Łukasiewicz Games

Enrico Marchioni
Institut de Recherche en Informatique de Toulouse
Université Paul Sabatier, France
enrico.marchioni@irit.fr

Michael Wooldridge
Dept of Computer Science
University of Oxford, UK
mjw@cs.ox.ac.uk

ABSTRACT

Boolean games provide a simple, compact, and theoretically attractive abstract model for studying multi-agent interactions in settings where players will act strategically in an attempt to achieve personal goals. A standard critique of Boolean games, however, is that the binary nature of goals (satisfied or unsatisfied) inevitably trivialises the nature of such strategic interactions: a player is assumed to be indifferent between all outcomes that satisfy his goal, and indifferent between all outcomes that do not satisfy his goal.

In this paper, we introduce Łukasiewicz Games, which overcome this limitation by considering goals to be specified using finitely-valued Łukasiewicz logics. The significance of this is that formulae of Łukasiewicz logic can express every continuous piecewise linear polynomial function with integer coefficients over \([0, 1]^n\), thereby allowing goal formulae in Łukasiewicz Games to naturally express a much richer range of utility functions. 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 the properties of Łukasiewicz Games. In particular, after investigating the complexity of decision problems in Łukasiewicz Games, we give a logical characterisation of the existence of Nash equilibria in such games.

Categories and Subject Descriptors

I.2.4 [Distributed Artificial Intelligence]: Multiagent Systems
I.2.11 [Knowledge representation formalisms and methods]

General Terms

Theory

Keywords

Boolean games, Łukasiewicz Games, Łukasiewicz logic

1. INTRODUCTION

Boolean games provide a simple, compact, and theoretically attractive abstract model for studying multi-agent interactions in settings where players will act strategically in an attempt to achieve private goals. 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 a gross simplification for many situations, a concern which led researchers to extend the original Boolean games model with costs, leading to richer and more realistic preference structures for agents [11]. While these refinements make it possible to model much richer types of interaction, the inherently dichotomous nature of preferences in Boolean games is surely one of their most debated features, and the work in the present paper is directly motivated by this limitation.

Specifically, we introduce Łukasiewicz Games, which overcome this limitation of Boolean games by allowing goals to be specified as formulae of finitely-valued Łukasiewicz logics [3]. The rationale for using Łukasiewicz logic in this way is given by the McNaughton Theorem, which says that every continuous piecewise linear polynomial function with integer coefficients over \([0, 1]^n\) can be expressed as a formula of Łukasiewicz logic in \(n\) variables (see, e.g., [3]). Thus, Łukasiewicz logic provides a natural, compact, formally well-defined and expressive logical representation language for payoff functions, allowing much richer preference structures than is easily possible in conventional Boolean games.

The present paper makes three main contributions. First, we provide a formal definition of Łukasiewicz Games. Second, we present a number of detailed worked examples, which illustrate how a range of strategic scenarios can be formalized within this framework. In particular, we argue, these scenarios cannot naturally be formalized using conventional Boolean games. Third, we investigate properties of Łukasiewicz Games. We show that despite their expressive power, the key decision problems for Łukasiewicz Games are no more complex than for conventional Boolean games; in addition, we give a logical characterisation for the existence of equilibria in Łukasiewicz Games.

In the remainder of the paper, when we refer to Łukasiewicz logic, it should be understood that we are referring specifically to finitely-valued Łukasiewicz logics, as distinct from the many other extensions of Łukasiewicz logic that have been investigated in the literature [3].
2. PRELIMINARY DEFINITIONS

Since Łukasiewicz logic is fundamental to our present work, but is not as widely known as the classical logics that underpin conventional Boolean games, we begin by introducing the concepts of Łukasiewicz logic that are used in the remainder of the paper. The language of Łukasiewicz logic $L_k$ is built from a countable set of variables $V = \{p_1, p_2, \ldots\}$, the binary connective $\to$, and the truth constant $\bar{0}$ (for falsity). Further connectives are defined as follows:

- $\neg \varphi$ is $\varphi \to 0$.
- $\varphi \land \psi$ is $\varphi \land ((\varphi \to \psi) \to (\psi \to \varphi))$.
- $\varphi \lor \psi$ is $\neg (\neg \varphi \land \neg \psi)$.
- $\varphi \rightarrow \psi$ is $\varphi \land (\psi \to (\varphi \to \psi))$.

We also write $\varphi \land \cdots \land \varphi$, with $n \geq 1$.

Let $\text{Form}$ denote the set of formulae of Łukasiewicz logic. A valuation $e$, is a mapping $e : V \to [0,1]$, which assigns to all propositional variables a value from the real unit interval; we require that $e(\bar{0}) = 0$. The semantics of Łukasiewicz logic are 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 $\to$ operator (as we can define the remaining operators in terms of this operator and $\bar{0}$), we present the complete ruleset in the interest of clarity:

- $e(\varphi \to \psi) = \min(1 - e(\varphi) + e(\psi), 1)$
- $e(\neg \varphi) = 1 - e(\varphi)$
- $e(\varphi \land \psi) = \max(0, e(\varphi) + e(\psi) - 1)$
- $e(\varphi \lor \psi) = \max(1, e(\varphi) + e(\psi))$
- $e(\varphi \land \psi) = \min(e(\varphi), e(\psi))$
- $e(\varphi \rightarrow \psi) = \max(0, e(\varphi) - e(\psi))$
- $e(\varphi \lor \psi) = \max(e(\varphi), e(\psi))$
- $e(\varphi \leftrightarrow \psi) = 1 - |e(\varphi) - e(\psi)|$
- $e(\varphi(\psi)) = \max(0, e(\varphi) - e(\psi))$

A valuation $e$ is a model for a formula $\varphi$ if $e(\varphi) = 1$. A valuation $e$ is a model for a theory $T$, if $e(\varphi) = 1$, for every $\varphi \in T$.

