})) = s_i \right\} \neq \emptyset.$$

From now on, we will tacitly assume each game to be normalised. Also, notice that, so far, we have been denoting by $f_{\phi_i}(s_i, \vec{s}_{-i})$ the value of the function f_{ϕ_i} given the strategy profile $(s_i, \vec{s}_{-i})$. As mentioned above, this actually is an abuse of notation since the strategy profile $(s_i, \vec{s}_{-i})$ corresponds to a specific assignment to all the variables in the game, but for the valuation of f_{ϕ_i} only the assignments to the variables actually occurring in f_{ϕ_i} are taken into account. Since every game can be considered normalised, the use of this notation can now be regarded as correct.

In the remainder of this section, we are now going to explore two properties related to the existence of equilibria. The first one has to do with a special class of games, called satisfiable, for which equilibria always exist.

We call Łukasiewicz game $\mathcal{G}$ *satisfiable* if there exists a strategy profile $(s_1, \ldots, s_n)$ such that

$$f_{\phi_i}(s_1, \ldots, s_n) = 1$$

for all i.

The following proposition is an immediate consequence:

Proposition 6.4. *Every satisfiable Łukasiewicz game $\mathcal{G}$ admits a pure strategy Nash equilibrium.*

Proof. By definition, there is a strategy profile $(s_1, \ldots, s_n)$ such that $f_{\phi_i}(s_1, \ldots, s_n) = 1$ for all i. Therefore, the profile $(s_1, \ldots, s_n)$ guarantees the maximum payoff to each player, and trivially corresponds to a Nash equilibrium.

The notion of a satisfiable game will play an important role in the next section where we will show that for every finite game $\mathcal{G}$ having a Nash equilibrium, there always exists a satisfiable game equivalent to $\mathcal{G}$.

The second general property of Łukasiewicz games is the fact that the existence of equilibria can always be expressed through a first-order sentence of the theory of the related standard MV-algebra. Indeed, take any Łukasiewicz game $\mathcal{G}$ on Ł, let $\vec{x}_i$, $\vec{y}_i$ denote tuples of variables assigned to player i, and define the following sentence:

$$\begin{aligned} \Phi_{\mathsf{NE}} := \; & \exists \vec{x}_1, \ldots, \vec{x}_n \forall \vec{y}_1, \ldots, \vec{y}_n \sqcap_{i=1}^{n} \big(\Phi_i(\vec{x}_1, \ldots, \vec{x}_{i-1}, \vec{y}_i, \vec{x}_{i+1}, \ldots, \vec{x}_n) \\ \leq \; & \Phi_i(\vec{x}_1, \ldots, \vec{x}_{i-1}, \vec{x}_i, \vec{x}_{i+1}, \ldots, \vec{x}_n) \big), \end{aligned}$$

where each Φ_i is the first-order term obtained from the payoff formula ϕ_i by simply replacing the propositional variables $\vec{p}_1, \ldots, \vec{p}_n$ with the variables $\vec{x}_1, \ldots, \vec{x}_n, \vec{y}_1, \ldots, \vec{y}_n$. It is then easy to check that:

Proposition 6.5. *A Łukasiewicz game $\mathcal{G}$ on* Ł *admits a pure strategy Nash equilibrium if and only if Φ_{NE} holds over $\mathbb{L}$.*

We will make explicit use of the previous result both in Section 7 and Section 8 to prove that the existence of equilibria in finite and infinite games is equivalent to the satisfiability of a formula of Ł. While Proposition 6.5 will play a fundamental role in proving that result for games with finite and infinite strategy spaces, both cases will require radically different techniques.

7. FINITE GAMES: LOGICAL CHARACTERISATIONS

The aim of this section is to provide a characterisation of the existence of pure strategy Nash equilibria in finite Łukasiewicz games by laying out a step-by-step proof of the following theorem.

THEOREM 7.1. *Let* L *be any finite Łukasiewicz logic. For each Łukasiewicz game* $\mathcal{G}$ *on* L *there exists an* L*-formula* $\phi_{\mathsf{NE}}(\vec{p}_1, \ldots, \vec{p}_n)$ *such that, for all* $(\vec{a}_1, \ldots, \vec{a}_n) \in (L_k)^m$

$$(\vec{a}_1, \ldots, \vec{a}_n) \in Sat(\phi_{\mathsf{NE}}(\vec{p}_1, \ldots, \vec{p}_n)) \quad \textit{iff} \quad (\vec{a}_1, \ldots, \vec{a}_n) \in \mathsf{NE}(\mathcal{G}).$$

Moreover, there exists an L*-formula* ϕ_{eq} *so that the following statements are equivalent:*

(1) $\mathcal{G}$ *admits a pure strategy Nash equilibrium.*
(2) ϕ_{eq} *is satisfiable.*
(3) $\phi_{\mathsf{NE}}(\vec{p}_1, \ldots, \vec{p}_n)$ *is satisfiable.*
(4) *There exists a satisfiable normalised game* $\mathcal{G}'$ *equivalent to* $\mathcal{G}$.

We begin by showing that the existence of equilibria for a finite Łukasiewicz game is equivalent to the satisfiability of a special formula ϕ_{eq}. We will define ϕ_{eq} by introducing exponentially many new variables, thus generating a formula whose length is exponential in the number of the original variables of the game. Still, it is interesting to notice that ϕ_{eq} can be recursively defined given any game $\mathcal{G}$.

Next, we prove that for any finite Łukasiewicz game $\mathcal{G}$, there exists another special formula ϕ_{NE} whose satisfiability set coincides with the set of equilibria of the game. A trivial consequence of this fact is that $\mathcal{G}$ admits an equilibrium if and only if ϕ_{NE} is satisfiable. This result is a refinement of the previous one, since ϕ_{NE} includes occurrences exclusively of each and every one of the variables in $\mathcal{G}$. Still, in spite of its simplicity, we will see that this formula cannot be generated as easily as ϕ_{eq}, since its definition requires the elimination of quantifiers.

We conclude the proof of Theorem 7.1 by showing that a finite Łukasiewicz game $\mathcal{G}$ admits an equilibrium if and only if we can define from it an equivalent satisfiable normalised game $\mathcal{G}'$. We will explicitly show how to build $\mathcal{G}'$ from $\mathcal{G}$. What is interesting about this is the fact that $\mathcal{G}'$ is satisfiable, and in this particular case, every strategy profile that belongs to the set of equilibria satisfies and so maximises each of the payoff functions of the game.

Proof of Theorem 7.1: (1) ⇔ (2)

We now show that the existence of equilibria for an arbitrary finite game is equivalent to the satisfiability of a special finite-valued formula. Notice that for $Ł_k^c$, given the presence of constants in the language, such a formula always exists. In fact, for each variable p, we can encode a valuation $e(p) = \frac{j}{k}$ by using constants through formulae of the form

$$p \leftrightarrow \overline{\frac{j}{k}},$$

which are satisfiable if and only if $e(p)$ does equal $\frac{j}{k}$. Therefore, we can build a formula that expresses the fact that a Nash equilibrium actually exists by encoding all possible strategy profiles and all possible changes of strategy by each player. Still, we are going to show that it is still possible to write such a formula even without truth constants, apart from $\bar{0}$, for both $Ł_k$ and $Ł_k^c$.

In order to show how, we need some preliminary results. We begin by proving that valuations can be encoded by formulae.

LEMMA 7.2. *For every propositional variable p and every valuation $e : \{p\} \to L_k$ there exists a formula $\psi(p)$ of a finite-valued Łukasiewicz logic Ł over L_k such that*

$$e(p) = \frac{j}{k} \qquad \textit{iff} \qquad e(\psi(p)) = 1.$$[12]

PROOF. We assume j and k to be coprime. If that is not the case then we have that

$$e\left(\psi_{\frac{j'}{k'}}(p)\right) = 1 \quad \text{iff} \quad e(p) = \frac{j'}{k'},$$

where j' and k' are coprime and $\frac{j}{k} = \frac{j'}{k'}$.

Let $\mathsf{q}_{j,k}$ and $\mathsf{r}_{j,k}$ denote the quotient and the remainder, respectively, of the Euclidean division of k by j.

If $e(p) = 0$, let

$$\psi_0(p) := \neg p.$$

Then

$$e(p) = 0 \qquad \text{iff} \qquad e(\neg p) = 1.$$

If $e(p) = \frac{1}{k}$, then let

$$\psi_{\frac{1}{k}}(p) := \neg d(\neg((k-1)p), p).$$

It is easy to check that

$$e(p) = \frac{1}{k} \qquad \text{iff} \qquad e(\neg d(\neg((k-1)p), p)) = 1.$$

In fact,

$$e(\neg d(\neg((k-1)p), p)) = 1 - |(1-(k-1)x) - x|$$

and

$$1 - |(1-(k-1)x) - x| = 1 \qquad \text{iff} \qquad x = \frac{1}{k}.$$

For $e(p) = \frac{j}{k}$, with $j \geq 2$, the proof proceeds by induction. For j and k coprime, let

$$\psi_{\frac{j}{k}}(p) = \psi_{(\mathsf{r}_{j,k},k)}(\neg(\mathsf{q}_{j,k}p)),$$

while for j and k not coprime, take j' and k' coprime such that and $\frac{j}{k} = \frac{j'}{k'}$ and let

$$\psi_{\frac{j}{k}}(p) = \psi_{\frac{j'}{k'}}(p) = \psi_{(\mathsf{r}_{j',k'},k')}(\neg(\mathsf{q}_{j',k'}p)).$$

Notice that $\mathsf{r}_{j,k} < j$. So, for instance, if $j = 2$, then

$$\psi_{\frac{2}{k}}(p) := \psi_{\frac{1}{k}}(\neg(\mathsf{q}_{j,k}p)) = \neg d(\neg((k-1)(\neg(\mathsf{q}_{j,k}p))), \neg(\mathsf{q}_{j,k}p)).$$

This concludes the proof of the lemma. □

[12]Notice that the following proof translates into logical terms the algebraic proof of Lemma 19 in Lenzi and Marchioni [2014], whose context and content are significantly different from those of the present article. Also, it is worth pointing out that the same result is a consequence of the McNaughton Theorem. However, we prefer to offer here an independent constructive proof that does not rely on the notion of a McNaughton function.

