Łukasiewicz Logics for Cooperative Games

Enrico Marchioni

School of Electronics and Computer Science, University of Southampton, UK

Michael Wooldridge

Department of Computer Science, University of Oxford, UK

Abstract

Coalitional resource games (CRGs) provide a natural abstract framework with which to model scenarios in which groups of agents cooperate by pooling resources in order to carry out tasks or achieve individual goals. In this work, we introduce a richer and more general framework, called *Łukasiewicz resource games* (ŁRG), which is based on many-valued Łukasiewicz logics, whose formulae make it possible to specify the class of piecewise linear polynomial functions with integer and rational coefficients on $[0,1]^n$. The use of Łukasiewicz logics provides a new approach to the representation of the scenario/situations modelled by CRGs. In ŁRGs, each agent is endowed with resources that can be allocated over a set of tasks, where the outcome of a task depends on the profile of resources that are allocated to it. We specify task outcomes using formulae of Łukasiewicz logic. In addition, agents have payoff functions over task outcomes, which are also specified by Łukasiewicz formulae. After motivating and introducing ŁRGs, we formalise notions of coalition structures and the core for ŁRGs and investigate the non-emptiness of the core both from a logical and computational perspective. We prove that ŁRGs are a proper generalisation of CRGs by showing how any CRG can be translated into a ŁRG that is strategically equivalent, in the sense that the former has a non-empty core if and only if so does the latter.

Keywords: Cooperative games, Non-transferable Utility, Łukasiewicz Logics
2010 MSC: 00-01, 99-00

1. Introduction

The past decade has witnessed a substantial growth of interest in issues surrounding the use of concepts and models from cooperative game theory in computer science generally, and artificial intelligence in particular – see, e.g., [12] for a detailed survey. Broadly speaking, two main sets of issues relating to cooperative games have been investigated in the literature: the development of *representations* for cooperative games (and in particular, *compact* representations); and the development of techniques for computing solution concepts such as the core and the Shapley value. This paper focuses on the former issue. The key problem is that naive representations of cooperative games require space exponential in the number of players in the game [12]. Two main approaches have been adopted with respect to compact representations: the development of *general* representations (such as Ieong and Shoham's marginal contribution nets [22]), and the development of game models for specific applications (such as cooperative games interpreted on networks [6]). One approach to the development of compact representations that has been widely adopted within the artificial intelligence community generally, but which has had comparatively little impact on representations for cooperative games, is the use of *logic* as a representation formalism. Against this background, our aims in the present paper are twofold.

First, we introduce a natural and compelling generalisation of the *Coalitional Resource Games* (CRGs) model of Wooldridge and Dunne [36]. The basic idea in CRGs is that agents are endowed with certain resources, and may cooperate with each other by pooling resources to achieve certain goals. Each agent desires the satisfaction of some goal, and cooperation allows agents to achieve goals that could not be achieved if they acted in isolation. Our new model assumes a finite set of players, each of which is endowed with a set of resources that they may allocate over a set of *tasks*. The outcome of each task is dependent on the profile of resources allocated to it by the agents. In other words, the outcome of a task is the result of how the players decide to allocate their resources. Each player also has a payoff that depends on those task outcomes, which represents that player's preferences over outcomes. The goal of each agent is then primarily to maximise this payoff, but also to minimise the costs they incur in doing so.

Second, we show how this model can be succinctly captured using *Łukasiewicz logics*, whose formulae make it possible to specify the class of piecewise linear polynomial functions with integer and rational coefficients on $[0,1]^n$. Łukasiewicz logics were proposed as a representation language for non-cooperative games by Marchioni and Wooldridge [24], who presented Łukasiewicz games as a generalisation of the classic logic-based Boolean games model of Harrenstein et al. [20]. In conventional Boolean games, each player controls a finite set of propositional variables, and seeks the satisfaction of a goal, specified as a formula of propositional logic over the total set of propositional variables in the game. The Łukasiewicz games of Marchioni and Wooldridge generalise Boolean games by specifying goals as formulae of some many-valued Łukasiewicz logic, thereby permitting the definition of a much richer class of utility functions than is possible in Boolean games based on classical propositional logic. Our present paper can be seen as an extension of this work to cooperative games.

The work presented within this article provides a mathematical framework through which an important class of cooperative games can be given a declarative logical representation using a formalism that is well-known and widely studied within the knowledge representation community – viz., Łukasiewicz logic. Using our model, problems related to reasoning about cooperative games can therefore be reduced to logical reasoning problems, such as satisfiability checking or theorem proving.

The remainder of this work is structured as follows.

- In the following section, we summarise the background material on Łukasiewicz logics that is used throughout the paper.

- In Section 3, we formally introduce Łukasiewicz Resource Games (ŁRG) as a logic-based model of cooperative games.

- In Section 4, we formalise notions of cooperation, coalition structures, and the core, investigate some of their main properties, and give a characterisation of the structure of the core.

- Section 5 introduces the CRG model of Wooldridge and Dunne [36] and shows that finite ŁRGs are a proper generalisation of this model. In particular we show that for every CRG there exists a finite ŁRG so that the core of the former is non-empty if and only if so is the core of the latter.

- In Section 6, we present a number of computational results related to computing the core of ŁRGs.
- We conclude with some final remarks.

2. Łukasiewicz Logics

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 summarising the relevant concepts of Łukasiewicz logics and their related class of functions that will be extensively used in the remainder of the paper. The interested reader can find detailed treatments of Łukasiewicz logics and their semantics in [11, 13, 29].

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

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

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

A *valuation*, s, is a mapping $s : \textsc{Var} \rightarrow [0, 1]$, 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 s to complex formulae. Although strictly speaking we only need to state the rules for the connective $\rightarrow$ and the truth-constant $\overline{0}$ (as the remaining connectives can be defined in terms of these), we present the complete ruleset in the interest of clarity:

$$\begin{array}{rcl}
s(\overline{0}) & = & 0 \\
s(\phi \rightarrow \psi) & = & \min(1 - s(\phi) + s(\psi), 1) \\
s(\neg\phi) & = & 1 - s(\phi) \\
s(\overline{1}) & = & 1 \\
s(\phi \odot \psi) & = & \max(0, s(\phi) + s(\psi) - 1) \\
s(\phi \wedge \psi) & = & \min(s(\phi), s(\psi)) \\
s(\phi \vee \psi) & = & \max(s(\phi), s(\psi)) \\
s(\phi \oplus \psi) & = & \min(1, s(\phi) + s(\psi)) \\
s(\phi \leftrightarrow \psi) & = & 1 - |s(\phi) - s(\psi)| \\
s(\phi \ominus \psi) & = & \max(0, s(\phi) - s(\psi)) \\
s(d(\phi, \psi)) & = & |s(\phi) - s(\psi)|
\end{array}$$

We say that a formula ϕ is *satisfiable* if there exists a valuation s such that $s(\phi) = 1$. Given a formula $\phi(p_1, \ldots, p_n)$ over variables $p_1, \ldots, p_n$, we let $Sat(\phi(p_1, \ldots, p_n))$ denote the *satisfiability set* of $\phi(p_1, \ldots, p_n)$, i.e.:

$$Sat(\phi(p_1, \ldots, p_n)) = \Big\{(a_1, \ldots, a_n) \in [0,1]^n \;\Big|\; s(\phi(p_1, \ldots, p_n)) = 1, \text{ and } s(p_1) = a_1, \ldots, s(p_n) = a_n\Big\}.$$

As usual, a *theory*, Γ, is a set of formulae. An evaluation s is a model for Γ if it satisfies every $\psi \in \Gamma$. We call a formula ϕ a *tautology* if $s(\phi) = 1$ for every valuation s. Note that these notions of satisfiabilty, satisfiability set, and tautology will be used also for the various other logics introduced below, with the obvious modifications.

We make use of a number of extensions of Ł$_\infty$:

- *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, i.e., $s(\overline{c}) = c$, 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 number $n \geq 1$. Each δ_n functions as a *divisibility* operator. It has the following interpretation, for all valuations s into $[0,1]$:

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

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

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

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

- For each natural number $k \geq 1$, the *Finite-valued Łukasiewicz logics* (Ł$_k$) share the same language as infinite-valued Łukasiewicz logic Ł$_\infty$. In such logics, it is assumed that the domain of all valuations of each propositional variable 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 for Ł$_\infty$, but restricted to L_k, which is closed under all Łukasiewicz operations. Notice that 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 s interpret such constants in the natural way: $s(\overline{c}) = c$.

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

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

Thus, a formula of the form $\Delta\phi$ always evaluates to a Boolean value (i.e., 0 or 1).

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. Formally, L is the logic of all tautologies in the L-language, i.e., of all the L-formulae ϕ such that $s(\phi) = 1$ for every s in the related class of valuations.[1] 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 we mentioned above, each Łukasiewicz logic L can be shown to be the logic of a certain class of functions: we will now flesh out this observation. Given an L-formula $\phi(p_1, \dots, p_n)$ we can obviously define a real-valued function $f_\phi(x_1, \dots, x_n)$ so that for each assignment s to the propositional variables $p_1, \dots, p_n$,

$$f_\phi(s(p_1), \dots, s(p_n)) = s(\phi(p_1, \dots, p_n)).$$

The formula $\phi(p_1, \dots, p_n)$ is said to *realise* the function $f_\phi(x_1, \dots, x_n)$.[2]

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

Definition 1 (McNaughton Function)**.** *A function*

$$f : [0,1]^n \to [0,1]$$

is called a (rational) 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* $\mathsf{p}_1, \dots, \mathsf{p}_k$ *with integer (rational) coefficients,*

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

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* $f(\vec{y}) = \mathsf{p}_i(\vec{y})$*.*

Infinite-valued Łukasiewicz logic and Rational Łukasiewicz logic are the logics whose class of functions coincides with the whole set of McNaughton and rational McNaughton functions.

[1] All the Łukasiewicz logics we have introduced have a specific axiomatisation, and their axiom systems are complete w.r.t. to the semantics given here. Giving the precise axiomatisation is beyond the scope of this work. All the details can be found in [11, 18, 19, 16].

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

Theorem 2 ([27, 18])**.** *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 $\phi(p_1, \dots, p_n)$.*[3] *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 $\phi(p_1, \dots, p_n)$.*[4]

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 $c_1, \dots, c_m \in \mathbb{Q} \cap [0,1]$.

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 to a formula $\phi(p_1, \dots, p_n)$ of Ł$_k$ is obtained by taking the function $f_{\phi(p_1,\dots,p_n)}$ 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 $n \geq 0$.[5]

Notation 2. *From now on we will generally refer to the functions associated to the logics introduced above simply as* Lukasiewicz functions*.*

Definition 3. *Any first-order formula in one of the following languages*

$$\mathcal{L} = \langle \to, \{\delta_n\}_{n>1}, 0 \rangle, \qquad \mathcal{L}_k = \langle \to, \{c\}_{c \in L_k} \rangle$$

is called a quantified Łukasiewicz formula *(sentence if the formula contains no free variables). A quantified Łukasiewicz formula is called* existential *if it contains only existential quantifiers.*

It is easy to see that every quantified Łukasiewicz formula has the form

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

where $\Phi(x_1, \dots, x_n, y_1, \dots, y_m)$ is a Boolean combination of equalities and strict inequalities in $\mathcal{L}$ or $\mathcal{L}_k$ and each Q_i is either an existential or universal quantifier.

[3] See also [28], [11] and [2].

[4] See also [2].

[5] In fact, for any function $f : (L_k)^n \to L_k$, we can define a formula realising f as follows:

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

Let $\mathbb{L}$ and $\mathbb{L}_k$ be the following first-order structures

$$\mathbb{L} = \Big\langle [0,1], \rightarrow, \{\delta_n\}_{n>1}, 0 \Big\rangle, \qquad \mathbb{L}_k = \Big\langle L_k, \rightarrow, \{c\}_{c \in L_k} \Big\rangle$$

where, for all x, y and natural numbers $n > 1$

$$x \rightarrow y = \min(1 - x + y, 1), \qquad \delta_n x = \frac{1}{n} x.$$

The following Theorem summarises the complexity of key decision questions associated with the logics we have discussed, and is used later.

Theorem 4 ([11, 24])**.** *Checking the satisfiability of a Łukasiewicz propositional formula for any Łukasiewicz logic is* NP*-complete. Checking whether:*

1. *a quantified Łukasiewicz sentence in $\mathcal{L}$ holds over $\mathbb{L}$ is in* 2-EXPTIME*,*
2. *a quantified Łukasiewicz sentence in $\mathcal{L}_k$ holds over $\mathbb{L}_k$ is* PSPACE*-complete,*
3. *an existential Łukasiewicz sentence in $\mathcal{L}$ holds over $\mathbb{L}$ is in* NP*,*
4. *an existential Łukasiewicz sentence in $\mathcal{L}_k$ holds over $\mathbb{L}_k$ is* NP*-complete.*

3. Łukasiewicz Resource Games

We begin by informally introducing Łukasiewicz Resource Games (ŁRG), the cooperative game model that we work with throughout the paper, and then go on to show how the model is formalised using Łukasiewicz logic. The ŁRGs model is a natural and compelling generalisation of Coalitional Resource Games, which were introduced in [36], and extended in [15].

ŁRGs are populated by a set of n players. Each player is endowed with a certain amount of *resources* and must decide how to allocate these resources across a set of m different *tasks*. Usually, there will be *constraints* on how these resources can be allocated: for example, the most obvious type of constraint will be that an agent has some bound on the total amount that they can contribute. The outcome of each task will then be a function of the resources that were allocated to that task by the players. Finally, each player receives a payoff, which is a function of the outcomes of all tasks, and encodes what the player would like the outcomes and their relationships to be. Ultimately the goal of each agent is to maximise their payoff, and, secondarily, to minimise the costs they incur in doing so.

A little more formally, let $\textsc{Ag} = \{A_1, \ldots, A_n\}$ be the set of players. A task, τ, is then a function $\tau : \mathbb{R}^n \rightarrow \mathbb{R}$, where the i'th argument of τ will represent how much player i contributes to the task. Let $\textsc{Task} = \{\tau_1, \ldots, \tau_m\}$ denote the set of tasks. A utility function for player i is given by a function $u_i : \mathbb{R}^m \rightarrow \mathbb{R}$, where the j'th argument of u_i corresponds to the outcome of the j'th task. Such a game is played as follows:

1. Each player A_i decides how much to contribute to each task τ_j. Let α_{ij} denote the amount that A_i contributes to τ_j. (As we noted, there will typically be some constraints on the values α_{ij}.)
2. The result of each task τ_j is then given by $\tau_j(\alpha_{1j}, \ldots, \alpha_{nj})$.
3. Each player A_i then receives utility $u_i(\tau_1(\alpha_{11}, \ldots, \alpha_{n1}), \ldots, \tau_m(\alpha_{1m}, \ldots, \alpha_{nm}))$.