In this paper, we restrict our attention to finite-valued Łukasiewicz logics $L_k$. In such logics, it is assumed that the domain is a set of $k$ elements $\mathbb{N}_k = \{0, \frac{1}{k}, \ldots, \frac{k-1}{k}, 1\}$. The notions of valuation and model for $L_k$ are defined analogously just replacing $[0,1]$ by $\mathbb{N}_k$ as set of truth values for the logic, that is, valuations are functions with the signature $e : V \to \mathbb{N}_k$. It is sometimes useful to introduce constants in addition to $\bar{0}$ that will denote values in the domain $\mathbb{N}_k$. Specifically, we will denote by $L_{\bar{0}}$ the Łukasiewicz logic obtained by adding constants $\mathbb{N}_k$ for every value $e \in L_{\bar{0}}$. We assume that valuation functions $e$ interpret such constants in the natural way: $e(\mathbb{N}_k) = e$.

A well-known result by McNaughton established that every continuous piecewise linear polynomial function with integer coefficients over $[0,1]^n$ (called a McNaughton function) is definable by a formula in Łukasiewicz logic [3].

In the case of finite-valued Łukasiewicz logics $L_{\bar{0}}$, the functions defined by a formula are a combination of the restrictions of McNaughton functions over $(L_{\bar{0}})^n$ and the constant functions for each $e \in L_{\bar{0}}$. Notice that the class of functions definable by $L_{\bar{0}}$-formulas exactly coincides with the class of all functions $f : (L_{\bar{0}})^n \to L_{\bar{0}}$, for every $n \geq 0$. In this sense, we can associate to every formula $\varphi(p_1, \ldots, p_n)$ from $L_{\bar{0}}$ a function $f_\varphi : (L_{\bar{0}})^n \to L_{\bar{0}}$.

Finally, note that the satisfiability problem for finite Łukasiewicz logics is no more complex than for classical propositional logic: the problem is NP-complete [7,3].

3. ŁUKASIEWICZ GAMES

We now introduce the framework of Łukasiewicz games. First, let $V = \{p_1, \ldots, p_m\}$ be a finite set of propositional variables, as above. Our games are populated by a finite set of players $P = \{P_1, \ldots, P_n\}$ (also referred to as “agents”). Note that throughout this paper, we assume that $|P| = n$. Each player $P_i$ controls a subset of propositional variables $V_i \subseteq V$, so that the sets $V_i$ form a partition of $V$. The fact that player $P_i$ is in control of the set $V_i$ of propositional variables 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 $L_k = \{0, \frac{1}{k}, \ldots, \frac{k-1}{k}, 1\}$.

A strategy for an agent $P_i$ is a function $s_i : V_i \to L_k$, 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 $e : V \to L_k$ and vice versa; we find it convenient to abuse notation a little by treating strategy profiles as valuations and valuations as strategy profiles.

As in conventional Boolean games, we assume that each player is associated with a $L_k$-formula $\varphi_i$, with propositional variables from $V$, whose valuation is interpreted as the payoff function for player $P_i$. That is, the player $P_i$ seeks a valuation $e$ that maximises the value of the corresponding function $f_{\varphi_i}$. Of course, not all the variables in $\varphi_i$ 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.

Łukasiewicz Games can be seen as a generalization of Boolean Games [14, 2], since the latter are obviously a special case of the former. However, Łukasiewicz Games also incorporate the notion of cost/effort, given that each player’s strategic choice can be seen as an assignment to each controlled variable carrying an intrinsic cost. This makes it possible to formalise situations in which agents aim at a better tradeoff between the costs of making certain choices and the resulting payoff.

Example 1. Consider the following example with two players $P_1$ and $P_2$, who control variables $\{x\}$ and $\{y, z\}$, respectively. The payoffs are given by the following formulae:

$\varphi_1 := x \land \neg y \land z$ and $\varphi_2 := x \lor z$.

In order for $P_2$ to maximize the payoff, it suffices to assign $s_2(z) = 1$, no matter what $P_1$ does. Player $P_2$, however, has no chance of maximizing the payoff except by assigning $s_1(x) = 1$ and hoping that $s_2(y) = 1$ and $s_2(y) = 0$. The latter makes sense, since, given $s_2(z) = 1$, any assignment to $y$ does not change the value of $\varphi_2$, so $P_2$ has no incentive in assigning any other value to $y$ but $0$ and raising the costs. Still, $P_2$ knows that $P_1$ needs $s_1(x) = 1$ and consequently can get the maximum payoff by simply keeping the costs at minimum with $s_2(z) = 0$ and $s_2(y) = 0$.

We now formally define Łukasiewicz games.

