Artificial Intelligence 173 (2009) 45–79

Contents lists available at ScienceDirect

Artificial Intelligence

www.elsevier.com/locate/artint

Reasoning about coalitional games

Thomas Ågotnes [a,*], Wiebe van der Hoek [b], Michael Wooldridge [b]

[a] *Department of Computer Engineering, Bergen University College, Norway*
[b] *Department of Computer Science, University of Liverpool, UK*

ARTICLE INFO

Article history:
Received 15 September 2007
Received in revised form 6 August 2008
Accepted 10 August 2008
Available online 23 August 2008

Keywords:
Multi-agent systems
Knowledge representation
Coalitional games
Modal logic

ABSTRACT

We develop, investigate, and compare two logic-based knowledge representation formalisms for reasoning about coalitional games. The main constructs of *Coalitional Game Logic* (CGL) are expressions for representing the ability of coalitions, which may be combined with expressions for representing the preferences that agents have over outcomes. *Modal Coalitional Game Logic* (MCGL) is a normal modal logic, in which the main construct is a modality for expressing the preferences of groups of agents. For both frameworks, we give complete axiomatisations, and show how they can be used to characterise solution concepts for coalitional games. We show that, while CGL is more expressive than MCGL, the former can only be used to reason about coalitional games with finitely many outcomes, while MCGL can be used to reason also about games with infinitely many outcomes, and is in addition more succinct. We characterise the computational complexity of satisfiability for CGL, and give a tableaux-based decision procedure.

1. Introduction

Coalitional games are games in which agents can potentially benefit by cooperating [24, pp. 257–298]. Such games provide a natural and compelling model through which to understand cooperative action, and have been widely studied in the context of both natural and artificial multi-agent systems. In the game theory literature, two basic questions are asked in the context of coalitional games: *Which coalitions will form*? and *How will the benefits of cooperation be shared within a coalition*? With respect to the first question, solution concepts such as *the core* have been proposed, which try to capture the idea of rational participation in a coalition [24, p. 258]. With respect to the second question, solution concepts such as the Shapley value have been proposed, which attempt to define a "fair" distribution of the benefits of cooperation to agents within a coalition [24, p. 291].

In the context of multi-agent systems and artificial intelligence, the use of coalitional game models and cooperative solution concepts raises a number of important issues. Perhaps the most fundamental issues are those of *representing* coalitional games, and *reasoning* with such representations. A number of researchers have developed models for coalitional games, and, given such models, have investigated the complexity of associated solution concepts, e.g., [10,12,13,20,23,40,41]. However, very little work has considered the kinds of logical, declarative representation schemes that are commonly used in the knowledge representation community [38]. There is good reason to suppose that such logic-based representations will be of value in reasoning about coalition games. For example, they can be used together with tools and techniques developed in AI and computer science [39]:

- As query languages, for expressing properties φ of coalitional games. Checking whether a game has property φ reduces to the *model checking* problem.

* Corresponding author.
E-mail addresses: tag@hib.no (T. Ågotnes), wiebe@csc.liv.ac.uk (W. van der Hoek), mjw@liv.ac.uk (M. Wooldridge).

0004-3702/$ – see front matter
doi:10.1016/j.artint.2008.08.004

- For directly reasoning about coalitional games via *theorem proving*.
- For expressing desirable properties φ of a coalitional game we want to *synthesise*. This corresponds to a constructive proof of satisfiability for φ.

Moreover, logical representation schemes are frequently succinct, compared to the alternatives. Logical representations of coalitional games might also be of value in game theory itself, as they open the door for automated reasoning tools such as theorem provers.

Our aim in this paper is thus to develop and study logic-based knowledge representation formalisms for coalitional games (more precisely, coalitional games without transferable payoffs [24, p. 268]). We develop two logical languages that are interpreted directly as statements of such games. We study the axiomatisation and computational complexity of these logical languages, and demonstrate how they can be used to characterise and reason about coalitional games:

- First, we develop a *Coalitional Game Logic* (CGL). Syntactically, CGL contains modal cooperation expressions of the form $\langle C\rangle\varphi$, meaning that coalition C can achieve an outcome satisfying φ. In addition, CGL includes operators that make it possible to explicitly represent an agent's preferences over outcomes. The inclusion of an explicit mechanism for representing preferences makes CGL very different to cooperation logics such as ATL [3] and Coalition Logic [27], which otherwise might, at first sight, seem to be similar to CGL. However, the differences go much deeper than just introducing a way of representing preferences: we show in Section 5 that CGL is fundamentally incomparable to these logics. Note that we interpret formulae of CGL directly with respect to coalitional games without transferable payoff, thereby establishing an explicit link between formulae of the logic and properties of coalitional games.
- Second, we develop a *Modal Coalitional Game Logic* (MCGL), a normal modal logic interpreted directly in coalitional games by using the preference relations in coalitional games as modal accessibility relations.

Both logics can be used to characterise and reason about many important properties of coalitional games, such as non-emptiness of the core. They differ, however, in that CGL can only express such properties under the assumption that the possible outcomes of the games are finite, while MCGL does not have this restriction. On the other hand, if we make the finiteness assumption, CGL is more expressive than MCGL, while the latter can often express interesting properties such as non-emptiness of the core much more succinctly.

The remainder of this article is organised as follows. In the next section, we discuss related work, and introduce the basic mathematical framework of coalitional games and solution concepts for such games. CGL is introduced in Section 3. Following a presentation of the syntax and semantics of the logic, we give a number of technical results relating to it, as follows. First, we prove that the logic is expressively complete with respect to finite coalitional games without transferable payoff, in the sense that for any two different finite coalitional games, there exists a formula of CGL that will be true in one game and false in the other. We then give an axiomatisation of CGL, and show that it is sound and complete with respect to finite coalitional games. With respect to model checking and satisfiability, we show that while model checking for the logic is tractable, the satisfiability problem for CGL is NP-complete. We present a tableau-based decision procedure for the logic, and to illustrate the use of the logic, we show how to axiomatically characterise a number of well-known solution concepts for coalitional games, including, for example, non-emptiness of the core of finite coalitional games. Finally, we give some examples of formal deductions in the logic. In Section 4 we introduce MCGL. After presenting the syntax and semantics, we present an axiomatisation, and prove that it is sound and complete with respect to all coalitional games. We illustrate the logic by expressing properties such as non-emptiness of the core of general (not necessarily finite) coalitional games, and by formally deriving some properties of coalitional games as theorems within the proof system. Finally, in Section 5, we compare CGL and MCGL, first to each other, and then to Coalition Logic, and we conclude in Section 6.

2. Background

Originally developed within the game theory community [4,15,22], models of cooperative or coalitional games entered multi-agent systems and artificial intelligence research in the 1990s. Initial research focused on approaches to coalition formation. For example, Shehory and Kraus developed algorithms for coalition formation, in which agents were modelled as having different capabilities, and were assumed to benevolently desire some overall task to be accomplished, where this task had some complex (plan-like) structure [30–32]. Sandholm considered the closely-related problem of coalition structure generation, i.e., the problem of partitioning an overall set of agents into mutually disjoint coalitions, so that social welfare (the sum of individual coalition values) is maximised [29].

More recently, the issue of *representing* coalitional games, and the complexity of computing with these representations, has received attention. The issue of representation—which is of course central to the field of artificial intelligence—is of particular importance in the context of coalitional games, as the obvious representations for them have completely unrealistic space requirements (see, e.g., the discussion in [40, pp. 34–41]). Some effort has therefore been devoted to developing *succinct* representations for coalitional games. There are roughly two main lines of attack with respect to the problem of representing coalitional games, as follows.

- First, one can try to find some representation scheme that is guaranteed to be succinct (i.e., will lead to representations that are of size polynomial in the number of agents), but which are *incomplete*, by which we mean that there will be some coalitional games that simply cannot be captured within the representation. Such a representation will be of interest if it can represent games in the domain of interest to us. An example is the "induced subgraph" representation of Deng and Papadimitriou [12].
- Second, one can try to find a representation that is complete but not guaranteed to be succinct. That is, although the representation might be of exponential size in the worst case, it will be useful if it can be used to represent succinctly games in the domain of interest to us. An example is the "marginal contribution nets" representation of Ieong and Shoham [20].

Given a specific representation scheme, it is possible to ask concrete questions about, for example, the complexity of computing solution concepts. Deng and Papadimitriou were perhaps the first to investigate the complexity of cooperative solution concepts, for several representations of coalitional games [12]; this led to work by Conitzer and Sandholm [8–10], Ieong and Shoham [20], Wooldridge and Dunne [40,41], and Elkind et al. [13]. A survey of related results in this area is [5].

Although logic-based approaches to knowledge representation are widely used in artificial intelligence, there has been little work on logic-based approaches to representing coalitional games.[1] There are many arguments in favour of a logic-based approach to knowledge representation for coalitional games.[2] For example, such languages can be used as expressive, general, semantically well-defined query languages for model checkers and the like [28,39]. Alternatively, and adopting a more traditional AI approach, such formalisms can be used to explicitly model an agent's environment, and the agent can then use theorem proving to make decisions about what action to perform (e.g., whether to join a particular coalition). And finally, of course, logical representations are very often succinct, and so a logic of coalitional games might be of interest as a succinct representation scheme for coalitional games. Perhaps surprisingly, little work has attempted to use logic as a succinct representation scheme for coalitional games. Ieong and Shoham's *marginal contribution nets* use a logic-based rule notation to represent the characteristic function of a coalitional game [20]. Wooldridge and Dunne proposed a propositional logic representation for their "Qualitative Coalitional Games", explicitly arguing that this representation would be more succinct in many cases of interest than an explicit representation [40]. However, in both of these cases, logic was used as part of the underlying coalitional game model, rather than as a tool for representing properties of coalitional games overall.

Given these concerns, the existing formalisms most closely related to our interests are *Alternating-time Temporal Logic* (ATL) [3] and Coalition Logic (CL) [27], which can be regarded as a strict fragment of ATL. In both of these formalisms, the main construct is an expression of the form $\langle C\rangle\varphi$, with the intended meaning that coalition C has the ability to achieve φ. Such coalition logics have proved to have many important applications, for example in the specification and verification of social choice mechanisms [26], and for knowledge representation in multi-agent systems where one seeks to represent the strategic structure of multi-agent environments [34]. The semantic structures underpinning coalition logic are *effectivity functions*, which have been studied in the social choice literature [1]. But, while Coalition Logic, ATL, and their many variants have proved to be intuitive, powerful, and practical tools for understanding the properties of game-like multi-agent systems [39], they have several limitations for reasoning about coalitional games. In particular, they do not provide any mechanism for representing and reasoning about the preferences of agents, which is of course a fundamental requirement for modelling rational action. (We investigate the potential use of Coalition Logic for coalitional games in detail in Section 5.)

2.1. Coalitional games

We here provide a summary of the relevant necessary definitions and concepts from cooperative game theory, but we refer the interested reader to, e.g., [24] for more detail and discussion. A *coalitional game* (without transferable payoff) is an $(m+3)$-tuple [24, p. 268]:

$$\Gamma = \langle N, \Omega, V, \sqsupseteq_1, \ldots, \sqsupseteq_m\rangle$$

where:

- $N = \{1, \ldots, m\}$ is a non-empty set of *players* (or *agents*);
- Ω is a non-empty set of *outcomes*;
- $V : (2^N \setminus \emptyset) \to 2^\Omega$ is the *characteristic function* of Γ, which for every non-empty coalition C defines the choices $V(C)$ available to C, so that $\omega \in V(C)$ means C can choose outcome ω; and
- $\sqsupseteq_i \subseteq \Omega \times \Omega$ is a complete, reflexive, and transitive *preference relation*, for each agent $i \in N$.

We let $\omega \sqsupset_i \omega'$ denote the fact that ω is strictly preferred over ω' by agent i (i.e., $\omega \sqsupseteq_i \omega'$ but not $\omega' \sqsupseteq_i \omega$). We sometimes refer to the set N, of all agents, as the *grand* coalition.

[1] In contrast, a number of researchers have investigated logics for representing and reasoning about *non-cooperative* games: see, e.g., [17] for one example of such work, and [37] for a detailed survey.

[2] We will not restate all the well-known arguments in favour of logic for knowledge representation—see, e.g., [38] for detailed discussions.

The definition above is a very *general* definition of coalitional games. It gives no indication of what the outcomes Ω are, or where the preference relations $\sqsupseteq_i$ come from. There are several points to note here:

- The first is that readers familiar with temporal logic [14] or state transition systems may be tempted to interpret outcomes Ω as possible system states; and readers familiar with Alternating-time Temporal Logic [3] or Coalition Logic [27] may be tempted to interpret V as an effectivity function [1]. However, this is *not* the intended interpretation. Indeed, we show in Section 5.2 that there is no direct mapping between the state-based models of [27] and coalitional games by interpreting outcomes Ω as states and V as an effectivity function. For example, it is perfectly consistent in a coalitional game that, at the same time, disjoint coalitions have the abilities to choose *different* outcomes.
- The second point to note is that a more common *but less general* definition of coalitional game is that of a coalitional game *with transferable payoff*, formalised as a pair $\langle N, v\rangle$ consisting of the set of players N together with a function $v : 2^N \to \mathbb{R}$ associating a real number $v(C)$ with every group of players C. The intended interpretation of a game $\langle N, v\rangle$ is that $v(C)$ represents the payoff, or utility, that C could obtain should they choose to cooperate. It is easy to see that coalitional games without transferable utility are more general than coalitional games with transferable utility: we simply interpret $V(C)$ as being the set of possible distributions of payoff $v(C)$ to members of C.[3]

Henceforth, we use the term "coalitional game" to refer to the more general concept of a coalitional game without transferable utility, as defined above. A *finite* coalitional game is a coalitional game with a finite set of outcomes.

Example 1 *(Dinner Game[4]).* Three agents must decide to go out for either Indian (I), Chinese (C) or Japanese (J) food. The majority will go where the majority decides: if all three agents decide to go for Indian they will do that; if two of the agents wants to go for Chinese the two agents will do that and the third agent will stay at home. To simplify the presentation in this example, we will define the preferences of agents by way of *utilities*: for every agent i we assume a utility function $u_i : \Omega \to \mathbb{R}$, so that $u_i(\omega)$ denotes the utility agent i gets from outcome ω. These utility functions then induce preference relations in the obvious way.