This model is easily seen to generalise many other cooperative game models. For example, it can capture coalition structure formation [12], by having $2^{|\textsc{Ag}|}$ tasks, each one corresponding to a possible coalition. The value α_{ij} can then be interpreted as the amount of effort that A_i devotes to coalition j.

We emphasise that our model does not impose any specific interpretation on the values that players contribute to tasks, but nevertheless a natural interpretation is that players are contributing resources, in which case a natural constraint is that a player is endowed with some fixed amount of resources.

We now show how this model is formalised within Łukasiewicz logic. First, let L be a subset of $[0,1]$ closed under Łukasiewicz operations. The overall set of tasks is denoted by $\textsc{Task} = \{\tau_1, \ldots, \tau_m\}$; we formally define how these tasks are represented below. Each player A_i controls a set

$$\textsc{Var}_i = \{p_{i1}, \ldots, p_{im}\}$$

of variables, which means that A_i has the unique ability to assign to each variable p_{ij} a value from L: this assignment will indicate the amount of resource allocated by agent A_i to task τ_j. Formally, the allocation of resources for a player i to tasks is specified by valuations s_i from the set of propositional variables $\textsc{Var}_i$ controlled by i into L.

To represent constraints on how each agent allocates resources, we use Łukasiewicz formulae θ_i over the set of variables $\textsc{Var}_i$. The idea is that each player i may only choose valuations for their variables that satisfy θ_i. Note that by setting θ_i to a tautology, we remove any constraints: the player controlling the variables in θ_i is free to choose any assignment.

Each task τ_j has an associated outcome, depending on the resources assigned to it by the agents. This outcome is represented by a Łukasiewicz formula

$$\tau_j(p_{1j}, \ldots, p_{nj})$$

with associated Łukasiewicz function

$$f_{\tau_j} : L^n \to L.$$

Thus, $f_{\tau_j}(x_{1j}, \ldots, x_{nj})$ will be a numeric value indicating the outcome of the task τ_j given the resource allocation as specified by the variables $x_{1j}, \ldots, x_{nj}$.

The overall payoff a player A_i receives will be a function of the outcomes of all tasks. The payoff function for a player A_i is encoded by a Łukasiewicz formula

$$\chi_i(\tau_1, \ldots, \tau_m)$$

with associated Łukasiewicz function

$$f_{\chi_i} : L^{n \cdot m} \to L.$$

Thus, agent A_i seeks to maximise the value of f_{χ_i}.

Collecting these components together, we have:

Definition 5 (Łukasiewicz Resource Games). *Let* Ł *be a Łukasiewicz logic with set of truth-values L, and let $n, m \in \mathbb{N}$ be such that $n, m > 0$. A* Łukasiewicz Resource Game *on* Ł *is given by a structure*

$$\Gamma = \langle \textsc{Ag}, \textsc{Var}, \textsc{Const}, \textsc{Val}, \textsc{Task}, \textsc{Payoff} \rangle$$

where

1. $\textsc{Ag} = \{A_1, \dots, A_n\}$ *is a set of agents (also referred to as "players").*[6]
2. $\textsc{Var} = \{\textsc{Var}_1, \dots, \textsc{Var}_n\}$ *contains the sets*

$$\textsc{Var}_i = \{p_{i1}, \dots, p_{im}\}$$

of propositional variables assigned to each agent A_i*, so that the* $\textsc{Var}_i$*'s are mutually disjoint;*

3. $\textsc{Const} = \{\theta_1, \dots, \theta_n\}$ *is a set of Łukasiewicz formulae, one for each agent* A_i*, representing constraints – each formula* θ_i *contains only variables over* $\textsc{Var}_i$*;*
4. $\textsc{Val} = \{\textsc{Val}_1, \dots, \textsc{Val}_n\}$ *contains the sets* $\textsc{Val}_i$*, one for each agent* A_i*, of all valuations* s_i *that* A_i *can assign to the variables in* $\textsc{Var}_i$*, i.e.*

$$\textsc{Val}_i = \{s_i : \textsc{Var}_i \to L \mid s_i(\theta_i) = 1\}.$$

5. $\textsc{Task} = \{\tau_1, \dots, \tau_m\}$ *is a set of Łukasiewicz formulae*

$$\tau_j(p_{1j}, \dots, p_{nj}),$$

called task formulae*, each realising a Łukasiewicz function*

$$f_{\tau_j}(x_{1j}, \dots, x_{nj}).$$

6. $\textsc{Payoff} = \{\chi_1, \dots, \chi_n\}$ *is a set of Łukasiewicz formulae*

$$\chi_i(\tau_1, \dots, \tau_m),$$

called payoff formulae*, each realising a Łukasiewicz function*

$$f_{\chi_i}\big(f_{\tau_1}, \dots, f_{\tau_m}\big).$$

A game Γ is said to be *finite* if it is based on a finite-valued Łukasiewicz logic, and *infinite* otherwise.

We can now define formally how ŁRGs are played:

1. each player A_i chooses a valuation $s_i : \textsc{Var}_i \to L$ for the variables $\textsc{Var}_i$ that the player controls, which must satisfy the constraint that $s_i(\theta_i) = 1$
2. each task τ_j then has an outcome

$$f_{\tau_j}(s_1(p_{1j}), \dots, s_n(p_{nj}))$$

3. each player A_i receives a payoff of

$$f_{\chi_i}\big(f_{\tau_1}(s_1(p_{11}), \dots, s_n(p_{n1})), \dots, f_{\tau_m}(s_1(p_{1m}), \dots, s_n(p_{nm}))\big).$$

We now provide an example of a coalition game that can be formalised in the framework of ŁRGs. Throughout the whole paper we will be using this example to show how the technical notions we introduce work in the context of ŁRGs.

[6] Notice that we will sometimes refer to an agent/player A_i simply as "i".

Example 6. *Three children, Charlie, Marcie and Pattie, want to buy ice cream and each have ten £1 coins to spend. There are four different flavours: chocolate, strawberry, banana, and vanilla. The amount of ice cream to be bought depends on how much money is spent on it, up to a maximum of £10. In other words, each flavour comes in the same limited quantity, and the amount that can be acquired is proportional to the money allocated to it and cannot exceed £10. Charlie wants to have as much chocolate as possible and at least as much chocolate as the remaining flavours. Marcie wants to taste a bit of everything and would then prefer an equal distribution of flavours but she also wants to have at least £10 worth of ice cream. Pattie only wants as much chocolate and strawberry as possible.*

The game has the following formalisation obtained by using

$$L = \{0, 1, \ldots, 9, 10\}$$

as the set of truth values of finite-valued Łukasiewicz logic Ł_{10}.[7] *Let* C, M, P *be our agents and let* $\textsc{Var}_C$, $\textsc{Var}_M$, *and* $\textsc{Var}_P$ *be the sets of variables assigned to each agent, respectively, defined as follows:*

$$\begin{aligned} \textsc{Var}_C &= \{p_{Cc}, p_{Cs}, p_{Cb}, p_{Cv}\}, \\ \textsc{Var}_M &= \{p_{Mc}, p_{Ms}, p_{Mb}, p_{Mv}\}, \\ \textsc{Var}_P &= \{p_{Pc}, p_{Ps}, p_{Pb}, p_{Pv}\}. \end{aligned}$$

So, for example, the amount that Charlie contributes to buying chocolate ice cream is defined by the value of the variable p_{Cc}*, while the amount he contributes to buying strawberry ice cream is given by the value of* p_{Cs}*, and so on.*

We require that the resources available to each agent amount to no more than £10. This is formalised by constraints θ_i*, for each child* A_i*:*

$$\begin{aligned} \theta_i := \; & \left(\bigoplus_{j' \in \{s,b,v\}} p_{ij'} \to \neg p_{ic}\right) \wedge \left(\bigoplus_{j'' \in \{c,b,v\}} p_{ij''} \to \neg p_{is}\right) \wedge \\ & \left(\bigoplus_{j''' \in \{c,s,v\}} p_{ij'''} \to \neg p_{ib}\right) \wedge \left(\bigoplus_{j'''' \in \{c,b,s\}} p_{ij''''} \to \neg p_{iv}\right). \end{aligned}$$

In fact, θ_i *is satisfiable if and only if all its conjuncts are satisfied at the same time, which happens if and only if the sum of the values allocated to any triple of flavours is less than or equal to the negation of the value allocated to the fourth one. In other words, for each* θ_i*:*

$$s_i(\theta_i) = 1 \qquad \textit{iff} \qquad s_i(p_{is}) + s_i(p_{ib}) + s_i(p_{iv}) + s_i(p_{ic}) \leq 10.$$

The formulae

$$\begin{aligned} \tau_c &:= p_{Cc} \oplus p_{Mc} \oplus p_{Pc} & \tau_s &:= p_{Cs} \oplus p_{Ms} \oplus p_{Ps} \\ \tau_b &:= p_{Cb} \oplus p_{Mb} \oplus p_{Pb} & \tau_v &:= p_{Cv} \oplus p_{Mv} \oplus p_{Pv} \end{aligned}$$

[7] Notice that, for the sake of clarity, we are actually using a set of truth values that is order-isomorphic to the standard set for Ł$_{10}$

$$\left\{0, \frac{1}{10}, \ldots, \frac{9}{10}, 1\right\},$$

along with the linear transformation of the Łukasiewicz operations over L. Notice that satisfiability for a formula ϕ now means the existence of a valuation s such that $s(\phi)$=10.

encode the amount of ice cream, for each flavour, obtained depending on the resources invested by the players. In fact, for each $j \in \{c, b, s, v\}$ and each valuation s_i with $i \in \{C, M, P\}$,

$$f_{\tau_j} = \min(s_C(p_{Cj}) + s_M(p_{Mj}) + s_P(p_{Pj}), 10).$$

The formulae

$$\begin{array}{rcl} \chi_C & := & \tau_c \wedge ((\tau_s \vee \tau_b \vee \tau_v) \to \tau_c) \\ \chi_M & := & \left(\bigoplus_{j\in\{c,b,s,v\}} \tau_j\right) \wedge ((\tau_c \leftrightarrow \tau_s) \wedge (\tau_s \leftrightarrow \tau_b) \wedge (\tau_b \leftrightarrow \tau_v)) \\ \chi_P & := & (\tau_c \wedge \tau_s) \wedge ((\tau_b \vee \tau_v) \to (\tau_c \wedge \tau_s)) \end{array}$$

specify each agent's requirements, i.e.:

$$f_{\chi_C} = \min(f_{\tau_c}, \min(10 - \max(f_{\tau_s}, f_{\tau_b}, f_{\tau_v}) + f_{\tau_c}, 10)),$$

$$f_{\chi_M} = \min(\min(f_{\tau_c} + f_{\tau_s} + f_{\tau_b} + f_{\tau_v}, 10), 10 - |f_{\tau_c} - f_{\tau_s}|, 10 - |f_{\tau_s} - f_{\tau_b}|, 10 - |f_{\tau_b} - f_{\tau_v}|),$$

$$f_{\chi_P} = \min(\min(f_{\tau_c}, f_{\tau_s}), \min(10 - \max(f_{\tau_b}, f_{\tau_v}) + \min(f_{\tau_c}, f_{\tau_s}), 10)).$$

The key idea is that each child wants to get as much satisfaction as possible from their purchase and this is achieved by maximising the value of the functions defined by the formulae above.

4. Coalitions

As mentioned above, the primary objective of each agent A_i is to maximise the value of their payoff formula χ_i. Player A_i can try to achieve this in isolation or by cooperating with other agents and forming a coalition. Given a ŁRG, a *coalition* $C \subseteq \textsc{Ag}$ is a subset of agents. The coalition containing all agents ($C = \textsc{Ag}$) is called the *grand coalition*.

Given a coalition C, with $|C| = n'$ and $n' \leq n$, we denote by $\mathbf{s}_C$ the tuple of valuations

$$(s_{i'}, s_{i''}, s_{i'''}, \dots),$$

seen as a vector in $L^{n' \cdot m}$. In other words, we see a valuation both as an assignment of values to the propositional variables of the players and as a tuple of exactly those values. While s_i refers to a specific valuation $s_i : \textsc{Var}_i \to L$ for player A_i, $\mathbf{s}_i$ denotes the same valuation seen as a vector. For a coalition C, every $\mathbf{s}_C$ is called an *allocation*. We denote by $\mathbf{s}_{-C}$ any allocation of all the players that are not members of C. By $\pi_i(\mathbf{s}_C)$ we mean player A_i's specific allocation $\mathbf{s}_i$ in $\mathbf{s}_C$. We use both $(\mathbf{s}_C, \mathbf{s}_{-C})$ and $\mathbf{s}$ to refer to an allocation of all the players in the game. Given a payoff function f_{χ_i}, both $f_{\chi_i}(\mathbf{s}_C, \mathbf{s}_{-C})$ and $f_{\chi_i}(\mathbf{s})$ denote the outcome of the choice made by all players for the given allocation. For a coalition C, S_C denotes the set of all allocations for C. When an allocation $\mathbf{s}_C$ comprises only rational values, i.e., $\mathbf{s}_C \in \mathbb{Q}^{m \cdot |C|}$, we call it a *rational allocation*.

Given a ŁRG Γ, we say that a coalition C is satisfiable if there exists an allocation $\mathbf{s} = (\mathbf{s}_1, \dots, \mathbf{s}_n)$ so that $f_{\chi_i}(\mathbf{s}) = 1$ for all $i \in C$. A coalition C is then satisfiable if there exists an allocation of resources by all players so that all agents in C *fully maximise their utility*, i.e., each player receives the largest possible payoff.

We say a ŁRG Γ is *satisfiable* if it admits a non-empty satisfiable coalition. Γ is *unsatisfiable* if it is not satisfiable.

Example 6 (continued)**.** *Going back to our example, it is easy to check that the grand coalition is not satisfiable. In fact, to satisfy both Pattie and Charlie, the children need to maximise the amount of chocolate and strawberry, obtaining £10 of each. Marcie wants an equal distribution of flavours, but given the others' requirements this can be achieved only by acquiring £10 of each ice cream. Then for all the children to be satisfied they must spend a total of £40, which is impossible, since none of them have more than £10.*

If every child was allowed to spend up to £10 for each flavour, not in total, then there would exist several possible allocations to make the grand coalition satisfiable. As an example, if every child paid £4 for each flavour, they would get the maximum amount of ice cream of each type, then satisfying everybody's requirements.

The coalition formed by Pattie and Charlie is satisfiable. In fact, if Pattie spends £10 for strawberry and Charlie £10 for chocolate, they both maximise their respective functions no matter what choice Marcie makes. As a consequence, the game is satisfiable.

Satisfiability is quite a general notion and is a result of choices made by all players no matter what coalition they actually belong to. The fact that a coalition C is satisfiable simply means that there is an allocation of resources for which all the agents in the coalition maximise their payoff. There is no guarantee though that those agents that do not belong to C will make that specific choice, and a change in their allocation could affect the satisfiability of C. Knowing that a coalition C is satisfiable then gives us little to no information on whether for each of its members being in C would be better than joining forces with other players. However, if a game is unsatisfiable, we know that there is no possible allocation of resources that would satisfy any of the possible non-empty coalitions.