Definition 1 (Łukasiewicz Games). A Łukasiewicz game $G$ on $L_{\bar{0}}$ is given by a structure:

$G = \langle P, V, \{V_i\}, \{S_i\}, \{\varphi_i\} \rangle$

where:

1. $P = \{P_1, \ldots, P_n\}$ is a finite set of players.
2. $V = \{p_1, \ldots, p_m\}$ is a finite set of propositional variables taking values from

$L_{\bar{0}} = \{0, \frac{1}{k}, \ldots, \frac{k-1}{k}, 1\}$.
3. \( V_i \subseteq V \) is the set of propositional variables under control of player \( P_i \), so that the sets \( V_i \) form a partition of \( V \), i.e.:
\[
V = \bigcup_{i=1}^{n} V_i, \quad \text{and for} \ i \neq j, V_i \cap V_j = \emptyset.
\]

4. \( S_i \) is the strategy set for player \( i \) that includes all valuations \( s_i : V_i \rightarrow L_k \) of the propositional variables in \( V_i \), i.e.
\[
S_i = \{ s_i \mid s_i : V_i \rightarrow L_k \}.
\]

5. \( \varphi_i \) is a \( L_k^\omega \)-formula built from variables in \( V \), whose associated function
\[
f_{\varphi_i}(s_1, \ldots, s_n) \quad \text{and} \quad f_{\varphi_i}(s_i, s_{-i})
\]
corresponds to the payoff function (also called utility function) of \( P_i \), and whose value will be determined by the valuations in \( \{ S_1, \ldots, S_n \} \).

A strategy profile \( \bar{s} \) for \( G \) is a tuple \( \bar{s} = (s_1, \ldots, s_n) \), with each \( s_i \in S_i \) being the strategy selection for the corresponding player in \( G \). Given a strategy \( s_i \) for \( P_i \), we denote by \( s_{-i} \) the collection of strategies \( (s_1, \ldots, s_{i-1}, s_{i+1}, \ldots, s_n) \) not including \( s_i \), and \( S_{-i} \) is the set of all \( s_{-i} \). With an abuse of notation, we use
\[
f_{\varphi_i}(s_1, \ldots, s_n) \quad \text{and} \quad f_{\varphi_i}(s_i, s_{-i})
\]
to denote \( P_i \)’s payoff under the strategy profile \( (s_1, \ldots, s_n) \): recall that \( \varphi_i \) defines a payoff function \( f_{\varphi_i} \), and a strategy profile \( (s_1, \ldots, s_n) \) corresponds to a valuation \( e : V \rightarrow L_k \).

We now introduce the concepts of dominance, best response and Nash equilibrium adapted to our framework. In what follows, we always assume we are working with an arbitrary \( n \)-player Lukasiewicz game \( G \) on \( L_k^\omega \).

**Definition 2. (Dominance).** Let \( G \) be a Lukasiewicz game. A strategy \( s_i \in S_i \) for player \( P_i \) is called strictly dominated (weakly dominated) if there exists a strategy \( s_i' \in S_i \) such that for all \( s_{-i} \in S_{-i} \),
\[
f_{\varphi_i}(s_i, s_{-i}) < f_{\varphi_i}(s_i', s_{-i}) \quad \text{or} \quad f_{\varphi_i}(s_i, s_{-i}) \leq f_{\varphi_i}(s_i', s_{-i}), \quad \text{resp.}
\]

**Definition 3. (Best Response).** Let \( G \) be Lukasiewicz game and let \( (s_1, \ldots, s_n) \) be a strategy profile for \( G \). The strategy \( s_i \) for \( P_i \) is called a best response whenever, fixing \( s_{-i} \), there exists no strategy \( s_i' \) such that
\[
f_{\varphi_i}(s_i, s_{-i}) > f_{\varphi_i}(s_i', s_{-i}).
\]

**Definition 4. (Pure Strategy Nash Equilibrium).** Let \( G \) be a Lukasiewicz game. A strategy profile \( (s_1, \ldots, s_n) \) is called a pure strategy Nash Equilibrium (NE) for \( G \) whenever \( s_i^* \) is a best response to \( s_{-i} \) for each \( 1 \leq i \leq n \).

Complexity results are not at all the main focus of the present paper, but it is natural to ask whether the enriched framework of Lukasiewicz games leads to a blow-up in computational complexity compared with conventional Boolean games. We will assume in what follows that we are working with games defined on a fixed domain \( L_k \) for \( k \in \mathbb{N} \) such that \( k > 0 \). Let us refer to the Evaluation Problem for Lukasiewicz games as the problem of computing, for a given valuation function \( e : V \rightarrow L_k \), and a formula \( \varphi \) of Lukasiewicz logic on \( L_k \), the value \( e(\varphi) \). Since we are dealing with rationals in a fixed domain, this problem is trivially solvable in polynomial time. The Membership problem is the problem of determining, for a given Lukasiewicz game \( G \) and strategy profile \( \bar{s} \) for \( G \), whether \( \bar{s} \) forms a pure strategy Nash equilibrium for \( G \).

The NON-EMPTINESS problem is the problem of determining for a given game \( G \) whether there exist any pure strategy Nash equilibria for the game. The problems are co-NP-complete and \( \Sigma_2^P \)-complete for conventional Boolean games. And, perhaps surprisingly, we have:

**Theorem 1.** The Membership and Non-Emptiness problems for Lukasiewicz games are \( \text{co-NP-complete} \) and \( \Sigma_2^P \)-complete, respectively.