The utilities of the agents with respect to the outcome I, C, and J are as follows:

- $u_1(I) = u_2(C) = u_3(J) = 4$,
- $u_1(C) = u_2(J) = u_3(I) = 2$,
- $u_1(J) = u_2(I) = u_3(C) = 0$,
- each agent gets an additional unit of utility for each other agent that joins him,
- an agent who stays at home has a utility of 0.

The situation can be modelled by the following coalitional game:

- $N = \{1, 2, 3\}$,
- $\Omega = \{I_{12}, I_{13}, I_{23}, C_{12}, C_{13}, C_{23}, J_{12}, J_{13}, J_{23}, I_{123}, C_{123}, J_{123}\}$. I_{12} is the outcome where the majority $1, 2$ goes out for Indian; C_{123} where all agents go for Chinese, etc.,
- ◦ $V(1) = V(2) = V(3) = \emptyset$,
  ◦ $V(1, 2) = \{I_{12}, C_{12}, J_{12}\}$,
  ◦ $V(1, 3) = \{I_{13}, C_{13}, J_{13}\}$,
  ◦ $V(2, 3) = \{I_{23}, C_{23}, J_{23}\}$,
  ◦ $V(1, 2, 3) = \{I_{123}, C_{123}, J_{123}\}$
- $I_{123} \sqsupset_1 I_{12} \sqsupseteq_1 I_{13} \sqsupset_1 C_{123} \sqsupset_1 C_{12} \sqsupseteq_1 C_{13} \sqsupset_1 J_{123} \sqsupset_1 J_{12} \sqsupseteq_1 J_{13} \sqsupset_1 I_{23} \sqsupseteq_1 C_{23} \sqsupseteq_1 J_{23}$. Similarly for agents 2 and 3.

The following is a variant of the three-player majority game with an infinite set of outcomes.

Example 2 *(Cake Game).* Three agents must decide how to divide a cake amongst them. Any majority can decide a division of the cake amongst themselves. We assume that each agent only cares about the amount of cake she gets: the more cake the better. The situation can be modelled by the following coalitional game:

- $N = \{1, 2, 3\}$,
- $\Omega = \mathbb{R}^3$,
- $V(1) = V(2) = V(3) = \emptyset$,
- $V(C) = \{\langle x_1, x_2, x_3\rangle\colon \sum_{i\in C} x_i = 1, \forall_{i\in N\setminus C} x_i = 0\}$,
- $\langle x_1, x_2, x_3\rangle \sqsupseteq_i \langle y_1, y_2, y_3\rangle$ iff $x_i \geqslant y_i$.

[3] Of course, Ω will be infinite in this case.
[4] Based on an example given by Vincent Conitzer.

2.2. Solution concepts for coalitional games

Game theory defines and studies different concepts related to coalitional games, in particular *solution concepts* such as, e.g., the core. It is desirable that formal logics for coalitional games, such as those introduced in the next two sections, are able to express and reason about such concepts. We discuss here three solution concepts from the theory of coalitional games, viz. *the core* [15], *stable sets* [22] and *the bargaining set* [4]. Our formulation of these solution concepts follows that of [24]; there the two latter solution concepts are however defined only for games with real numbered payoffs and transferable utility and below we translate the definitions to the more general games with preference relations over general outcomes and non-transferable utility.

A *C-feasible outcome* is an outcome which can be chosen by the coalition C; thus ω is C-feasible if $\omega \in V(C)$. A *feasible outcome* is a N-feasible outcome, where N is the grand coalition. Thus, for example, the set $\{I_{123}, C_{123}, J_{123}\}$ represents the feasible outcomes for the Dinner Game (Example 1).

The first solution concept we consider is the core: the core of a game is a (possibly empty) set of outcomes.

Definition 1 *(Core).* The core of a coalitional game is the set of feasible outcomes ω for which there exists no coalition C and C-feasible outcome ω' such that $\omega' \sqsupset_i \omega$ for all $i \in C$.

An important property of a coalitional game is whether the core is empty or not. In the Dinner Game (Example 1) the core is empty: for example, if the three agents choose to go for Japanese food together, agents 1 and 2 would benefit from going for Chinese on their own instead. Similarly, the core of the Cake Game (Example 2) is empty. If the core of a game is empty, then the grand coalition is unstable, since by definition it means that some coalition could benefit by defecting from the grand coalition. Thus, the question "is the grand coalition stable" reduces to the question "is the core non-empty". Intuitively, if the core is non-empty, then we can think of the members of the core as being candidates for an outcome that the grand coalition might choose.[5]

Like the core, a *stable set* is a set of outcomes. A coalitional game may have several stable sets, but need not necessarily have any. We characterise stable sets in terms of *imputations* and *objections*. Because we will discuss a different objection concept in the context of the bargaining set below, we will here use the term *s-objection* (stable set objection) to avoid confusion. An imputation is a feasible outcome that for each agent i is as least as good as any outcome the singleton coalition $\{i\}$ can choose on his own. For example, in the Dinner Game (Example 1) the set of imputations is the set of all feasible outcomes $\{I_{123}, C_{123}, J_{123}\}$. Also for the Cake Game (Example 2) all feasible outcomes are imputations. An imputation ω is a C-s-objection to an imputation ω' if every agent in C strictly prefers ω over ω' and the coalition C can choose an outcome which for every agent in C is as least as good as ω. The imputation ω is an s-objection to ω' if ω is a C-s-objection to ω' for some coalition C.

Definition 2 *(Stable Set).* A set of imputations Y is a stable set if it satisfies:

Internal stability If $\omega \in Y$, there is no s-objection to ω in Y.
External stability If ω is an imputation and $\omega \notin Y$, there is an s-objection to ω in Y.

In the Dinner Game (Example 1), the set of all imputations $\{I_{123}, C_{123}, J_{123}\}$ is a stable set. An example of a stable set for the Cake Game (Example 2) is $\{(\frac{1}{2}, \frac{1}{2}, 0), (\frac{1}{2}, 0, \frac{1}{2}), (0, \frac{1}{2}, \frac{1}{2})\}$.

Finally, we focus on the notion of a bargaining set of a coalitional game which is, like a stable set, a set of imputations, but, unlike a stable set, is unique and always exists. The bargaining set of a game can be defined in terms of *objections* and *counterobjections*, but the former concept is not the same as in the definition of stable sets. Let ω be an imputation:

b-objection: A pair consisting of a coalition C and a C-feasible outcome ω' is a b-objection of an agent $i \in C$ against an agent $j \notin C$ to ω if every agent in C strictly prefers ω' over ω.

b-counterobjection: A pair consisting of a coalition D and a D-feasible outcome v is a b-counterobjection to an b-objection (ω', C) of i against j to ω, if D includes j but not i, every agent in $D \setminus C$ thinks v is as least as good as ω and every agent in $D \cap C$ thinks v is as least as good as ω'.

We can now formulate the bargaining set.

Definition 3 *(Bargaining Set).* The bargaining set of a coalitional game is the set of all imputations ω such that there exists a b-counterobjection to every b-objection of any player i against any player j to ω.

[5] The theory of the core does not dictate which should be chosen, however, and the fact that there may be many elements in the core is frequently considered to be a limitation of the core as a solution concept. The Shapley value (which was developed primarily for coalitional games with transferable utility), has an advantage over the core in this respect, since it is unique [24, p. 293].

For example, the bargaining set of the Cake Game (Example 2) is $\{(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})\}$.

3. A logic for finite coalitional games

We now define *Coalitional Game Logic* (CGL): a formalism for reasoning about coalitional games, which can express all interesting properties of *finite* coalitional games. We first define the syntax and the semantics of the logic. We then study its expressive power; axiomatisation; computational properties; and give a decision procedure for satisfiability. Finally we use the logic to characterise the solution concepts discussed above.

3.1. CGL: Syntax and semantics

In the language of CGL, we are able to explicitly refer to agents in N, coalitions in 2^N, and outcomes in Ω. The language must therefore contain symbols standing for elements of each of these domains. We let Σ_N be a set of symbols for agents, Σ_C be a set of symbols for coalitions, and Σ_Ω be a set of symbols for outcomes, and let $\Sigma = \Sigma_N \cup \Sigma_C \cup \Sigma_\Omega$. Given these atomic building blocks, the language of CGL is defined in two parts.

- First, given a set of outcome symbols Σ_Ω, we have an *outcome language* $\mathcal{L}_o$, defined by the grammar φ_o, below, which expresses the properties of outcomes. The outcome symbols themselves are the main constructs of this language; a formula such as $\omega_1 \vee \omega_2$ (where $\omega_1, \omega_2 \in \Sigma_\Omega$) means that the outcome corresponds to either ω_1 or ω_2.
- Second, given a set of agent symbols Σ_N and a set of coalition symbols Σ_C, we have a *cooperation language* $\mathcal{L}_c$, for expressing the properties of coalitional cooperation, and the preferences that agents have over possible outcomes. This language is generated by the grammar φ_c below. $\mathcal{L}_c$ has two main constructs. First, $\omega_1 \succeq_i \omega_2$ (where $\omega_1, \omega_2 \in \Sigma_\Omega$, $i \in \Sigma_N$) expresses the fact that agent i either prefers outcome ω_1 over outcome ω_2, or is indifferent between the two. Second, $\langle C \rangle \varphi$ (where $C \in \Sigma_C$) says that C can choose an outcome in which the formula φ will be true.[6]

Formally, the grammar of CGL is defined as follows:

$$\varphi_o ::= \sigma_\omega \mid \neg\varphi_o \mid \varphi_o \vee \varphi_o$$

$$\varphi_c ::= (\sigma_\omega \succeq_{\sigma_i} \sigma_{\omega'}) \mid \langle \sigma_C \rangle \varphi_o \mid \neg\varphi_c \mid \varphi_c \vee \varphi_c$$

where $\sigma_i \in \Sigma_N$ is an agent symbol, $\sigma_C \in \Sigma_C$ is a coalition symbol, and $\sigma_\omega, \sigma_{\omega'} \in \Sigma_\Omega$ are outcome symbols.

To simplify our subsequent presentation, we exploit the direct correspondence between symbols for outcomes/agents/coalitions and the outcomes/agents/coalitions that appear in games. Let $\mathcal{C} = 2^N \setminus \emptyset$ denote the set of non-empty coalitions, henceforth simply called coalitions. In this paper, we will henceforth assume a one-to-one correspondence between Σ_Ω and Ω, between Σ_N and N and between Σ_C and the set of coalitions $\mathcal{C}$. So we assume that $\Sigma_\Omega = \{\sigma_\omega\colon \omega \in \Omega\}$, $\Sigma_N = \{\sigma_i\colon i \in N\}$ and $\Sigma_C = \{\sigma_C\colon C \in \mathcal{C}\}$. The languages are parameterised by the sets $\Sigma_\Omega, \Sigma_N, \Sigma_C$, i.e., as a consequence of the assumption, by some set N of agents and set Ω of outcomes. In the following, we assume that these two parameters—and thus the languages—are fixed.

The language of Coalitional Game Logic is $\mathcal{L}_c$; it expresses statements about coalitional games. An $\mathcal{L}_c$ formula γ is interpreted in a coalitional game Γ as follows, where $\Gamma \models \gamma$ means that γ is true in Γ. As mentioned it is assumed that the agents N and outcomes Ω are fixed, so when we talk about coalitional games in the context of CGL we implicitly mean coalitional games over agents N and outcomes Ω. Said in another way, the formulae of the language are defined as statements about games with a certain set of agents and a certain set of outcomes, and formally interpreted in such games.

First, we define the satisfaction of an $\mathcal{L}_o$ formula α in an outcome ω of a coalitional game Γ over N and Ω, written $\Gamma, \omega \models \alpha$:

$$\Gamma, \omega \models \sigma_{\omega'} \quad \text{iff} \quad \omega = \omega'$$

$$\Gamma, \omega \models \neg\varphi \quad \text{iff not} \quad \Gamma, \omega \models \varphi$$

$$\Gamma, \omega \models \varphi \vee \psi \quad \text{iff} \quad \Gamma, \omega \models \varphi \text{ or } \Gamma, \omega \models \psi$$

Satisfaction of $\gamma \in \mathcal{L}_c$ in Γ is then defined as follows:

$$\Gamma \models (\sigma_{\omega_1} \succeq_{\sigma_i} \sigma_{\omega_2}) \quad \text{iff} \quad (\omega_1 \sqsupseteq_i \omega_2)$$

$$\Gamma \models \langle \sigma_C \rangle \varphi \quad \text{iff} \quad \exists \omega \in V(C) \text{ such that } \Gamma, \omega \models \varphi$$

$$\Gamma \models \neg\varphi \quad \text{iff not} \quad \Gamma \models \varphi$$

$$\Gamma \models \varphi \vee \psi \quad \text{iff} \quad \Gamma \models \varphi \text{ or } \Gamma \models \psi$$

[6] This construct may seem syntactically similar to counterparts in ATL and Coalition Logic, but it here means something fundamentally different. We discuss this formally, and in detail, in Section 5.

When $\Gamma \models \varphi$ for every Γ over N and Ω, we write $\models \varphi$ and say that φ is *valid*.

To simplify the text that follows, we abuse notation somewhat, and write ω for both an outcome (a semantic construct) and the corresponding symbol σ_ω in the language (a syntactic construct). Similarly, we will write i instead of σ_i for agents in the language, and C instead of σ_C for coalitions. So, we will just write $\langle C\rangle\omega$ for $\langle\sigma_C\rangle\sigma_\omega$, although the reader should be aware of the distinction between our object language $\mathcal{L}_c$ and the objects that live in the semantics: outcomes, agents and their preferences, and coalitions.

We will use the usual derived propositional connectives: $\top$ is the usual constant for truth, $\varphi \wedge \psi$ stands for $\neg(\neg\varphi \vee \neg\psi)$, $\varphi \rightarrow \psi$ for $\neg\varphi \vee \psi$ and $\varphi \leftrightarrow \psi$ for $(\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi)$, as well as $\varphi \triangledown \psi$ for $(\varphi \vee \psi) \wedge \neg(\varphi \wedge \psi)$ (exclusive or) and $[C]\varphi$ for $\neg\langle C\rangle\neg\varphi$. We also write $(\omega_1 \succ_i \omega_2)$ to abbreviate $(\omega_1 \succeq_i \omega_2) \wedge \neg(\omega_2 \succeq_i \omega_1)$ and $\omega_1 =_i \omega_2$ to abbreviate $(\omega_1 \succeq_i \omega_2) \wedge (\omega_2 \succeq_i \omega_1)$.