4.1. Cooperation Structures

So far, we have been talking about satisfiability for coalitions from an external point of view, disregarding how they are formed and whether agents prefer to join forces with some players rather than others to achieve their goals. We now give a precise definition of what it means for an agent to cooperate successfully with others. For this we introduce the notion of a *cooperation structure* (cf. [15]).

Definition 7 (Cooperation Structure)**.** *For any Łukasiewicz Resource Game* Γ*, let* C *be a coalition in* Γ *and let* $\mathbf{s}_C$ *be an allocation for* C*. Then the structure* $\lambda = \langle C, \mathbf{s}_C \rangle$ *is called a* cooperation structure*.*

Thus, a cooperation structure defines a coalition, together with the resources that the coalition collectively allocate to tasks. In other words, a cooperation structure defines a commitment made by agents to form a coalition and allocate a specific amount of resources. Any cooperation structure $\lambda = \langle C, \mathbf{s}_C \rangle$ where $\mathbf{s}_C$ is a rational allocation is called a *a rational cooperation structure.*

It is worth pointing out that the set of cooperation structures in a game Γ is closed under disjoint union. In fact, take $\lambda_1 = \langle C_1, \mathbf{s}_{C_1} \rangle$ and $\lambda_2 = \langle C_2, \mathbf{s}_{C_2} \rangle$ with $C_1 \cap C_2 = \emptyset$. The structure

$$\lambda_1 \uplus \lambda_2 = \langle C_1 \cup C_2, (\mathbf{s}_{C_1}, \mathbf{s}_{C_2}) \rangle$$

clearly is a cooperation structure. We will make use of this fact later in the article.

A cooperation structure can be regarded as successful for a player A_i whenever the resources allocated by its members maximise A_i's outcome independently from the choices of the external agents.

Definition 8 (Successful Cooperations)**.** *We say that a cooperation structure $\lambda = \langle C, \mathbf{s}_C \rangle$, in a Łukasiewicz Resource Game Γ, is*

1. successful for agent A_i, *if for all allocations $\mathbf{s}_{-C}$, we have $f_{\chi_i}(\mathbf{s}_C, \mathbf{s}_{-C}) = 1$.*
2. globally successful *if it is successful for all agents $A_i \in C$.*

The set of successful cooperation structures for agent A_i is denoted by succ(i).

It is possible to define an inclusion relation between cooperation structures as follows. First, given coalitions $C_1 \subseteq C_2$ and their related allocation sets S_1, S_2, the projection

$$\pi_{C_1} : S_2 \to S_1$$

is the function sending each allocation $\mathbf{s}_{C_2}$ to the allocation $\mathbf{s}_{C_1}$ obtained by eliminating the elements in $\mathbf{s}_{C_2}$ that belong to the members of $C_2 \setminus C_1$, respecting the order. For instance, suppose

$$C_1 = \{A_1, A_3\} \quad \text{and} \quad C_2 = \{A_1, A_2, A_3, A_4\}.$$

The projection π_{C_1} of the tuple $(\mathbf{s}_1, \mathbf{s}_2, \mathbf{s}_3, \mathbf{s}_4)$ is the tuple $(\mathbf{s}_1, \mathbf{s}_3)$.

We say that for two cooperation structures $\lambda_1 = \langle C_1, \mathbf{s}_{C_1} \rangle$ and $\lambda_2 = \langle C_2, \mathbf{s}_{C_2} \rangle$,

$$\lambda_1 \sqsubseteq \lambda_2 \text{ if and only if } C_1 \subseteq C_2 \text{ and } \pi_{C_1}(\mathbf{s}_{C_2}) = \mathbf{s}_{C_1}.$$

The following proposition shows that if a cooperation structure is successful for a player i, the player incurs no losses if new agents are accepted into the coalition.

Proposition 9. *In a Łukasiewicz Resource Game Γ, for any two cooperation structures $\lambda_1 = \langle C_1, \mathbf{s}_{C_1} \rangle$ and $\lambda_2 = \langle C_2, \mathbf{s}_{C_2} \rangle$, if $\lambda_1 \sqsubseteq \lambda_2$ and $\lambda_1 \in$ succ(i) then $\lambda_2 \in$ succ(i).*

Proof. If $\lambda_1 \in$ *succ*(i), then for player A_i, $f_{\chi_i}(\mathbf{s}_{C_1}, \mathbf{s}_{-C_1}) = 1$ for all allocations $\mathbf{s}_{-C_1}$. Since, for all $\lambda_2 \sqsupseteq \lambda_1$, $\pi_{C_1}(\mathbf{s}_{C_2}) = \mathbf{s}_{C_1}$, trivially, we have $f_{\chi_i}(\mathbf{s}_{C_2}, \mathbf{s}_{-C_2}) = 1$ for all $\mathbf{s}_{-C_2}$. □

Example 6 (continued)**.** *As shown above, the coalition formed by Pattie and Charlie is satisfiable. In fact, if Pattie spends £10 for strawberry and Charlie £10 for chocolate, then they both maximise their respective functions. This can be achieved no matter what Marcie plays. As a consequence, the cooperation structure $\lambda = \langle \{P, C\}, (\mathbf{s}_P, \mathbf{s}_C) \rangle$ where $\mathbf{s}_P$ and $\mathbf{s}_C$ are such that*

$$s_P(p_{Pc}) = s_P(p_{Pb}) = s_P(p_{Pv}) = s_C(p_{Cs}) = s_C(p_{Cs}) = s_C(p_{Cv}) = 0,$$
$$s_P(p_{Ps}) = s_C(p_{Cc}) = 10,$$

is globally succesful.

Notice, however, that Charlie alone is successful, while Pattie alone is not. Charlie can simply decide to spend £10 for chocolate fully achieving his goal. On the contrary, Pattie must spend her £10 for strawberry, but needs Charlie (or Marcie) to do the same for chocolate if she wants to maximise her function.

Marcie can never be part of a cooperation structure that is successful for her unless she involves both Charlie and Pattie. In fact, while there are cooperations with one of the other children with allocations that can maximise her payoff, the cooperation cannot be successful for Marcie, because the player external to the coalition can always make the distribution of flavours unequal. The only possible successful cooperation structures for Marcie are those based on the grand coalition. However, no such cooperation structure is successful for Charlie and Pattie. As seen above, the grand coalition is not satisfiable, so, if Marcie is in a cooperation structure based on the grand coalition that satisfies her formula, at least for Charlie or Pattie (as a matter of fact for both) the cooperation cannot be successful.

4.2. Costs and Preferences

The previous discussion has set out the basic model of cooperation in our framework. However, the notion of a cooperation structure does not provide an agent with the tools to choose one coalition over others. In fact, if what matters is simply to maximise their payoff function, under the concept of successful cooperation structure, a player can only distinguish which coalitions have an allocation that makes them successful from those that do not. But recall that a player's secondary concern is to minimise the resources they allocate. This idea leads to defining a concept of cost related to the allocation of resources and is what will make it possible to define a proper notion of preference between cooperation structures for the players. First then, we introduce a notion of cost.

Definition 10 (Cost Function)**.** *Given a Łukasiewicz Resource Game* Γ*, for any agent* i *and allocation* $\mathbf{s}_i$*, the function*

$$cost_i(\mathbf{s}_i) = \sum_{j=1}^{m} s_i(p_{ij})$$

is called a cost function*.*

Notice that, whenever the total amount of resources available to i is 1, then

$$cost_i(\mathbf{s}_i) = \bigoplus_{j=1}^{m} s_i(p_{ij}).$$

We can now define a notion of preference between cooperation structures. Agents will always prefer a cooperation structure that is successful for them to those that are not. Whenever a player must choose between cooperation structures that are successful or between cooperation structures that are not successful, then, they will always choose those that minimise their costs. More formally, we have:

Definition 11 (Preference)**.** *In a Łukasiewicz Resource Game* Γ*, a cooperation structure* λ_1 *is preferred to* λ_2 *by player* i*, denoted*

$$\lambda_2 \preceq_i \lambda_1$$

whenever any of the following conditions is satisfied

1. $\lambda_1, \lambda_2 \in succ(i)$*, and* $cost_i(\pi_i(\mathbf{s}_{C_1})) \leq cost_i(\pi_i(\mathbf{s}_{C_2}))$*;*
2. $\lambda_1, \lambda_2 \notin succ(i)$*, and* $cost_i(\pi_i(\mathbf{s}_{C_1})) \leq cost_i(\pi_i(\mathbf{s}_{C_2}))$*;*
3. $\lambda_1 \in succ(i)$ *and* $\lambda_2 \notin succ(i)$*.*

The relation $\prec_i$ is defined in the obvious way.

Given a cooperation structure $\lambda = \langle C, \mathbf{s}_C \rangle$, we can define a utility function u_i for each $i \in C$ as follows:

$$u_i(\lambda) = \begin{cases} m + 1 - cost_i(\pi_i(\mathbf{s}_C)) & \lambda \in succ(i) \\ -cost_i(\pi_i(\mathbf{s}_C)) & \text{otherwise} \end{cases},$$

where m is the number of tasks in the game. Notice that for all allocations $\mathbf{s}_i$, this utility function can be used to faithfully represents an agent's preference between cooperation structures, as shown below.

Proposition 12. *In any Łukasiewicz Resource Game* Γ*, for any two cooperation structures* λ_1, λ_2 *and player* i*,*

$$\lambda_1 \prec_i \lambda_2 \quad \textit{if and only if} \quad u_i(\lambda_1) < u_i(\lambda_2).$$

Proof. We have the following cases to consider:

1. $\lambda_1, \lambda_2 \in \mathit{succ}(i)$, and $\mathit{cost}_i(\pi_i(\mathbf{s}_{C_1})) > \mathit{cost}_i(\pi_i(\mathbf{s}_{C_2}))$, if and only if,
$$u_i(\lambda_1) = m + 1 - \mathit{cost}_i(\pi_i(\mathbf{s}_{C_1})) < m + 1 - \mathit{cost}_i(\pi_i(\mathbf{s}_{C_2})) = u_i(\lambda_2).$$
2. $\lambda_1, \lambda_2 \notin \mathit{succ}(i)$, and $\mathit{cost}_i(\pi_i(\mathbf{s}_{C_1})) > \mathit{cost}_i(\pi_i(\mathbf{s}_{C_2}))$, if and only if,
$$u_i(\lambda_1) = -\mathit{cost}_i(\pi_i(\mathbf{s}_{C_1})) < -\mathit{cost}_i(\pi_i(\mathbf{s}_{C_2})) = u_i(\lambda_2).$$
3. $\lambda_2 \in \mathit{succ}(i)$ and $\lambda_1 \notin \mathit{succ}(i)$, if and only if,
$$u_i(\lambda_1) = -\mathit{cost}_i(\pi_i(\mathbf{s}_{C_1})) < m + 1 - \mathit{cost}_i(\pi_i(\mathbf{s}_{C_2})) = u_i(\lambda_2).$$

□

Whenever a cooperation structure is formed, new agents can join the coalition. Each player in the original cooperation structure will prefer the new, enlarged structure at least as much as the old one.

Proposition 13. *Given a Łukasiewicz Resource Game* Γ*, let* $\lambda_1 = \langle C_1, \mathbf{s}_{C_1}\rangle$ *and* $\lambda_2 = \langle C_2, \mathbf{s}_{C_2}\rangle$ *be cooperation structures such that* $C_1 \cap C_2 = \emptyset$*. Then, for all* $i \in C_j$*, with* $j \in \{1, 2\}$*,*

$$\lambda_j \preceq_i \lambda_1 \uplus \lambda_2.$$

Proof. Without any loss of generality, suppose that for some player $i \in C_1$, $\lambda_1 \uplus \lambda_2 \prec_i \lambda_1$ (the case for $i \in C_2$ is completely analogous). By Definition 11 we have the following cases:

1. If $\lambda_1 \in \mathit{succ}(i)$ and $\lambda_1 \uplus \lambda_2 \notin \mathit{succ}(i)$, then we have a contradiction. In fact, $\lambda_1 \sqsubset \lambda_1 \uplus \lambda_2$ and by Proposition 9, it must be the case that $\lambda_1 \uplus \lambda_2 \in \mathit{succ}(i)$.
2. If $\lambda_1, \lambda_1 \uplus \lambda_2 \in \mathit{succ}(i)$, or $\lambda_1, \lambda_1 \uplus \lambda_2 \notin \mathit{succ}(i)$, then
$$\mathit{cost}_i(\pi_i(\mathbf{s}_{C_1})) < \mathit{cost}_i(\pi_i(\mathbf{s}_{C_1 \cup C_2})).$$
By construction $\pi_i(\mathbf{s}_{C_1 \cup C_2}) = \pi_i(\mathbf{s}_{C_1}, \mathbf{s}_{C_2}) = \pi_i(\mathbf{s}_{C_1})$, since $i \in C_1$, i.e.: a contradiction.

□

Example 6 (continued)**.** *In our example, Pattie prefers to cooperate with Charlie as long as he spends £10 on chocolate, i.e. she prefers the cooperation structure* $\langle\{P, C\}, (\mathbf{s}_P, \mathbf{s}_C)\rangle$ *where*

$$s_P(p_{Pc}) = s_P(p_{Pb}) = s_P(p_{Pv}) = s_C(p_{Cs}) = s_C(p_{Cb}) = s_C(p_{Cv}) = 0,$$
$$s_P(p_{Ps}) = s_C(p_{Cc}) = 10,$$

to $\langle\{P\}, \mathbf{s}_P\rangle$ *since the former is successful for her, while the latter is not.*

We have seen that Marcie can cooperate with Charlie and Pattie, so that, for instance, the structure $\langle\{C, M, P\}, (\mathbf{s}_C, \mathbf{s}_M, \mathbf{s}_P)\rangle$ *where*

$$s_M(p_{Mc}) = s_M(p_{Mb}) = s_M(p_{Mv}) = s_M(p_{Ms}) = 0,$$
$$s_C(p_{Cc}) = s_C(p_{Cs}) = 2 \quad s_C(p_{Cb}) = s_C(p_{Cv}) = 1,$$
$$s_P(p_{Pc}) = s_P(p_{Ps}) = 1 \quad s_P(p_{Pb}) = s_P(p_{Pv}) = 2,$$

is successful for her. Actually, in this situation, Marcie maximises her payoff and incurs no costs. Therefore, the above is one of her preferred cooperation structures.

4.3. Coalition Structures and the Core

We now introduce the notion of a coalition structure, which formalises the idea of players dividing into disjoint teams (cf. [15]). We will then formulate a version of the standard cooperative game theory concept of the core, and explore some properties of these definitions.