Now we are ready to show that for each finite game the existence of an equilibrium is equivalent to the existence of a special satisfiable formula ϕ_{eq}. We prove this by giving an explicit construction of ϕ_{eq}. As before, we assume the game to be normalised.

Notice that as an immediate consequence of Lemma 7.2, we have:

COROLLARY 7.3. *In every Łukasiewicz game $\mathcal{G}$ on a finite Łukasiewicz logic* Ł, *for every strategy profile* $(s_1, \ldots, s_n)$ *there exists an* Ł*-formula* ψ *so that*

$$f_\psi(s'_1, \ldots, s'_n) = 1 \qquad \textit{iff} \qquad s_i = s'_i$$

for all i.

In fact, let $\mathsf{V}_i = \{p_{1_i}, \ldots, p_{m_i}\}$ be the set of variables in control of player i, and let

$$(\alpha_{1_i}, \ldots, \alpha_{m_i}) \in (L_k)^{m_i}.$$

The formula

$$\psi_{\alpha_{1_i}}\left(p_{1_i}\right)$$

encodes the assignment by player i of the value α_{1_i} to the variable p_{1_i}, that is,

$$e(\psi_{\alpha_{1_i}}\left(p_{1_i}\right)) = 1 \qquad \text{iff} \qquad e(p_{1_i}) = \alpha_{1_i}.$$

So, the formula

$$\psi_{\alpha_{1_i}}\left(p_{1_i}\right) \wedge \cdots \wedge \psi_{\alpha_{m_i}}\left(p_{m_i}\right)$$

encodes player i's strategy

$$(\alpha_{1_i}, \ldots, \alpha_{m_i}),$$

and the formula

$$\bigwedge_{i=1}^{n}\left(\psi_{\alpha_{1_i}}\left(p_{1_i}\right) \wedge \cdots \wedge \psi_{\alpha_{m_i}}\left(p_{m_i}\right)\right) \tag{1}$$

encodes the strategy profile

$$\left(\alpha_{1_1}, \ldots, \alpha_{m_1}, \ldots, \alpha_{1_i}, \ldots, \alpha_{m_i}, \ldots, \alpha_{1_n}, \ldots, \alpha_{m_n}\right).$$

To avoid any possible confusion, notice that for $j \neq i$, we might have that $m_i \neq m_j$, since i and j might be in control of a different number of variables.

Take, for each player i the set of all strategies

$$\mathsf{S}_i = \left\{s_i \mid s_i = \left(\beta_{1_i}, \ldots, \beta_{m_i}\right) \in (L_k)^{m_i}\right\}.$$

Assign to each player i a new set of variables

$$\mathsf{V}^{s_i} = \left\{q_{1_i}^{\beta_{1_i}}, \ldots, q_{m_i}^{\beta_{m_i}}\right\},$$

for each $s_i \in (L_k)^{m_i}$. This means that if a player controls m_i variables, she has $(k+1)^{m_i}$ different strategy profiles and is therefore assigned $m_i \cdot (k+1)^{m_i}$ new variables.

Proceeding as above take the formula

$$\psi_{\beta_{1_i}}\left(q_{1_i}^{\beta_{1_i}}\right)$$

that encodes the assignment by player i of the value β_{1_i} to the variable $q_{1_i}^{\beta_{1_i}}$, so that the formula

$$\psi_{\beta_{1_i}}\left(q_{1_i}^{\beta_{1_i}}\right) \wedge \cdots \wedge \psi_{\beta_{m_i}}\left(q_{m_i}^{\beta_{m_i}}\right) \tag{2}$$

encodes player i's strategy

$$(\beta_{1_i}, \ldots, \beta_{m_i}).$$

Let

$$\phi_i(p_{1_1}, \ldots, p_{m_1}, \ldots, p_{1_{i-1}}, \ldots, p_{m_{i-1}}, \ldots, p_{1_i}, \ldots, p_{m_i}, \\ p_{1_{i+1}}, \ldots, p_{m_{i+1}}, \ldots p_{1_n}, \ldots, p_{m_n}) \tag{3}$$

be player i's payoff formula, and let

$$\phi_i(p_{1_1}, \ldots, p_{m_1}, \ldots, p_{1_{i-1}}, \ldots, p_{m_{i-1}}, \ldots, q_{1_i}^{\beta_{1_i}}, \ldots, q_{m_i}^{\beta_{m_i}}, \\ p_{1_{i+1}}, \ldots, p_{m_{i+1}}, \ldots p_{1_n}, \ldots, p_{m_n}) \tag{4}$$

be the formula obtained from (3) by replacing the variables

$$\{p_{1_i}, \ldots, p_{m_i}\}$$

with the new variables

$$\left\{q_{1_i}^{\beta_{1_i}}, \ldots, q_{m_i}^{\beta_{m_i}}\right\}.$$

So, using (3) and (4), the satisfiability of the formula

$$\begin{aligned} &\phi_i(p_{1_1}, \ldots, p_{m_1}, \ldots, p_{1_{i-1}}, \ldots, p_{m_{i-1}}, \ldots, q_{1_i}^{\beta_{1_i}}, \ldots, q_{m_i}^{\beta_{m_i}}, \\ &\qquad p_{1_{i+1}}, \ldots, p_{m_{i+1}}, \ldots p_{1_n}, \ldots, p_{m_n}) \to \\ &\phi_i(p_{1_1}, \ldots, p_{m_1}, \ldots, p_{1_{i-1}}, \ldots, p_{m_{i-1}}, \ldots, p_{1_i}, \ldots, p_{m_i}, \\ &\qquad p_{1_{i+1}}, \ldots, p_{m_{i+1}}, \ldots p_{1_n}, \ldots, x_{m_n}) \end{aligned} \tag{5}$$

encodes the fact that player i's payoff does not increase. To simplify the notation, we denote the formula (5) by χ.

Define the formula ϕ_{eq}, where each

$$\vec{s} \in (L_k)^{\sum_{i=1}^{n} m_i}$$

is a strategy profile:

$$\begin{aligned} \phi_{eq} :=& \bigvee_{\vec{s}\in(L_k)^{\sum_{i=1}^{n} m_i}} \Bigg[\bigwedge_{i=1}^{n} \left(\psi_{\alpha_{1_i}}\left(p_{1_i}\right) \wedge \cdots \wedge \psi_{\alpha_{m_i}}\left(p_{m_i}\right)\right) \wedge \\ & \bigwedge_{i=1}^{n} \Bigg[\bigwedge_{s_i \in (L_k)^{m_i}} \left[\psi_{\beta_{1_i}}\left(q_{1_i}^{\beta_{1_i}}\right) \wedge \cdots \wedge \psi_{\beta_{m_i}}\left(q_{m_i}^{\beta_{m_i}}\right) \wedge \chi \right]\Bigg]\Bigg]. \end{aligned} \tag{6}$$

From the above construction, it is easy to check that ϕ_{eq} actually encodes the existence of equilibria. In fact, ϕ_{eq} is a disjunction indexed by all possible strategy profiles. The existence of an equilibrium requires at least one of the disjuncts to be satisfiable. Each disjunct is a conjunction of formulae encoding the requirement that for a given strategy profile and for every player, every change of strategy does not result in any payoff increase. So, if any such disjunct is satisfiable, the related strategy profile actually corresponds to a Nash equilibrium.

LEMMA 7.4. *A finite Łukasiewicz game $\mathcal{G}$ admits a pure strategy Nash equilibrium if and only if ϕ_{eq} is satisfiable.*

Consequently, we have proved the equivalence between the first two conditions of Theorem 7.1.

Proof of Theorem 7.1: (1) ⇔ (3)

We are now going to refine the previous result by showing that we can obtain a formula whose set of satisfiable elements coincides with the set of equilibria of the game. First, we need the following preliminary result that proves that every set definable by a quantifier-free formula in the first-order theory of a standard finite MV-algebra with or without constants can be defined by a formula of the corresponding Łukasiewicz logic.

LEMMA 7.5. *Let* L *be a finite-valued Łukasiewicz logic and let* $\mathbb{L}$ *be its corresponding standard* MV-*algebra. For every quantifier-free formula* $\Phi(x_1, \ldots, x_n)$ *in the language* $\mathcal{L}_{\mathbb{L}}$ *there exists an* L*-formula* $\phi(p_1, \ldots, p_n)$ *such that, for all* $(a_1, \ldots, a_n) \in (L_k)^n$

$$\{(a_1, \ldots, a_n) \mid \mathbb{L} \models \Phi(a_1, \ldots, a_n)\} \quad \textit{iff} \quad (a_1, \ldots, a_n) \in Sat(\phi(p_1, \ldots, p_n)).$$

PROOF. Let $Terms_{\mathbb{L}}$ be the set of terms t in the language $\mathcal{L}_{\mathbb{L}}$ and let $Form_{\mathrm{L}}$ be the set of L-formulae. Define a mapping $\tau : Terms_{\mathbb{L}} \to Form_{\mathrm{L}}$ such that

(1) if t is a variable x, then $\tau(t) = p$;
(2) if t is constant c, then $\tau(t) = \bar{c}$;
(3) if t is $t' \oplus t''$, then $\tau(t) = \tau(t') \oplus \tau(t'')$;
(4) if t is $\neg t'$, then $\tau(t) = \neg\tau(t')$.

Every quantifier-free formula Φ of in $\mathcal{L}_{\mathbb{L}}$ is a Boolean combination of equalities and (strict) inequalities between terms. So, define a new mapping $\lambda : Form_{\mathbb{L}} \to Form_{\mathrm{L}}$, where $Form_{\mathbb{L}}$ is the set if quantifier-free formulae in $\mathcal{L}_{\mathbb{L}}$, as follows:

(1) If Φ is $(t = t')$, then $\lambda(\Phi) = \Delta(\tau(t) \leftrightarrow \tau(t'))$.
(2) If Φ is $(t < t')$, then $\lambda(\Phi) = \neg\Delta(\tau(t') \to \tau(t))$.
(3) If Φ is $\Phi' \sqcap \Phi''$, then $\lambda(\Phi) = \lambda(\Phi') \wedge \lambda(\Phi'')$.
(4) If Φ is $\sim\Phi'$, then $\lambda(\Phi) = \neg\lambda(\Phi')$.