Note that $\langle C\rangle\top$ is true iff C can at least bring about something: $V(C) \neq \emptyset$. $[C]\varphi$ means that $\neg\langle C\rangle\neg\varphi$, i.e., every choice of C must involve φ. As an example, suppose $\Omega = \{\omega_1, \omega_2, \omega_3, \omega_4\}$ and $V(C) = \{\omega_1, \omega_2\}$. Then the following formula is true:

$$\langle C\rangle\omega_1 \wedge \langle C\rangle(\omega_1 \vee \omega_3) \wedge \neg\langle C\rangle\omega_3 \wedge [C](\omega_1 \vee \omega_2) \wedge \neg[C](\omega_1 \vee \omega_3)$$

Note that if $\omega_1 \neq \omega_2$, then we can have $\langle C\rangle\omega_1 \wedge \langle C\rangle\omega_2$, but the formula $\langle C\rangle(\omega_1 \wedge \omega_2)$ can never be true.

Example 3. Let Γ be the Dinner Game from Example 1. We have the following:

- $\Gamma \models \langle 1,2\rangle I_{12} \wedge [1]\neg I_{12}$. Agents 1 and 2 can choose to go for Indian together, but 1 cannot choose on his own that they go together.
- $\Gamma \models \bigwedge_{\omega\in\Omega} \neg\langle 1\rangle\omega$. Agent 1 cannot choose any outcome on his own.
- $\Gamma \models \bigvee_{\omega\in\Omega}(\langle 1,2,3\rangle\omega \wedge \omega \succ_1 C_{12})$. The agents can together choose something which for 1 is better than going for Indian together with 2.
- $\Gamma \models \neg\langle 1,3\rangle C_{123} \wedge \bigvee_{\omega\in\Omega}(\omega \succ_1 C_{123} \wedge \omega \succ_3 C_{123})$. Agents 1 and 3 cannot choose that all agents go for Chinese, but they can choose something which is strictly better for both of them (e.g., going for Indian on their own).
- $\Gamma \models \bigwedge_{\omega\in\Omega} \neg\omega \succ_1 I_{123}$. Agent 1 prefers nothing better than having Indian with his two friends.
- $\Gamma \models \bigvee_{\omega\in\Omega}(\langle 2,3\rangle\omega \wedge I_{123} \succ_1 \omega)$. Agents 2 and 3 can choose some option which for 1 is worse than all three agents going for Indian.

Let us, for any coalition C and finite set of outcome symbols Δ, suggestively write $\langle[C]\rangle\Delta$ for $\bigwedge_{\delta\in\Delta}\langle C\rangle\delta \wedge [C]\bigvee_{\delta\in\Delta}\delta$.[7] A formula of this form is said to *fully describe C's choices*. It is easy to see that we have the following. Let $\Delta \subseteq \Omega$ be finite.

$$\Gamma \models \langle[C]\rangle\Delta \quad \text{iff} \quad V(C) = \Delta$$

A conjunction $\bigwedge_{\delta\in\emptyset}\varphi$ is, by convention, equal to $\top$, in the same way that $\bigvee_{\delta\in\emptyset}\varphi$ equals $\bot$, so that, indeed, we get $\Gamma \models \langle[C]\rangle\emptyset$ iff $\Gamma \models [C]\bot$ iff $\Gamma \models \neg\langle C\rangle\top$ iff $V(C) = \emptyset$.

Exclusive disjunctions $\varphi \triangledown \psi$ will play an important role in the proofs which follow. Note that the negation $\neg(\varphi \triangledown \psi)$ is the same as $(\neg\varphi \wedge \neg\psi) \vee (\varphi \wedge \psi)$. Moreover, if Φ is a set of formulas, then we define $\triangledown_{\varphi\in\Phi}\varphi$ to be true iff exactly one of the φ's is true. Formally, for any set $\Phi = \{\varphi_1, \ldots, \varphi_k\}$,[8]

$$\triangledown_{\varphi\in\Phi}\varphi \equiv \left(\bigvee_{i\leqslant k}\varphi_i \wedge \bigwedge_{j\neq i, j\leqslant k}\neg\varphi_j\right)$$

Note that $\Gamma \models [C](\omega_i \vee \omega_j) \leftrightarrow [C](\omega_i \triangledown \omega_j)$ when $i \neq j$: using the definition of $[C]$ and contraposition this is the same as $\Gamma \models \langle C\rangle\neg(\omega_i \triangledown \omega_j) \leftrightarrow \langle C\rangle\neg(\omega_i \vee \omega_j)$. Now, syntactically, $\langle C\rangle\neg(\omega_i \triangledown \omega_j)$ is equivalent to $\langle C\rangle((\omega_i \wedge \omega_j) \vee \neg(\omega_i \vee \omega_j))$. But, inspecting the truth-definition of $\langle C\rangle$, this is again equivalent to $\langle C\rangle\neg(\omega_i \vee \omega_j)$ since the $\mathcal{L}_o$ formula $\omega_i \wedge \omega_j$ is never true.

So, which properties of a coalitional game can be expressed with our cooperation language? The answer, given by the following theorem, is "all", when we restrict the possible outcomes of a game to a finite set.

Theorem 1. *The logic* CGL *is expressively complete with respect to finite coalitional games. That is, for any two finite coalitional games Γ_1, Γ_2 over N and Ω such that $\Gamma_1 \neq \Gamma_2$, there exists a* CGL *formula ζ such that $\Gamma_1 \models \zeta$ and $\Gamma_2 \not\models \zeta$.*

Proof. Our proof is constructive. Given a finite game Γ, we define a formula ζ_Γ that *completely characterises* Γ. ζ_Γ is constructed from two conjuncts, Π_Γ, which characterises the preference relations of Γ, and Ξ_Γ, which characterises the cooperative properties of Γ. Let $\mathcal{C} = 2^N \setminus \emptyset$ collect all non-empty coalitions from N.

$$\zeta_\Gamma \equiv \Pi_\Gamma \wedge \Xi_\Gamma, \quad \Pi_\Gamma \equiv \bigwedge_{i\in N}\Bigg(\bigwedge_{\substack{\omega,\omega'\in\Omega \\ \omega \sqsupseteq_i \omega'}}(\omega \succeq_i \omega') \wedge \bigwedge_{\substack{\omega,\omega'\in\Omega \\ \omega \not\sqsupseteq_i \omega'}}\neg(\omega \succeq_i \omega')\Bigg), \quad \Xi_\Gamma \equiv \Bigg(\bigwedge_{C\in\mathcal{C}}\Bigg(\bigwedge_{\omega\in V(C)}\langle C\rangle\omega\Bigg) \wedge [C]\Bigg(\bigvee_{\omega\in V(C)}\omega\Bigg)\Bigg)$$

[7] Note that the $\langle[C]\rangle$ modality plays the same role w.r.t. $[C]$ as Levesque's [21] *only knowing* operator plays w.r.t. the traditional belief operator.

[8] Note that $\triangledown_{p,q,r}$ is not the same as $p \triangledown (q \triangledown r)$!

Table 1
The logic CGL

Taut	$\vdash \varphi$ where φ is an instance of a prop. tautology
Lin	$\vdash (\omega_1 \succeq_i \omega_2) \vee (\omega_2 \succeq_i \omega_1)$
Ref	$\vdash (\omega \succeq_i \omega)$
Trans	$\vdash (\omega_1 \succeq_i \omega_2) \wedge (\omega_2 \succeq_i \omega_3) \rightarrow (\omega_1 \succeq_i \omega_3)$
K	$\vdash [C](\varphi \rightarrow \psi) \rightarrow (([C]\varphi) \rightarrow ([C]\psi))$
Func	$\vdash [C](\triangledown_{\omega\in\Omega}\omega)$
Nec	$\vdash_c \varphi_o \;\Rightarrow \vdash [C]\varphi_o$
MP	$\vdash \varphi, \vdash \varphi \rightarrow \psi \;\Rightarrow \vdash \psi$

By construction, for any Γ_1, we have $\Gamma_1 \models \zeta_{\Gamma_1}$. Moreover, for any coalitional game $\Gamma_2 \neq \Gamma_1$, we have that $\Gamma_2 \not\models \zeta_{\Gamma_1}$. □

Given a formula φ, let $coal(\varphi)$ denote the set of coalitions named in φ; let $ag(\varphi)$ denote the set of agents named in cooperation expressions or preference expressions in φ; and let $out(\varphi)$ be the set of outcomes named in φ.

3.2. Axioms and completeness

We now present an axiomatic system in the language $\mathcal{L}_c$, and prove its soundness and completeness with respect to the class of all finite coalitional games without transferable payoff.

From now on we assume that Ω is finite.

Table 1 summarises the axioms and rules of our logic CGL. Formally, CGL is the set of all $\mathcal{L}_c$-formulas derivable under $\vdash$. In the axioms, $\vdash_c$ denotes derivability of classical logic, and $\varphi_o \in \mathcal{L}_o, \varphi, \psi \in \mathcal{L}_c$. The axiom *Taut* and rule *MP* guarantee that we extend classical logic. On top of that, the axiom K and rule *Nec* determine $[C]$ to be a normal necessity operator. Then, *Lin*, *Ref*, and *Trans* determine the preference of each i to be complete, reflexive and transitive, respectively. The only specific cooperation axiom, *Func*, says that whatever a coalition in the end will chose, it must be a unique alternative from Ω.

We now continue by proving results about the CGL proof system, including its completeness. Examples of derivations can be found in Section 3.6.

The following lemma tells us that in the scope of modal operators, disjunctions over *different* outcomes behave the same as exclusive disjunctions over outcomes. Note that this is in general not true for arbitrary disjunctions: $\langle C\rangle(\omega_1 \vee \omega_1)$ is not the same as $\langle C\rangle(\omega_1 \triangledown \omega_1)$; the latter is equivalent to $\langle C\rangle\bot$.

Lemma 1. *Let $\emptyset \neq C \subseteq N$ and $\Delta \subseteq \Omega$.*

(1) *The following are equivalent, in* CGL: (i) $\langle C\rangle\top$, (ii) $\langle C\rangle \bigvee_{\omega\in\Omega}\omega$, *and* (iii) $\langle C\rangle \triangledown_{\omega\in\Omega}\omega$.
(2) *In the scope of $\langle C\rangle$ and $[C]$ when exchanging arbitrary occurrences of $\triangledown_{\omega\in\Delta}\omega$ with that of $\bigvee_{\omega\in\Delta}\omega$ in a formula φ, the result is equivalent to φ.*
(3) $\vdash \bigwedge_{\omega\in\Delta} \neg\langle C\rangle\omega \rightarrow (\langle C\rangle\top \rightarrow \bigvee_{\omega\in\Omega\setminus\Delta}\langle C\rangle\omega)$.
(4) $\vdash (\bigwedge_{\delta\in\Delta} \neg\langle C\rangle\delta) \leftrightarrow [C]\bigvee_{\delta'\in\Omega\setminus\Delta}\delta'$.

Proof.

(1) Since $\triangledown_{\omega\in\Omega}\omega \;\Rightarrow\; \bigvee_{\omega\in\Omega}\omega \;\Rightarrow\; \top$, and the $\langle C\rangle$ is a normal diamond operator, we have (iii) $\Rightarrow$ (ii) $\Rightarrow$ (i). By *Func*, we have (i) $\Rightarrow$ (iii).
(2) With induction over φ. The only interesting case $\varphi = \langle C\rangle\psi$ follows from the previous item.
(3) From axiom *Func* follows $\vdash \langle C\rangle\top \rightarrow \langle C\rangle \triangledown_{\omega\in\Omega}\omega$. By item 1, we have $\vdash \langle C\rangle\top \rightarrow \langle C\rangle \bigvee_{\omega\in\Omega}\omega$. Since $\langle C\rangle$ is a normal diamond, we have $\vdash \langle C\rangle\top \rightarrow \bigvee_{\omega\in\Omega}\langle C\rangle\omega$. Applying *Taut* to this and the assumption $\bigwedge_{\omega\in\Delta}\neg\langle C\rangle\omega$, gives the desired property.
(4) Follows directly from *Func*, modal reasoning and the previous item. □

Lemma 2.

(1) *$C \neq \emptyset$ be a coalition. Then* $\vdash \bigvee_{\Delta\subseteq\Omega}\langle[C]\rangle\Delta$.
(2) *Let $C \neq \emptyset$ be a coalition. Then* $\vdash \triangledown_{\Delta\subseteq\Omega}\langle[C]\rangle\Delta$.
(3) $\vdash \bigwedge_{C\in\mathcal{C}} \bigvee_{\Delta\subseteq\Omega}\langle[C]\rangle\Delta$.
(4) $\vdash \bigwedge_{C\in\mathcal{C}} \triangle_{\Delta\subseteq\Omega}\langle[C]\rangle\Delta$.
(5) $\vdash \bigvee_{\langle\Delta_C:\, C\in\mathcal{C}\rangle\subseteq\Omega^{|\mathcal{C}|}} \bigwedge_{C\in\mathcal{C}}\langle[C]\rangle\Delta_C$
(*the disjunction is over the set of possible tuples consisting of one set of outcomes Δ_C for each coalition $C \in \mathcal{C}$*).

Proof.

(1) Note that $\langle C\rangle\omega$ is just an atom in the coalitional language. Let $M=\{1,\dots,n\}$. Then, even by propositional reasoning, $\vdash\bigvee_{I\cup J=M,\, I\cap J=\emptyset}(\bigwedge_{i\in I}\langle C\rangle\omega_i\wedge\bigwedge_{j\in J}\neg\langle C\rangle\omega_j)$. Now, take a fixed I and J with $I\cap J=\emptyset$ and $I\cup J=M$. If $I=\emptyset$, then $\bigwedge_{i\in I}\langle C\rangle\omega_i\wedge\bigwedge_{j\in J}\neg\langle C\rangle\omega_j$ equals $\bigwedge_{\omega\in\Omega}\neg\langle C\rangle\omega$, which is equivalent to $\langle[C]\rangle\emptyset$. With Lemma 1 item 4, we have, for fixed I and J, each disjunct in this is equivalent to $\langle[C]\rangle\Delta_I$, with $\Delta_I=\{\omega_i\colon i\in I\}$.

(2) Note that, using the notation of the previous item, we even have

$$\vdash\triangle_{I\cup J=M,\, I\cap J=\emptyset}\Bigl(\bigwedge_{i\in I}\langle C\rangle\omega_i\wedge\bigwedge_{j\in J}\neg\langle C\rangle\omega_j\Bigr)$$

From this the statement follows directly.