Definition 14 (Coalition Structure)**.** *For any given Łukasiewicz Resource Game* Γ*, a* coalition structure

$$\sigma = \{\lambda_1, \dots, \lambda_w\}$$

is a set of cooperation structures so that the coalitions $\{C_1, \dots, C_w\}$ *form a partition of* $\textsc{Ag}$*.*

A coalition structure is then the outcome of a game Γ, where each player commits to either joining forces with others or remaining alone and allocating a certain amount of resources. We denote the set of coalition structures in a game Γ by $\mathbf{cs}(\Gamma)$.

Given a coalition structure σ,

$$\lambda_{\sigma(i)} = \langle C_{\sigma(i)}, \mathbf{s}_{C_{\sigma(i)}} \rangle$$

denotes the cooperation structure in σ of which i is a member. If σ contains only the cooperation structure with the grand coalition, we call it the *grand coalition structure.* A coalition structure formed by all rational cooperation structures is called *rational.*

Definition 15 (Stability, Core)**.** *In a Łukasiewicz Resource Game* Γ*, a cooperation structure* $\lambda = \langle C, \mathbf{s}_C \rangle$ blocks *a coalition structure* σ *if, for all* $i \in C$

$$\lambda_{\sigma(i)} \prec_i \lambda.$$

A coalition structure is said to be stable *if it is not blocked by any cooperation structure. The set of all stable coalition structures is called* the core *of* Γ *and is denoted by* $\boldsymbol{core}(\Gamma)$*.*

Example 6 (continued)**.** *The coalition structure*

$$\{\langle\{P, C\}, (s_P, s_C)\rangle, \langle\{M\}, s_M\rangle\}$$

where

$$s_P(p_{Ps}) = 10, \qquad s_C(p_{Cc}) = 10, \text{ and } \qquad s_M(p_{Mj}) = 0,$$

for all $j \in \{c, b, s, v\}$ *is stable (i.e., Charlie and Pattie join forces, with Charlie spending £10 on chocolate and Pattie £10 on strawberry, while Marcie is left alone and spends no money at all). In fact, Charlie and Pattie form a globally successful cooperation structure, and the only way for them to be in a cooperation structure they would strictly prefer is to involve Marcie to form the grand coalition and decrease their costs. In that case though, the costs would increase for Marcie. Also, Marcie certainly prefers other cooperation structures that are successful for her, as seen above. However, those cooperations need to be based on the grand coalition, which cannot be successful for every player in it, since it is not satisfiable. So the above coalition structure cannot be blocked and belongs to the core.*

Definition 16. *Let $\sigma = \{\lambda_1, \dots, \lambda_w\}$ be a coalition structure in a Łukasiewicz Resource Game. We call $\mathbf{union}(\sigma)$ the set of all coalition structures generated by taking unions of the cooperation structures in σ, i.e.*

$$\mathbf{union}(\sigma) = \Bigg\{ \sigma' = \{\lambda'_1, \dots, \lambda'_t\} \quad \Big| \quad \sigma, \sigma' \in \mathbf{cs}(\Gamma), \sigma = \{\lambda_1, \dots, \lambda_w\}, \\ \forall \lambda'_i, \exists \lambda_{1_i}, \dots, \lambda_{t_i} \in \{\lambda_1, \dots, \lambda_w\} \text{ such that } \lambda'_i = \biguplus_{j=1_i}^{t_i} \lambda_j \Bigg\}.$$

The next proposition shows stability is preserved when a new coalition structure is generated from a stable one by taking unions of its cooperation structures.

Proposition 17. *Let $\sigma = \{\lambda_1, \dots, \lambda_w\} \in \boldsymbol{core}(\Gamma)$ in a Łukasiewicz Resource Game Γ. Then, any coalition structure $\upsilon \in \mathbf{union}(\sigma)$ is such that $\upsilon \in \boldsymbol{core}(\Gamma)$.*

Proof. Suppose $\upsilon \in \mathbf{union}(\sigma)$ is blocked by some $\lambda^* = \langle C^*, \mathbf{s}_{C^*} \rangle$. Then, for all $i \in C^*$,

$$\lambda_{\upsilon(i)} \prec_i \lambda^*,$$

i.e., player i prefers the cooperation structure λ^* to the cooperation structure i belongs to in υ.

Now, by definition, one of the following cases holds true:

1. $\lambda^* \in \mathit{succ}(i)$ and $\lambda_{\upsilon(i)} \notin \mathit{succ}(i)$. By construction $\lambda_{\upsilon(i)}$ is a disjoint union of cooperation structures from σ, only one of which, $\lambda_{\sigma(i)}$, contains i (i.e. $\lambda_{\sigma(i)} \sqsubseteq \lambda_{\upsilon(i)}$). By Proposition 9, if $\lambda_{\upsilon(i)} \notin \mathit{succ}(i)$, then $\lambda_{\sigma(i)} \notin \mathit{succ}(i)$, so $\lambda^* \in \mathit{succ}(i)$ and $\lambda_{\sigma(i)} \notin \mathit{succ}(i)$.
2. $\lambda^*, \lambda_{\upsilon(i)} \in \mathit{succ}(i)$ and
$$\mathit{cost}_i\left(\pi_i\left(\mathbf{s}_{C^*}\right)\right) < \mathit{cost}_i\left(\pi_i\left(\mathbf{s}_{C_{\upsilon(i)}}\right)\right).$$
Again, by construction, $\lambda_{\upsilon(i)}$ is a disjoint union of cooperation structures from σ so that i's allocation is the same as the one in $\lambda_{\sigma(i)}$. This means that
$$\mathit{cost}_i\left(\pi_i\left(\mathbf{s}_{C^*}\right)\right) < \mathit{cost}_i\left(\pi_i\left(\mathbf{s}_{C_{\upsilon(i)}}\right)\right) = \mathit{cost}_i\left(\pi_i\left(\mathbf{s}_{C_{\sigma(i)}}\right)\right).$$
3. $\lambda^*, \lambda_{\upsilon(i)} \notin \mathit{succ}(i)$ and
$$\mathit{cost}_i\left(\pi_i\left(\mathbf{s}_{C^*}\right)\right) < \mathit{cost}_i\left(\pi_i\left(\mathbf{s}_{C_{\upsilon(i)}}\right)\right).$$
The case is completely analogous to case (2).

As a consequence, λ^* blocks σ, so σ does not belong to the core. □

We now want to show that the core of a ŁRG is non-empty if and only if every stable coalition structure is generated by stable grand coalition structures. For this, we introduce the notion of splitting.

Definition 18. *Given a stable coalition structure σ in a Łukasiewicz Resource Game, the* splitting *of σ, denoted by $\mathbf{split}(\sigma)$, is the set of all stable coalition structures σ' such that $\sigma \in \mathbf{union}(\sigma')$.*

For example, if

$$\sigma = \{\langle\{A_1, A_2, A_3\}, (\mathbf{s}_1, \mathbf{s}_2, \mathbf{s}_3)\rangle, \langle\{A_4, A_5, A_6\}, (\mathbf{s}_4, \mathbf{s}_5, \mathbf{s}_6)\rangle\}$$

is stable and so is

$$\sigma' = \{\langle\{A_1, A_2\}, (\mathbf{s}_1, \mathbf{s}_2)\rangle, \langle\{A_3\}, (\mathbf{s}_3)\rangle, \langle\{A_4, A_5, A_6\}, (\mathbf{s}_4, \mathbf{s}_5, \mathbf{s}_6)\rangle\}$$

then σ' belongs to the splitting of σ. Clearly, the splitting of a stable coalition structure σ is never empty since it contains at least σ itself.

Let $\mathbf{sgr}(\Gamma)$ denote the set of all grand coalition structures in a game Γ that are stable. It turns out that the every member of the core can be obtained from the splitting of the structures in $\mathbf{sgr}(\Gamma)$.

Theorem 19. *For every Łukasiewicz Resource Game Γ, we have*

$$\boldsymbol{core}(\Gamma) = \bigcup_{\sigma \in \boldsymbol{sgr}(\Gamma)} \mathbf{split}(\sigma),$$

and

$$|\boldsymbol{core}(\Gamma)| = \sum_{\sigma \in \boldsymbol{sgr}(\Gamma)} |\mathbf{split}(\sigma)|.$$

Proof. The fact that

$$\mathbf{core}(\Gamma) \supseteq \bigcup_{\sigma \in \mathbf{sgr}(\Gamma)} \mathbf{split}(\sigma)$$

is trivial.

Conversely, if the core is non-empty, then it contains at least a stable structure σ. By Proposition 17, for every $\sigma \in \mathbf{core}(\Gamma)$, $\mathbf{union}(\sigma)$ contains a grand coalition structure σ', which is stable, and trivially, by construction, σ belongs to the splitting of σ' and so

$$\mathbf{core}(\Gamma) \subseteq \bigcup_{\sigma \in \mathbf{sgr}(\Gamma)} \mathbf{split}(\sigma).$$

Finally, it is easy to see that a stable coalition structure σ cannot belong to the splitting of two different grand coalition structures. Therefore, each element of the core belongs to the splitting of exactly one stable grand coalition structure, and, clearly,

$$|\mathbf{core}(\Gamma)| = \sum_{\sigma \in \mathbf{sgr}(\Gamma)} |\mathbf{split}(\sigma)|.$$

□

Notice that the $\mathbf{sgr}(\Gamma)$ may be an uncountable set, and, in that case, the core's cardinality would be $\aleph_1$, under the continuum hypothesis.

While Theorem 19 gives us some information concerning the structure of the core, it is important to ask whether there are any games with an empty or a non-empty core. Section 5 will answer the former question and show the existence of ŁRGs with an empty core. The following theorem shows that there is a class of games whose core is always non-empty. That is the class of unsatisfiable games whose core has finite cardinality and contains all possible coalition structures where no agent allocates any resources.

Proposition 20. *Let Γ be any unsatisfiable Łukasiewicz Resource Game with n players. Then:*

1. *For every coalition structure $\sigma = \{\lambda_1, \dots, \lambda_w\}$ with cooperation structures $\lambda_j = \langle C_j, \mathbf{s}_{C_j} \rangle$, we have that:*

   $\sigma \in \boldsymbol{core}(\Gamma)$ *if and only if for all $j \in \{1, \dots, w\}$ and all players $i \in C_j$, $cost_i(\pi_i(\mathbf{s}_{C_j})) = 0$;*

2. $|\boldsymbol{core}(\Gamma)| = B_n$*, where B_n is the nth Bell number.*[8]

Proof. Γ is unsatisfiable, which means it admits no satisfiable non-empty coalition. So, for every player i, there are no cooperation structures λ such that $\lambda \in succ(i)$, i.e., for all i, $succ(i) = \emptyset$.

To prove (1), suppose that for some cooperation structure $\lambda_j = \langle C_j, \mathbf{s}_{C_j} \rangle$ in σ and some player $i \in C_j$,

$$cost_i(\pi_i(\mathbf{s}_{C_j})) > 0.$$

Player i cannot take part in any successful cooperation structure, but they can form their own individual cooperation structure $\lambda' = \left\langle \{i\}, \mathbf{s}_{\{i\}} \right\rangle$, where

$$cost_i\left(\mathbf{s}_{\{i\}}\right) = 0,$$

so that

$$\lambda_j \prec_i \lambda'.$$

Consequently, σ is not in the core.

Conversely, take any coalition structure σ such that $\sigma \notin \mathbf{core}(\Gamma)$. Then, there exists a cooperation structure $\lambda' = \langle C', \mathbf{s}_{C'} \rangle$ blocking σ. Therefore, for all $i \in C'$

$$\lambda_{\sigma(i)} \prec_i \lambda',$$

and, since there exists no successful cooperation structure, we have, for all $i \in C'$

$$cost_i(\pi_i(\mathbf{s}_{C'})) < cost_i(\pi_i(\mathbf{s}_{C_{\sigma(i)}})).$$

Consequently, there exist some i and some cooperation structure $\lambda_{\sigma(i)}$ in σ for which

$$cost_i(\pi_i(\mathbf{s}_{C_{\sigma(i)}})) > 0.$$

This concludes the proof of (1).

To prove (2), notice that the only grand coalition structure σ' in the core is the one where every agent assigns 0 to each single variable. The splitting of σ' contains all coalition structures based on any possible partition of the set of players. Therefore, there are as many coalition structures in the core as there are partitions of AG, whose number corresponds to the nth Bell number B_n. □

Given any ŁRG Γ, we can associate to it a set of systems of linear inequalities so that at least one of the systems admits a solution if and only if Γ has non-empty core.

[8] The nth Bell number, denoted B_n, [33] is the number of partitions of a set with n elements and is given by the formula

$$B_n = \frac{1}{e} \sum_{k=0}^{\infty} \frac{k^n}{k!}.$$

Theorem 21. *For every Łukasiewicz Resource Game* Γ *there exists a finite set of systems of linear inequalities so that at least one system has a solution if and only if* $\textbf{\textit{core}}(\Gamma) \neq \emptyset$.

Proof. We start with a proof for infinite games. Let $\mathbb{L}$ be the following first-order structure

$$\mathbb{L} = \Big\langle [0,1], \to, \{\delta_n\}_{n>1}, 0 \Big\rangle,$$

where, for all $x, y \in [0,1]$ and natural number $n > 1$

$$x \to y = \min(1 - x + y, 1), \qquad \delta_n x = \frac{1}{n}x.$$

As shown in [24, Lemma 8.3][9], for every quantifier-free formula Φ in the language

$$\mathcal{L} = \langle \to, \{\delta_n\}_{n>1}, 0 \rangle,$$

there exists a formula Φ^* in the language of ordered groups with rational constants

$$\mathcal{L}_{\mathbb{Q}} = \langle +, -, <, \{c\}_{c\in\mathbb{Q}} \rangle$$

such that

$$\mathbb{L} \models \Phi \qquad \text{iff} \qquad \mathbb{R} \models \Phi^*.$$

So, for instance, let

$$f_{\chi_i}(\bar{x}_1, \ldots, \bar{x}_n) = 1,$$

where each $\bar{x}_j$ is a tuple of variables, be the quantifier-free formula in $\mathcal{L}$ defining the set of allocations that satisfy χ_i, where f_{χ_i} is the McNaughton function associated to χ_i written in the language $\mathcal{L}$. There exists a formula $f^*_{\chi_i}$ in $\mathcal{L}_{\mathbb{Q}}$ that defines exactly the same set over the reals.[10]

Then, let $\Phi_{\mathbf{core}}$ be the following first-order formula:

$$\bigsqcap_{C\in\mathcal{C}(\Gamma)} \left(\bigsqcup_{i\in C} \Big(\big((f^*_{\chi_i}(\bar{x}_1, \ldots, \bar{x}_n) = 1) \sqcap \big(\exists \bar{y}_C \forall \bar{y}_{-C}\ (f^*_{\chi_i}(\bar{y}_C, \bar{y}_{-C}) = 1) \Rightarrow (\textstyle\sum \bar{x}_i \leq \sum \bar{y}_i)\big)\big) \sqcup \right.$$
$$\left. \big((f^*_{\chi_i}(\bar{x}_1, \ldots, \bar{x}_n) < 1\big) \sqcap (\sim\exists \bar{z}_C \forall \bar{z}_{-C}\ (f^*_{\chi_i}(\bar{z}_C, \bar{z}_{-C}) = 1) \sqcap (\textstyle\sum \bar{x}_i = 0))) \Big) \right),$$

where:

1. $\sqcap, \sqcup, \Rightarrow, \sim$ denote the classical metalanguage conjunction, disjunction, implication and negation, respectively.
2. $\mathcal{C}(\Gamma)$ is the set of all possible coalitions $C \subseteq$ Ag.
3. $\bar{y}_C, \bar{z}_C$ and $\bar{y}_{-C}, \bar{z}_{-C}$ are the tuples of variables under control of the agents in C and not in C, respectively.