It is easy to check that for every formula $\Phi(x_1, \ldots, x_n)$ and all $(a_1, \ldots, a_n) \in (L_k)^n$

$$\{(a_1, \ldots, a_n) \mid \mathbb{L} \models \Phi(a_1, \ldots, a_n)\} \quad \text{iff} \quad (a_1, \ldots, a_n) \in Sat(\lambda(\Phi(x_1, \ldots, x_n))),$$

and so $\Phi(x_1, \ldots, x_n)$ is satisfiable if and only if so is $\lambda(\Phi(x_1, \ldots, x_n))$. In fact, on the one hand, every function and constant symbol in $\mathcal{L}_{\mathbb{L}}$ has an interpretation in $\mathbb{L}$ corresponding to the interpretation of the related connective and constants in L. On the other hand, the use of the operator Δ forces each formula of the form $\Delta(\tau(t) \leftrightarrow \tau(t'))$ and $\neg\Delta(\tau(t') \to \tau(t))$ to behave like a Boolean formula, making compositions of such formulae into Boolean combinations. □

The next lemma finally shows that the set of equilibria of a finite game can be defined by a formula of the corresponding logic.

LEMMA 7.6. *For every Łukasiewicz game* $\mathcal{G}$ *on a finite-valued Łukasiewicz logic* L *there exists an* L*-formula* $\phi_{\mathsf{NE}}(\vec{p}_1, \ldots, \vec{p}_n)$ *such that, for all* $(\vec{a}_1, \ldots, \vec{a}_n) \in (L_k)^m$

$$(\vec{a}_1, \ldots, \vec{a}_n) \in \mathsf{NE}(\mathcal{G}) \quad \textit{iff} \quad (\vec{a}_1, \ldots, \vec{a}_n) \in Sat(\phi_{\mathsf{NE}}).$$

PROOF. Once again, let L be a finite-valued Łukasiewicz logic and let $\mathbb{L}$ be its corresponding standard MV-algebra.

Take any game $\mathcal{G}$ on L. From Proposition 6.5, we know that $\mathcal{G}$ has a pure strategy Nash equilbrium if and only if Φ_{NE} holds over $\mathbb{L}$. By dropping the existential quantifiers in Φ_{NE}, we obtain the formula

$$\forall \vec{y}_1, \ldots, \vec{y}_n \sqcap_{i=1}^{n} \left(\Phi_i(\vec{x}_1, \ldots, \vec{x}_{i-1}, \vec{y}_i, \vec{x}_{i+1}, \ldots, \vec{x}_n) \leq \Phi_i(\vec{x}_1, \ldots, \vec{x}_{i-1}, \vec{x}_i, \vec{x}_{i+1}, \ldots, \vec{x}_n) \right), \quad (7)$$

so that Φ_{NE} belongs to $\mathsf{Th}(\mathbb{L})$ if and only if the set defined by (7), that is,

$$\Big\{\vec{a}_1,\dots,\vec{a}_n \mid \mathbb{L} \models \forall \vec{y}_1,\dots,\vec{y}_n \ \sqcap_{i=1}^{n} \Big(\Phi_i(\vec{a}_1,\dots,\vec{a}_{i-1},\vec{y}_i,\vec{a}_{i+1},\dots,\vec{a}_n) \leq \Phi_i(\vec{a}_1,\dots,\vec{a}_{i-1},\vec{a}_i,\vec{a}_{i+1},\dots,\vec{a}_n)\Big)\Big\},$$

is not empty.

Each $\mathsf{Th}(\mathbb{L})$ has quantifier elimination in $\mathcal{L}_{\mathbb{L}}$. Hence, there exists a quantifier-free formula

$$\Phi(\vec{x}_1,\dots,\vec{x}_n) \tag{8}$$

that is equivalent to (7), which means that $\mathcal{G}$ has a pure strategy Nash equilibrium if and only if

$$\{\vec{a}_1,\dots,\vec{a}_n \mid \mathbb{L} \models \Phi(\vec{a}_1,\dots,\vec{a}_n)\} \neq \emptyset.$$

Finally, from Lemma 7.5, we know that there exists an L-formula $\lambda(\Phi(\vec{x}_1,\dots,\vec{x}_n))$ whose satisfiability set coincides with the set defined by $\Phi(\vec{x}_1,\dots,\vec{x}_n)$, which is the set of equilibria of the game.

The above lemma proves the equivalence of the first and third conditions in Theorem 7.1.

Proof of Theorem 7.1: (1) ⇔ (4)

As said above, we want to show that every finite game $\mathcal{G}$ admits a pure strategy Nash equilibrium if and only if it is equivalent to a satisfiable game. We prove this in the following lemma by making use of the fact that the set of equilibria can be encoded through the satisfiability set of a finite-valued formula.

LEMMA 7.7. *A finite Łukasiewicz game $\mathcal{G}$ admits a Nash equilibrium if and only if it is equivalent to a normalised satisfiable game.*

PROOF. Let $\mathcal{G}$ be a Łukasiewicz game on any finite Łukasiewicz logic L. One direction is trivial. In fact, if, given $\mathcal{G}$, there exists an equivalent satisfiable game, by definition of equivalence and the fact that each satisfiable game has a Nash equilibrium, that is, Lemma 6.4, we immediately obtain that $\mathcal{G}$ has a Nash equilibrium as well.

We now prove the converse statement. From Lemma 7.6, we know that there is an L-formula $\phi_{\mathsf{NE}}(\vec{p}_1,\dots,\vec{p}_n)$ such that, for all $(\vec{a}_1,\dots,\vec{a}_n) \in (L_k)^m$

$$(\vec{a}_1,\dots,\vec{a}_n) \in \mathsf{NE}(\mathcal{G}) \quad \text{iff} \quad (\vec{a}_1,\dots,\vec{a}_n) \in Sat(\phi_{\mathsf{NE}}).$$

Then, define the following game

$$\mathcal{G}' = \langle \mathsf{P}, \mathsf{V}, \{\mathsf{V}_i\}_{i\in\mathsf{P}}, \{\mathsf{S}_i\}_{i\in\mathsf{P}}, \{\phi'_i\}_{i\in\mathsf{P}}\rangle,$$

where

$$\phi'_i := \phi_{\mathsf{NE}}(\vec{p}_1,\dots,\vec{p}_n) \vee \phi_i.$$

Suppose that $\mathcal{G}$ admits a Nash equilibrium. This means that there exists a strategy profile $(s_1,\dots,s_n)$ that corresponds to a valuation satisfying $\phi_{\mathsf{NE}}(\vec{p}_1,\dots,\vec{p}_n)$, which implies that each

$$\phi_{\mathsf{NE}}(\vec{p}_1,\dots,\vec{p}_n) \vee \phi_i$$

is satisfiable. Consequently, $\mathcal{G}'$ is a satisfiable game and every strategy profile that corresponds to a Nash equilibrium of $\mathcal{G}$ also is an equilibrium for $\mathcal{G}'$.

Take now a strategy profile $(s'_1, \ldots, s'_n)$ that is not an equilibrium for $\mathcal{G}$. This means that the corresponding valuation is such that

$$f_{\phi_{\mathsf{NE}}(\vec{p}_1,\ldots,\vec{p}_n)}(s'_1, \ldots, s'_n) = 0.$$

So, for each player i the payoff given by ϕ'_i corresponds to the payoff given by ϕ_i, which is not a Nash equilibrium.

Therefore, the existence of equilibria for $\mathcal{G}$ is equivalent to the existence of the satisfiable game $\mathcal{G}'$ that is equivalent to $\mathcal{G}$. □

The above lemma proves the equivalence of the first and fourth conditions in Theorem 7.1, whose proof is therefore complete.

8. INFINITE GAMES: LOGICAL CHARACTERISATIONS

This section is devoted to a characterisation of the existence of equilibria for infinite Łukasiewicz games similar to the one given in the previous section.

As mentioned in Section 3, the unary connective Δ is definable in every finite-valued Łukasiewicz logic. The presence of Δ has a central role in proving the results contained in Theorem 7.1. Δ, however, is not realised by any formula in infinite-valued Łukasiewicz logics, since its associated function is not continuous. So, to prove a characterisation of the existence of NE similar to the finite case, we need notions and techniques that are particular to infinite-valued Łukasiewicz logics.[13] Our aim is to prove the following theorem.

THEOREM 8.1. *For every Łukasiewicz game $\mathcal{G}$ on an infinite-valued Łukasiewicz logic* L*, there exists an* L*-formula $\phi_{\mathsf{NE}}(\vec{p}_1, \ldots, \vec{p}_n)$ such that, for all $(\vec{a}_1, \ldots, \vec{a}_n) \in [0, 1]^m$*

$$(\vec{a}_1, \ldots, \vec{a}_n) \in Sat(\phi_{\mathsf{NE}}(\vec{p}_1, \ldots, \vec{p}_n)) \quad \textit{iff} \quad (\vec{a}_1, \ldots, \vec{a}_n) \in \mathsf{NE}(\mathcal{G}),$$

and the following statements are equivalent:

(1) *$\mathcal{G}$ admits a pure strategy Nash equilibrium.*
(2) *$\mathcal{G}$ admits a rational pure strategy Nash equilibrium, i.e.: $\mathsf{NE}(\mathcal{G}) \cap \mathbb{Q}^m \neq \emptyset$.*
(3) *ϕ_{NE} is satisfiable.*

Moreover, for every infinite Łukasiewicz game on either $Ł_\infty$ or $RPŁ_\infty$, the above conditions are equivalent to the following one:

(4) *There exists $k \in \mathbb{N}$ so that, for all $k' \geq k$ such that $L_k \subseteq L_{k'}$, $\mathcal{G}$ has a pure strategy Nash equilibrium on each finite-valued Łukasiewicz logic or finite-valued Łukasiewicz logic with constants, respectively, over $L_{k'}$.*

Proof of Theorem 8.1: (1) ⇔ (2)

Our first step is to prove that if the set of equilibria is not empty, then it must always contain a rational equilibrium, that is, a valuation such that the values are all rational numbers in [0, 1]. This is a consequence of the following lemma.