(3) This is immediate from item 1: if for an arbitrary C we have $\vdash\langle[C]\rangle\varphi_C$, then also $\vdash\bigwedge_{C\subseteq Z}\langle[C]\rangle\varphi_C$, for any $Z\in\mathcal{C}$.

(4) Follows from item 2 in the same way as 3 follows from 1.

(5) This follows immediately from item 1 and propositional reasoning: note that for every two coalitions C_1 and C_2 we derive $\vdash\bigvee_{\Delta_1\subseteq\Omega}\langle[C_1]\rangle\Delta_1\wedge\bigvee_{\Delta_2\subseteq\Omega}\langle[C_2]\rangle\Delta_2$, and use $(p\vee q)\wedge r\equiv(p\wedge r)\vee(q\wedge r)$. □

Definition 4.

(1) For any agent i, we say that a formula is a preference literal for i if it is either $\omega\succeq_i\omega'$ or $\neg(\omega\succeq_i\omega')$, for some ω and ω'. We say that π_i fully describes i's preferences, if π_i is of the form $\bigwedge_{\omega,\omega'\in\Omega}(\neg)(\omega\succeq_i\omega')$. We then say that $\pi\in PossPref(i)$.

(2) Given that we have m agents, a conjunction $\Pi=(\pi_1\wedge\cdots\wedge\pi_m)$, (where each π_i fully describes i's preferences) is said to fully describe the preferences of all the agents. From now on, we let K denote the set of all possible such conjunctions Π.

(3) Recall that $\langle[C]\rangle\Delta$, where C is a coalition and Δ is a set of (atoms for) outcomes $\omega_1,\dots,\omega_u$, is said to fully describe C's choices. Now let $\mathcal{C}=2^N\setminus\emptyset$, and let, for each $C\in\mathcal{C}$, Δ_C be a set of outcomes. Then $\bigwedge_{C\in\mathcal{C}}\langle[C]\rangle\Delta_C$ is said to fully describe all of N's choices. (Similarly for subsets of N.) We often will denote such a full description by Ξ.

Lemma 3.

(1) *Let $\Pi_1,\Pi_2,\dots,\Pi_d$ be all full descriptions of N's preferences. Then*: $\vdash\bigvee_{k\leqslant d}\Pi_k$.

(2) *Let $\Pi_1,\Pi_2,\dots,\Pi_d$ be all full descriptions of N's preferences. Moreover, let $\mathcal{C}=2^N\setminus\emptyset$ and let*

$$\Bigl(\bigwedge_{C\in\mathcal{C}}\langle[C]\rangle\Delta_C\Bigr)_1,\dots,\Bigl(\bigwedge_{C\in\mathcal{C}}\langle[C]\rangle\Delta_C\Bigr)_z$$

enumerate all possible full descriptions of all choices of all coalitions (note that $z=(2^n)^m$). From now on, we will let T denote the set of all such descriptions. Then:

$$\vdash\bigvee_{k\leqslant d,t\leqslant z}\Bigl(\Pi_k\wedge\bigwedge_{C\in\mathcal{C}}\langle[C]\rangle\Delta_C\Bigr)_t$$

Theorem 2. *Let φ be a formula of the cooperation language. Let T and K be as in Definition* 4 *and Lemma* 3.

(1) *Let N be the set of agents, Ω the set of outcomes, and let $\mathcal{C}=2^N\setminus\emptyset$. Then φ is equivalent to a formula of the form*

$$\bigvee_{k\in K,t\in T}\Bigl(\Pi_k\wedge\Bigl(\bigwedge_{C\in\mathcal{C}}\langle[C]\rangle\Delta_C\Bigr)_t\Bigr)$$

where each $\Pi_k=(\pi_1\wedge\cdots\wedge\pi_m)$ fully describes N's preferences, i.e., each π_i fully describes i's preferences, and each $(\bigwedge_{C\in\mathcal{C}}\langle[C]\rangle\Delta_C)_t$ describes fully what N can choose.

(2) *The same holds if we take $\Omega=out(\varphi)$, $N=ag(\varphi)$ and we let C range over all coal(φ).*

Complex as it may appear, our normal form is nothing more than an enumeration of possible full preferences combined with full descriptions of choices. The range of these possibilities is determined by the index sets K and T, which act like constraints: the smaller those index sets, the smaller the possible models for the formula. As a reading guide, note that

$$\bigvee_{k\in\{1,2\},\,t\in\{a,b\}}(A_k\wedge B_t)$$

equals $(A_1\wedge B_a)\vee(A_1\wedge B_b)\vee(A_2\wedge B_a)\vee(A_2\wedge B_b)$.

Proof. Note that the theorem is semantically obvious, the point is that we should be able to *syntactically prove* it from the axioms given. This is done by induction over φ.

Suppose $\varphi = \omega_1 \succeq_k \omega_2$. Let us say that a π_k is k-compatible with $(\omega_1 \succeq_k \omega_2)$ if the latter occurs as a conjunct in π_k. We then write $(\omega_1 \succeq_k \omega_2) \in \pi_k$. Then, φ is (in propositional logic) equivalent to

$$\bigvee_{\pi_i \in PossPref(i)\ (i \neq k),\ (\omega_1 \succeq_k \omega_2) \in \pi_k} (\pi_1 \wedge \cdots \wedge \pi_k \wedge \cdots \wedge \pi_m)$$

This if of the form $\bigvee_{k \in K} \Pi_k$. Since by Lemma 2, item 5 we have that $\top$ is equivalent to $\bigvee_{\Delta_C \subseteq \Omega, C \in \mathcal{C}} \bigwedge_{C \in \mathcal{C}} \langle [C] \rangle \Delta_C$ we see that φ is equivalent to

$$\bigvee_{k \in K} \Pi_k \wedge \bigvee_{\Delta_C \subseteq \Omega,\ C \in \mathcal{C}} \bigwedge_{C \in \mathcal{C}} \langle [C] \rangle \Delta_C$$

which, by propositional reasoning, is equivalent to

$$\bigvee_{k \in K,\ \Delta_C \subseteq \Omega,\ C \in \mathcal{C}} \Pi_k \wedge \bigwedge_{C \in \mathcal{C}} \langle [C] \rangle \Delta_C$$

Suppose $\varphi = \langle E \rangle \varphi_0$. Formula φ_0 regards outcomes, and is equivalent to a disjunction $\bigvee \alpha$ where each α is of the form

$$\alpha = \big((\neg)\omega_1 \wedge (\neg)\omega_2 \wedge \cdots \wedge (\neg)\omega_n\big)$$

Using rule *Nec* and axiom K, we derive $\vdash \langle E \rangle \varphi_0 \leftrightarrow \langle E \rangle \bigvee \alpha$. Using that $\langle \cdot \rangle$ is a diamond, we then obtain that

$$\vdash \langle E \rangle \varphi_0 \leftrightarrow \bigvee \langle E \rangle \alpha$$

Now we use axiom *Func* to get rid of every α that contains more than one positive literal ω_i: let β range over all the those α's with at most one positive literal. Then $\vdash \langle E \rangle \varphi_0 \leftrightarrow \bigvee \langle E \rangle \beta$. Now, again in propositional logic, note that every disjunction $\bigvee_{i \in M} \psi_i$ is equivalent to

$$\bigvee_{I \cup J = M,\ I \cap J = \emptyset} \left(\bigwedge_{i \in I} \psi_i \wedge \bigwedge_{j \in J} \neg \psi_j \right)$$

In our case, letting the β's range over $\beta_1, \ldots, \beta_M$,

$$\vdash \langle E \rangle \varphi_0 \leftrightarrow \bigvee_{I \cup J = M,\ I \cap J = \emptyset} \left(\bigwedge_{i \in I} \langle E \rangle \beta_i \wedge \bigwedge_{j \in J} \neg \langle E \rangle \beta_j \right)$$

But, for every fixed $I, J \subseteq M$, $(\bigwedge_{i \in I} \langle E \rangle \beta_i \wedge \bigwedge_{j \in J} \neg \langle E \rangle \beta_j)$ is equivalent to $\langle [E] \rangle \Delta_I$, with $\Delta_I = \{\omega_i\colon\ i \in I\}$. Hence, we find that $\vdash \langle E \rangle \varphi_0 \leftrightarrow \bigvee_{I \subseteq M} \langle [E] \rangle \Delta_I$. This only limits the abilities of C, and not those of the others: using Lemma 2 item 5 once again:

$$\bigvee_{I \subseteq M} \langle [E] \rangle \Delta_I \leftrightarrow \bigvee_{I \subseteq M,\ D \neq E,\ \Delta_D \subseteq \Omega} \langle [E] \rangle \Delta_I \wedge \bigwedge_{D \in \mathcal{C}} \langle [D] \rangle \Delta_D$$

of which the r.h.s. is of the form $\bigvee_{C \in \mathcal{C}, t \in T} (\langle [C] \rangle \Delta_C)_t$, for some index set T. Since $\top$ is provably equivalent with $\bigvee_{k \in K} \Pi_k$ (where now K gives all possible full preference descriptions), the result follows.

Suppose φ is of the form $\varphi_1 \vee \varphi_2$. We can assume that $\varphi_i \equiv \bigvee_{k_i \in K_i, t_i \in T_i} (\Pi_{k_i} \wedge \Xi_{t_i})$, then

$$\varphi \equiv \bigvee_{k \in K_1 \cup K_2,\ t \in T_1 \cup T_2} (\Pi_k \wedge \Xi_t)$$

which is of the required form.

Let $\varphi = \neg \varphi_1$. Then $\varphi_1 = \bigvee_{k \in K, t \in T} (\Pi_k \wedge \Xi_t)$, for some index sets K and T. From Lemma 3, item 2, we know that $\vdash \bigvee_{k \leqslant d, t \leqslant z} (\Pi_k \wedge \Xi_t)$. In words: we know that a big disjunction is valid, but also that φ excludes some of them. Then, using propositional reasoning again, we obtain:

$$\varphi \equiv \bigvee_{k \leqslant d,\ k \notin K,\ t \leqslant z,\ t \notin T} (\Pi_k \wedge \Xi_t) \qquad \square$$

Theorem 3 *(Completeness). We have, for all $\varphi \in \mathcal{L}_c$: $\models \varphi \Rightarrow \vdash \varphi$.*

Proof. Suppose $\nvdash \varphi$, i.e., $\neg \varphi$ is consistent. We know that

$$\vdash \neg \varphi \leftrightarrow \bigvee_{k \in K,\ t \in T} \left(\Pi_k \wedge \left(\bigwedge_{C \in \mathcal{C}} \langle [C] \rangle \Delta_C \right)_t \right)$$

for some index sets K and T. Call the right-hand side of this equivalence φ'. Let X be a maximal consistent set around φ'. By virtue of maximal consistent sets, we know that for some k and t, $\varphi'' = (\Pi_k \wedge (\bigwedge_{C \in \mathcal{C}} \langle [C] \rangle \Delta_C)_t) \in X$. But now we can read off the finite game $\Gamma = \langle N, \Omega, V, \sqsupseteq_1, \ldots, \sqsupseteq_m \rangle$ from φ'' immediately:

(1) N and Ω are already given;
(2) let $V(C) = \Delta$, where $\langle [C] \rangle \Delta$ is part of φ';
(3) every $\sqsupseteq_i$ relation is immediately read off from the component π_i for Π_k in φ''.

Now, it is easy to see, for every sub-formula ψ of φ'':

$$\Gamma \models \psi \quad \Leftrightarrow \quad \psi \in X \qquad \square$$

3.3. Model checking and the complexity of satisfiability

It is easy to see that the *model checking problem* for CGL (i.e., the problem of determining, for any given game Γ and φ, whether or not $\Gamma \models \varphi$ [7]) may be solved in deterministic polynomial time: an obvious recursive algorithm for this problem can be directly extracted from the semantic rules of the language.

The *satisfiability problem* is the problem of checking whether or not, for any given φ there exists a game Γ such that $\Gamma \models \varphi$. For most modal logics, the corresponding satisfiability problem has a trivial NP-hard lower bound, since such logics subsume propositional logic, for which satisfiability is the defining NP-complete problem [6, p. 374]. However, our logic is specialised for reasoning about coalitional games, and it is not so obvious that it subsumes propositional logic, since we do not have primitive propositions. We must therefore prove NP-hardness from first principles.

For the proof, we need a few additional constructions. A *partial coalitional game* is a structure $\langle N, \Omega, V, \sqsupseteq_1, \ldots, \sqsupseteq_m \rangle$ where all the components are as in regular coalitional games, except that V is a *partial* function, i.e., it is not required to be defined for every possible coalition. Given a partial game $\Gamma = \langle N, \Omega, V, \sqsupseteq_1, \ldots, \sqsupseteq_m \rangle$, we can use the semantic rules for CGL to interpret some formulae (although because V is not defined for all coalitions, we cannot necessarily interpret all formulae over N, Ω). Where Γ is a partial game and φ is a formula, let us write $\Gamma \models_p \varphi$ to mean that (i) it is possible to evaluate φ with respect to Γ, and (ii) φ is true under this evaluation. Now, we can prove the following.

Lemma 4. *A CGL formula φ is satisfiable iff there exists a partial game $\Gamma = \langle N, \Omega, V, \sqsupseteq_1, \ldots, \sqsupseteq_m \rangle$ such that*:

(1) $N = ag(\varphi)$,
(2) $|\Omega| = |out(\varphi)| + 1$ *and* $out(\varphi) \subseteq \Omega$,
(3) $\mathrm{dom} V = coal(\varphi)$, *and*
(4) $\Gamma \models_p \varphi$.