[9] Notice that in [24], this result is proven with respect to the structure

$$\Big\langle [0,1], \oplus, \neg, \{\delta_n\}_{n>1}, 0 \Big\rangle,$$

whose operations are definable from those of $\mathbb{L}$, as shown in Section 2. Conversely, $\to$ is definable from $\oplus$ and $\neg$, since for all $x \in [0,1]$, $x \to y = \neg x \oplus y$.

[10] Notice that, as shown in [24], $f^*_{\chi_i}$ can be built from f_{χ_i} in polynomial time.

4. the formulae
$$f^*_{\chi_i}(\bar{x}_1, \ldots, \bar{x}_n) = 1 \quad \text{and} \quad f^*_{\chi_i}(\bar{x}_1, \ldots, \bar{x}_n) < 1$$
in the language $\langle +, -, <, \{c\}_{c\in\mathbb{Q}}\rangle$ define the satisfiability set of the Łukasiewicz function f_{χ_i} and its complement, respectively.

5. the formulae
$$\exists \bar{y}_C \forall \bar{y}_{-C} \ (f^*_{\chi_i}(\bar{y}_C, \bar{y}_{-C}) = 1 \quad \text{and} \quad {\sim}\exists \bar{z}_C \forall \bar{z}_{-C} \ (f^*_{\chi_i}(\bar{z}_C, \bar{z}_{-C}) = 1)$$
in the language $\langle +, -, <, \{c\}_{c\in\mathbb{Q}}\rangle$ encode the fact that there exists and there does not exist (respectively) an allocation $\bar{y}_C$ ($\bar{z}_C$, resp.) such that for all $\bar{y}_{-C}$ ($\bar{z}_{-C}$, resp.) the payoff of i is maximised.

6. the terms
$$\sum \bar{x}_i \quad \text{and} \quad \sum \bar{y}_i$$
define the sum of the variables controlled by player i in the tuples $\bar{x}_i$ and $\bar{y}_i$, respectively.

$\Phi_{\mathbf{core}}$ is such that
$$\big\{(\mathbf{s}_1, \ldots, \mathbf{s}_n) \mid \mathbb{R} \models \Phi_{\mathbf{core}}(\mathbf{s}_1, \ldots, \mathbf{s}_n)\big\} \quad \text{iff} \quad \big(\langle \text{AG}, (\mathbf{s}_1, \ldots, \mathbf{s}_n)\rangle\big) \in \mathbf{core}(\Gamma).$$
In fact, $\Phi_{\mathbf{core}}$ is satisfied by all tuples $(\mathbf{s}_1, \ldots, \mathbf{s}_n) \in \mathbb{R}^{m\cdot n}$, such that, for all possible coalitions $C \in \mathcal{C}(\Gamma)$ and at least one agent $i \in C$:

1. $f_{\chi_i}(\mathbf{s}_1, \ldots, \mathbf{s}_n) = 1$, i.e. i's payoff is maximised, and if there exists a joint allocation $\bar{y}_i$ making the cooperation in C successful for i, then the cost $\sum \bar{x}_i$ for agent i is less than or equal to $\sum \bar{y}_i$,
   or
2. $f_{\chi_i}(\mathbf{s}_1, \ldots, \mathbf{s}_n) < 1$, i.e. i's payoff is not maximised, i incurs no cost, i.e., $\sum \bar{x}_i = 0$, and there exists no joint allocation $\bar{z}_i$ making the cooperation in C successful for i, i.e.
$$ {\sim}\exists \bar{z}_C \forall \bar{y}_{-C} \ (f^*_{\chi_i}(\bar{z}_C, \bar{z}_{-C})) = 1.$$

Now, $\Phi_{\mathbf{core}}$ is a formula in the language of ordered groups with rational constants and so is equivalent to a quantifier free formula $\Phi^{\mathrm{fr}}_{\mathbf{core}}$ in the same language [24, 25, 21]. $\Phi^{\mathrm{fr}}_{\mathbf{core}}$ can be rewritten in disjunctive normal form, so that it is equivalent to a disjunction of conjunctions of linear inequalities, i.e.:
$$\bigsqcup_{i=1}^{t} \left(\prod_{j=1}^{r} \mathsf{p}_{ij}(\bar{x}_1, \ldots, \bar{x}_n) \triangledown 0 \right),$$
where each
$$\mathsf{p}_{ij}(\bar{x}_1, \ldots, \bar{x}_n) \triangledown 0$$
is a linear polynomial inequality[11] with integer coefficients and $\triangledown \in \{<, >, \leq, \geq\}$. Therefore, $\Phi^{\mathrm{fr}}_{\mathbf{core}}$'s satisfiability is equivalent to the existence of a solution of at least one of the systems
$$\prod_{j=1}^{r} \mathsf{p}_{ij}(\bar{x}_1, \ldots, \bar{x}_n) \triangledown 0.$$

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

It then follows that non-emptiness of the core is equivalent to the satisfiability of $\Phi^{\text{fr}}_{\mathbf{core}}$.

As for finite games, let $\mathbb{L}_k$ be the following first-order structure

$$\mathbb{L}_k = \left\langle L_k, \to, \{c\}_{c \in L_k} \right\rangle,$$

where, for all $x, y \in L_k$

$$x \to y = \min(1 - x + y, 1).$$

Following the reasoning of [24, Lemma 8.3], it is easy to show that for every quantifier-free formula Φ in the language

$$\mathcal{L}_k = \langle \to, \{c\}_{c \in L_k} \rangle,$$

there exists a formula Φ^* in the language

$$\mathcal{L}_{\mathbb{Z}} = \langle +, -, <, \{X_n\}_{n>2}, \{c\}_{c \in \mathbb{Z}} \rangle$$

of Presburger Arithmetic (see [25]) with constants for each integer and predicates X_n for the set of integers divisible by n, such that

$$\mathbb{L}_k \models \Phi \qquad \text{iff} \qquad \mathbb{Z} \models \Phi^*.$$

The rest of the proof then follows exactly the above argument for infinite games using the fact that Presburger Arithmetic admits elimination of quantifiers in the above language. □

The following is a trivial consequence of the above theorem, given the fact that every formula in $\mathcal{L}_{\mathbb{Q}}$ defines a non-empty set over $\mathbb{R}$ if and only if so it does on $\mathbb{Q}$ [21].

Corollary 22. *The core of a Łukasiewicz Resource Game is non-empty if and only if it contains at least a rational coalition structure.*

5. Translating Coalitional Resource Games into Finite Łukasiewicz Resource Games

In the introduction, we claimed that, in a precise formal sense, Łukasiewicz Resource Games can be understood to be a generalisation of the Coalitional Resource Games model (CRG) of Wooldridge and Dunne [36]. The aim of this section is to make good this claim. We show that any CRG can be translated into a finite ŁRG that is strategically equivalent in that it preserves the non-emptiness of the core. We start by recalling the definition of a CRG and some of their main properties[12], along with an example. We then show how to build a ŁRG from an arbitrary CRG. This will then make it possible to show that there exist ŁRGs with an empty core, by exploiting a translation of a CRG game not having any stable coalitions.

Notation 3. *In the rest of this section, when necessary, we will be using the superscripts* $^{\text{crg}}$ *and* $^{\text{lrg}}$ *to make the context clear for some notions that are similar for both CRGs and ŁRGs.*

[12] Bear in mind that the concepts and terminology we introduce here are specific to CRGs. Their difference w.r.t. similar notions for ŁRGs should be clear from the context.

5.1. Coalitional Resource Games

Coalitional Resource Games are models of strategic interaction where each player has a finite set of goals and is endowed with a finite set of resources. Each agent's objective is to achieve at least one of their goals, and to do this, each agent must contribute a certain quantity of each resource. In order to do so, players may cooperate by joining forces with others, and while their main objective is the achievement of some of their goals, their secondary aim is to minimise their costs in doing so.

More formally, we have:

Definition 23 (Coalitional Resource Games [36, 15]). *A* Coalitional Resource Game $\mathcal{G}$ *is a tuple*

$$\mathcal{G} = \langle \textsc{Ag}, \textsc{Goal}, \textsc{Res}, \textsc{Goal}_1, \ldots, \textsc{Goal}_n, \mathrm{en}, \mathrm{req} \rangle$$

where

1. $\textsc{Ag} = \{A_1, \ldots, A_n\}$ *is a set of agents;*
2. $\textsc{Goal} = \{g_1, \ldots, g_m\}$ *is a set of possible goals;*
3. $\textsc{Res} = \{r_1, \ldots, r_t\}$ *is a set of resources;*
4. *for each* $i \in \textsc{Ag}$,
$$\textsc{Goal}_i = \{g_{i1}, \ldots, g_{ih}\} \subseteq \textsc{Goal}$$
is a set of goals;
5. $\mathrm{en} : \textsc{Ag} \times \textsc{Res} \to \mathbb{N}$ *is an endowment function, with the intended interpretation that if* $\mathrm{en}(i, r) = c$*, then agent* $i \in \textsc{Ag}$ *is endowed with quantity* $c \in \mathbb{N}$ *of resource* $r \in \textsc{Res}$*; and*
6. $\mathrm{req} : \textsc{Goal} \times \textsc{Res} \to \mathbb{N}$ *is a requirement function, with the intended interpretation that if* $\mathrm{req}(g, r) = c$*, then to achieve a goal* $g \in \textsc{Goal}$*, it is necessary to expend quantity* $c \in \mathbb{N}$ *of resource* $r \in \textsc{Res}$*.*

The above definition comes with the assumption that trying to achieve a goal always requires the expenditure of at least one unit, i.e.

$$\forall g \in \textsc{Goal}, \exists r \in \textsc{Res} \text{ such that } \mathrm{req}(g, r) > 0.$$

The endowment function can be extended to coalitions via the function $en : 2^{\textsc{Ag}} \times \textsc{Res} \to \mathbb{N}$:

$$en(C, r) = \sum_{i \in C} \mathrm{en}(i, r).$$

Similarly, the requirement function can be extended to sets of goals via the function $req : 2^{\textsc{Goal}} \times \textsc{Res} \to \mathbb{N}$:

$$req(\textsc{Goal}', r) = \sum_{g \in \textsc{Goal}'} \mathrm{req}(g, r).$$

With a small abuse of notation, we also use $req(\textsc{Goal}')$ to denote the total cost of resources that are required to satisfy the set of goals $\textsc{Goal}'$:

$$req(\textsc{Goal}') = \sum_{r \in \textsc{Res}} req(\textsc{Goal}', r).$$

Definition 24. *For any Coalitional Resource Game $\mathcal{G}$, a set of goals* $\textsc{Goal}'$ satisfies *agent i if* $\textsc{Goal}_i \cap \textsc{Goal}' \neq \emptyset$*;* $\textsc{Goal}'$ *satisfies a coalition $C \subseteq \textsc{Ag}$ if it satisfies every member of C.*

A set of goals $\textsc{Goal}'$ *is* feasible *for coalition C if that coalition is endowed with sufficient resources to achieve all the goals in* $\textsc{Goal}'$*. We define a function*

$$\mathrm{sf} : 2^{\textsc{Ag}} \to 2^{2^{\textsc{Goal}}}$$

to return the set of goal sets that both satisfy and are feasible for a given coalition:

$$\mathrm{sf}(C) = \{\textsc{Goal}' \subseteq \textsc{Goal} : \textsc{Goal}' \textit{ is feasible for and satisfies } C\}.$$

We say that a coalition C is successful *if* $\mathrm{sf}(C) \neq \emptyset$*. If C is successful, then C is endowed with resources to bring about some goal set* $\textsc{Goal}'$ *such that* $\textsc{Goal}'$ *will satisfy every member of C.*[13]

Example 25 ([15])**.** *Consider the following example. There are four agents* $\textsc{Ag} = \{A_1, A_2, A_3, A_4\}$*, with possible goals* $\textsc{Goal} = \{g_1, g_2, g_3, g_4\}$*, and three resources* $\textsc{Res} = \{r_1, r_2, r_3\}$*. The goal sets for each agent are as follows:*

$$\textsc{Goal}_1 = \{g_1\} \qquad \textsc{Goal}_2 = \{g_2\} \qquad \textsc{Goal}_3 = \{g_3\} \qquad G_4 = \{g_4\}.$$

The requirement and endowment functions req *and* en *are defined in the following tables:*

req	r_1	r_2	r_3
g_1	4	0	1
g_2	4	1	0
g_3	4	1	0
g_4	4	0	2

en	r_1	r_2	r_3
A_1	8	1	0
A_2	8	0	1
A_3	8	0	1
A_4	4	2	0

As shown in [15], this CRG has an empty core.

We recall now the notions of a cooperation structure, coalition structure and core of CRGs.

Definition 26. *Given a Coalitional Resource Game $\mathcal{G}$, a* resource vector *for agent i is an element*

$$\boldsymbol{\xi}_i = (\alpha_{i1}, \ldots, \alpha_{it}) \in \mathbb{N}^t$$

with, for all $r \in \textsc{Res}$, $\alpha_{ir} \leq \mathrm{en}(i, r)$. Given a coalition $C = \{A_1, \ldots, A_{n'}\}$, a contribution vector *$\boldsymbol{\xi}_C$ is a tuple of resource vectors, i.e.*

$$\boldsymbol{\xi}_C = (\boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_{n'}).$$

A contribution vector indicates the allocation to each specific resource of each player i in the coalition. Clearly, this contribution must always be less than or equal to i's endowment for a particular resource.

[13] Notice that the notion of being successful for a coalition in a CRG differs from the one of successful cooperation structure as defined for ŁRGs.