The proof of this can be adapted from proofs of the corresponding results for conventional Boolean games: the key point is that strategies for Lukasiewicz games are “small witnesses” in the terminology of computational complexity. A strategy for a player simply consists of a value from the domain \( L_k \) for every variable controlled by that player. Since values in \( L_k \) are rational numbers that can be assumed to be expressed in binary, they have a compact representation.

4. **Examples**

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

4.1 Generalized Matching Pennies

The first game we present is a generalization of Matching Pennies\(^3\), 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.

Now, 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
\[
S_1 = \{ s_1 \mid s_1 : \{ p_1 \} \rightarrow L_k \}, \quad S_2 = \{ s_2 \mid s_2 : \{ p_2 \} \rightarrow L_k \}.
\]

Now, recall that the Lukasiewicz logic expression \( d(p_1, p_2) \), defined in Section 2, gives the difference between its arguments \( p_1 \) and \( p_2 \). Using this expression, we can define the payoff function for \( P_1 \) as the formula \( \neg d(p_1, p_2) \), while \( P_2 \)'s payoff is defined by the formula \( d(p_1, p_2) \). The game is formally defined as follows: \( G = \{ (p_1, p_2), \{ p_1 \}, \{ p_2 \}, S_1, S_2, \neg d(p_1, p_2), d(p_1, p_2) \} \).

Table 1 shows the payoff matrix for this generalized version of Matching Pennies with \( k = 5 \).

4.2 Weak-Link Games

Weak-link games\(^4\) are a class of coordination games, where the players benefit from mutually coordinating on the same strategy. The original version of the game (see [4]) 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:
\[
u_i(x_1, \ldots, x_n) = a + b \cdot \min(x_1, \ldots, x_n) - c \cdot (x_i - \min(x_1, \ldots, x_n)),
\]
where \( x_i \) is the choice made by player \( i \), and \( a, b, c \) are positive.

\(^3\) Note that a similar generalization, based on infinite-valued Lukasiewicz logic, was first presented in [9].

\(^4\) Nothing to do with the popular TV game show “The Weakest Link”.

839
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.

Notice that in weak-link games, the payoff function’s domain is a strict subset of the function’s range. Therefore, we introduce a variant of such games where domain and range coincide in order to make it possible to define a representation in terms of Łukasiewicz logic Ł-formulae.

We formalize $n$-player weak-link games in the following game:

$$\mathcal{G} = (\mathcal{P}, \mathcal{V}, \{V_i\}, \{S_i\}, \{\varphi_i\})$$

where:

1. $\mathcal{V} = \{p_1, \ldots, p_n\}$.
2. $V_i = \{p_i\}$.
3. The strategy space is defined as follows, for each $i$:
   $$S_i = \{s_i : s_i : \{p_i\} \rightarrow L_k\}.$$  
4. Each player $P_i$’s utility is given by the formula
   $$\varphi_i(p_1, \ldots, p_n) := \bigwedge_{j=1}^n p_j \oplus \left(p_i \bigwedge_{j=1}^n p_j\right)$$
   whose associated payoff function is
   $$f_{\varphi_i} = \max\left(\min_{j=1}^n x_j - \max_{j=1}^n x_j, 0\right), 0).$$

Table 2 shows the payoff matrix for $P_1$ for the two-player version of the weak-link game with $k = 7$ ($P_2$’s payoff is obtained by replacing $P_1$ by $P_2$ and vice versa), where

$$\varphi(p_1, p_2) := (p_1 \land p_2) \lor (p_1 \land (p_1 \land p_2))$$

with payoff function

$$f_{\varphi} = \max(\min(x_1, x_2) - \max(x_1 - \min(x_1, x_2), 0), 0).$$

Notice that in this game there is no strictly dominated strategy, while the strategy $s_1(p_1) = 0$ is weakly dominated.

### 4.3 Traveler’s Dilemma

The Traveler’s Dilemma was introduced by Basu [1] in order to illustrate the tension between the rational solution suggested by the existence of a Nash Equilibrium, and apparently reasonable behavior based on intuition. We introduce (a slightly modified variant of) the Dilemma and show how to formalize 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. An agent of the airline promises a refund for the inconvenience, but, not knowing the exact value of the objects, she puts forward the following proposal. Both travelers must privately write down on paper a number between 0 and 100, corresponding to the cost of the antiques. 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 minus one unit (with a minimum of 0). 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 (with a maximum of 100). The other player, who is regarded by the agent as dishonest, will receive $x$ with a penalty of two units (with a minimum of 0).

Payoffs in the Traveler’s Dilemma are defined by the following function, whose payoff matrix is shown in Table 3:

$$f(x, y) = \begin{cases} 
\max(x - 1, 0) & x = y \\
\min(x + 2, 100) & x < y \\
\max(y - 2, 0) & y < x 
\end{cases}$$

Given that each player wants to maximize her/his payoff, what choices should they make? If both travelers choose 100, they both get 99, which is almost the exact value of their items. However, each traveler soon realizes 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 100. 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 98. 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 0, thus gaining nothing in the process. The strategy profile $c$ is the unique pure strategy Nash equilibrium of the game. This, however, clearly clashes with what intuition would lead us to accept as a rational choice. It seems unreasonable that two individuals would follow the above line of reasoning and rationally come to the conclusion that the best solution is ending up empty-handed.