Proof. The right-to-left direction is obvious, so consider the left-to-right direction, and let $\Gamma = \langle N, \Omega, V, \sqsupseteq_1, \ldots, \sqsupseteq_m \rangle$ be a game such that $\Gamma \models \varphi$. Let $A = \Omega \setminus out(\varphi)$, i.e., A is the set of outcomes in Γ not named in φ. Let ω^* be an outcome such that $\omega^* \notin \Omega$, and define a partial game $\Gamma^* = \langle N^*, \Omega^*, V^*, \sqsupseteq_1^*, \ldots, \sqsupseteq_o^* \rangle$ as follows:

- $N^* = ag(\varphi)$;
- $\Omega^* = out(\varphi) \cup \{\omega^*\}$;
- The relation $\sqsupseteq_i^*$ is obtained by first restricting $\sqsupseteq_i$ to $out(\varphi)$, and then defining $\omega^* \sqsupset_i \omega$ for all $\omega \in out(\varphi)$;
- V^* is the partial function such that V^* is only defined for coalitions named in φ (i.e., $C \in \mathrm{dom} V^*$ iff $C \in coal(\varphi)$);
- $V^*(C) = \begin{cases} V(C) & \text{if } V(C) \subseteq out(\varphi) \\ (V(C) \setminus A) \cup \{\omega^*\} & \text{otherwise.} \end{cases}$

Notice that Γ^* satisfies conditions (1)–(3) of the lemma. We now prove that Γ^* satisfies condition (4). More precisely, we show that for all sub-formulae ψ of φ: $\Gamma \models \psi$ iff $\Gamma^* \models \psi$. The inductive base is where $\varphi = (\omega_1 \succeq_i \omega_2)$, and is obvious, since $i \in ag(\varphi)$ and $\{\omega_1, \omega_2\} \subseteq out(\varphi)$, and hence $\omega_1 \sqsupseteq_i^* \omega_2$ iff $\omega_1 \sqsupseteq_i \omega_2$.

For the inductive assumption, assume the result is proved for all sub-formulae; in the inductive step, the significant case is where $\varphi = \langle C \rangle \psi$. If $\Gamma \models \langle C \rangle \psi$ then $\exists \omega \in V(C)$ such that $\Gamma, \omega \models \psi$. There are two possibilities: either $\omega \in out(\varphi)$ (in which case $V^*(C) = V(C)$, and the result is obvious), or else $\omega \notin out(\varphi)$. In the latter case, $V^*(C) = (V(C) \setminus A) \cup \{\omega^*\}$; we claim that $\Gamma^*, \omega^* \models \psi$. To see this, assume w.l.o.g. that ψ is in Conjunctive Normal Form. Now, since $\omega \notin out(\varphi)$, then no positive literals can be satisfied by Γ, ω: only negative literals. But such literals must also be satisfied by Γ^*, ω^*, and so $\Gamma^*, \omega^* \models \psi$.

The case for $\Gamma \not\models \langle C \rangle \psi$ implies $\Gamma^* \not\models \langle C \rangle \psi$ is similar. $\square$

Given this, we can prove:

Theorem 4. *The satisfiability problem for CGL formulae is NP-complete, even for CGL formulae φ such that $|ag(\varphi)| = 1$.*

Proof. For membership of NP, we know that φ is satisfiable iff it has a "certificate" for this in the form of a partial game Γ as in Lemma 4. This partial game is of size linear in the size of the formula φ. Since we can check whether $\Gamma \models_p \varphi$ in polynomial time, we conclude that CGL satisfiability is in NP.

For NP-hardness, we reduce SAT, the problem of determining whether a formula $\varphi(x_1, \ldots, x_k)$ of propositional logic, over Boolean variables $x_1, \ldots, x_k$, has a satisfying assignment [25]. The basic idea is to map variables x_i to outcomes ω_i, to introduce an additional outcome $\omega_\perp$ to correspond to the truth value "false", so that $(\omega_x \succ_1 \omega_\perp)$ will mean "x takes the value 'true'". Formally, let $\varphi^\#$ denote the CGL formula obtained from the propositional logic formula φ by systematically replacing every Boolean variable p by the corresponding CGL expression $(\omega_p \succ_1 \omega_\perp)$. Now, we claim that $\varphi^\#$ is CGL satisfiable iff the input SAT instance φ is a satisfiable formula of propositional logic.

For the $\Rightarrow$ direction, assume $\varphi^\#$ is CGL satisfiable, and consider the associated preference relation $\sqsupseteq_1$ in any Γ such that $\Gamma \models \varphi$. From this relation, extract a valuation ξ for the variables $x_1, \ldots, x_k$ as follows: each variable x_i is true under ξ if $\omega_{x_i} \sqsupset_1 \omega_\perp$, and false otherwise. The interpretation ξ is consistent, since we cannot have both $\omega_{x_i} \sqsupset_1 \omega_\perp$ and $\omega_\perp \sqsupseteq_1 \omega_{x_i}$. The interpretation ξ satisfies φ by a trivial induction on the structure of φ.

For $\Leftarrow$, assume φ is a satisfiable formula of propositional logic, and let ξ be a valuation that satisfies φ. Then we can reconstruct a game Γ_ξ such that $\Gamma_\xi \models \varphi^\#$, as follows. Γ_ξ contains a single agent, (agent 1), and an outcome ω_{x_i} for each variable x_i appearing in φ. We also define an additional outcome $\omega_\perp$. The preference relation $\sqsupseteq_i$ is then defined as follows:

- For each Boolean variable p such that p is true under ξ, define $\omega_p \sqsupset_i \omega_\perp$.
- For each pair of Boolean variables p_1, p_2 such that p_1 and p_2 are both true or both false under ξ, define $\omega_{p_1} =_i \omega_{p_2}$.
- For each Boolean variable p such that p is false under ξ, define $\omega_\perp \sqsupseteq_i \omega_p$.
- For each pair of Boolean variables p_1, p_2 such that p_1 is true (respectively, false) and p_2 is false (respectively, true) under ξ, define $\omega_{p_1} \sqsupset_i \omega_{p_2}$ (respectively, $\omega_{p_2} \sqsupset_i \omega_{p_1}$).

An induction on $\varphi^\#$ proves that $\Gamma_\xi \models \varphi^\#$. $\square$

3.4. A decision procedure for satisfiability

Although Theorem 4 classifies the complexity of the satisfiability problem for CGL, it does not give us a direct decision procedure for establishing satisfiability. Now, the nature of NP-completeness immediately indicates one possible decision procedure: iterate through all possible partial models for the formula, checking each one in turn to see whether it satisfies the input formula. However, such an approach would surely be completely impracticable. In this section, we present an algorithm for satisfiability checking that was specifically developed for CGL. This approach is based on the method of *analytic tableaux*. The basic idea of such methods is to use the structure of a formula to guide the search for a satisfying model; while in the worst case tableaux methods do no better than naive search, in many cases they prove highly efficient. Tableaux methods have been widely applied for satisfiability checking in both classical and non-classical logics (see, e.g., [19,33]), and in particular have proven to be very effective in practice for modal and temporal logics [18].

To introduce the decision procedure, we need some subsidiary definitions and assumptions. First, throughout the procedure, we assume that double negations ($\neg\neg$) are eliminated whenever they occur by the standard logical equivalence $\varphi \leftrightarrow \neg\neg\varphi$. Second, we will assume that bi-conditionals have been eliminated by expanding them according to their definition, leaving only the classical operators $\vee, \wedge, \rightarrow$, and $\neg$ in a formula.

Now, we say an *α-formula* of $\mathcal{L}_o$ or $\mathcal{L}_c$ is one in which the outermost logical operators of the formula define a conjunction, while a *β-formula* is one where the outermost logical operators define a disjunction. Given an α-formula φ, we denote the conjuncts of this formula by $\alpha_1(\varphi)$ and $\alpha_2(\varphi)$, respectively, and we similarly assume $\beta_1(\varphi)$ and $\beta_2(\varphi)$ give the disjuncts of β-formula φ. The following table defines the functions α_i and β_i (see, e.g., [33, p. 21]).

φ	$\alpha_1(\varphi)$	$\alpha_2(\varphi)$	φ	$\beta_1(\varphi)$	$\beta_2(\varphi)$
$\psi \wedge \chi$	ψ	χ	$\neg(\psi \wedge \chi)$	$\neg\psi$	$\neg\chi$
$\neg(\psi \vee \chi)$	$\neg\psi$	$\neg\chi$	$\psi \vee \chi$	ψ	χ
$\neg(\psi \rightarrow \chi)$	ψ	$\neg\chi$	$\psi \rightarrow \chi$	$\neg\psi$	χ

If S is a set of $\mathcal{L}_o$ or $\mathcal{L}_c$ formulae, then we say that S is:

- *α-closed* if for every $\varphi \in S$, if φ is an α-formula then both $\alpha_1(\varphi) \in S$ and $\alpha_2(\varphi) \in S$;
- *β-closed* if for every $\varphi \in S$, if φ is a β-formula then *either* $\beta_1(\varphi) \in S$ or $\beta_2(\varphi) \in S$;
- *proper* if whenever $\varphi \in S$, we have $\neg\varphi \notin S$, and *improper* otherwise.

Phase 1: Decomposing the $\mathcal{L}_c$ formula

We can now present the first stage of the satisfiability checking algorithm. The basic idea is to systematically decompose an $\mathcal{L}_c$ formula using α- and β-functions, in much the same way as classical tableaux [33]. In the classical case, the process results in a set of sets of propositional literals; each such set represents one way that the original formula could be satisfied, and each such set of literals is itself easily checked for satisfiability. In our case, we do not end up with sets of propositional literals, but with sets of preference literals and coalition literals; nevertheless, each such set represents one possible way the original input formula might be satisfied. Later stages check these sets for satisfiability, dealing with preference literals and coalition literals.

We will say a *root* is a set of $\mathcal{L}_c$ formulae together with a *labelling*, which tells us for every member of the set whether the formula is *treated* or *untreated*. A *pre-tableaux* is a set of roots.

Given an input formula $\varphi \in \mathcal{L}_c$, we start with a pre-tableaux $R' = \{\{\varphi\}\}$ in which φ is untreated. We then iteratively repeat the following steps:

(1) If R' contains a root S such that S is improper, then delete S from R.
(2) If R' contains a root S with an untreated α-formula φ, then:
   (a) create a new root S' from S, with $S' = S \cup \{\alpha_1(\varphi), \alpha_2(\varphi)\}$ where φ is now labelled as "treated" and $\alpha_1(\varphi)$ and $\alpha_2(\varphi)$ are labelled as "untreated";
   (b) remove S from R' and add S' to R'.
(3) If R' contains a root S with an untreated β-formula φ, then:
   (a) create a new root S_1, set $S_1 := S \cup \{\beta_1(\varphi)\}$ with φ now labelled as "treated" and $\beta_1(\varphi)$ as "untreated";
   (b) create a new root S_2, set $S_2 := S \cup \{\beta_2(\varphi)\}$ with φ now labelled as "treated" and $\beta_2(\varphi)$ as "untreated";
   (c) remove S from R' and add S_1 and S_2 to R'.
(4) If R' contains no roots to which the above rules can be applied, then quit.

Let $R'(\varphi)$ be the set of roots obtained by applying the above procedure to input CGL formula φ. No member of $R'(\varphi)$ will contain any untreated α- or β-formula. It is easy to see that, for any $S \in R'(\varphi)$, the only untreated formulae will be of the form:

- *preference literals*: $(\omega \succeq_i \omega')$ or $\neg(\omega \succeq_i \omega')$;
- *coalition literals*: $\langle C\rangle\psi$ or $\neg\langle C\rangle\psi$.

Essentially, the next two phases of the procedure deal with these two classes of formulae.

Phase 2: Dealing with preference literals

The second stage of the decision procedure deals with preference literals. The idea is simply to eliminate any roots that correspond to inherently contradictory sets of preference literals. Given a set S of $\mathcal{L}_c$ formula, we denote by *prlit*(S) the set of preference literals it contains. Now, clearly, *prlit*(S) will induce a collection of preference relations, one for each agent named in an atom in *prlit*(S). At this stage what we simply need to check is that the relations induced in this way are indeed preference relations: they are complete, reflexive, and transitive. To do this we use a *closure* procedure. The idea of the closure procedure is to take a set of preference literals, and then from this generate sets of preference literals corresponding to every possible preference ordering. For *prlit*(S) to be satisfiable, it must be consistent with some such set. We generate the closure in two parts: the first deals with the requirement that preference relations must be complete, the second deals with reflexivity and transitivity.

The completeness requirement on preference relations dictates that, for every agent i and every pair of outcomes ω, ω' we have either $(\omega \succeq_i \omega')$ or $(\omega' \succeq_i \omega)$. Thus, we start by generating the set of all such combinations. The algorithm for this is as follows:

(1) $Y := \{\emptyset\}$
(2) for each $i \in ag(\varphi)$ do
      for each $\omega, \omega' \in out(\varphi)$ do
         for each $Z \in Y$ do
            $Z_1 := Z \cup \{(\omega \succeq_i \omega')\}$
            $Z_2 := Z \cup \{(\omega' \succeq_i \omega)\}$
            $Y := (Y \setminus Z) \cup \{Z_1, Z_2\}$.

Let $Y(\varphi)$ denote the set of sets of preference literals obtained from CGL formula φ by this process. Next, we replace each root $S \in R'(\varphi)$ with S combined with every combination of sets of preference literals in $Y(\varphi)$. The idea is that, if a root $S \in R'(\varphi)$ is satisfiable, then it will be compatible with at least one of the sets in $Y(\varphi)$.

(1) $R''(\varphi) := \emptyset$
(2) for each $S \in R'(\varphi)$ do
      for each $X \in Y(\varphi)$ do
         $R''(\varphi) := R''(\varphi) \cup \{S \cup X\}$.

Finally, we exhaustively apply the following rules to $R''(\varphi)$, until no new applications of these rules are possible.

(1) for each $S \in R''(\varphi)$ do
for each $i \in ag(\varphi)$ and $\omega \in out(\varphi)$
$S := S \cup \{(\omega \succeq_i \omega)\}$
for each $i \in ag(\varphi)$, $(\omega_3 \succeq_i \omega_2) \in prlit(S)$, $(\omega_2 \succeq_i \omega_1) \in prlit(S)$
$S := S \cup \{(\omega_3 \succeq_i \omega_1)\}$.

Intuitively, the first step corresponds to generating the reflexive closure of the preference relations, while the second step corresponds to taking the transitive closure. Where X is a set of preference literals, then we say X is *closed for preference literals* iff:

(1) for every pair of outcomes ω, ω' named in X, and every agent i named in X, either $(\omega \succeq_i \omega')$ or $(\omega' \succeq_i \omega)$;
(2) for every outcome ω named in X, and every agent i named in X, $(\omega \succeq_i \omega) \in X$;
(3) for every agent i named in X, if $(\omega \succeq_i \omega') \in X$ and $(\omega' \succeq_i \omega'') \in X$ then $(\omega \succeq_i \omega'') \in X$.

Now, the point about closure for preference literals is the following.