LEMMA 8.2. *An infinite Łukasiewicz game $\mathcal{G}$ admits a Nash equilibrium if and only if it admits a rational Nash equilibrium.*

[13] It is possible to expand infinite-valued Łukasiewicz logics by adding Δ to their language (see Esteva et al. [2011]). The functions definable in the expanded logics are all piecewise linear polynomial functions with integer and rational coefficients but are not necessarily continuous [Montagna and Panti 2001]. The addition of Δ makes it possible to prove that the equivalence of conditions (1), (3), and (4) from Theorem 7.1 also holds for infinite games. A proof of these facts can be obtained by using the same arguments used in the proof of Theorem 7.1.

Proof. Let $\mathbb{L}$ be any of the standard infinite MV-algebras defined above. By Proposition 6.5, we know that the existence of equilibria can be expressed through a first-order sentence Φ_{NE} of $\mathsf{Th}(\mathbb{L})$. As proved in Baaz and Veith [1999], Caicedo [2007], and Lenzi and Marchioni [2014], the substructure $\mathbb{L}^{\mathbb{Q}}$ of $\mathbb{L}$ in the same signature $\mathcal{L}_{\mathbb{L}}$ defined on $\mathbb{Q} \cap [0, 1]$ is a model of $\mathsf{Th}(\mathbb{L})$ and satisfies the same sentences as $\mathbb{L}$.[14] Therefore, Φ_{NE} is valid for $\mathbb{L}$ if and only if it is valid for $\mathbb{L}^{\mathbb{Q}}$. □

Proof of Theorem 8.1: (1) ⇔ (3)

As shown in the previous section, the set of equilibria for games based on a finite-valued Łukasiewicz logic L can be encoded through the satisfiability set of an L-formula. We are now going to show that the same result holds for games based on infinite-valued logics. The key will be to show that the set of pure strategy Nash equilibria in any infinite Łukasiewicz game is a rational polyhedron. The result will then be a consequence of Lemma 3.3.

Recall that the first-order theory $\mathsf{Th}(\mathbb{R})$ of the ordered group of real numbers is the set of sentences in the language of ordered groups $\langle +, -, 0, < \rangle$ that hold over $\mathbb{R}$ (see Hodges [1993]). $\mathsf{Th}(\mathbb{R})$ has quantifier elimination in $\langle +, -, 0, < \rangle$. Trivially, $\mathsf{Th}(\mathbb{R})$ also admits quantifier elimination in the language $\langle +, -, \{c\}_{c\in\mathbb{Q}}, < \rangle$ obtained by expanding $\langle +, -, 0, < \rangle$ with a constant for every rational number. We show that every quantifier-free formula of a standard infinite MV-algebra can be translated into a formula in the language $\langle +, -, \{c\}_{c\in\mathbb{Q}}, < \rangle$, so that both formulae define the same set.

Lemma 8.3. *Let* L *be any infinite Łukasiewicz logic and* $\mathbb{L}$ *be its corresponding standard* MV-*algebra. For every quantifier-free formula* $\Phi(x_1, \ldots, x_n)$ *in* $\mathcal{L}_{\mathbb{L}}$ *there exists a quantifier-free formula* $\Phi^{\diamond}(x_1, \ldots, x_n)$ *in the language* $\langle +, -, \{c\}_{c\in\mathbb{Q}}, < \rangle$ *such that for all* $(a_1, \ldots, a_n) \in [0, 1]^n$,

$$\mathbb{L} \models \Phi(a_1, \ldots, a_n) \qquad \textit{iff} \qquad \mathbb{R} \models \Phi^{\diamond}(a_1, \ldots, a_n).$$

Proof. Every quantifier-free Łukasiewicz formula $\Phi(x_1, \ldots, x_n)$ is a Boolean combination

$$\mathbf{B}^m_{j=1}(\Theta_j), \tag{9}$$

where each $\Theta_j(x_1, \ldots, x_n)$[15] is an equality or inequality between terms in $\mathcal{L}_{\mathbb{L}}$, that is, it is an atomic formula of the form

$$t \diamond t'$$

with $\diamond \in \{<, >, =\}$.

By an unnested atomic formula in $\mathcal{L}_{\mathbb{L}}$ we mean one of the following formulae:

$$t = t', \qquad t < t' \qquad t = \neg t', \qquad t = t' \oplus t'', \qquad t = \delta_n t',$$

where t, t', t'' are either variables or constants [Hodges 1993]. A formula is called unnested if all its atomic subformulae are unnested. As shown in Hodges [1993, Theorem 2.6.1], every formula in some language $\mathcal{L}$ is logically equivalent to an unnested formula in the same language. So, this result trivially holds for $\mathcal{L}_{\mathbb{L}}$ as well.

Let

$$T = \bigcup_{j=1}^{m} \{t_{1_j}, \ldots, t_{q_j}\}$$

[14] In model-theoretic terms: all the models of $\mathsf{Th}(\mathbb{L})$ are elementarily equivalent to each other, and the structure $\mathbb{L}^{\mathbb{Q}}$ whose domain is the set of rational numbers in $[0, 1]$ is a prime model, that is, it can be elementarily embedded into every other model of the theory.

[15] Without any loss of generality, we can assume that all variables $x_1, \ldots, x_n$ occur in each Θ_j.

be the set of terms in (9), where each t_{i_j} denotes the i_j-th term in atomic formula Θ_j. Assign a new propositional variable z to each element of T (same variables for same terms), and write a new formula for every $t \in T$ as follows:

(1) if t is a variable x, then write $z = x$;
(2) if t is a constant c, then write $z = c$;
(3) if t is $t' \oplus t''$, then write $z = z' \oplus z''$;
(4) if t is $\neg t'$, then write $z = \neg z'$;
(5) if t is $\delta_n t'$, then write $z = \delta_n z'$.

Let $\Psi(\vec{x}, \vec{z})$ be the conjunction of all the previous formulae, where

$$\vec{x} = x_1, \ldots, x_n, \qquad \vec{z} = z_1, \ldots, z_{m'},$$

and m' is the number of newly introduced variables. Define the formula

$$\exists z_1, \ldots \exists z_{m'} \ \Psi(\vec{x}, \vec{z}) \sqcap \mathbf{B}^\star, \tag{10}$$

where $\mathbf{B}^\star$ is the formula obtained from $\mathbf{B}^m_{j=1}(\Theta_j)$ by replacing each atomic formula $t \diamond t'$ with the formula $z \diamond z'$, where z, z' are the variables assigned to t, t', respectively.

We now translate (10) into a formula in the language $\langle +, -, \{c\}_{c \in \mathbb{Q}}, < \rangle$, in the following way:

(1) Replace in (10) every occurrence of unnested atomic formulae with $\oplus$, $\neg$, and δ_n as follows:

$$\begin{array}{rcl} t = \neg t & \mapsto & t = 1 - t', \\ t = t' \oplus t'' & \mapsto & ((t' + t'' \leq 1) \sqcap (t = t' + t'')) \sqcup ((t' + t'' > 1) \sqcap (t = 1), \\ t = \delta_n t' & \mapsto & \underbrace{t + \cdots + t}_{n} = t'. \end{array}$$

The newly introduced formulae define the graphs of the basic Łukasiewicz functions in $\mathbb{R}$, for all the values in $[0, 1]$.

(2) Let Ψ' be the conjunction of formulae of the form

$$(0 \leq z) \sqcap (z \leq 1)$$

for every newly introduced variable z.

(3) Write the following formula:

$$\exists z_1, \ldots \exists z_{m'} \ \Psi' \sqcap [(\Psi(\vec{x}, \vec{z}) \sqcap \mathbf{B}^\star)]. \tag{11}$$

It is easy to see that (11) is a formula in $\langle +, -, \{c\}_{c \in \mathbb{Q}}, < \rangle$ such that, for all $(a_1, \ldots, a_n) \in [0, 1]^n$

$$\mathbb{L} \models \Phi(a_1, \ldots, a_n) \quad \text{iff} \quad \mathbb{R} \models \exists z_1, \ldots \exists z_{m'} \ \Psi' \sqcap [(\Psi(\vec{a}, \vec{z}) \sqcap \mathbf{B}^\star)].$$

As mentioned above $\mathsf{Th}(\mathbb{R})$ admits quantifier elimination in $\langle +, -, \{c\}_{c \in \mathbb{Q}}, < \rangle$, and so there exists a quantifier-free formula $\Phi^\diamond(x_1, \ldots, x_n)$ that is equivalent to (11).

Consequently, for all $(a_1, \ldots, a_n) \in [0, 1]^n$,

$$\mathbb{L} \models \Phi(a_1, \ldots, a_n) \qquad \text{iff} \qquad \mathbb{R} \models \Phi^\diamond(a_1, \ldots, a_n).$$

An easy inspection of the above proof shows that the translation of any quantifier-free Łukasiewicz formula (9) into (11) requires only polynomial time in the length of (9).

The following example illustrates how the above translation works.

Example 8.4. Consider the following quantifier-free formula

$$(x \oplus x) \oplus y < \frac{1}{2}.$$

The set of its subterms is the following:

$$T = \left\{ x,\ y,\ x \oplus x,\ (x \oplus x) \oplus y,\ \frac{1}{2} \right\}.$$

Assign new variables to each subterm as follows:

$$x \mapsto z_1, \qquad y \mapsto z_2, \qquad x \oplus x \mapsto z_3, \qquad (x \oplus x) \oplus y \mapsto z_4, \qquad \frac{1}{2} \mapsto z_5.$$

Write the following unnested formula, equivalent to the one we started with:

$$\exists z_1 \exists z_2 \exists z_3 \exists z_4 \exists z_5 \ \Bigg[(x = z_1) \sqcap (y = z_2) \sqcap (z_1 \oplus z_1 = z_3) \sqcap (z_3 \oplus z_2 = z_4) \sqcap \left(\tfrac{1}{2} = z_5 \right) \Bigg] \sqcap (z_4 < z_5).$$

Finally, replace the unnested atomic formulae where the symbol $\oplus$ appears with their corresponding formulae in $\langle +, -, \{c\}_{c \in \mathbb{Q}}, < \rangle$, and add the conjunction of the formulae that specify that each variable must belong to $[0, 1]$:

$$\begin{aligned}\exists z_1 \exists z_2 \exists z_3 \exists z_4 \exists z_5 \ & \left[\bigsqcap_{i=1}^{5} ((0 \le z_i) \sqcap (z_i \le 1) \right] \sqcap \Bigg[(x = z_1) \sqcap (y = z_2) \sqcap \\ & \Big(((z_1 + z_1 \le 1) \sqcap (z_3 = z_1 + z_1)) \sqcup ((z_1 + z_1 > 1) \sqcap (z_3 = 1)) \Big) \sqcap \\ & \Big(((z_3 + z_2 \le 1) \sqcap (z_4 = z_3 + z_2)) \sqcup ((z_3 + z_2 > 1) \sqcap (z_4 = 1)) \Big) \sqcap \\ & \left(\frac{1}{2} = z_5 \right) \Bigg] \sqcap \big(z_4 < z_5 \big).\end{aligned}$$

This concludes the example.