The Traveler’s Dilemma can be formalized as a Łukasiewicz game over $L_{100}$ as follows. Define the following game

$$\mathcal{G} = (\{T_1, T_2\}, \{p_1, p_2\}, \{s_1\}, \{s_2\}, \{\varphi(p_1, p_2), \varphi_2(p_1, p_2)\}),$$

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 $T_2$; the payoff formulas are defined as follows$^5$

$^5$The connective $\Delta$ is defined as follows in each finite-valued Łukasiewicz logic $L_k$:

$$\Delta\varphi := \neg(k(\neg\varphi))$$
These objective formulae define the following payoff functions:

$$\varphi_1(p_1, p_2) := (p_1 \leftrightarrow p_2) \land (p_1 \lor \frac{1}{100}) \lor 
(p_2 \rightarrow p_1) \land (p_2 \lor \frac{1}{100}) \lor ,
\neg (p_1 \rightarrow p_2) \land (p_2 \lor \frac{1}{100})
$$

$$\varphi_2(p_1, q_2) := (p_1 \leftrightarrow p_2) \land (p_1 \lor \frac{1}{100}) \lor 
(p_2 \rightarrow p_1) \land (p_2 \lor \frac{1}{100}) \lor ,
\neg (p_1 \rightarrow p_2) \land (p_2 \lor \frac{1}{100})
$$

These objective formulae define the following payoff functions:

$$f_{\varphi_1}(x_1, x_2) = \begin{cases} 
\max (x_1 - \frac{1}{100}, 0) & x_1 = x_2 \\
\min (x_1 + \frac{1}{100}, 1) & x_1 < x_2 \\
\max (x_2 - \frac{1}{100}, 0) & x_2 < x_1
\end{cases}
$$

$$f_{\varphi_2}(x_1, x_2) = \begin{cases} 
\max (x_1 - \frac{1}{100}, 0) & x_1 = x_2 \\
\min (x_2 + \frac{1}{100}, 1) & x_2 < x_1 \\
\max (x_2 - \frac{1}{100}, 0) & x_2 < x_1
\end{cases}
$$

Both $f_{\varphi_1}$ and $f_{\varphi_2}$ can be easily seen to be the linear transformation into $L_{100}$ of the payoff function for the Traveler's Dilemma over the set $\{0, 1, \ldots, 99, 100\}$.

### 5. Characterising Nash Equilibria

The aim of this section is to provide different characterizations of the existence of pure strategy Nash equilibria in Łukasiewicz games. We first need some additional terminology and preliminary results. In particular, we need payoff functions to be defined exactly on the same domain and have as inputs not only the same set of variables but the whole set of variables occurring in a game. This requirement leads to introducing the concept of a normalized Łukasiewicz game.

**Definition 5** (Normalized Game). A Łukasiewicz game $G = \langle P, V, \{V_i\}, \{S_i\}, \{\varphi_i\} \rangle$, where $V = \{p_1, \ldots, p_m\}$, is called normalized whenever each payoff function $\varphi_i$ is of the form $\varphi_i(p_1, \ldots, p_m)$, i.e., all the variables from $V$ occur in each $\varphi_i$.

As mentioned above, not every Łukasiewicz game is normalized, since each payoff function might contain a different subset of variables. However, we now show that each game can be transformed into a normalized one. This amounts to showing that any payoff formula can be rewritten in an equivalent form in the whole set $V$ of variables.

**Definition 6.** Let $\varphi(p_1, \ldots, p_m)$ be a $L_k$-formula. We say that $\varphi(p_1, \ldots, p_m)$ has an equivalent extension $\varphi^e(p_1, \ldots, p_m, q_1, \ldots, q_v)$ in $\{q_1, \ldots, q_v\}$, if, for every $(a_1, \ldots, a_m) \in (L_k)^m$,

$$f_{\varphi}(a_1, \ldots, a_m) = f_{\varphi^e}(a_1, \ldots, a_m, b_1, \ldots, b_v)$$

for all $(b_1, \ldots, b_v) \in (L_k)^v$.

The next Lemma shows that for any arbitrary Łukasiewicz formula over $L_k$, we can always find an equivalent extension in any set of variables.

**Lemma 1.** Let $\varphi(p_1, \ldots, p_m)$ be a be any $L_k$-formula. For any set of variables $\{q_1, \ldots, q_v\}$, there exists an equivalent extension of $\varphi(p_1, \ldots, p_m)$ in $\{q_1, \ldots, q_v\}$.
The above Lemma makes it possible to assume every Łukasiewicz game is normalized. To make this precise, we introduce first a notion of equivalence for games.

**Definition 7 (Equivalent Games).** Let \( G = \langle P, V, \{V_i\}, \{S_i\}, \{\varphi_i\}\rangle \) and \( G' = \langle P', V', \{V'_i\}, \{S'_i\}, \{\varphi'_i\}\rangle \) be two Łukasiewicz games over \( L_k^n \). We say that \( G \) and \( G' \) are equivalent whenever:

1. \( P = P' \),
2. \( V = V' \),
3. For each \( i \), \( V_i = V'_i \) and \( S_i = S'_i \),
4. \( (s^1_i, \ldots, s^n_i) \) is a NE for \( G \) if and only if \( (s^1_i, \ldots, s^n_i) \) is a NE for \( G' \).

Given Lemma 1, it is straightforward to prove that all Łukasiewicz games have an equivalent normalized counterpart.