Lemma 5. *Let X be a set of preference literals. Then the* CGL *formula $\bigwedge_{\rho \in X} \rho$ is* CGL *satisfiable iff there exists a set X' of* CGL *formulae such that X' is proper and closed for preference literals, and $X \subseteq X'$.*

So, to conclude the second stage of the decision procedure, we delete from $R''(\varphi)$ any set S that is no longer proper as a result of the closure procedure. Let $R(\varphi)$ be the structure obtained from $R''(\varphi)$ by applying the closure procedure and then deleting from it any roots that are no longer proper.

It should be clear that what we are doing in the above procedure is taking each root S, and then systematically trying to find an extension of S that is proper and closed for preference literals. The search is systematic in the sense that it considers all possible alternatives.

Phase 3: Creating branches from choices

We now move into the third stage of the procedure, where we deal with modal literals. At this point, our structures become richer, not simply having roots, but *branches*, where each branch will correspond to a choice of a set of agents. To see how this stage works, consider that an existential cooperation modality $\langle C \rangle \psi$ asserts the existence of a choice for coalition C (they have a choice satisfying ψ), while a universal modality, of the form $\neg\langle C \rangle \psi$, asserts a constraint on the choices of C: every choice they have satisfies $\neg\psi$. This stage of the procedure thus attempts to construct choices corresponding to each existential modality, where these choices must respect the constraints imposed by the relevant universal cooperation modalities.

Some more definitions are needed. An *outcome literal* is either an outcome symbol ω or the negation of such a symbol, $\neg\omega$. We say an $\mathcal{L}_o$ formula ψ is outcome-satisfiable if for some Γ, ω we have $\Gamma, \omega \models \psi$. We say a set of outcome formulae Y is outcome-proper if it is proper and moreover for no ω_1 and ω_2 is it the case that both $\omega_1 \in Y$ and $\omega_2 \in Y$. The key point about outcome-proper sets of outcome literals is the following readily proved result.

Lemma 6. *Let Y be a set of outcome literals. Then the outcome formula $\bigwedge_{\mu \in Y} \mu$ is outcome-satisfiable iff it is outcome-proper.*

A *tableau*, Υ, is a set of pairs (S, B), where $S \subseteq \mathcal{L}_c$ and $B \subseteq 2^{ag(\varphi)} \times 2^{\mathcal{L}_o}$ is a set of *branches*; S will be a root of $R(\varphi)$, while B will be a set of choices corresponding to the modal formulae in S.

Given a structure $R(\varphi)$ as generated from a CGL formula φ by the above procedure, we create a tableau $\Upsilon'(\varphi)$ using the following procedure.

(1) initialise $\Upsilon'(\varphi)$ to be empty;
(2) for each $S \in R(\varphi)$, add a pair, (S, B_S) to $\Upsilon'(\varphi)$, where B_S is constructed as follows:
  (a) initialise B_S to $\emptyset$,
  (b) for each coalition $C \in coal(\varphi)$, and for each formula $\langle C \rangle \psi \in S$, let

$$B_S = B_S \cup \big(C, \{\psi, \neg\chi \colon \neg\langle C \rangle \chi \in S\}\big)$$

We now apply α- and β-rules as introduced earlier in each branch of $\Upsilon'(\varphi)$. Now, however, we delete any branches obtained that are not outcome-proper. (Intuitively, a branch corresponds to a choice of a coalition, and a choice is a single member of Ω.) Let $\Upsilon(\varphi)$ be the tableau obtained from $\Upsilon'(\varphi)$ by deleting outcome-improper branches.

Phase 4: Checking the tableau

We say a tableau Υ is *finished* for the CGL formula φ iff:

(1) For some $(S, B) \in \Upsilon$, $\varphi \in S$.

(2) For each $(S, B) \in \Upsilon$, S is:
   (a) proper;
   (b) α-closed;
   (c) β-closed.
(3) For each $(S, B) \in \Upsilon$, S is closed with respect to preference literals.
(4) For each $(S, B) \in \Upsilon$, and for each $(C, Y) \in B$, Y is:
   (a) α-closed;
   (b) β-closed;
   (c) outcome-proper.
(5) For each $(S, B) \in \Upsilon$, if $\langle C\rangle\psi \in S$ then there exists a branch $(C, Y) \in B$ such that $\psi \in Y$.
(6) For each $(S, B) \in \Upsilon$, if $\neg\langle C\rangle\psi \in S$ then for each branch $(C, Y) \in B$, $\neg\psi \in Y$.

The key point about finished tableaux is the following.

Theorem 5. *A* CGL *formula φ is satisfiable iff there exists a finished tableaux for φ.*

Proof. ($\Rightarrow$) Assume φ is satisfiable, and let Γ be the witness to this. Then we can immediately construct a finished tableau Υ_Γ for φ from Γ. ($\Leftarrow$) Assume Υ is a finished tableau for φ. Then we can immediately "read off" from Υ a partial game Γ_Υ that satisfies φ. □

So, given a CGL formula φ, which we want to test for satisfiability, our decision procedure is then as follows:

(1) Generate the tableau $\Upsilon(\varphi)$ using the procedure described above.
(2) Check that the tableau $\Upsilon(\varphi)$ generated in this way is finished; if so, announce "φ is satisfiable", otherwise announce "φ is unsatisfiable".

That the procedure is correct and is guaranteed to terminate is immediate from construction.

3.5. Characterising coalitional games

We characterise the three solution concepts from the theory of coalitional games discussed in Section 2.1.

We write $CM(\omega)$ to mean that ω is in the core.

$$CM(\omega) \equiv \langle N\rangle\omega \wedge \neg\Big[\bigvee_{C\subseteq N}\bigvee_{\omega'\in\Omega}(\langle C\rangle\omega') \wedge \bigwedge_{i\in C}(\omega' \succ_i \omega)\Big]$$

CNE will then mean that the core is non-empty:

$$CNE \equiv \bigvee_{\omega\in\Omega} CM(\omega)$$

Theorem 6. *The core of a finite coalitional game Γ over N and Ω is non-empty iff $\Gamma \models CNE$.*

Moving on to stable sets, the CGL formula $IMP(\omega)$ is true whenever ω is an imputation:

$$IMP(\omega) \equiv \langle N\rangle\omega \wedge \bigwedge_{\omega'\in\Omega}\bigwedge_{i\in N}(\langle\{i\}\rangle\omega' \to \omega \succeq_i \omega')$$

Next, $OBJ(\omega, \omega', C)$ expresses that outcome ω is an C-s-objection to outcome ω', when both ω and ω' are imputations:

$$OBJ(\omega, \omega', C) \equiv \Big(\bigwedge_{i\in C}\omega \succ_i \omega'\Big) \wedge \bigvee_{\omega''\in\Omega}\Big(\langle C\rangle\omega'' \wedge \bigwedge_{i\in C}\omega'' \succeq_i \omega\Big)$$

Given a set of outcomes $Y \subseteq \Omega$, the CGL formula $STABLE(Y)$ expresses the fact that Y is a stable set:

$$\begin{aligned}
&STABLE(Y) \equiv \\
&\quad \bigwedge_{\omega\in Y} IMP(\omega) \\
&\quad \wedge\Big(\bigwedge_{\omega\in Y}\bigwedge_{C\subseteq N}\bigwedge_{\omega'\in Y}\neg OBJ(\omega', \omega, C)\Big) \\
&\quad \wedge\Big(\bigwedge_{\omega\in\Omega\setminus Y} IMP(\omega) \to \Big(\bigvee_{C\subseteq N}\bigvee_{\omega'\in Y} OBJ(\omega', \omega, C)\Big)\Big)
\end{aligned}$$

Theorem 7. *Y is a stable set of a finite coalitional game Γ over N and Ω iff $\Gamma \models STABLE(Y)$.*

Proof. Given a finite coalitional game, let $\mathcal{I}$ denote the set of all imputations. First, we argue that $IMP(\omega)$ and $OBJ(\omega, \omega', C)$ have the correct meaning. Every $\omega \in Y$ is an imputation iff $\omega \in V(N)$ (feasibility) and $\omega \succeq_i \omega'$ for all i and $\omega' \in V(\{i\})$ which is exactly when $IMP(\omega)$ holds. If $\omega, \omega' \in \mathcal{I}$ and $C \subseteq N$, ω is a C-s-objection to ω' iff $\omega \succ_i \omega'$ for every $i \in C$ and there is a $\omega'' \in V(C)$ such that $\omega'' \succeq_i \omega$ for every $i \in C$, which is exactly when $OBJ(\omega, \omega', C)$ holds.

For the main proof, let $Y \subseteq \Omega$. If there is an ω in Y which is not an imputation, Y is not a stable set and $IMP(\omega)$ is not true and we are done, so assume that Y is a set of imputations. Let

$$\hat{Y} = \{\omega \in \mathcal{I}\colon \text{ there is no s-objection to } \omega \text{ in } Y\}$$

It is easy to see that Y is a stable set iff $Y = \hat{Y}$. We argue that the second and third main conjuncts of the formula $STABLE(Y)$ is true whenever $Y \subseteq \hat{Y}$ and $\hat{Y} \subseteq Y$ hold, respectively, and the theorem follows (the first conjunct is true under the assumption that Y are imputations). The second conjunct is true exactly when for every member of Y there is no C-s-objection to ω in Y for any C, which is exactly when $Y \subseteq \hat{Y}$ holds. The third conjunct is true iff every imputation which is not in Y has an s-objection in Y or, contrapositively, that every imputation which does not have an s-objection in Y is included in Y which is the same as $\hat{Y} \subseteq Y$. □

Existence of a stable set can then be expressed as:

$$ES \equiv \bigvee_{Y \subseteq \Omega} STABLE(Y)$$

Corollary 1. *A finite coalitional game Γ over N and Ω has a stable set iff $\Gamma \models ES$.*

Finally, we consider the bargaining set. The CGL formula $OBJB(\omega', C, \omega)$ means that (ω', C) is an b-objection of any $i \in C$ against any $j \notin C$ to ω.

$$OBJB(\omega', C, \omega) \equiv \langle C \rangle \omega' \wedge \bigwedge_{k \in C} \omega' \succ_k \omega$$

$ECO(\omega', C, i, j, \omega)$ means that there exists a b-counterobjection to the b-objection (ω', C) of i against j to ω.

$$ECO(\omega', C, i, j, \omega) \equiv \bigvee_{v \in \Omega} \bigvee_{D' \subseteq N \setminus \{i\}} \left(\langle D' \cup \{j\} \rangle v \wedge \left(\left(\bigwedge_{k \in (D' \cup \{j\}) \setminus C} v \succeq_k \omega \right) \wedge \left(\bigwedge_{k \in (D' \cup \{j\}) \cap C} v \succeq_k \omega' \right) \right) \right)$$

$INBARG(\omega)$ means that outcome $\omega \in \Omega$ is in the bargaining set:

$$INBARG(\omega) \equiv IMP(\omega) \wedge \bigwedge_{C \subseteq N} \bigwedge_{i \in C} \bigwedge_{j \in N \setminus C} \bigwedge_{\omega' \in \Omega} \left[OBJB(\omega', C, \omega) \rightarrow ECO(\omega', C, i, j, \omega) \right]$$

Theorem 8. *ω is a member of the bargaining set of a finite coalitional game Γ over N and Ω iff $\Gamma \models INBARG(\omega)$.*

Proof. It is easy to see that when ω is an imputation, C is a coalition, $i \in C$, $j \notin C$ and ω' an outcome, $\Gamma \models OBJB(\omega', C, \omega)$ iff (ω', C) is an b-objection of i against j to ω. To see that there exist a b-counterobjection to the b-objection (ω', C) of i against j to ω iff $\Gamma \models ECO(\omega', C, i, j, \omega)$, observe that (v, D) is a b-counterobjection iff the disjunct given by v and $D' = D \setminus \{j\}$ is true. The theorem follows immediately. □

We can now define $BS(Y)$, $Y \subseteq \Omega$, to express the fact that Y is the bargaining set.

$$BS(Y) \equiv \bigwedge_{\omega \in Y} INBARG(\omega) \wedge \bigwedge_{\omega \in \Omega \setminus Y} \neg INBARG(\omega)$$

Corollary 2. *Y is the bargaining set of a finite coalitional game Γ over N and Ω iff $\Gamma \models BS(Y)$.*

3.6. Proof examples

We briefly illustrate the proof system by formally deducing some well known properties of coalitional games.

Example 4.

(1) Every member of the core is an imputation:

$$CM(\omega) \rightarrow IMP(\omega)$$

	Let $i \in N$, $\omega' \in \Omega$	
1	$CM(\omega) \to (\langle i\rangle\omega' \to \neg\omega' \succ_i \omega)$	*Taut*
2	$(CM(\omega) \wedge \langle i\rangle\omega') \to \neg(\omega' \succeq_i \omega \wedge \neg\omega \succeq_i \omega')$	1, *Taut*
3	$(CM(\omega) \wedge \langle i\rangle\omega') \to (\omega' \succeq_i \omega \to \omega \succeq_i \omega')$	2, *Taut*
4	$\neg\omega' \succeq_i \omega \to \omega \succeq_i \omega'$	*Lin*, *Taut*
5	$CM(\omega) \wedge \langle i\rangle\omega' \to \omega \succeq_i \omega'$	3, 4, *Taut*
6	$CM(\omega) \to (\langle i\rangle\omega' \to \omega \succeq_i \omega')$	5, *Taut*
7	$CM(\omega) \to \bigwedge_{\omega' \in \Omega} \bigwedge_{i \in N}(\langle i\rangle\omega' \to \omega \succeq_i \omega')$	Rep. 1–6 for any $\omega' \in \Omega$, $i \in N$, *Taut*
8	$CM(\omega) \to \langle N\rangle\omega$	*Taut*
9	$CM(\omega) \to IMP(\omega)$	7, 8, *Taut*

Fig. 1. Every member of the core is an imputation: a deductive proof.