We now use Lemma 8.3 to prove that the set of equilibria of any infinite Łukasiewicz game can be encoded through the satisfiability set of a Łukasiewicz formula.

LEMMA 8.5. *For every Łukasiewicz game $\mathcal{G}$ on an infinite Łukasiewicz logic* Ł*, there exists an* Ł*-formula $\phi_{\mathsf{NE}}(\vec{p}_1, \ldots, \vec{p}_n)$ such that, for all $(\vec{a}_1, \ldots, \vec{a}_n) \in [0, 1]^m$*

$$(\vec{a}_1, \ldots, \vec{a}_n) \in Sat(\phi_{\mathsf{NE}}(\vec{p}_1, \ldots, \vec{p}_n)) \quad \textit{iff} \quad (\vec{a}_1, \ldots, \vec{a}_n) \in \mathsf{NE}(\mathcal{G}).$$

Consequently, $\mathcal{G}$ admits a pure strategy Nash equilibrium if and only if $\phi_{\mathsf{NE}}(\vec{p}_1, \ldots, \vec{p}_n)$ is satisfiable.

PROOF. Reasoning as in Lemma 7.7, the set of equilibria of $\mathcal{G}$ coincides with the set defined by the formula

$$\forall \vec{y}_1, \ldots, \vec{y}_n \prod_{i=1}^{n} \big(\Phi_i(\vec{x}_1, \ldots, \vec{x}_{i-1}, \vec{y}_i, \vec{x}_{i+1}, \ldots, \vec{x}_n) \le \Phi_i(\vec{x}_1, \ldots, \vec{x}_{i-1}, \vec{x}_i, \vec{x}_{i+1}, \ldots, \vec{x}_n) \big). \quad (12)$$

$\mathsf{Th}(\mathbb{L})$ has quantifier elimination in the related language. Hence, there exists a quantifier-free formula $\Phi(\vec{x}_1, \ldots, \vec{x}_n)$ that is equivalent to (12).

As proved in the previous lemma, there exists a quantifier-free formula $\Phi^{\diamond}(\vec{x}_1, \ldots, \vec{x}_n)$ in the language $\langle +, -, \{c\}_{c \in \mathbb{Q}}, < \rangle$ such that, for all $(\vec{a}_1, \ldots, \vec{a}_n) \in [0, 1]^m$:

$$\mathbb{L} \models \Phi(\vec{a}_1, \ldots, \vec{a}_n) \qquad \text{iff} \qquad \mathbb{R} \models \Phi^{\diamond}(\vec{a}_1, \ldots, \vec{a}_n).$$

This means that both $\Phi(\vec{x}_1, \dots, \vec{x}_n)$ and $\Phi^\diamond(\vec{x}_1, \dots, \vec{x}_n)$ define the same set of elements of $[0, 1]^m$.

The formula $\Phi^\diamond(\vec{x}_1, \dots, \vec{x}_n)$ can be equivalently rewritten in disjunctive normal form, i.e.:

$$\bigsqcup_{i=1}^{s} \left(\prod_{j=1}^{r} \mathbf{p}_{ij}(\vec{x}_1, \dots, \vec{x}_n) \triangledown 0 \right)$$

where each

$$\mathbf{p}_{ij}(\vec{x}_1, \dots, \vec{x}_n) \triangledown 0$$

is a linear polynomial inequality[16] with integer coefficients and

$$\triangledown \in \{<, >, \leq, \geq\}.$$

The set of solutions $X_i \subseteq [0, 1]^m$ of each conjunction

$$\prod_{j=1}^{r} \mathbf{p}_{ij}(\vec{x}_1, \dots, \vec{x}_n) \triangledown 0$$

is a rational polyhedron. This means that the set of solutions of $\Phi^\diamond(\vec{x}_1, \dots, \vec{x}_n)$ is a finite union

$$X = \bigcup_{i=1}^{s} X_i$$

of rational polyhedra X_i, which also is a rational polyhedron.

Consequently, the set of equilibria of $\mathcal{G}$ is a rational polyhedron, since

$$(\vec{a}_1, \dots, \vec{a}_n) \in \mathsf{NE}(\mathcal{G}) \qquad \text{iff} \qquad (\vec{a}_1, \dots, \vec{a}_n) \in X$$

for all $(\vec{a}_1, \dots, \vec{a}_n) \in [0, 1]^m$.

By Lemma 3.3, there exists a Łukasiewicz formula $\phi_{\mathsf{NE}}(\vec{p}_1, \dots, \vec{p}_n)$ such that, for all $(\vec{a}_1, \dots, \vec{a}_n) \in [0, 1]^m$:

$$(\vec{a}_1, \dots, \vec{a}_n) \in Sat(\phi_{\mathsf{NE}}(\vec{p}_1, \dots, \vec{p}_n)) \qquad \text{iff} \qquad (\vec{a}_1, \dots, \vec{a}_n) \in X.$$

Therefore,

$$(\vec{a}_1, \dots, \vec{a}_n) \in Sat(\phi_{\mathsf{NE}}(\vec{p}_1, \dots, \vec{p}_n)) \qquad \text{iff} \qquad (\vec{a}_1, \dots, \vec{a}_n) \in \mathsf{NE}(\mathcal{G}),$$

and so $\phi_{\mathsf{NE}}(\vec{p}_1, \dots, \vec{p}_n)$ is satisfiable if and only if $\mathcal{G}$ admits an equilibrium. □

As an immediate consequence of this result, we obtain:

COROLLARY 8.6. *For any infinite Łukasiewicz game $\mathcal{G}$, the set of pure strategy Nash equilibria is a rational polyhedron.*

Proof of Theorem 8.1: (1) ⇔ (4)

We now conclude the proof of Theorem 8.1 and show that an infinite game $\mathcal{G}$ based on $Ł_\infty$ or $RPŁ_\infty$ has an equilibrium if and only if $\mathcal{G}$ has an equilibrium on some finite logic L on L_k, and there exist infinitely many versions of the same game, each on some finite logic L′ on $L_{k'}$, such that $L_k \subseteq L_{k'}$, also having an equilibrium.

[16]Without any loss of generality, we can assume that each polynomial includes occurrences of all variables $\vec{x}_1, \dots, \vec{x}_n$.

LEMMA 8.7. *An infinite Łukasiewicz game $\mathcal{G}$ on either $Ł_\infty$ or $RPŁ_\infty$ admits a pure strategy Nash equilibrium if and only if there exists some $k \in \mathbb{N}$ such that, for all $k' \geq k$ so that $L_k \subseteq L_{k'}$, $\mathcal{G}$ admits an equilibrium on $Ł_{k'}$ ($Ł^c_{k'}$, respectively).*

PROOF. We give a proof for $RPŁ_\infty$. The proof for $Ł_\infty$ is essentially identical.

Suppose $\mathcal{G}$ on $RPŁ_\infty$ has an equilibrium. Then by Lemma 8.2, $\mathcal{G}$ must admit a rational equilibrium $\vec{a}_1, \ldots, \vec{a}_n$. This means that the following universal sentence holds on the standard MV-algebra with constants $\mathbb{MV}^{\mathbb{Q}}_\infty$, whose domain is $\mathbb{Q} \cap [0, 1]$:

$$\forall \vec{y}_1, \ldots, \vec{y}_n \sqcap_{i=1}^n \left(\Phi_i(\vec{a}_1, \ldots, \vec{a}_{i-1}, \vec{y}_i, \vec{a}_{i+1}, \ldots, \vec{a}_n) \leq \Phi_i(\vec{a}_1, \ldots, \vec{a}_{i-1}, \vec{a}_i, \vec{a}_{i+1}, \ldots, \vec{a}_n)\right). \quad (13)$$

Let $\mathbb{MV}^c_k$ be the finite standard MV-algebra with constants generated by $\vec{a}_1, \ldots, \vec{a}_n$ and the remaining rational constants occurring in (13), i.e., the structure obtained by taking the closure under the operations $\oplus$ and $\neg$ of the set including $\vec{a}_1, \ldots, \vec{a}_n$ and the rational constants in (13). Since (13) is a universal sentence, it trivially holds for $\mathbb{MV}^c_k$, and for every $\mathbb{MV}^c_{k'}$ such that $L_k \subseteq L_{k'}$. Therefore, if $\mathcal{G}$ on $RPŁ_\infty$ has an equilibrium, it also has an equilibrium on every $Ł^c_{k'}$ obtained as above.

Conversely, suppose that $\mathcal{G}$ does not have an equilibrium. This means that for every rational tuple $\vec{a}_1, \ldots, \vec{a}_n$, it is always possible to find rationals $\vec{b}_1, \ldots, \vec{b}_n$ such that

$$\prod_{i=1}^{n} \left(\Phi_i(\vec{a}_1, \ldots, \vec{a}_{i-1}, \vec{b}_i, \vec{a}_{i+1}, \ldots, \vec{a}_n) > \Phi_i(\vec{a}_1, \ldots, \vec{a}_{i-1}, \vec{a}_i, \vec{a}_{i+1}, \ldots, \vec{a}_n)\right) \quad (14)$$

holds over the standard finite MV-algebra with constants generated by $\vec{a}_1, \ldots, \vec{a}_n, \vec{b}_1, \ldots, \vec{b}_n$ plus the remaining constants in (14). In other words, for every k and standard finite MV-algebra with constants $\mathbb{MV}^c_k$, we can always find some k' and $\mathbb{MV}^c_{k'}$ such that $L_k \subseteq L_{k'}$ and (14) holds over $\mathbb{MV}^c_{k'}$. □

The key idea in the above proof is that the formulae (13) and (14) can be evaluated over a standard finite MV-algebra. For $\mathbb{MV}_\infty$ and $\mathbb{MV}_k$, this is possible because they have the same language (and so do their corresponding logics). In the case of $\mathbb{MV}^c_\infty$, there obviously is no finite standard MV-algebra with the same language, since $\mathbb{MV}^c_\infty$ contains all rational constants in $[0, 1]$. Still, as the proof of Lemma 8.7 shows, it suffices to take the standard finite MV-algebra $\mathbb{MV}^c_k$ whose domain is the closure of the set of elements that satisfy (13) and (14) along with the rational constants appearing in such formulae. (13) and (14) can then be properly evaluated over $\mathbb{MV}^c_k$, since this structure contains all the constants occurring in both formulae. It is worth pointing out though, that Lemma 8.7 does not hold for $RŁ_\infty$. In fact, if (13) and (14) contain occurrences of the divisibility operators δ_n, they cannot be evaluated over a standard finite MV-algebra, since δ_n is not definable in the finite setting.