**Lemma 2.** Every Łukasiewicz game is equivalent to a normalized game.

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

### 5.1 Equilibria and Satisfiability

We now show that the existence of equilibria for an arbitrary game over \( L_k^n \) is equivalent to the satisfiability of a special \( L_k^n \)-formula. Notice that, if we restrict to the language of \( L_k^n \), such a formula always exists. In fact, for each variable \( p \), we can encode a valuation \( e(p) = \frac{1}{k} \) by using constants through formulas of the form
\[
p \leftrightarrow \frac{1}{k},
\]
which are satisfiable if and only if \( e(p) \) does equal \( \frac{1}{k} \). Therefore, we can build a formula that expresses the fact that a NE actually exists by encoding all possible strategy combinations and all possible changes of strategy by each player. Still, we are going to show that even if we do not have additional constants in our logic, it is still possible to write such a formula.

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

**Lemma 3.** For every propositional variable \( p \) and every valuation \( e : \langle p \rangle \rightarrow L_k \) there exists a formula \( \psi(p) \) such that
\[
e(p) = \frac{1}{k} \quad \text{if and only if} \quad e(\psi(p)) = 1.6
\]

**Proof.** We assume \( j \) and \( k \) to be coprime. If that is not the case then we have that
\[
e(p) = \frac{1}{k} \quad \text{if and only if} \quad e(\psi(p)) = 1.
\]

\[6\]Notice that the following proof translates into logical terms the algebraic proof of Lemma 19 in [10], whose context and content are significantly different from those of the present article.

Then
\[
e(p) = \frac{1}{k} \quad \text{if and only if} \quad e(\psi(p)) = 1.
\]

If \( e(p) = \frac{1}{k} \), then let
\[
\psi(p) := \varphi(p).
\]

It is easy to check that
\[
e(p) = \frac{1}{k} \quad \text{if and only if} \quad e(\varphi(p)) = 1.
\]

In fact,
\[
e(\varphi(p)) = 1 \quad \text{iff} \quad x = \frac{1}{k}.
\]

For \( e(p) = \frac{1}{k} \), with \( j \geq 2 \), the proof proceeds by induction. For \( j \) and \( k \) coprime, let
\[
\psi(p) := \varphi(p).
\]

while, for \( j \) and \( k \) not coprime, take \( j' \) and \( k' \) coprime such that \( \frac{j}{k} = \frac{j'}{k'} \) and let
\[
\psi(p) := \varphi(p).
\]

Notice that \( r_{j,k} < j \). So, for instance, if \( j = 2 \), then
\[
\psi(p) := \varphi(p).
\]

This concludes the proof of the Lemma.

Now we are ready to show that for each game the existence of an equilibrium is equivalent to the existence of a special satisfiable formula \( \varphi \). We prove this by giving an explicit construction of \( \varphi \). As before, we assume the game to be normalized.

Notice that as an immediate consequence of Lemma 3, we have:

**Corollary 1.** In every Łukasiewicz game \( G \) on \( L_k^n \), for every strategy combination \( (s_1, \ldots, s_n) \) there exists a \( L_k^n \)-formula \( \psi \) so that
\[
f_{\psi}(s^1, \ldots, s^n) = 1 \quad \text{if and only if} \quad s_i = s'_i
\]
for all \( i \).

In fact, let \( V_i = \{p_{i1}, \ldots, p_m\} \) be the set of variables in control of player \( i \), and let
\[
(\alpha_{i1}, \ldots, \alpha_{im}) \in (L_k)^m.
\]

The formula
\[
\psi_{\alpha_{i1}}(p_{i1}) = 1 \quad \text{iff} \quad e(p_{i1}) = \alpha_{i1},
\]
encodes the assignment by player \( i \) of the value \( \alpha_{i1} \) to the variable \( p_{i1} \), i.e.
So, the formula
\[ \psi_{\alpha_1}(p_{1_i}) \land \cdots \land \psi_{\alpha_{m_i}}(p_{m_i}) \]
encodes player \( i \)'s strategy
\[ \{ \alpha_1, \ldots, \alpha_{m_i} \}, \]
and the formula
\[ \bigwedge_{i=1}^{n} \left( \psi_{\alpha_1}(p_{1_i}) \land \cdots \land \psi_{\alpha_{m_i}}(p_{m_i}) \right) \tag{1} \]
encodes the strategy combination
\[ \{ \alpha_1, \ldots, \alpha_{m_i}, \ldots, \alpha_{m_j}, \ldots, \alpha_{m_j} \}. \]
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.

Now, take, for each player \( i \) the set of all strategies
\[ S_i = \{ s_i \mid s_i = (\beta_{1_i}, \ldots, \beta_{m_i}) \in (L_k)^{m_i} \}. \]
Assign to each player \( i \) a new set of variables
\[ \mathcal{V}^i = \{ q_{1_i}, q_{1_i}, \ldots, q_{m_i} \}, \]
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 combinations and is therefore assigned \( m_i \cdot (k + 1)^{m_i} \) new variables.