	Let $\omega' \in Y$ and $C \subseteq N$	
	Let $\omega'' \in \Omega$	
1	$(CM(\omega) \wedge \langle C\rangle\omega'') \to \bigvee_{i \in C} \neg\omega'' \succ_i \omega$	*Taut*
2	$OBJ(\omega', \omega, C) \to \bigwedge_{i \in C} \omega' \succ_i \omega$	*Taut*
	Let $i \in C$	
3	$(\omega' \succ_i \omega \wedge \omega'' \succeq_i \omega') \to \omega'' \succeq_i \omega$	*Trans* + *Taut*
4	$(\omega'' \succeq_i \omega' \wedge \omega \succeq_i \omega'') \to \omega \succeq_i \omega'$	*Trans*
5	$(\neg\omega \succeq_i \omega' \wedge \omega'' \succeq_i \omega') \to \neg\omega \succeq_i \omega''$	4, *Taut*
6	$(\omega' \succ_i \omega \wedge \omega'' \succeq_i \omega') \to \neg\omega \succeq_i \omega''$	5, *Taut*
7	$(\omega' \succ_i \omega \wedge \omega'' \succeq_i \omega') \to \omega'' \succ_i \omega$	3, 6, *Taut*
8	$(\neg\omega'' \succ_i \omega \wedge \omega' \succ_i \omega) \to \neg\omega'' \succeq_i \omega'$	7, *Taut*
9	$(\bigvee_{i \in C} \neg\omega'' \succ_i \omega \wedge \bigwedge_{i \in C} \omega' \succ_i \omega) \to \bigvee_{i \in C} \neg\omega'' \succeq_i \omega'$	Rep. 3–8 for any $i \in C$
10	$(CM(\omega) \wedge \langle C\rangle\omega'' \wedge OBJ(\omega', \omega, C)) \to \bigvee_{i \in C} \neg\omega'' \succeq_i \omega'$	1, 2, 9, *Taut*
11	$(OBJ(\omega', \omega, C) \wedge CM(\omega)) \to (\langle C\rangle\omega'' \to \bigvee_{i \in C} \neg\omega'' \succeq_i \omega')$	10, *Taut*
12	$(OBJ(\omega', \omega, C) \wedge CM(\omega)) \to \bigwedge_{\omega'' \in \Omega}(\langle C\rangle\omega'' \to \bigvee_{i \in C} \neg\omega'' \succeq_i \omega')$	Rep. 1–11 for any $\omega'' \in \Omega$
13	$(OBJ(\omega', \omega, C) \wedge CM(\omega)) \to \neg \bigvee_{\omega'' \in \Omega}(\langle C\rangle\omega'' \wedge \bigwedge_{i \in C} \omega'' \succeq_i \omega')$	12, *Taut*
14	$OBJ(\omega', \omega, C) \to \bigvee_{\omega'' \in \Omega}(\langle C\rangle\omega'' \wedge \bigwedge_{i \in C} \omega'' \succeq_i \omega')$	Def. of $OBJ(\cdot)$, *Taut*
15	$(OBJ(\omega', \omega, C) \wedge CM(\omega)) \to \bot$	13, 14, *Taut*
16	$CM(\omega) \to \neg OBJ(\omega', \omega, C)$	15, *Taut*
17	$CM(\omega) \to \neg \bigvee_{C \subseteq N} \bigvee_{\omega' \in Y} OBJ(\omega', \omega, C)$	Rep. 1–16 for any $\omega' \in Y, C \subseteq N$
18	$CM(\omega) \to IMP(\omega)$	The first part
19	$CM(\omega) \to \neg(IMP(\omega) \to \bigvee_{C \subseteq N} \bigvee_{\omega' \in Y} OBJ(\omega', \omega, C))$	17, 18, *Taut*
20	$CM(\omega) \to \neg \bigwedge_{\omega \in \Omega \setminus Y}(IMP(\omega) \to \bigvee_{C \subseteq N} \bigvee_{\omega' \in Y} OBJ(\omega', \omega, C))$	19, $\omega \in \Omega \setminus Y$, *Taut*
21	$CM(\omega) \to \neg STABLE(Y)$	20, *Taut*

Fig. 2. The core is a subset of any stable set: a deductive proof.

(2) The core is a subset of any stable set:

$$CM(\omega) \to \neg STABLE(Y) \quad \text{whenever } \omega \notin Y$$

Proof. The first part of the proof is shown in Fig. 1, the second in Fig. 2. □

4. A modal logic for coalitional games

In this section we study the logic of coalitional games from an alternative perspective: modal logic. We introduce *Modal Coalitional Game Logic* (MCGL), a modal logic interpreted in coalitional games. The language is similar to the language of CGL—in particular one of the main constructs of the language is an operator $\langle C\rangle$ for each coalition. The meaning of this operator is however radically different.

After we have introduced the language and its interpretation, introduced an axiomatic system, and shown that the system is sound and complete, we show how the logic can be used to express concepts like the core for *general* (not necessarily finite) coalitional games. The relationship between CGL and MCGL is studied in detail in Section 5.

4.1. MCGL*: Syntax and semantics*

The language of MCGL is interpreted as statements about an outcome in a coalitional game. One of the main constructs is modalities $\langle C\rangle$ for each coalition C. A formula of the form $\langle C\rangle\varphi$ is intended to mean that C *prefers* φ. In more detail, $\langle C\rangle\varphi$, interpreted in outcome ω of coalitional game Γ, is intended to mean that there exists an outcome ω' (possibly different from ω), which is weakly preferred over ω by each agent in C and in which φ is true. Similar modalities are used by [16] in the context of non-cooperative game theory. Another main construct is atomic propositions of the form p_C, where C is a coalition. p_C is intended to mean that coalition C can choose the current outcome, i.e., that the outcome is C-feasible. So, for example, $\langle C\rangle p_C$ means that there exists a C-feasible outcome which is weakly preferred over the current outcome by each member of C. There are also operators $\langle C^s\rangle$ standing for *strict*, rather than weak, preference. Furthermore, there are *converse* operators $\langle C^c\rangle$ and $\langle C^{sc}\rangle$; $\langle C^c\rangle\varphi$ ($\langle C^{sc}\rangle\varphi$) is intended to mean that there is an outcome over which the current outcome is weakly (strictly) preferred and in which φ is true. Finally, there is an operator $\langle D\rangle$ such that $\langle D\rangle\varphi$ is intended to mean that φ is true in some outcome different from the current one. In addition to the atomic propositions p_C for each coalition C, the language is parameterised by a set Θ' of general primitive atomic proposition symbols. These stand for arbitrary propositions which can be true or false statements about an outcome in a game.

Let N be a set of agents, and let $\mathcal{C} = 2^N \setminus \emptyset$ be the coalitions (we henceforth assume that a coalition is never empty). The language if *Modal Coalitional Game Logic* (MCGL), denoted $\mathcal{L}^M(N, \{p_C\colon C \in \mathcal{C}\} \cup \Theta')$, where Θ' is a countably infinite set of primitive propositions, is defined as follows. Let

$$\Theta = \Theta' \cup \{p_C\colon C \in \mathcal{C}\}$$

be the atomic propositions in the language. The language is defined by the following grammar:

$$\varphi ::= p \mid \langle i\rangle\varphi \mid \langle i^s\rangle\varphi \mid \langle C\rangle\varphi \mid \langle C^s\rangle\varphi \mid \langle D\rangle\varphi \mid \langle i^c\rangle\varphi \mid \langle i^{sc}\rangle\varphi \mid \langle C^c\rangle\varphi \mid \langle C^{sc}\rangle\varphi \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2$$

where $p \in \Theta$, $C \in \mathcal{C}$, $i \in N$. Derived: $[\cdot], [\cdot^s], [D], [\cdot^c], [\cdot^{sc}]$ are the duals of $\langle\cdot\rangle, \langle\cdot^s\rangle, \langle D\rangle, \langle\cdot^c\rangle, \langle\cdot^{sc}\rangle$, respectively.

The following definitions will be useful:

$$\Xi = \{i, C, i^s, C^s, i^c, C^c, i^{sc}, C^{sc}\}$$

$$\Xi^{+D} = \Xi \cup \{D\}$$

$$\mathit{Diamonds} = \{\langle\xi\rangle\colon \xi \in \Xi^{+D}\}$$

$$\mathit{Boxes} = \{[\xi]\colon \xi \in \Xi^{+D}\}$$

Let $\Gamma = (N, \Omega, V, \sqsupseteq_1, \ldots, \sqsupseteq_m)$ be a coalitional game, and let π be a valuation of Θ' in Ω, i.e., $\pi(p) \subseteq \Omega$ for each $p \in \Theta'$. The fact that a formula φ is satisfied by the combination of Γ, π and an outcome $w \in \Omega$, denoted $\Gamma, \pi, w \models \varphi$, is defined recursively as follows:

- $\Gamma, \pi, w \models p_C$ iff $w \in V(C)$,
- $\Gamma, \pi, w \models p$ iff $w \in \pi(p)$, when $p \in \Theta'$,
- $\Gamma, \pi, w \models \langle i\rangle\varphi$ iff there is a v such that $v \sqsupseteq_i w$ and $\Gamma, \pi, v \models \varphi$,
- $\Gamma, \pi, w \models \langle i^s\rangle\varphi$ iff there is a v such that $v \sqsupseteq_i w$ and not $w \sqsupseteq_i v$ and $\Gamma, \pi, v \models \varphi$,
- $\Gamma, \pi, w \models \langle C\rangle\varphi$ iff there is a v such that for every $i \in C$, $v \sqsupseteq_i w$, and $\Gamma, \pi, v \models \varphi$,
- $\Gamma, \pi, w \models \langle C^s\rangle\varphi$ iff there is a v such that for every $i \in C$, $v \sqsupseteq_i w$ and not $w \sqsupseteq_i v$, and $\Gamma, \pi, v \models \varphi$,
- $\Gamma, \pi, w \models \langle D\rangle\varphi$ iff $\Gamma, \pi, v \models \varphi$ for some $v \neq w$,
- $\Gamma, \pi, w \models \langle i^c\rangle\varphi$ iff there is a v such that $w \sqsupseteq_i v$ and $\Gamma, \pi, v \models \varphi$,
- $\Gamma, \pi, w \models \langle i^{sc}\rangle\varphi$ iff there is a v such that $w \sqsupseteq_i v$ and not $v \sqsupseteq_i w$ and $\Gamma, \pi, v \models \varphi$,
- $\Gamma, \pi, w \models \langle C^c\rangle\varphi$ iff there is a v such that for every $i \in C$, $w \sqsupseteq_i v$, and $\Gamma, \pi, v \models \varphi$,
- $\Gamma, \pi, w \models \langle C^{sc}\rangle\varphi$ iff there is a v such that for every $i \in C$, $w \sqsupseteq_i v$ and not $v \sqsupseteq_i w$, and $\Gamma, \pi, v \models \varphi$.

We write $\Gamma, w \models \varphi$ for the fact that $\Gamma, \pi, w \models \varphi$ for arbitrary π. Note that when φ have no occurrences of any $p \in \Theta'$, $\Gamma, w \models \varphi$ iff $\Gamma, \pi, w \models \varphi$ for any π. Furthermore, we write $\Gamma \models \varphi$ whenever $\Gamma, w \models \varphi$ for any w. Thus, we have an interpretation of our formulae as statements about outcomes in coalitional games, and about coalitional games themselves, respectively.

Example 5. Let Γ be the Dinner Game from Example 1. We have the following (ref. also Example 3):

- $\Gamma, I_{12} \models p_{\{1,2\}} \wedge \neg p_1$. Agents 1 and 2 can choose to go for Indian together, but 1 cannot choose on his own that they go together.
- $\Gamma, I_{123} \models \neg p_1 \wedge \neg\langle D\rangle p_1$. Agent 1 cannot choose any outcome on his own.
- $\Gamma, I_{12} \models \langle 1^s\rangle p_{\{1,2,3\}}$. The agents can choose something which for 1 is better than going for Indian together with 2.
- $\Gamma, C_{123} \models (\neg p_{\{1,3\}}) \wedge \langle\{1,3\}^s\rangle p_{\{1,3\}}$. Agents 1 and 3 cannot choose that all agents go for Chinese, but they can choose something which is strictly better for both of them (e.g., going for Indian on their own).
- $\Gamma, I_{123} \models [1^s]\bot$. Agent 1 prefers nothing better than having Indian with his two friends.
- $\Gamma, I_{123} \models \langle 1^{sc}\rangle p_{\{2,3\}}$. If the three agents go for Indian, there is some option which is worse for 1 which 2 and 3 could choose.

Table 2
The logic MCGL

Taut	φ	where φ is an instance of a prop. tautology
$K(\Box)$	$\Box(p \to q) \to (\Box p \to \Box q)$	$\Box \in Boxes$
T	$[i]p \to p$	
4	$[i]p \to [i][i]p$	
$Converse_1(\xi)$	$p \to [\xi]\langle \xi^c \rangle p$	$\xi \in \Xi$
$Converse_2(\xi)$	$p \to [\xi^c]\langle \xi \rangle p$	$\xi \in \Xi$
$Trichotomy$	$(p \wedge [i]q) \to [D](q \vee p \vee \langle i \rangle p)$	
D_1	$p \to [D]\langle D \rangle p$	*symmetry*
D_2	$\Diamond_1 \cdots \Diamond_k p \to (p \vee \langle D \rangle p)$	$\Diamond_i \in Diamonds$
$Strict_1$	$p \wedge \langle i \rangle (q \wedge [i]\neg p) \to \langle i^s \rangle q$	
$Strict_2$	$(p \wedge [D]\neg p \wedge \langle i^s \rangle q) \to \langle i \rangle (q \wedge \neg \langle i \rangle p)$	
$Strict_3$	$\langle i^s \rangle p \to \langle D \rangle p$	
$Intersect_1$	$((p \wedge [D]\neg p) \vee \langle D \rangle (p \wedge [D]\neg p)) \to (\bigwedge_{i \in C} \langle i \rangle p \to \langle C \rangle p)$	
$Intersect_2$	$((p \wedge [D]\neg p) \vee \langle D \rangle (p \wedge [D]\neg p)) \to (\bigwedge_{i \in C} \langle i^s \rangle p \to \langle C^s \rangle p)$	
$Intersect_3$	$\langle C \rangle p \to \langle i \rangle p$	$i \in C$
$Intersect_4$	$\langle C^s \rangle p \to \langle i^s \rangle p$	$i \in C$
$Nec(\Box)$	$\vdash \varphi \ \Rightarrow \vdash \Box \varphi$	$\Box \in Boxes$
$USub$	$\vdash \varphi \Rightarrow \vdash \psi$ where ψ is the result of uniformly replacing atomic propositions in φ with arbitrary formulae	
D-rule	$\vdash (p \wedge \neg \langle D \rangle p) \to \theta \Rightarrow \vdash \theta$	p not in θ
MP	$\vdash \varphi, \vdash \varphi \to \psi \ \Rightarrow \vdash \psi$	

4.2. Axiomatisation

In this section we construct a sound and complete axiomatisation by viewing the logic as a normal modal logic. In particular, the basic accessibility relations in this logic are the individual preference relations $\sqsupseteq_i$.