9. INFINITE GAMES: ADDITIONAL RESULTS

In this section, we provide some additional results concerning infinite Łukasiewicz games. We start by showing that there is a special class of games that always admits a NE. This is the class of Łukasiewicz games whose payoff formulae are built only from literals and the operators $\oplus, \wedge, \delta_n$ (up to provable equivalence). Next, we show that Łukasiewicz games based on rational McNaughton functions are expressive enough to capture an approximate notion of equilibria in all games based on continuous functions with values from $[0, 1]$.

9.1. Łukasiewicz Games with Concave McNaughton Functions

Recall that a function $f : \mathbb{R}^n \to \mathbb{R}$ is called concave in a variable x_i if, for all

$$(a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_n) \in \mathbb{R}^{n-1},$$

the set

$$\left\{(b, c) \in \mathbb{R}^2 \mid c \leq f(a_1, \ldots, a_{i-1}, b, a_{i+1}, \ldots, a_n)\right\},$$

called the *hypograph* of $f(a_1, \ldots, a_{i-1}, x_i, a_{i+1}, \ldots, a_n)$, is convex.

We call a propositional variable p, a constant $\overline{c}$, or their negation $\neg p, \neg\overline{c}$, a *literal*. Any $\phi(p_1, \ldots, p_n)$ provably equivalent to an L-formula only built from literals with the connectives

$$\oplus, \quad \wedge, \quad \{\delta_n\}_{n>1},$$

is called a $\langle\oplus, \wedge, \delta_n\rangle$-formula.

LEMMA 9.1. *Let* L *be any infinite Łukasiewicz logic and let* $\phi(p_1, \ldots, p_n)$ *be any* $\langle\oplus, \wedge, \delta_n\rangle$*-formula of* L. *Then* $f_\phi : [0, 1]^n \to [0, 1]$ *is concave in each* x_i.

PROOF. First, notice that for any Łukasiewicz literal l, the associated function f_l is obviously concave. The sum and minimum of two concave functions are concave and so is the product of a concave function by a rational number. So for all literals l, l', the formulae

$$l \oplus l', \qquad l \wedge l', \qquad \delta_n l$$

define concave McNaughton functions.

The result then follows by induction on the complexity of the formulae, since, by assumption, all formulae must be provably equivalent to an expression of the following form:

$$\phi \oplus \phi', \qquad \phi \wedge \phi', \qquad \delta_n \phi,$$

where ϕ, ϕ' are $\langle\oplus, \wedge, \delta_n\rangle$-formulae. □

We call an infinite Łukasiewicz game a $\langle\oplus, \wedge, \delta_n\rangle$ game whenever the following conditions are satisfied, for all $1 \leq i \leq n$:

(1) $\mathsf{V} = \{p_1, \ldots, p_n\}$.
(2) $\mathsf{V}_i = \{p_i\}$.
(3) ϕ_i is a $\langle\oplus, \wedge, \delta_n\rangle$-formula.

We want to show that every $\langle\oplus, \wedge, \delta_n\rangle$-game admits an equilibrium. This will be an easy consequence of the following general result:

THEOREM 9.2 ([NIKAIDÔ AND ISODA 1955]). *Let* $\mathcal{G}$ *be an n-person game where*

(1) *each player i's strategy space* S_i *is a nonempty, convex, compact subset of a finite dimensional Euclidean space;*
(2) *each payoff function* $f_i(x_1, \ldots, x_n)$ *is continuous;*
(3) *each payoff function* $f_i(x_1, \ldots, x_n)$ *is concave in* x_i.

Then $\mathcal{G}$ *always admits a pure strategy Nash equilibrium.*

We can now prove:

PROPOSITION 9.3. *Every infinite Łukasiewicz* $\langle\oplus, \wedge, \delta_n\rangle$*-game* $\mathcal{G}$ *admits a pure strategy Nash equilibrium.*

PROOF. The result is an easy consequence of Theorem 9.2. In fact, each player's strategy space corresponds to $[0, 1]$, which of course is a convex compact subset of $\mathbb{R}$. Moreover, each payoff function is trivially continuous, being a McNaughton function. Finally, the fact that each payoff function $f_i(x_1, \ldots, x_n)$ is concave in x_i, follows by Lemma 9.1. □

9.2. Approximation of Continuous Games

Łukasiewicz games provide a game-theoretic model that is more expressive than Boolean games. However, this model only captures games with payoff functions given by piecewise linear polynomials. This is certainly a limitation of the model, and many other classes of games cannot be represented in this framework. Still, rational McNaughton functions can be used to approximate any continuous function, as shown in Aguzzoli and Mundici [2003] and Amato and Porto [2000].

THEOREM 9.4 ([AMATO AND PORTO 2000]). *Let $g : [0,1]^n \to [0,1]$ be a continuous function and let $0 < \epsilon \in \mathbb{R}$. There exists a rational McNaughton function $f : [0,1]^n \to [0,1]$ such that $|f(\vec{x}) - g(\vec{x})| < \epsilon$ for every $\vec{x} \in [0,1]^n$.*

A rational McNaughton function that satisfies the above property is said to *ϵ-approximate g*. We exploit the above result to show that we can capture an approximate notion of equilibrium in games based on any continuous functions over $[0,1]^n$.

A continuous game $\mathcal{C}$ is given by a structure

$$\mathcal{C} = \langle \mathsf{P}, \mathsf{V}, \{\mathsf{V}_i\}_{i\in\mathsf{P}}, \{\mathsf{S}_i\}_{i\in\mathsf{P}}, \{g_i\}_{i\in\mathsf{P}}\rangle,$$

where

(1) $\mathsf{P} = \{P_1, \ldots, P_n\}$ is a finite set of *players*;
(2) $\mathsf{V} = \{x_1, \ldots, x_m\}$ is a finite set of variables taking values from $[0,1]$;
(3) $\mathsf{V}_i \subseteq \mathsf{V}$ is the set of variables under control of player P_i, so that the sets V_i form a partition of V;
(4) S_i is the strategy set for player i that includes all mappings $s_i : \mathsf{V}_i \to [0,1]$ of the variables in V_i, that is,

$$\mathsf{S}_i = \{s_i \mid s_i : \mathsf{V}_i \to [0,1]\};$$

(5) $g_i : [0,1]^m \to [0,1]$ is a continuous payoff function.

Infinite Łukasiewicz games can be seen to be a special class of continuous games.

Recall that, for any $0 < \epsilon \in \mathbb{R}$, a continuous game $\mathcal{C}$ is said to have an *ϵ-equilibrium* if there exists a strategy profile $(\vec{a}_1, \ldots, \vec{a}_n) \in [0,1]^m$ such that for every player i there is no strategy $\vec{b}_i$ such that

$$g_i(\vec{a}_1, \ldots, \vec{a}_{i-1}, \vec{b}, \vec{a}_{i+1}, \ldots, \vec{a}_n) - g_i(\vec{a}_1, \ldots, \vec{a}_n) \geq \epsilon.$$

The notion of ϵ-equilibrium weakens the usual notion of pure strategy Nash equilibrium by tolerating deviations in the outcome when a player changes her strategy. Still, these deviations are accepted only when the difference between resulting payoff and the original is strictly smaller than ϵ.

We can use this notion to show that Łukasiewicz games on $\mathrm{RŁ}_\infty$ can be used to approximate continuous games and also their ϵ-equilibria in the sense detailed by the following proposition.

PROPOSITION 9.5. *Let $\mathcal{C}$ be a continuous game. Then, for all $\epsilon \in (0,1]$, there exists a Łukasiewicz game $\mathcal{G}$ on $\mathrm{RŁ}_\infty$ such that for all strategy profiles $\vec{a} = (\vec{a}_1, \ldots, \vec{a}_n) \in [0,1]^m$,*

(1) *if $\vec{a}$ is an ϵ-equilibrium for $\mathcal{C}$, then it is a 2ϵ-equilibrium for $\mathcal{G}$;*
(2) *if $\vec{a}$ is an ϵ-equilibrium for $\mathcal{G}$, then it is a 2ϵ-equilibrium for $\mathcal{C}$.*

PROOF. We only prove (1), since the proof of (2) is essentially identical.

Fix an $\epsilon \in (0,1]$ and define a Łukasiewicz game $\mathcal{G}$ with the same number of players, same number of variables, and same distribution of variables among the players as in $\mathcal{C}$. Then, for each player i, take the formula ϕ_i, whose corresponding rational McNaughton function f_{ϕ_i} is such that, for all $(\vec{x}_1, \ldots, \vec{x}_n) \in [0, 1]^m$

$$|f_{\phi_i}(\vec{x}_1, \ldots, \vec{x}_n) - g(\vec{x}_1, \ldots, \vec{x}_n)| < \frac{\epsilon}{2}.$$

This means that each payoff function f_i in the Łukasiewicz game $\mathcal{G}$ $\frac{\epsilon}{2}$-approximate the corresponding function g_i in $\mathcal{C}$.