Definition 27. *A* cooperation structure *in a Coalitional Resource Game $\mathcal{G}$ is a triple*

$$\lambda^{\text{crg}} = \langle C, \textsc{Goal}_C, \boldsymbol{\xi}_C \rangle,$$

where $C \subseteq \textsc{Ag}, \textsc{Goal}_C \subseteq \textsc{Goal}$ and $\boldsymbol{\xi}_C$ is a contribution vector, such that these components must satisfy the feasibility constraint that the coalition is endowed with sufficient resources to achieve the goals it commits to, i.e.:

$$\forall r \in \textsc{Res}, \quad \sum_{i \in C} \alpha_{ir} \geq req(\textsc{Goal}_C, r).$$

The intended interpretation is that coalition C will cooperate to achieve the set of goals $\textsc{Goal}_C$, and that each agent i will contribute resources $\boldsymbol{\xi}_i = (\alpha_{i1}, \ldots, \alpha_{it})$ towards this joint effort. Given a cooperation structure λ^{crg}, we denote by $succ^{\text{crg}}(\lambda^{\text{crg}})$ the set of agents in C that would have some of their goals satisfied by λ^{crg}.

Definition 28. *In a Coalitional Resource Game $\mathcal{G}$, a* cost function *for agent i with respect to a resource vector $\boldsymbol{\xi}_i = (\alpha_{i1}, \ldots, \alpha_{it})$ is defined as the sum of the contribution to each resource, i.e.*

$$cost_i^{\text{crg}}(\boldsymbol{\xi}_i) = \sum_{r \in R} \alpha_{ir}.$$

$cost_i^{\text{crg}}(\lambda^{\text{crg}})$ denotes the cost for agent i w.r.t. their contribution vector within coalition C.

For every agent i it is possible to define a preference relation $\preceq_i^{\text{crg}}$ between cooperation structures as follows.

Definition 29. *Given two cooperation structures $\lambda_1^{\text{crg}}, \lambda_2^{\text{crg}}$ containing agent i in a Coalitional Resource Game $\mathcal{G}$, we define*

$$\lambda_2^{\text{crg}} \preceq_i^{\text{crg}} \lambda_1^{\text{crg}},$$

whenever one of these conditions is satisfied:

1. *$i \notin succ^{\text{crg}}(\lambda_1^{\text{crg}}), i \notin succ^{\text{crg}}(\lambda_2^{\text{crg}})$, and $cost_i^{\text{crg}}(\lambda_1^{\text{crg}}) \leq cost_i^{\text{crg}}(\lambda_2^{\text{crg}})$.*
2. *$i \in succ^{\text{crg}}(\lambda_1^{\text{crg}}), i \in succ^{\text{crg}}(\lambda_2^{\text{crg}})$, and $cost_i^{\text{crg}}(\lambda_1^{\text{crg}}) \leq cost_i^{\text{crg}}(\lambda_2^{\text{crg}})$.*
3. *$i \in succ^{\text{crg}}(\lambda_1^{\text{crg}})$ and $i \notin succ^{\text{crg}}(\lambda_2^{\text{crg}})$.*

The relation $\prec_i^{\text{crg}}$ is defined in the obvious way.

Observe that the preference relations defined above are quasi-dichotomous: a player prefers all cooperation structures that result in their goal being achieved over all those that do not, but secondarily prefers to minimise costs.

Definition 30. *A* coalition structure *in a Coalitional Resource Game $\mathcal{G}$*

$$\sigma^{\text{crg}} = \{\lambda_1^{\text{crg}}, \ldots, \lambda_{m'}^{\text{crg}}\}$$

is a set of cooperation structures, so that $C_1, \ldots, C_{m'}$ form a partition of $\textsc{Ag}$, where C_i is the coalition in the cooperation structure λ_i^{crg}.

Definition 31. *In a Coalitional Resource Game $\mathcal{G}$, we say that a cooperation structure $\lambda^{\text{crg}} = \langle C, G', \boldsymbol{\xi} \rangle$* blocks *a coalition structure σ^{crg} if for all $i \in C$*

$$\lambda_{\sigma^{\text{crg}}(i)}^{\text{crg}} \prec_i^{\text{crg}} \lambda^{\text{crg}},$$

where $\lambda_{\sigma^{\text{crg}}(i)}^{\text{crg}}$ is the cooperation structure in σ^{crg} to which i belongs. A coalition structure is stable *if it is not blocked by any cooperation structure. The* core *of a Coalitional Resource Game $\mathbf{core}^{\text{crg}}(\mathcal{G})$ is the set of all stable coalition structures.*

5.2. Translation

We now present our core-preserving translation from CRGs to ŁRGs. Let

$$\mathcal{G} = \langle \textsc{Ag}, \textsc{Goal}, \textsc{Res}, \textsc{Goal}_1, \ldots, \textsc{Goal}_n, \mathrm{en}, \mathrm{req} \rangle$$

be a CRG. Let k be the maximum between the total amount of resources required to achieve all goals and the total amount of resources available to all agents, i.e.:

$$k = \max\left(\mathit{req}(\textsc{Goal}), \sum_{r \in \textsc{Res}} \mathit{en}(\textsc{Ag}, r) \right).$$

Take the finite valued Łukasiewicz logic with constants $Ł_k^c$, and define the following ŁRG:

$$\Gamma(\mathcal{G}) = \langle \textsc{Ag}, \textsc{Var}, \textsc{Const}, \textsc{Val}, \textsc{Task}, \textsc{Payoff} \rangle,$$

where

1. $\textsc{Ag}$ is the same set of agents as in $\mathcal{G}$.
2. $\textsc{Var} = \{\textsc{Var}_1, \ldots, \textsc{Var}_n\}$, where each

   $$\textsc{Var}_i = \{p_i^{r_1}, \ldots, p_i^{r_t}\}$$

   contains one propositional variable per resource $r \in \textsc{Res}$ for player i.
3. $\textsc{Const} = \{\theta_1, \ldots, \theta_n\}$ is a set of $Ł_k^c$-formulae, with

   $$\theta_i := \bigwedge_{j=1}^{t} p_i^{r_j} \to \overline{c}_i^{r_j},$$

   where

   $$c_i^{r_j} = \frac{\mathrm{en}(i, r_j)}{k}.$$
4. $\textsc{Val} = \{\textsc{Val}_1, \ldots, \textsc{Val}_n\}$, where, for each player i

   $$\textsc{Val}_i = \{s_i : \textsc{Var}_i \to L_k \mid s_i(\theta_i) = 1\},$$

   i.e., $\textsc{Val}_i$ is the set of all valuations of $\{p_i^{r_1}, \ldots, p_i^{r_t}\}$ such that $\forall r_j \in \textsc{Res}$,

   $$s_i(p_i^{r_j}) \cdot k \leq \mathrm{en}(i, r_j).$$
5. $\textsc{Task} = \{\tau_1, \ldots, \tau_t\}$ is a set of $Ł_k^c$-formulae such that, for each $1 \leq j \leq t$,

   $$\tau_j := \bigoplus_{i=1}^{n} p_i^{r_j}$$
6. $\textsc{Payoff} = \{\chi_1, \ldots, \chi_n\}$ is a set of $Ł_k^c$-formulae such that, for each i,

   $$\chi_i := \bigvee_{g \in \textsc{Goal}_i} \left(\Delta \left(\bigwedge_{j=1}^{t} \left(\overline{d}_g^{r_j} \to \tau_j \right) \right) \right),$$

   where $\tau_j \in \textsc{Task}$, and

   $$d_g^{r_j} = \frac{\mathrm{req}(g, r_j)}{k}.$$

The above construction deserves some clarification. First of all, given any $\mathcal{G}$, $\Gamma(\mathcal{G})$ is easily seen to be a ŁRG on Ł^c_k. Every allocation of resources of amount $n \in \mathbb{N}$ from $\mathcal{G}$ is transformed into an allocation of amount $\frac{n}{k}$ in $\Gamma(\mathcal{G})$. The choice of k as the maximum of $req(\textsc{Goal})$ and $\sum_{r \in \textsc{Res}} en(\textsc{Ag}, r)$ is due to the fact that we want to be able to define both the values of the endowment and requirement functions in $\mathcal{G}$ and the sum of these values in a finite-valued Łukasiewicz logic overcoming the limitation intrinsic to the use of the bounded sum operator $\oplus$, which behaves like $+$ only if the sum of its arguments is less than or equal to 1. The choice of k must then be done to ensure that, for the cases we are interested in, $\oplus$ functions as the regular addition of natural numbers.

$\Gamma(\mathcal{G})$ is built from the same set of agents as $\mathcal{G}$. Each agent i is in charge of as many propositional variables as there are resources and is assigned a formula θ_i that determines how much of these resources are available to i. So, the set of all possible allocations for an agent i is the set of all possible valuations s_i that satisfy θ_i, meaning that the agent is actually allowed to use only up to the available amount of each resource. The task formulae τ_j encode the total quantity of resource j allocated by all agents. Finally, the payoff formulae χ_i encode the fact that player i maximises their payoff if at least one of their goals is achieved. Each χ_i is satisfiable if and only if for at least one goal $g \in \textsc{Goal}_i$, the combined allocation τ_j of all players of each resource r_j needed to achieve g is at least as great as the required amount $d_g^{r_j} = \frac{\mathrm{req}(g, r_j)}{k}$.

Definition 32. *Given any Coalitional Resource Game $\mathcal{G}$, the structure $\Gamma(\mathcal{G})$ is called the* Łukasiewicz resource game defined from $\mathcal{G}$.

In the rest of this section, we will always use $\mathcal{G}$ to refer to an arbitrary CRG and $\Gamma(\mathcal{G})$ to denote the ŁRG defined from it.

Example 33. *To make the construction clear, we show how the translation of Example 25 works. A simple calculation shows that $req(\textsc{Goal}) = 21$ and $\sum_{r \in \textsc{Res}} en(\textsc{Ag}, r) = 33$. So we use the logic Ł^c_{33} and define the game*

$$\langle \textsc{Ag}, \textsc{Var}, \textsc{Const}, \textsc{Val}, \textsc{Task}, \textsc{Payoff} \rangle$$

as follows:

1. $\textsc{Ag} = \{A_1, A_2, A_3, A_4\}$.
2. $\textsc{Var} = \{\textsc{Var}_1, \textsc{Var}_2, \textsc{Var}_3, \textsc{Var}_4\}$, *where, for each i,*
$$V_i = \{p_i^{r_1}, p_i^{r_2}, p_i^{r_3}\}.$$
3. $\textsc{Const} = \{\theta_1, \theta_2, \theta_3, \theta_4\}$ *is a set of Ł^c_{33}-formulae, such that*
$$\theta_1 := \left(p_1^{r_1} \to \overline{\frac{8}{33}}\right) \wedge \left(p_1^{r_2} \to \overline{\frac{1}{33}}\right) \wedge \left(p_1^{r_3} \to \overline{0}\right)$$
$$\theta_2 := \left(p_2^{r_1} \to \overline{\frac{8}{33}}\right) \wedge \left(p_2^{r_2} \to \overline{0}\right) \wedge \left(p_2^{r_3} \to \overline{\frac{1}{33}}\right)$$
$$\theta_3 := \left(p_3^{r_1} \to \overline{\frac{8}{33}}\right) \wedge \left(p_3^{r_2} \to \overline{0}\right) \wedge \left(p_3^{r_3} \to \overline{\frac{1}{33}}\right)$$
$$\theta_4 := \left(p_4^{r_1} \to \overline{\frac{4}{33}}\right) \wedge \left(p_4^{r_2} \to \overline{\frac{2}{33}}\right) \wedge \left(p_4^{r_3} \to \overline{0}\right)$$

4. $\textsc{Val} = \{\textsc{Val}_1, \textsc{Val}_2, \textsc{Val}_3, \textsc{Val}_4\}$, *where, for each player i*

$$\textsc{Val}_i = \{s : \textsc{Var}_i \to L_{33} \mid s(\theta_i) = 1\},$$

i.e., $\textsc{Val}_i$ *is the set of all valuations of* $\{p_i^{r_1}, p_i^{r_2}, p_i^{r_3}\}$ *that satisfy* θ_i*.*

5. $\textsc{Task} = \{\tau_1, \tau_2, \tau_3\}$ *is a set of* $Ł_{33}^c$*-formulae such that, for each* $1 \leq j \leq 3$*,*

$$\tau_j := \bigoplus_{i=1}^{4} p_i^{r_j}.$$

6. $\textsc{Payoff} = \{\chi_1, \chi_2, \chi_3, \chi_4\}$ *is a set of* $Ł_{33}^c$*-formulae such that,*

$$\chi_1 := \Delta\left(\left(\overline{\frac{4}{33}} \to \tau_1\right) \wedge \left(\overline{\frac{1}{33}} \to \tau_3\right)\right)$$

$$\chi_2 := \Delta\left(\left(\overline{\frac{4}{33}} \to \tau_1\right) \wedge \left(\overline{\frac{1}{33}} \to \tau_2\right)\right)$$

$$\chi_3 := \Delta\left(\left(\overline{\frac{4}{33}} \to \tau_1\right) \wedge \left(\overline{\frac{1}{33}} \to \tau_2\right)\right)$$

$$\chi_4 := \Delta\left(\left(\overline{\frac{4}{33}} \to \tau_1\right) \wedge \left(\overline{\frac{2}{33}} \to \tau_3\right)\right)$$

5.3. Correspondence between the Cores

We are now going to show that the core of any CRG is non-empty if and only if so is the core of its related ŁRG. In order to do so, we need some preliminary results.

Lemma 34. *Let $\mathcal{G}$ be any Coalitional Resource Game such that core$(\mathcal{G}) \neq \emptyset$. Then there exists a stable coalition structure $\sigma^{\mathrm{crg}} = \langle \lambda_1^{\mathrm{crg}}, \ldots, \lambda_{n'}^{\mathrm{crg}} \rangle$ where for each $\lambda_j^{\mathrm{crg}} = \langle C_j, \textsc{Goal}_{C_j}, \boldsymbol{\xi}_{C_j} \rangle$ one of the following conditions holds true:*

1. $\textsc{Goal}_{C_j} \in \mathrm{sf}(C_j)$*,*
2. $|C_j| = 1$ *and* $\mathrm{sf}(C_j) = \emptyset$*.*

Proof. We have to prove that in any CRG with a non-empty core there always exists a stable coalition structure such that every cooperation structure in it either has all its members achieving some of their goals or it contains only one agent that is not successful.