Proceeding as above, take the formula
\[ \psi_{\beta_1}(q_{1_i}) \]
that encodes the assignment by player \( i \) of the value \( \beta_{1_i} \) to the variable \( q_{1_i} \), so that the formula
\[ \psi_{\beta_1}(q_{1_i}) \land \cdots \land \psi_{\beta_{m_i}}(q_{m_i}) \tag{2} \]
encodes player \( i \)'s strategy
\[ \{ \beta_1, \ldots, \beta_{m_i} \}. \]
Let
\[ \varphi_i(p_{1_i}, \ldots, p_{m_i}, \ldots, p_{1_i}, \ldots, p_{m_i}) \tag{3} \]
be player \( i \)'s payoff formula, and let
\[ \varphi_i(p_{1_i}, \ldots, p_{m_i}, \ldots, p_{1_i}, \ldots, p_{m_i}) \land \psi_{\beta_i}(q_{1_i}, \ldots, q_{m_i}) \land \psi_{\beta_{m_i}}(q_{m_i}) \tag{4} \]
be the formula obtained from (3) by replacing the variables
\[ \{ p_{1_i}, \ldots, p_{m_i} \} \]
with the variables
\[ \{ q_{1_i}, \ldots, q_{m_i} \}. \]
So, using (3) and (4), the formula
\[ \varphi_i(p_{1_i}, \ldots, p_{m_i}, \ldots, p_{1_i}, \ldots, p_{m_i}) \land \psi_{\beta_i}(q_{1_i}, \ldots, q_{m_i}) \land \psi_{\beta_{m_i}}(q_{m_i}) \rightarrow \varphi_i(p_{1_i}, \ldots, p_{m_i}, \ldots, p_{1_i}, \ldots, p_{m_i}), \]
encodes the fact that player \( i \)'s payoff does not increase. To simplify the notation, we denote the formula (5) by \( \chi \).

Now, define the formula \( \mathcal{E}_G \), where each
\[ s \in (L_k)^{\sum_{i=1}^{m_i}} \]
is a strategy combination:
\[ \mathcal{E}_G := \bigvee_{s \in (L_k)^{\sum_{i=1}^{m_i}}} \left( \bigwedge_{i=1}^{n} \left( \psi_{\alpha_1}(p_{1_i}) \land \cdots \land \psi_{\alpha_{m_i}}(p_{m_i}) \land \bigwedge_{i=1}^{n} \left( s_{\beta_1}(q_{1_i}) \land \cdots \land s_{\beta_{m_i}}(q_{m_i}) \land \chi \right) \right) \right) \tag{6} \]

From the above construction, it is easy to check that \( \mathcal{E}_G \) actually encodes the existence of equilibria. In fact, \( \mathcal{E}_G \) is a disjunction indexed by all possible strategy combinations. The existence of an equilibrium requires at least one of the disjuncts to be satisfiable. Each disjunct is a conjunction of formulas encoding the requirement that for a given strategy combination and for every player, every change of strategy does not result in any payoff increase. So, if any such a disjunct is satisfiable, the related strategy combination actually corresponds to a NE.  

**Theorem 2.** A Łukasiewicz game \( G \) on \( L_k \) admits a Nash Equilibrium if and only if \( \mathcal{E}_G \) is satisfiable.

### 5.2 Equilibria and Satisfiable Games

Our purpose now is to show that whenever a game \( G \) has a Nash equilibrium, there exists a special kind of game, called a satisfiable game, that is equivalent to \( G \).

**Definition 8 (Satisfiable Games).** A Łukasiewicz game \( G \) on \( L_k \) is called satisfiable if there exists a strategy combination \( (s_1, \ldots, s_n) \) such that \( f_{\varphi_i}(s_1, \ldots, s_n) = 1 \) for all \( i \).

The following lemma is an immediate consequence:

**Lemma 4.** Every satisfiable Łukasiewicz game \( G \) on \( L_k \) admits a Nash Equilibrium.

As said above, we want to show that every game \( G \) on \( L_k \) admits a NE if and only if it is equivalent to a satisfiable game. In order to do so, we need to use some concepts related to the algebraic semantics of Łukasiewicz logics and their model theory. Space limitations prevent us from defining these concepts; we refer the reader to [10].

Take any linearly ordered finite MV-algebra \( MV_k \). The first-order theory \( Th(MV_k) \) of \( MV_k \) is the set of first order sentences in the language \( (\oplus, \neg) \) that hold over \( MV_k \). As shown in [10], each theory \( Th(MV_k) \) admits quantifier elimination in \( (\oplus, \neg) \). This means that every formula of \( Th(MV_k) \) is equivalent to a quantifier-free formula.

Now, take any Łukasiewicz game \( G \) on \( L_k \), and let \( \vec{x}_i, \vec{q}_i \) denote tuples of variables assigned to player \( i \), and let \( \mathcal{E}_G \) be the following sentence in \( Th(MV_k) \) (where \( \ominus \) denotes the Boolean metalanguage conjunction and \( \leq \) the order relation):
\[
\exists \vec{x}_1, \ldots, \vec{x}_n \forall \vec{y}_1, \ldots, \vec{y}_n \left( \bigwedge_{i=1}^{n} \varphi_i(\vec{x}_1, \ldots, \vec{x}_{i-1}, \vec{y}_i, \vec{x}_{i+1}, \ldots, \vec{x}_n) \leq \bigwedge_{i=1}^{n} \varphi_i(\vec{x}_1, \ldots, \vec{x}_{i-1}, \vec{x}_i, \vec{x}_{i+1}, \ldots, \vec{x}_n) \right). \]

It is then easy to check that:
\[ MV_\text{algebras generalize Boolean algebras and form the algebraic semantics of Łukasiewicz logics.} \]
**Lemma 5.** A Łukasiewicz game \( G \) on \( L^n_k \) admits a Nash Equilibrium if and only if \( E_G \) holds in \( \text{Th}(\text{MV}_k) \).