Let $\vec{a} = (\vec{a}_1, \ldots, \vec{a}_n) \in [0, 1]^m$ be an ϵ-equilibrium for $\mathcal{C}$. Then for every player i and every strategy $\vec{b}$

$$g_i(\vec{a}_1, \ldots, \vec{a}_{i-1}, \vec{b}, \vec{a}_{i+1}, \ldots, \vec{a}_n) - g_i(\vec{a}_1, \ldots, \vec{a}_n) < \epsilon.$$

This means that for every i in $\mathcal{G}$ and every strategy $\vec{b}$

$$f_i(\vec{a}_1, \ldots, \vec{a}_{i-1}, \vec{b}, \vec{a}_{i+1}, \ldots, \vec{a}_n) - f_i(\vec{a}_1, \ldots, \vec{a}_n) < 2\epsilon,$$

since in the worst case scenario

$$g_i(\vec{a}_1, \ldots, \vec{b}, \ldots, \vec{a}_n) + \frac{\epsilon}{2} > f_i(\vec{a}_1, \ldots, \vec{b}, \ldots, \vec{a}_n) > g_i(\vec{a}_1, \ldots, \vec{b}, \ldots, \vec{a}_n)$$

and

$$g_i(\vec{a}_1, \ldots, \vec{a}_n) > f_i(\vec{a}_1, \ldots, \vec{a}_n) > g_i(\vec{a}_1, \ldots, \vec{a}_n) - \frac{\epsilon}{2},$$

so the difference between $f_i(\vec{a}_1, \ldots, \vec{a}_{i-1}, \vec{b}, \vec{a}_{i+1}, \ldots, \vec{a}_n)$ and $f_i(\vec{a}_1, \ldots, \vec{a}_n)$ cannot be greater than 2ϵ.

Therefore, $\vec{a} = (\vec{a}_1, \ldots, \vec{a}_n) \in [0, 1]^m$, is a 2ϵ-equilibrium for $\mathcal{G}$. □

10. COMPLEXITY

In this section, we study the complexity of deciding whether a Łukasiewicz game admits an equilibrium or whether a certain strategy profile does belong to the set of equilibria.

Recall that, for any standard MV-algebra $\mathbb{L}$, a quantified Łukasiewicz sentence has the form

$$Q_1 x_1 \ldots Q_n x_n \; \Phi(x_1, \ldots, x_n),$$

where $\Phi(x_1, \ldots, x_n)$ is a Boolean combination of equalities and inequalities in the language $\mathcal{L}_{\mathbb{L}}$, and each Q_i is either an existential or universal quantifier. If all Q_i's are existential quantifiers, we say the sentence is *existential*.

As we have seen in the previous sections, the existence of equilibria can be encoded through a quantified Łukasiewicz sentence. We then study the general problem of determining the validity of this kind of formula with respect to the related standard MV-algebra. We start with the finite case.

THEOREM 10.1. *Let $\mathbb{L}$ be any standard finite* MV-*algebra. Checking whether a quantified Łukasiewicz sentence in $\mathcal{L}_{\mathbb{L}}$ belongs to* Th($\mathbb{L}$) *is* PSPACE-*complete. Checking whether an existential Łukasiewicz sentence in $\mathcal{L}_{\mathbb{L}}$ belongs to* Th($\mathbb{L}$) *is* NP-*complete.*

PROOF. Let Φ be a quantified Łukasiewicz sentence with n variables. PSPACE-containment can be proved with an argument very similar to the one given for quantified Boolean formulae (see Arora and Barak [2009]). We describe a recursive procedure to determine the validity of Φ that requires only an amount of space that is polynomial in the number of variables and the size m of the formula.

If $n = 0$, then there are no variables, so Φ only contains constants. Clearly, in this case, checking the validity of Φ can be simply computed in polynomial time.

So, assume that $n > 0$. Let, for each $r \in L_k$, Φ_r be the formula obtained by eliminating the first quantifier Q_1 and replacing all occurrences of x_1 in Φ with r. If Q_1 is an existential quantifier $\exists$, then the algorithm outputs 1 if, when applied to the formula Φ_r, it also outputs 1 for at least some $r \in L_k$. If Q_1 is a universal quantifier $\forall$, then the algorithm outputs 1 if, when applied to the formula Φ_r, it also outputs 1 for all $r \in L_k$. The algorithm then runs recursively by reusing the same space and retaining only the output of a specific computation.

To prove hardness, take any quantified Boolean formula

$$Q_1 x_1 \dots Q_n x_n \ \Phi(x_1, \dots, x_n). \tag{15}$$

For each variable x_i, let

$$\Psi_0^1(x_i) := (x_i = 0) \sqcup (x_i = 1).$$

Rename $y_1, \dots, y_m$ and $z_1, \dots, z_{n-m}$ the variables from $\{x_1, \dots, x_n\}$ that are under the scope of a universal and an existential quantifier, respectively. Rewrite the above formula as follows:

$$Q_1 \dots Q_n \ \left(\prod_{j=1}^{n-m} \Psi_0^1(z_j)\right) \sqcap \left(\left(\prod_{i=1}^{m} \Psi_0^1(y_i)\right) \Rightarrow \Phi(y_1, \dots, y_m, z_1, \dots, z_{n-m})\right), \tag{16}$$

where $Q_1 \dots Q_n$ is the sequence of quantifiers with the same alternation as in (15) obtained by replacing the previous variables with the new ones.

It is clear that (16) can be obtained in polynomial time from (15). In addition, it is obvious by construction that (16) is valid over L_k if and only if (15) is valid over $\{0, 1\}$. In fact, notice that the values in (16) are restricted to only $\{0, 1\}$, and, under this restriction, the Łukasiewicz operations behave in the same way as the Boolean ones.

As for the existential case, take a sentence of the form

$$\exists_1 x_1 \dots \exists_n x_n \ \Phi(x_1, \dots, x_n)$$

and guess a tuple $(a_1, \dots, a_n) \in (L_k)^n$. Checking if

$$\mathbb{L} \models \Phi(a_1, \dots, a_n)$$

requires computing whether the equalities and inequalities in $\Phi(x_1, \dots, x_n)$ are true or false for the values $a_1, \dots, a_n$, and computing the value of the resulting Boolean combination. This can be done in polynomial time.

Hardness can be proved similar to the case of quantified sentences by exploiting the fact that checking the validity of an existential Boolean formula is in NP. □

We now focus on the problem of deciding quantified Łukasiewicz sentences for the infinite case.

THEOREM 10.2. *Let $\mathbb{L}$ be any standard infinite* MV-*algebra. Checking whether a quantified Łukasiewicz sentence in $\mathcal{L}_{\mathbb{L}}$ belongs to* $\mathsf{Th}(\mathbb{L})$ *is in* 2-EXPTIME. *Checking whether an existential Łukasiewicz sentence in $\mathcal{L}_{\mathbb{L}}$ belongs to* $\mathsf{Th}(\mathbb{L})$ *is in* NP.

PROOF. Take a quantified Łukasiewicz formula

$$Q_1 x_1 \dots Q_n x_n \ \Phi(x_1, \dots, x_n). \tag{17}$$

$\Phi(x_1, \dots, x_n)$ is a Boolean combination

$$\mathbf{B}_{j=1}^{m}(\Theta_j), \tag{18}$$

where each $\Theta_j(x_1, \dots, x_n)$ is an equality or inequality between terms in $\mathcal{L}_{\mathbb{L}}$.

Following the same reasoning as in Lemma 8.3, the formula

$$\exists z_1, \dots \exists z_{m'} \ Q_1 x_1 \dots Q_n x_n \ \Psi' \sqcap [(\Psi(\vec{x}, \vec{z}) \sqcap \mathbf{B}^\star)] \tag{19}$$

is a formula in $\langle +, -, \{c\}_{c\in\mathbb{Q}}, < \rangle$ that holds over $\mathbb{R}$ if and only if (17) holds over $\mathbb{L}$. Moreover, it is easy to check that (19) can be obtained in polynomial time in the length of (17).

As shown in Ferrante and Rackoff [1975], deciding the validity of an arbitrary quantified formula of length n in the theory $\mathsf{Th}(\mathbb{R})$ of the ordered group of real numbers requires at most deterministic time $2^{2^{pn}}$, for some fixed constant $c < 0$. Moreover, checking the validity of an existential sentence in $\mathsf{Th}(\mathbb{R})$ can be done in nondeterministic polynomial time (see Bradley and Manna [2007]). $\square$

We can now easily apply the previous discussion to determining the complexity of some game-theoretic problems.

Definition 10.3. For a Łukasiewicz game $\mathcal{G}$, the MEMBERSHIP problem is the problem of determining whether a rational strategy profile $(s_1, \ldots, s_n)$ belongs to the set of pure strategy Nash equilibria. The NONEMPTINESS problem is the problem of determining if the set of pure strategy Nash equilibria is not empty.

Given the previous results, it is then easy to prove the following theorem.

THEOREM 10.4. *For all Łukasiewicz games the* MEMBERSHIP *problem is in* CO-NP. *For finite Łukasiewicz games the* NONEMPTINESS *problem is in* PSPACE*, while for infinite Łukasiewicz games the* NONEMPTINESS *problem is in* 2-EXPTIME.

PROOF. By Proposition 6.5, the existence of equilibria can be expressed through a first-order sentence. Similarly, whether a rational strategy profile $\vec{a}_1, \ldots, \vec{a}_n \in \mathbb{Q}^m$ belongs to the set of equilibria can be eoncoded by the universal sentence

$$\forall \vec{y}_1, \ldots, \vec{y}_n \prod_{i=1}^{n} \Big(\Phi_i(\vec{a}_1, \ldots, \vec{a}_{i-1}, \vec{y}_i, \vec{a}_{i+1}, \ldots, \vec{a}_n) \leq \Phi_i(\vec{a}_1, \ldots, \vec{a}_{i-1}, \vec{a}_i, \vec{a}_{i+1}, \ldots, \vec{a}_n) \Big).$$

Therefore, the result follows from Theorem 10.1 and Theorem 10.2 . $\square$

11. CONCLUSIONS

A key challenge in the use of logic for game-theoretic reasoning is the development of techniques for representing the preferences or utilities of agents. In particular, the most widely used logic-based models for games assume that player's preferences are defined by logically specified Boolean goals. For many domains, this approach, leading to dichotomous preferences, is too simple. By using Łukasiewicz logics to specify player goals/utility functions, as we do in Łukasiewicz games, it becomes possible to express much richer preferences and utility functions, while at the same time staying within the attractive purely logical framework offered by Boolean games. In this article, we hope to have demonstrated the value of this approach.