So, take any stable coalition structure

$$\sigma^{\mathrm{crg}} = \langle \lambda_1^{\mathrm{crg}}, \ldots, \lambda_{n'}^{\mathrm{crg}} \rangle$$

and suppose that for some cooperation structure $\lambda_j^{\mathrm{crg}} = \langle C_j, \textsc{Goal}_{C_j}, \boldsymbol{\xi}_{C_j} \rangle$, $\textsc{Goal}_{C_j} \notin \mathrm{sf}(C_j)$. By Definition 24, this means that either $\textsc{Goal}_{C_j}$ is not feasible for C_j or it does not satisfy C_j. By Definition 27, $\textsc{Goal}_{C_j}$ must be feasible for C_j and so there must be some agent $A_i \in C_j$ for whom

$$\textsc{Goal}_{C_j} \cap \textsc{Goal}_i = \emptyset.$$

Then, i's contribution to the coalition must be zero, since σ^{crg} is stable. In fact, if i's contribution was positive, i.e., non-zero, then i would always prefer to spend nothing at all, given none of the player's goals are achieved by being in C_j. This would then mean that i alone would be able to block σ^{crg}, contradicting our assumption of stability (see also [15, Proposition 5]). So, if $\textsc{Goal}_{C_j} \notin \text{sf}(C_j)$, there exists some A_i who makes no contribution to λ_j^{crg} and whose goals are not in $\textsc{Goal}_{C_j}$. Then, λ_j^{crg} can be split into two cooperation structures

$$\lambda_j^{\text{crg}\prime} = \left\langle C_j \setminus \{A_i\}, \textsc{Goal}_{C_j}, \boldsymbol{\xi}_{C_j\setminus\{A_i\}} \right\rangle \qquad \lambda_j^{\text{crg}\prime\prime} = \left\langle \{A_i\}, \emptyset, \mathbf{0}_{\{A_i\}} \right\rangle,$$

where $\boldsymbol{\xi}_{C_j\setminus\{A_i\}}$ is the contribution vector obtained by excluding A_i's contribution from $\boldsymbol{\xi}_{C_j}$, and $\mathbf{0}_{\{A_i\}}$ is the contribution vector where agent i allocates no resources.

Take now the new coalition structure

$$\sigma^{\text{crg}\prime} = \langle \lambda_1^{\text{crg}}, \dots, \lambda_j^{\text{crg}\prime}, \lambda_j^{\text{crg}\prime\prime}, \dots \lambda_{n'}^{\text{crg}} \rangle$$

obtained from σ^{crg} by replacing λ_j^{crg} with $\lambda_j^{\text{crg}\prime}$ and $\lambda_j^{\text{crg}\prime\prime}$. It is trivial to see that σ^{crg} is, once again, stable, since any cooperation structure blocking $\sigma^{\text{crg}\prime}$ would have blocked σ^{crg} in the first place. Consequently, σ^{crg} can be recursively decomposed into a stable coalition structure where each cooperation structure either contains successful coalitions or it contains one agent who is not successful and allocates no resources. □

Something very similar can be proven for Łukasiewicz resource games defined from a CRG.

Lemma 35. *Let $\Gamma(\mathcal{G})$ be the Łukasiewicz resource game defined from a Coalitional Resource Game $\mathcal{G}$. If $\mathbf{core}^{\text{lrg}}(\Gamma(\mathcal{G})) \neq \emptyset$, then it contains a stable coalition structure*

$$\sigma^{\text{lrg}} = \langle \lambda_1^{\text{lrg}}, \dots, \lambda_w^{\text{lrg}} \rangle,$$

where each $\lambda_j^{\text{lrg}} = \langle C_j, \mathbf{s}_{C_j} \rangle$ is either globally successful or is such that $C_j = \{A_i\}$, for some agent i, and $\mathbf{s}_{C_j}$ is $\mathbf{0}_{A_i}$, i.e., the vector where i allocates no resources.

Proof. Suppose $\Gamma(\mathcal{G})$'s core is non-empty and take any stable coalition structure

$$\sigma^{\text{lrg}} = \langle \lambda_1^{\text{lrg}}, \dots, \lambda_w^{\text{lrg}} \rangle.$$

Suppose some λ_j^{lrg} is not globally successful. Then there is some agent i not maximising their payoff function. If this agent allocates any resources to λ_j^{lrg}, then the agent can always block σ^{lrg} by forming an individual cooperation structure that is still not successful for i but in which the agent does not allocate any resources, therefore incurring lower costs. This means that i's contribution to λ_j^{lrg} must be null, and so λ_j^{lrg} can be decomposed into a cooperation structure containing two disjoint structures

$$\lambda_j^{\text{lrg}\prime} = \langle C_j \setminus \{A_i\}, \pi_{C_j\setminus\{A_i\}}(\mathbf{s}_{C_j}) \rangle, \qquad \lambda_j^{\text{lrg}\prime\prime} = \langle \{A_i\}, \mathbf{0}_{A_i} \rangle.$$

By repeating the process, σ^{lrg} can be eventually split into a stable coalition structure σ'^{lrg} containing cooperations structures where either all their members maximise their payoff or having only one agent who does not allocate any resources. □

We can now show that the concept of a Łukasiewicz resource game properly generalises the concept of a CRG.

Theorem 36. *For any Coalitional Resource Game $\mathcal{G}$, let $\Gamma(\mathcal{G})$ be the Łukasiewicz resource game defined from it. Then:*

$$\mathbf{core}^{\mathrm{crg}}(\mathcal{G}) \neq \emptyset \qquad \textit{iff} \qquad \mathbf{core}^{\mathrm{lrg}}(\Gamma(\mathcal{G})) \neq \emptyset.$$

Proof. Suppose $\mathbf{core}^{\mathrm{crg}}(\mathcal{G}) \neq \emptyset$. Then, by Lemma 34, we know that there exists a stable coalition structure

$$\sigma^{\mathrm{crg}} = \langle \lambda_1^{\mathrm{crg}}, \ldots, \lambda_t^{\mathrm{crg}} \rangle$$

where each $\lambda_j^{\mathrm{crg}} = \langle C_j, \textsc{Goal}_{C_j}, \boldsymbol{\xi}_{C_j} \rangle$ is such that either $\textsc{Goal}_{C_j} \in \mathrm{sf}(C_j)$, or $C_j = \{A_i\}$, for some $A_i \in \textsc{Ag}$, and $\mathrm{sf}(C_j) = \emptyset$. For each λ_j^{crg}, define the following structure

$$\lambda_j^{\mathrm{lrg}} = \left\langle C_j,\ \frac{1}{k} \cdot \boldsymbol{\xi}_{C_j} \right\rangle.$$

Clearly, each λ_j^{lrg} is a well-defined cooperation structure of $\Gamma(\mathcal{G})$, and, by construction, is such that either

1. for each $i \in C_j$
$$f_{\chi_i}\left(\frac{1}{k} \cdot \boldsymbol{\xi}_{C_j}, \mathbf{s}_{-C_j}\right) = 1$$
for all allocations $\mathbf{s}_{-C}$ (i.e., λ_j^{lrg} is globally successful), or
2. C_j contains only one agent i, λ_j^{lrg} is not successful for i, and i allocates no resources.

We claim that the coalition structure

$$\sigma^{\mathrm{lrg}} = \left\langle \lambda_1^{\mathrm{lrg}}, \ldots, \lambda_t^{\mathrm{lrg}} \right\rangle$$

is stable. Suppose that is not the case, i.e., there exists a cooperation structure $\lambda^{\mathrm{lrg}\star} = \langle C^\star, \mathbf{s}_{C^\star} \rangle$ that blocks σ^{lrg}. This clearly means that for each $i \in C^\star$ and for all $\mathbf{s}_{-C^\star}$

$$f_{\chi_i}\left(\mathbf{s}_{C^\star}, \mathbf{s}_{-C^\star}\right) = 1$$

and one of the following cases holds:

(1) $\lambda^{\mathrm{lrg}}_{\sigma^{\mathrm{lrg}}(i)}, \lambda^{\mathrm{lrg}\star} \in \mathit{succ}(i)$ and

$$\mathit{cost}_i\left(\pi_i\left(\mathbf{s}_{C^\star}\right)\right) < \mathit{cost}_i\left(\pi_i\left(\mathbf{s}_{C_{\sigma^{\mathrm{lrg}}(i)}}\right)\right),$$

where $\pi_i\left(\mathbf{s}_{C_{\sigma^{\mathrm{lrg}}(i)}}\right)$ is i's allocation in the cooperation structure $\lambda^{\mathrm{lrg}}_{\sigma^{\mathrm{lrg}}(i)}$ the agent belongs to in σ^{lrg}.

(2) $\lambda^{\mathrm{lrg}}_{\sigma^{\mathrm{lrg}}(i)} \notin \mathit{succ}(i)$ and $\lambda^{\mathrm{lrg}\star} \in \mathit{succ}(i)$.

Define from $\lambda^{\mathrm{lrg}\star}$ the following structure:

$$\lambda^{\mathrm{crg}\star} = \langle C^\star, \textsc{Goal}^\star, k \cdot \mathbf{s}_{C^\star} \rangle,$$

where

$$\textsc{Goal}^\star = \bigcup_{i \in C^\star} \textsc{Goal}_i,$$

i.e., $\textsc{Goal}^\star$ is the set of all goals of the agents in $C^\star$. It is clear by construction that all the agents in $\lambda^{\mathrm{crg}\star}$ achieve some of their goals. However $\lambda^{\mathrm{crg}\star}$ blocks λ^{crg}, since each i would either achieve some goals by reducing their costs (1), or achieve some goals where they were unsuccessful in the first place (2). This though contradicts our assumption that σ^{crg} is stable. Consequently, σ^{lrg} must be a stable coalition structure and $\mathbf{core}^{\mathrm{lrg}}(\Gamma(\mathcal{G})) \neq \emptyset$.

To prove the converse, suppose $\mathbf{core}^{\mathrm{lrg}}(\Gamma(\mathcal{G}))$ is non-empty. By Lemma 35, we know the core contains a stable coalition structure

$$\sigma^{\mathrm{lrg}} = \langle \lambda_1^{\mathrm{lrg}}, \ldots, \lambda_w^{\mathrm{lrg}} \rangle,$$

where each $\lambda_j^{\mathrm{lrg}} = \langle C_j, \mathbf{s}_{C_j} \rangle$ is either globally successful or it only contains one agent not allocating any resources.

Define, for each $\lambda_j^{\mathrm{lrg}} = \langle C_j, \mathbf{s}_{C_j} \rangle$, a structure

$$\lambda_j^{\mathrm{crg}} = \langle C_j, \textsc{Goal}^*, k \cdot \mathbf{s}_{C_j} \rangle,$$

where C_j is the same coalition as in λ_j^{lrg} and G^* either is the set of all goals of the agents in C_j, if λ_j^{lrg} is globally successful, or it corresponds to $\emptyset$ otherwise. It is easy to see that each λ_j^{crg} is a well-defined cooperation structure of $\mathcal{G}$. Moreover, if λ_j^{lrg} is globally successful then all the agents in λ_j^{crg} achieve some of their goals, while if λ_j^{lrg} contains one agent not allocating any resource, then so does λ_j^{crg} and the agent does not achieve any goal.

We claim that

$$\sigma^{\mathrm{crg}} = \langle \lambda_1^{\mathrm{crg}}, \ldots, \lambda_w^{\mathrm{crg}} \rangle$$

is a stable coalition of $\mathcal{G}$. Suppose that is not the case, i.e., there exists some cooperation structure

$$\lambda^{\mathrm{crg}\star} = \langle C, \textsc{Goal}^\star, \boldsymbol{\xi}_C \rangle$$

that blocks σ^{crg}. This means that for every agent $i \in C$,

$$\lambda^{\mathrm{crg}}_{\sigma^{\mathrm{crg}}(i)} \prec_i^{\mathrm{crg}} \lambda^{\mathrm{crg}\star}$$

and i has some of her goals achieved. By an argument similar to the previous case, it is easy to see that this would contradict our assumption of stability for σ^{lrg}. □

From the above theorem and the fact that there exist CRGs with an empty core (see Example 25), we obtain the following:

Proposition 37. *There exists a Łukasiewicz resource game with an empty core.*

6. Complexity

In this section, we present some results about the computational complexity of decision problems on Łukasiewicz resource games.

Proposition 38. *For any Łukasiewicz resource game* Γ*, checking whether* Γ *is satisfiable is* NP-complete*.*

Proof. First, recall that checking satisfiability of a formula in any Łukasiewicz logic is NP-complete [11]. For membership, notice that any ŁRG Γ is satisfiable if and only if, for at least one player i, their payoff formula χ_i is satisfiable. In fact, if Γ is satisfiable, there is a non-empty coalition C and an allocation by all players so that each χ_i, with $i \in C$, is satisfiable. Conversely, if any χ_i is satisfiable there is a valuation s such that $s(\chi_i) = 1$. This valuation s can be trivially extended to an allocation of all players that satisfies χ_i. So, checking satisfiability for Γ is the same as checking satisfiability of χ_i for each player i in the related Łukasiewicz logic, which can be done in nondeterministic polynomial time.

As for hardness, we show that satisfiability for any Łukasiewicz logic can be reduced in polynomial time to satisfiability of a coalition in a ŁRG over the same logic. Take any Łukasiewicz formula $\phi(p_1, \ldots, p_m)$ and define the following ŁRG

$$\Gamma = \langle \textsc{Ag}, \textsc{Var}, \textsc{Const}, \textsc{Val}, \textsc{Task}, \textsc{Payoff} \rangle,$$

where

1. $\textsc{Ag} = \{A_1\}$ is a set including only one agent A_1.
2. $\textsc{Var} = \{\{p_1, \ldots, p_m\}\}$ includes the set of variables of A_1.
3. $\textsc{Const} = \{\top\}$, so player A_1 has no constraints.
4. $\textsc{Val} = \{\textsc{Val}_1\}$ contains the set $\textsc{Val}_1$ of all valuations from $\{p_1, \ldots, p_m\}$ into L.
5. $\textsc{Task} = \{\tau_1, \ldots, \tau_m\}$ is a set of Łukasiewicz formulae
$$\tau_j(p_j) := p_j.$$
6. $\textsc{Payoff} = \{\chi_1\}$ is a set including the Łukasiewicz formula
$$\chi_i(\tau_1, \ldots, \tau_m) := \phi(p_1, \ldots, p_n).$$

It is obvious by construction that the one and only (non-empty) coalition C is satisfiable in Γ if and only if so is $\phi(p_1, \ldots, p_n)$. □

We now investigate the complexity of checking whether a rational cooperation structure is globally successful, and whether any coalition can be made into a globally successful cooperation structure.