From Lemma 5, we obtain a characterization of the existence of NE in terms of the validity of a first-order formula \( E_G \). The fact that \( E_G \) is valid in \( \text{Th}(\text{MV}_k) \) is equivalent to the definability of a non-empty set that corresponds to the set of equilibria of \( G \). By quantifier-elimination, such a set is in turn definable by a quantifier-free formula \( \Phi(\vec{x}_1, \ldots, \vec{x}_n) \). In order to show the existence of an equivalent satisfiable game, we exploit the fact that this quantifier free-formula can be translated into a formula of the logic \( L^n_k \). This step will be a consequence of the following lemma.

**Lemma 6.** For every quantifier-free formula \( \Phi \) (with parameters) of \( \text{Th}(\text{MV}_k) \) in the language \( \langle \oplus, \lnot, 0 \rangle \) there exists a \( L^n_k \)-formula \( \varphi \) such that \( \Phi \) is satisfiable over \( \text{MV}_k \) if and only if there exists a valuation \( e \) such that \( e(\varphi) = 1 \).

**Proof.** Let \( \text{Terms}_{\text{MV}} \) be the set of terms \( t \) of \( \text{Th}(\text{MV}_k) \) in the language of \( \text{MV} \)-algebras and let \( \text{Form}_{L^n_k} \) be the set of \( L^n_k \)-formulas. Define a translation \( \tau : \text{Terms}_{\text{MV}} \rightarrow \text{Form}_{L^n_k} \) such that

1. If \( t = x \), then \( \tau(t) = p \).
2. If \( t = \frac{1}{k} \), then \( \tau(t) = \frac{1}{k} \).
3. If \( t = t' \oplus t'' \), then \( \tau(t) = \tau(t') \oplus \tau(t'') \).
4. If \( t = \lnot t' \), then \( \tau(t) = \lnot \tau(t') \).

Now, every quantifier-free formula \( \Phi \) of \( \text{Th}(\text{MV}_k) \) is a Boolean combination of equalities and (strict) inequalities between terms. So, define a new mapping \( \lambda : \text{Form}_{\text{MV}} \rightarrow \text{Form}_{L^n_k} \), where \( \text{Form}_{\text{MV}} \) is the set if quantifier-free formulas \( \Phi \) of \( \text{Th}(\text{MV}_k) \), as follows (where \( ~ \) denotes the Boolean metalanguage negation):

1. If \( \Phi \) is \( (t = t') \), then \( \lambda(\Phi) = \Delta(\tau(t) \leftrightarrow \tau(t')) \).
2. If \( \Phi \) is \( (t < t') \), then \( \lambda(\Phi) = \Delta(\tau(t') \rightarrow \tau(t)) \).
3. If \( \Phi = \Phi' \cap \Phi'' \), then \( \lambda(\Phi) = \lambda(\Phi') \land \lambda(\Phi'') \).
4. If \( \Phi = \lnot \Phi' \), then \( \lambda(\Phi) = \lnot \lambda(\Phi') \).

It is easy to check that every formula \( \Phi \) is satisfiable in \( \text{MV}_k \) if and only if \( \lambda(\Phi) \) is satisfiable. In fact, on the one hand, every function symbol in \( \langle \oplus, \lnot, 0 \rangle \) and every parameter as well has an interpretation in \( \text{MV}_k \) corresponding to the interpretation of the related connective (and constants) in \( L^n_k \). On the other hand, the use of the operator \( \Delta \) forces each formula of the form \( \Delta(\tau(t) \leftrightarrow \tau(t')) \) and \( \Delta(\tau(t') \rightarrow \tau(t)) \) to behave like a Boolean formula, making compositions of such formulas into Boolean combinations.

Given the previous lemma, and following the above reasoning, it is possible to prove that, by replacing each payoff formula \( \varphi_i \) in \( G \) with the formula

\[
\tau(\Phi(\vec{x}_1, \ldots, \vec{x}_n)) \lor \neg \varphi_i,
\]

we obtain a satisfiable game \( G' \) equivalent to \( G \).

**Theorem 3.** A Łukasiewicz game \( G \) on \( L^n_k \) admits a Nash Equilibrium if and only if it is equivalent to a satisfiable game.

### 6. Conclusions

Boolean games are a rich and natural model for understanding strategic behaviour in logic-based goal-oriented multi-agent systems. Although they have many desirable features and have attracted much interest, the dichotomous nature of the preferences in conventional Boolean games severely limits the types of scenarios that can easily be expressed using Boolean games based on classical (two valued) logic. By expressing goal formulae with Łukasiewicz logic, we are able to naturally and compactly express much richer objectives for agents, as we have demonstrated here.

Several questions suggest themselves for future work. For example, it is natural to consider whether we can adapt techniques for theorem proving with Łukasiewicz logic to solving Łukasiewicz games. In addition, further consideration of the computational complexity of Łukasiewicz games would be of interest.

**Acknowledgments.** Marchioni acknowledges support of the Marie Curie Intra-European Fellowship NAAMSI (FP7-PEOPLE-2011-IEF). Wooldridge was supported by the ERC Advanced Grant RACE (291528).

### 7. References