At this point, we should point out some possible links to related work, which may prove fertile ground for future research. One avenue to explore further is the link between Boolean games and logic programs. To pick one example, De Vos and Vermeir [1999] discuss the relationship between Nash equilibria and a semantic model for logic programs, namely, stable model semantics. Now, given that there has been much work recently on Łukasiewicz versions of logic programming (see, e.g., Schockaert et al. [2009, 2012]), it is natural to ask whether such results can be recast in a Łukasiewicz logic programming framework. For example, disjunctive linear programs have the same expressive power as Łukasiewicz logic, and this therefore suggests several novel approaches to Łukasiewicz games: for example, we might investigate the possibility of allowing a user to specify a game in Łukasiewicz logic, and then automatically translate this specification into disjunctive logic programming (or vice versa). With respect to solving games, given the natural connection between Łukasiewicz logics and linear polynomial functions, we expect to use some tools and techniques from linear and mixed integer programming for this task.

Many other avenues suggest themselves for future research. Extensions to *cooperative games* are one natural avenue for investigation. Another is to consider *iterated* games, in which players repeatedly meet in a strategic scenario: in this case, a key issue will be how to lift utility functions from individual games to sequences of games.

REFERENCES

S. Aguzzoli, S. Bova, and B. Gerla. 2011. Free algebras and functional representation for fuzzy logics. In *Handbook of Mathematical Fuzzy Logic*, Volume II, P. Cintula, P. Hájek, and C. Noguera (Eds.), College Publications, 713–792.

S. Aguzzoli and D. Mundici. 2003. Weierstrass approximation theorem and Lukasiewicz formulas with one quantified variable. In *Beyond Two: Theory and Applications of Multiple-Valued Logic*, M. Fitting et al. (Eds.), Springer-Verlag, Berlin, 315–335.

R. Alur and T. A. Henzinger. 1999. Reactive modules. *Formal Methods in System Design* 15, 11 (1999), 7–48.

P. Amato and M. Porto. 2000. An algorithm for the automatic generation of a logical formula representing a control law. *Neural Network World* 10 (2000), 777–786.

S. P. Anderson, J. K. Goereeb, and C. A. Holt. 2001. Minimum-effort coordination games: Stochastic potential and logit equilibrium. *Games and Economic Behavior* 34, 2 (2001), 177–199.

S. Arora and B. Barak. 2009. *Computational Complexity. A Modern Approach.* Cambridge University Press.

M. Baaz and H. Veith. 1999. Quantifier elimination in fuzzy logic. In *Computer Science Logic*, Lecture Notes in Computer Science, Vol. 1584, Springer-Verlag, Berlin, 399–414.

K. Basu. 1994. The traveler's dilemma: Paradoxes of rationality in game theory. *American Economic Review* 84, 2 (1994), 391–395.

E. Bonzon, M.-C. Lagasquie-Schiex, J. Lang, and B. Zanuttini. 2006. Boolean games revisited. In *Proceedings of the 17th European Conference on Artificial Intelligence (ECAI-2006)*.

E. Bonzon, M.-C. Lagasquie-Schiex, J. Lang, and B. Zanuttini. 2006. Compact preference representation and Boolean games. In *PRICAI 2006: Trends in Artificial Intelligence*, Lecture Notes in Computer Science, Vol. 4099, Springer-Verlag, Berlin, 41–50.

C. Boutilier, R. Brafman, C. Domshlak, H. Hoos, and D. Poole. 2004. CP-nets: A tool for representing and reasoning with conditional Ceteris Paribus preference statements. In *Journal of Artificial Intelligence Research* 21 (2004), 135–191.

A. R. Bradley and Z. Manna. 2007. *The Calculus of Computation*. Springer-Verlag, Berlin.

N. Bulling and V. Goranko. 2013. How to be both rich and happy: Combining quantitative and qualitative strategic reasoning about multi-player games. In *Electronic Proceedings in Theoretical Computer Science* 112 (2013), 33–41.

X. Caicedo. 2007. Implicit operations in DMV-algebras and the connectives of Łukasiewicz logic. In *Algebraic and Proof-Theoretic Aspects of Non-Classical Logics*, S. Aguzzoli, A. Ciabattoni, B. Gerla, C. Manara, and V. Marra (Eds.), Lecture Notes in Computer Science, Vol. 4460, Springer-Verlag, Berlin, 50–68.

R. Cignoli, I. M. L. D'Ottaviano, and D. Mundici. 2000. *Algebraic Foundations of Many-Valued Reasoning* (Trends in Logic Volume 7). Kluwer Academic Publishers, Dordrecht, The Netherlands.

M. De Vos and D. Vermeir. 1999. Choice logic programs and Nash equilibria in strategic games. In *Computer Science Logic*, Lecture Notes in Computer Science, Vol. 1683, Springer, Berlin, 266–276.

A. Di Nola and I. Leustean. 2011. Łukasiewicz logic and DMV-algebras. In *Handbook of Mathematical Fuzzy Logic*, Volume II, P. Cintula, P. Hájek, and C. Noguera (Eds.), College Publications, 469–584.

P. E. Dunne, S. Kraus, W. van der Hoek, and M. Wooldridge. 2008. Cooperative boolean games. In *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2008)*.

F. Esteva, L. Godo, and E. Marchioni. 2011. Fuzzy logics with enriched language. In *Handbook of Mathematical Fuzzy Logic*, Volume II, P. Cintula, P. Hájek, and C. Noguera (Eds.), College Publications, 627–712.

J. Ferrante and C. Rackoff. 1975. A decision procedure for the first order theory of real addition with order. *SIAM Journal on Computing* 4, 1 (1975), 69–76.

B. Gerla. 2001. *Many-Valued Logics of Continuous t-Norms and Their Functional Representation*. Ph.D. thesis, University of Milan, Italy.

M. Ghallab, D. Nau, and P. Traverso. 2004. *Automated Planning*. Morgan Kaufmann.

J. Grant, S. Kraus, M. Wooldridge, and I. Zuckerman. 2011. Manipulating boolean games through communication. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI-11)*.

P. Hájek. 1998. *Metamathematics of Fuzzy Logic* (Trends in Logic Volume 4). Kluwer Academic Publishers, Dordrecht, The Netherlands.

P. Harrenstein, W. van der Hoek, J.-J. Ch. Meyer, and C. Witteveen. 2001. Boolean games. In *Proceedings of the 8th Conference on Theoretical Aspects of Rationality and Knowledge (TARK VIII)*, J. van Benthem (Ed.), 287–298.

W. Hodges. 1993. *Model Theory*, Volume 42 of Encyclopedia of Mathematics and Its Applications. Cambridge University Press.

T. Kroupa and O. Majer. 2014. Optimal strategic reasoning with McNaughton functions. *International Journal of Approximate Reasoning* 55, 6 (2014), 1458–1468.

G. Lenzi and E. Marchioni. 2014. An algebraic characterization of o-minimal and weakly o-minimal MV-chains. *Journal of Pure and Applied Algebra* 218 (2014), 90–100.

V. Levit, T. Grinshpoun, and A. Meisels. 2013a. Boolean games for charging electric vehicles. In *Proceedings of the 2013 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technology (IAT)*.

V. Levit, T. Grinshpoun, A. Meisels, and A. Bazzan. 2013b. Taxation search in boolean games. In *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS'13)*.

E. Marchioni and M. Wooldridge. 2014. Łukasiewicz games. In *Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems (AAMAS'14)*. 837–844.

M. Maschler, E. Solan, and S. Zamir. 2013. *Game Theory*. Cambridge University Press.

M. Mavronicolas, B. Monien, and K. W. Wagner. 2007. Weighted boolean formula games. In *Internet and Network Economics*, Lecture Notes in Computer Science. Vol. 4858, Springer, Berlin, 469–481.

R. McNaughton. 1951. A theorem about infinite-valued sentential logic. *Journal of Symbolic Logic* 12, 1 (1951), 1–13.

F. Montagna and G. Panti. 2001. Adding structures to MV-algebras. *Journal of Pure and Applied Algebra* 164 (2001), 365–387.

D. Mundici. 1994. A constructive proof of McNaughton's theorem in infinite-valued logic. *Journal of Symbolic Logic* 59, 2 (1994), 596–602.

D. Mundici. 2011. *Advanced Łukasiewicz Calculus and MV-Algebras* (Trends in Logic Volume 35). Springer, The Netherlands.

H. Nikaidô and K. Isoda. 1955. Note on non-cooperative convex game. *Pacific Journal of Mathematics* 5, 5 (1955), 807–815.

N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani. 2007. *Algorithmic Game Theory*. Cambridge University Press.

M. J. Osborne and A. Rubinstein. 1994. *A Course in Game Theory*. MIT Press.

S. Parsons, J. A. Rodríguez-Aguilar, and M. Klein. 2011. Auctions and bidding: A guide for computer scientists. *ACM Computing Surveys* 43, 2, Article 10 (2011).

J. Pavelka. 1979. On fuzzy logic I, II, III. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik* 25 (1979), 45–52, 119–134, 447–464.

S. Schockaert, J. Janssen, D. Vermeir, and M. De Cock. 2009. Answer sets in a fuzzy equilibrium logic. In *Web Reasoning and Rule Systems*, Lecture Notes in Computer Science, Vol. 5837, Springer, Berlin, 135–149.

S. Schockaert, J. Janssen, and D. Vermeir. 2012. Fuzzy equilibrium logic: Declarative problem solving in continuous domains. *ACM Transactions on Computational Logic* 13, 4, Article 33.

J. R. Stallings 1967. *Lectures on Polyhedral Topology*. Tata Institute of Fundamental Research, Mumbay, India.

J. van Benthem. 2014. *Logic in Games*. MIT Press.

J. B. Van Huyck, R. C. Battalio, and R. O. Beil 1990. Tacit coordination games, strategic uncertainty, and coordination failure. *The American Economic Review* 80, 1 (1990), 234–248.

M. Wooldridge, U. Endriss, S. Kraus, and J. Lang. 2013. Incentive engineering for Boolean games. *Artificial Intelligence* 195 (2013), 418–439.

Received August 2014; revised March 2015; accepted May 2015