Proposition 39. *In a Łukasiewicz resource game* Γ*, checking whether a rational cooperation structure* $\lambda = \langle C, \mathbf{s}_C \rangle$ *is globally successful is in* co-NP*.*

Proof. Given a cooperation structure $\lambda = \langle C, \mathbf{s}_C \rangle$ with $\mathbf{s}_C \in \mathbb{Q}^{m \cdot |C|}$, λ is globally successful if and only if, for all possible allocations $\mathbf{s}_{-C}$ and all players $i \in C$,

$$f_{\chi_i}(\mathbf{s}_C, \mathbf{s}_{-C}) = 1.$$

Checking the complement of this problem is equivalent to checking whether the set

$$Y_i = \{\bar{y}_{-C} \mid f_{\chi_i}(\mathbf{s}_C, \bar{y}_{-C}) < 1\}$$

is non-empty for some $i \in C$. This is equivalent to checking the validity of the existential Łukasiewicz sentence

$$\exists y_{-C} \bigsqcup_{i \in C} (f_{\chi_i}(\mathbf{s}_C, \bar{y}_{-C}) < 1)\,,$$

which can be done in non-deterministic polynomial time both for finite and infinite games [24]. □

Proposition 40. *For a Łukasiewicz resource game* Γ*, checking if for a coalition* C *there exists an allocation* $\mathbf{s}_C$ *so that* $\lambda = \langle C, \mathbf{s}_C \rangle$ *is a globally successful cooperation is in* PSPACE *if* Γ *is finite, and it is in* 2-EXPTIME *if* Γ *is infinite.*

Proof. Checking if for a coalition C there exists an allocation $\mathbf{s}_C$ so that $\lambda = \langle C, \mathbf{s}_C \rangle$ is a globally successful cooperation is equivalent to checking if the following sentence holds over $\mathbb{L}$ or $\mathbb{L}_k$

$$\exists \bar{x}_C \forall \bar{y}_{-C} \bigsqcap_{i \in C} f_{\chi_i}(\bar{x}_C, \bar{y}_{-C}) = 1.$$

As shown in [24], if Γ is finite, the algorithm to check the validity of the above formula requires polynomial space, while it requires double exponential time if the game is infinite. □

The next proposition studies the complexity of checking if a rational coalition structure belongs to the core.

Proposition 41. *For a finite Łukasiewicz resource game* Γ*, checking if a rational coalition structure* σ *belongs to* ***core***(Γ) *is in* PSPACE. *For an infinite Łukasiewicz resource game* Γ*, checking if a rational coalition structure* σ *belongs to* ***core***(Γ) *is in* co-2-NEXPTIME.

Proof. Suppose Γ is a finite ŁRG and let $\sigma = \{\lambda_1, \ldots, \lambda_w\}$ be a coalition structure with cooperation structures $\lambda_j = \langle C_j, \mathbf{s}_{C_j} \rangle$ where each $\mathbf{s}_{C_j}$ is an allocation vector with values from L_k. Guess a cooperation structure $\lambda = \langle C, \mathbf{r}_C \rangle$, where $\mathbf{r}_C$ is an allocation vector with values from L_k. For every $i \in C$:

1. Check if

   $$\forall \bar{x}_{-C_{\sigma(i)}} \left(f_{\chi_i}(\mathbf{s}_{C_{\sigma(i)}}, \bar{x}_{-C_{\sigma(i)}}) = 1\right) \sqcap \forall \bar{y}_{-C}\ (f_{\chi_i}(\mathbf{r}_C, \bar{y}_{-C}) = 1)$$

   holds over $\mathbb{L}_k$ and check if

   $$cost_i(\pi_i(\mathbf{r}_C)) < cost_i(\pi_i(\mathbf{s}_{C_{\sigma(i)}}));$$

   else

2. Check if

   $$\exists \bar{x}_{-C_{\sigma(i)}} \left(f_{\chi_i}(\mathbf{s}_{C_{\sigma(i)}}, \bar{x}_{-C_{\sigma(i)}}) < 1\right) \sqcap \exists \bar{y}_{-C}\ (f_{\chi_i}(\mathbf{r}_C, \bar{y}_{-C}) < 1)$$

   holds over $\mathbb{L}_k$ and check if

   $$cost_i(\pi_i(\mathbf{r}_C)) < cost_i(\pi_i(\mathbf{s}_{C_{\sigma(i)}}));$$

   else

3. Check if
$$\exists \bar{x}_{-C_{\sigma(i)}} \ \left(f_{\chi_i}(\mathbf{s}_{C_{\sigma(i)}}, \bar{x}_{-C_{\sigma(i)}}) < 1\right) \sqcap \forall \bar{y}_{-C} \ (f_{\chi_i}(\mathbf{r}_C, \bar{y}_{-C}) = 1)$$
holds over $\mathbb{L}_k$.

It is easy to see that λ blocks σ if for all players in C any of the above cases is true. In fact:

1. if the first case holds then $\lambda_{\sigma(i)}, \lambda \in \mathit{succ}(i)$ and the cost for i in λ is strictly less than the cost in $\lambda_{\sigma(i)}$;
2. if the second case holds then $\lambda_{\sigma(i)}, \lambda \notin \mathit{succ}(i)$ and the cost for i in λ is strictly less than the cost in $\lambda_{\sigma(i)}$;
3. finally, if the third case holds then $\lambda_{\sigma(i)} \notin \mathit{succ}(i)$ and $\lambda \in \mathit{succ}(i)$.

Since checking the validity of a quantified sentence in $\mathcal{L}_k$ over $\mathbb{L}_k$ is in PSPACE, checking whether σ belongs to $\mathbf{core}(\Gamma)$ is in co-NPSPACE. Since co-PSPACE is equivalent to PSPACE and, by Savitch's Theorem NPSPACE=PSPACE (see [34]), we have that checking whether σ belongs to $\mathbf{core}(\Gamma)$ is in PSPACE.

Let Γ be any infinite ŁRG and let $\sigma = \{\lambda_1, \ldots, \lambda_w\}$ be a coalition structure with rational cooperation structures $\lambda_j = \langle C_j, \mathbf{s}_{C_j}\rangle$. Guess a rational cooperation structure $\lambda = \langle C, \mathbf{r}_C\rangle$. The proof now proceeds exactly as in the finite case. The result then follows from the fact that checking the validity of an $\mathcal{L}$-sentence over $\mathbb{L}$ is in 2-EXPTIME. □

To conclude, we study the complexity of checking whether a ŁRG admits a non-empty core.

Proposition 42. *For a Łukasiewicz resource game Γ, checking $\mathbf{core}(\Gamma) \neq \emptyset$ is in* 4-EXPTIME *if Γ is finite,* 3-EXPTIME *if Γ is infinite.*

Proof. Given an infinite ŁRG Γ, let $\exists\Phi_{\mathbf{core}}$ be the existential closure of the formula $\Phi_{\mathbf{core}}$ from Theorem 21, i.e.:

$$\exists \bar{x}_1 \ \ldots \exists \ \bar{x}_n \bigsqcap_{C \in \mathcal{C}(\Gamma)} \left(\bigsqcup_{i \in C} \ \Big(\big((f^*_{\chi_i}(\bar{x}_1, \ldots, \bar{x}_n) = 1) \sqcap \big(\exists \bar{y}_C \forall \bar{y}_{-C} \ (f^*_{\chi_i}(\bar{y}_C, \bar{y}_{-C}) = 1) \Rightarrow (\textstyle\sum \bar{x}_i \leq \sum \bar{y}_i)\big)\big) \sqcup \right.$$
$$\left. \big((f^*_{\chi_i}(\bar{x}_1, \ldots, \bar{x}_n) < 1\big) \sqcap (\sim\exists \bar{z}_C \forall \bar{z}_{-C} \ (f^*_{\chi_i}(\bar{z}_C, \bar{z}_{-C}) = 1) \sqcap (\textstyle\sum \bar{x}_i = 0))) \Big) \right).$$

Γ has a non-empty core if and only if $\exists\Phi_{\mathbf{core}}$ is valid over $\mathbb{R}$.

Now, given Γ, the length of $\exists\Phi_{\mathbf{core}}$ is exponential in the number of players, since it is built by taking into account all possible coalitions. As shown in [17], deciding the validity of an arbitrary quantified formula of length n in the theory of the ordered group of real numbers requires at most deterministic time $2^{2^{pn}}$ for some fixed constant $p > 0$. Consequently, checking whether the infinite game Γ has a non-empty core requires at most 3-EXPTIME.

The result for finite games follows by a similar argument and the fact that deciding the validity of an arbitrary quantified formula of length n in the theory of Presburger Arithmetics requires at most deterministic time $2^{2^{2^{pn}}}$, for some fixed constant $p > 0$ [31]. □

7. Final Remarks

In this work, we have introduced a symbolic approach to the compact representation of cooperative games that is a generalisation of the coalitional resource model of Wooldridge and Dunne [36, 15]. Our model makes use of Łukasiewicz logics, which make it possible to specify the class of piecewise linear polynomial functions with integer and rational coefficients on $[0,1]^n$ and their finite-valued restrictions. Łukasiewicz formulae are used to encode the outcome of the allocation of resources to tasks by players, as well as their constraints on such allocations, and overall payoffs. In this framework we have defined an appropriate notion of core and investigated some of its logical and computational properties.

Comparatively little other research has considered the use of logic in cooperative games. Ågotnes *et al* [1] developed two modal logics for reasoning about non-transferable utility games. For example, in the first of their formalisms, they used operators inspired by Alternating-time Temporal Logic (ATL) [3], writing for example $\langle C\rangle\omega$ to mean that the coalition C can cooperate to achieve outcome ω; they augmented these operators with preference operators $\omega \succeq_i \omega'$ to mean that agent i prefers outcome ω at least as much as ω'. For this logic they obtained completeness and complexity results, and showed how some standard theorems of cooperative game theory could be proved within the logic. A key difference with our work is that, by using Łukasiewicz logic, we can succinctly express rich utility functions. Closer to our work, Dunne *et al* [14] considered cooperative variations of Boolean games. The key difference with our work was the use of classical (two valued) logics in the specification of player's goals: as we have already argued, using Łukasiewicz logics permits much richer utility functions to be specified. Finally, Ieong and Shoham's marginal contribution net scheme makes use of logical formulae within weighted rules to express characteristic functions for games [22]. Again, the focus is on classical logic, and in marginal contribution nets, although logic is used within the representation, the weighted rules are not themselves expressions of a logical object language.

Several questions suggest themselves for future work. For example, in investigating the core, we have studied whether a coalition structure is stable against defection. This however does not address the issue of how a coalition is formed. In our future work then, we plan to study negotiation protocols for finding coalition structures (see [15]). In addition, we plan to consider whether we can adapt techniques for theorem proving with Łukasiewicz logics to solving Łukasiewicz resource games.

Acknowledgments

Marchioni and Wooldridge acknowledge support from the ERC Advanced Investigator Grant "RACE" (291528) and the EPSRC Standard Fellowship "Combining Qualitative and Quantitative Reasoning for Logic-based Games" (EP/M009130/1).

[1] T. Ågotnes, W. van der Hoek, and M. Wooldridge. Reasoning about coalitional games. *Artificial Intelligence*, 173(1):45–79, 2009.

[2] S. Aguzzoli, S. Bova, B. Gerla. 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, 2011.

[3] R. Alur, T. A. Henzinger, and O. Kupferman. Alternating-time temporal logic. *Journal of the ACM*, 49(5):672–713, September 2002.

[4] S. Arora, B. Barak. *Computational Complexity. A Modern Approach.* Cambridge University Press, Cambridge UK, 2009.

[5] M. Baaz, H. Veith. Quantifier elimination in fuzzy logic, In *Computer Science Logic*, Lecture Notes in Computer Science, Springer, Berlin Heidelberg, 399–414, 1999.

[6] Y. Bachrach and J. S. Rosenschein. Computing the Banzhaf power index in network flow games. In *Proceedings of the Sixth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2007)*, pages 335–341, Honolulu, Hawaii, 2007.

[7] J. van Benthem. *Logic in Games.* MIT Press, 2014.

[8] E. Bonzon, M.-C. Lagasquie-Schiex, J. Lang, and B. Zanuttini. Boolean games revisited. In *Proceedings of the Seventeenth European Conference on Artificial Intelligence (ECAI-2006)*, pages 265–269, Riva del Garda, Italy, 2006.

[9] A. R. Bradley, Z. Manna. *The Calculus of Computation.* Springer-Verlag Berlin Heidelberg, 2007.

[10] N. Bulling, V. Goranko. How to Be Both Rich and Happy: Combining Quantitative and Qualitative Strategic Reasoning about Multi-Player Games. In *Proc. Strategic Reasoning 2013*, Rome, April 2013

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

[12] G. Chalkiadakis, E. Elkind, and M. Wooldridge. *Computational Aspects of Cooperative Game Theory. Morgan & Claypool*, 2011.

[13] A. Di Nola, I. Leustean. Łukasiewicz Logic and MV-Algebras. In *Handbook of Mathematical Fuzzy Logic*, Volume II, P. Cintula, P. Hájek, and C. Noguera (Eds.), College Publications, 2011.

[14] P. E. Dunne, S. Kraus, W. van der Hoek, and M. Wooldridge. Cooperative Boolean games. In *Proceedings of the Seventh International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2008)*, pages 1015–1022, Estoril, Portugal, 2008.

[15] P. E. Dunne, S. Kraus, E. Manisterski, and M. Wooldridge. Solving coalitional resource games. *Artificial Intelligence*, 174: 20–50, 2010.

[16] F. Esteva, L. Godo, and E. Marchioni E. 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, 2011.

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

[18] B. Gerla. *Many-valued Logics of Continuous t-norms and Their Functional Representation.* Ph. D. thesis, University of Milan, 2001.

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

[20] P. Harrenstein, W. van der Hoek, J.-J.Ch. Meyer, and C. Witteveen. Boolean games. In *Proceedings of the Eighth Conference on Theoretical Aspects of Rationality and Knowledge (TARK VIII)*, pages 287–298, Siena, Italy, 2001.

[21] W. Hodges. *Model theory*, volume 42 of Encyclopaedia of Mathematics and its Applications. Cambridge University Press, Cambridge, 1993.

[22] S. Ieong and Y. Shoham. Marginal contribution nets: A compact representation scheme for coalitional games. In *Proceedings of the Sixth ACM Conference on Electronic Commerce (EC'05)*, Vancouver, Canada, 2005.

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

[24] E. Marchioni, M. Wooldridge. Łukasiewicz Games: A Logic-based Approach to Quantitative Strategic Interactions. *ACM Transactions on Computational Logic*, 16(4): Article 33, 2015.

[25] D. Marker. *Model theory. An Introduction.* Vol. 217 of *Graduate Texts in Mathematics.* Springer-Verlag, New York, 2002.

[26] M. Maschler, E. Solan, S. Zamir. *Game Theory.* Cambridge University Press, 2013.

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

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

[29] D. Mundici. *Advanced Lukasiewicz Calculus and MV-algebras.* Trends in Logic, Vol. 35, Springer, 2011.

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

[31] D.C. Oppen. A $2^{2^{2^{pn}}}$ upper bound on the complexity of Presburger arithmetic. *Journal of Computer and System Sciences*, 16(3): 323–332, 1978.

[32] M.J. Osborne, A. Rubinstein. *A Course in Game Theory.* MIT Press, 1994.

[33] G. C. Rota. The Number of Partitions of a Set. *The American Mathematical Monthly*, 71, 498–504, 1964.

[34] M. Sipser. *Introduction to the Theory of Computation.* Cengage Learning, 2012.

[35] M. Wooldridge, P.E. Dunne. On the computational complexity of qualitative coalitional games. *Artificial Intelligence*, 158(1): 27–73, 2004.

[36] M. Wooldridge, P.E. Dunne. On the computational complexity of coalitional resource games. *Artificial Intelligence*. 170(10): 853–871, 2006.