Let MCGL be the axiomatic system defined in Table 2.

Let us briefly reflect upon the axioms. Axiom $K(\Box)$ and rules $Nec(\Box)$ and $USub$ say that the logic is a normal modal logic. T and 4 say that each $\sqsupseteq_i$ is a pre-order. By *Trichonomy*, relation $\sqsupseteq_i$ moreover satisfies *completeness*. The $Converse_1(\xi)$ and $Converse_2(\xi)$ axioms ensure that the relation denoted by the $\langle \xi^c \rangle$ modality is the converse of the relation denoted by the $\langle \xi \rangle$ modality. The axioms for the D-operator, as well as the D-rule are relatively standard in modal logic (see [6]). The D-rule facilitates us to prove that $\vdash \langle i^s \rangle p \to \langle i \rangle p$, as one would, based on semantic considerations, expect. Also, our D_2 axiom generalises what is called *pseudo-transitivity* in [6], which is $\langle D \rangle \langle D \rangle p \to (p \vee \langle D \rangle p)$. The axioms $Strict_1$, $Strict_2$ and $Strict_3$ together ensure that the relation denoted by the $\langle i^s \rangle$ modality is the strict version of $\sqsupseteq_i$. Note that this implies that the former is irreflexive. Finally, taking $Intersect_k$ for $k = 1, 3$ guarantees that the relation denoted by the $\langle C \rangle$ modality is the intersection over all $i \in C$ of all $\sqsupseteq_i$, and for $k = 2, 4$ it ensures that the relation denoted by the $\langle C^s \rangle$ modality is the intersection over all $i \in C$ of the strict version of $\sqsupseteq_i$.

Examples of deductions in MCGL can be found in Section 4.4. The remainder of this section is concerned with proving that MCGL is sound and complete with respect to all coalitional games. The main result is found in Corollary 3.

4.2.1. Soundness and completeness

To make it more convenient to use standard modal logic tools, we will here interpret the logical language explicitly in a certain class of Kripke structures rather than in coalitional games. We show that MCGL is sound and complete with respect to this class, and it follows that it is sound and complete with respect to all coalitional games since these Kripke structures can be seen as coalitional games (Corollary 3 below).

Given a set of agents N and primitive propositions Θ', a model is a tuple:

$$M = \big(W, \{R_i\colon\ i \in N\}, \{R_i^s\colon\ i \in N\}, \{R_C\colon\ C \in \mathcal{C}\}, \{R_C^s\colon\ C \in \mathcal{C}\}, D, \{R_i^c\colon\ i \in N\}, \{R_i^{sc}\colon\ i \in N\}, \{R_C^c\colon\ C \in \mathcal{C}\}, \{R_C^{sc}\colon\ C \in \mathcal{C}\}, \pi\big)$$

where π is a valuation of $\Theta = \Theta' \cup \{p_C\colon\ C \in \mathcal{C}\}$ and

REFL $\forall_{i \in N} R_i$ is reflexive
TRANS $\forall_{i \in N} R_i$ is transitive
COMPL $\forall_{i \in N} R_i$ is complete
STRICT $\forall_{i \in N} R_i^s wu$ iff both $R_i wu$ and not $R_i uw$
DIFF $D = \{(w, u)\colon\ w \neq u\}$.
INTERSECTION $\forall_{C \in \mathcal{C}} R_C = \bigcap_{i \in C} R_i$
INTERSECTION-STRICT $\forall_{C \in \mathcal{C}} R_C^s = \bigcap_{i \in C} R_i^s$
CONVERSE Rwv iff $R^c vw$, for each relation $R \in \{R_i, R_i^s, R_C, R_C^s, D\}$

Defined this way, our representation of a model contain a lot of redundant information, but this definition makes it easy to use standard modal logic tools directly.

For the truth-definition, it is convenient to use the following notation.

Definition 5. For a diamond like $\langle C^s \rangle$, we call C^s its parameter. We use ξ for an arbitrary parameter. The set of parameters is Ξ. For every parameter $\xi \in \Xi$, we define its associated relation $\mathsf{Rel}(\xi)$ in a straightforward way, and, conversely, given an accessibility relation R, we define its parameter $\mathsf{Par}(R)$, such that $\mathsf{Rel}(\mathsf{Par}(R))$ is the identity function on accessibility relations, and $\mathsf{Par}(\mathsf{Rel}(\xi))$ being the identity function on parameters.

$\mathsf{Rel}(i) = R_i$	$\mathsf{Par}(R_i) = i$	
$\mathsf{Rel}(C) = R_C$	$\mathsf{Par}(R_C) = C$	
$\mathsf{Rel}(D) = D$	$\mathsf{Par}(D) = D$	
$\mathsf{Rel}(\xi^s) = \mathsf{Rel}(\xi)^s$	$\mathsf{Par}(R^s) = \mathsf{Par}(R)^s$	$\xi \in \{i, C\}$
$\mathsf{Rel}(\xi^c) = \mathsf{Rel}(\xi)^c$	$\mathsf{Par}(R^c) = \mathsf{Par}(R)^c$	$\xi \in \{i, C\}$
$\mathsf{Rel}(\xi^{sc}) = \mathsf{Rel}(\xi)^{sc}$	$\mathsf{Par}(R^{sc}) = \mathsf{Par}(R)^{sc}$	$\xi \in \{i, C\}$

The interpretation of a formula in a state w of a model M is defined as follows (Boolean connectives as usual), where we use the same satisfaction symbol $\models$ we used for the interpretation in games without risk of confusion:

- $M, w \models p$ iff $w \in \pi(p)$,
- $M, w \models \langle \xi \rangle \varphi$ iff there is a v such that $\mathsf{Rel}(\xi)wv$ and $M, v \models \varphi$, for all $\xi \in \Xi$.

A frame is a tuple $F = (W, \{R_i\colon\ i \in N\})$. A model M is based on a frame F if it is the result of extending the frame with some valuation function, i.e., $M = (F, \pi)$ for some π. $M \models \varphi$ means that $M, w \models \varphi$ for any state w in M, $F \models \varphi$ means that $M \models \varphi$ for any model M based on the frame F. Let us call the set of all models just defined $\mathcal{MCGL}$. Then $\models \varphi$ is shorthand for $\mathcal{MCGL} \models \varphi$, which in its turn means that for any model $M \in \mathcal{MCGL}$, we have $M \models \varphi$.

The models carry a lot of dependencies between different accessibility relations. It is now the challenge of the soundness and completeness proof that we can characterise these dependencies in our object language.

Theorem 9 *(Soundness). MCGL is sound w.r.t. the models:* $\forall\varphi\colon\ \mathsf{MCGL} \vdash \varphi \Rightarrow \mathcal{MCGL} \models \varphi$.

Proof. First, consider *Trichotomy*. Let $M, w \models p \wedge [i]q$. Now take a world $u \neq w$, we have to show that $q \vee p \vee \langle i \rangle p$ is true in u. Since R_i is complete, we either have $R_i wu$ (in which case $M, u \models q$) or else $R_i uw$ (which gives $M, u \models \langle i \rangle p$). Next, consider *Strict*$_1$. Suppose $M, w \models p \wedge \langle i \rangle (q \wedge [i]\neg p)$. Then for some u, we have $M, u \models q$ and $R_i wu$ but not $R_i uw$. By STRICT, then $R_i^s wu$, and hence $M, w \models \langle i^s \rangle q$. For *Strict*$_2$, suppose $M, w \models (p \wedge [D]\neg p \wedge \langle i^s \rangle q)$. This implies that w is the only p-world and also that for some u with $R_i wu$, $M, u \models q$. By STRICT, this means that we have $R_i wu$ but not $R_i uw$. Hence, in u, we also have $\neg\langle i \rangle p$. For *Strict*$_3$, suppose $M, w \models \langle i^s \rangle p$. There is a u such that $R_i^s wu$ and $M, u \models p$. Since R_i^s is strict, it is impossible that $w = u$. Thus $M, w \models \langle D \rangle p$. Consider the intersection axioms. It is obvious that *Intersect*$_3$ and *Intersect*$_4$ are valid. For the two others, note that $((p \wedge [D]\neg p) \vee \langle D \rangle (p \wedge [D]\neg p))$ is true in a world w if and only if there is a unique world in which p is true. Let us abbreviate it to $\langle \exists! \rangle p$. Then *Intersect*$_1$ becomes $\langle \exists! \rangle p \rightarrow (\bigwedge_{i \in C} \langle i \rangle p \rightarrow \langle C \rangle p)$. Now suppose $M, w \models \langle \exists! \rangle p$. Furthermore, suppose $M, w \models \bigwedge_{i \in C} \langle i \rangle p$. This implies that there are worlds u_i for which $R_i wu_i$ and $M, u_i \models p$ $(i \in C)$. Since there is a unique p-world, all u_i must be the same. By INTERSECTION, we have $M, w \models \langle C \rangle p$. The argument for *Intersect*$_2$ is similar. As for the D-rule, it pays off to consider the contrapositive: suppose that θ is not valid, i.e., $\neg\theta$ is satisfiable. This means that for some M and w, we have $M, w \models \neg\theta$. Since p does not occur in θ, $\neg\theta$ remains true in M, w if we change the valuation only for p, in fact it is not hard to see that we can change it in such a way that we obtain a model M' for which $M', w \models p \wedge [D]\neg p \wedge \neg\theta$. We leave validity of the other axioms, and preservation of validity of the rules, to the reader. □

The rest of this section is devoted to completeness of MCGL. Completeness of a logic LOG with respect to a semantics $\mathcal{SEM}$ amounts to showing that for any φ, if $\mathcal{SEM} \models \varphi$, then $\mathsf{LOG} \vdash \varphi$. Using contraposition, this is the same as proving that for any LOG-consistent formula φ, there is some $\mathcal{SEM}$ model M with a state s such that $M, s \models \varphi$. A common approach to prove the latter is by constructing a canonical model $\mathsf{can}(M)$, in which the states $\mathsf{can}(W)$ are all maximal LOG-consistent sets, and for any $\Diamond$ in the language with associated relation R, one defines this $\mathsf{can}(R)$ to hold between two m.c. sets, i.e., $\mathsf{can}(R)\Sigma\Delta$ iff for any δ, if $\delta \in \Delta$, then $\Diamond\delta \in \Sigma$, or, alternatively, for any δ, if $\Box\delta \in \Sigma$, then $\delta \in \Delta$. By construction, one obtains $\mathsf{can}(M), \Phi \models \varphi$ for any $\varphi \in \Phi$, which demonstrates satisfiability of φ. However, to show that φ is $\mathcal{SEM}$-satisfiable, one has to show that $\mathsf{can}(M)$ is indeed a $\mathcal{SEM}$-model.

If the latter is not the case, one might still obtain completeness, in an indirect way. One way to proceed would be to massage the canonical model $\mathsf{can}(M)$ into a model M' that still satisfies φ at s, but that *is* now a member of $\mathcal{SEM}$. This would typically work if $\mathcal{SEM}$ has some property P that is not modally definable, and for which no canonical formula exists. The latter means that $\mathsf{can}(M)$ will not have property P, but the first might facilitate transforming $\mathsf{can}(M)$ into M' that does have the property P, while M' having the same validities as $\mathsf{can}(M)$. An example of such a property P would be INTERSECTION, say that $R_3 = R_1 \cap R_2$ (cf. [35]). The axiom $A : \Diamond_3 p \to (\Diamond_1 p \wedge \Diamond_2 p)$ ensures that $R_3 \subseteq (R_1 \cap R_2)$, but equality here is not definable, and indeed, the canonical model of the logic $\mathsf{K} + A$ does not satisfy it. However, one may apply a *copying technique* in the canonical model as follows: for every two states s and t such that R_1st, R_2st, but not R_3st, replace t by two copies t_1, t_2, and put R_1st_1 and R_2st_2. In the resulting model, exactly the same formulas are true, and also $R_3 = R_1 \cap R_2$ holds (see [35]).

However, if there are other relations or properties that need to be maintained or achieved at the same time, such a manipulation of the canonical model can become extremely difficult. For an example of this, we refer to [36], where the canonical model for a system with individual knowledge (with relation R_i), Everybody's knowledge (with R_E, the union of the R_i's), Common knowledge (R_C, the transitive closure of R_E) and Distributed Knowledge (R_D, the intersection of the R_i's) neither satisfies the desired property for R_C, nor for R_D, and, although there are well-known techniques to solve the problem for R_C and for R_D individually, those techniques applied here fix the problem for one relation, while ruining it for the other.

Rather than starting with the big canonical model for a logical system and then unravel it, duplicate worlds, or look at sub-models, [6] presents a 'step by step' technique, which we will follow here for our completeness proof. This technique is extremely useful in case one can go "back-and-forth" in a model, i.e., having a relation that allows one to look in *both directions* of the Kripke model. We will now first briefly sketch the procedure before the proof is given in detail below. Take a consistent formula φ. Following [6], we build a *network* $\mathcal{N}$ to satisfy it. Such a network is a set of states with relations between them, but we start small, building up the network gradually. As a start, we add φ to some type $\Lambda(s)$ (such a type collects all the formulas that should be true at s) of an initial state in an initial network $\mathcal{N}_1$. But if φ implies some $\langle\xi\rangle\psi$ formula, this $\mathcal{N}_1$ is not good enough of course, our network has a *defect*, and we need to extend the network with a node t with $\psi \in \Lambda(t)$ and mark, using a labelling d, that s and t are $\mathsf{Rel}(\xi)$ related. So, if some $\mathcal{N}_i$ has a defect, we either add a node, or an edge, or a formula to some type, to obtain a network $\mathcal{N}_j$ $(j > i)$ that does not have the defect. Special attention has to be paid to the D-operator: in the end, we have to make sure that the arcs in $\mathcal{N}$ that are labelled with D, can be interpreted as inequality. In particular, D must not be reflexive, and it must be 'almost universal': hold between *any* two different nodes. To this end, we make sure that the network that we build is *named*: for every node s that we allow in $\mathcal{N}_i$, we make sure that there is some formula $\psi \wedge \neg\langle D\rangle\psi \in \Lambda(s)$ in some network $\mathcal{N}_j$ $(j > i)$. The network that we are after is $\mathcal{N} = \bigcup_i \mathcal{N}_i$.

So don't we need the canonical model $\mathsf{can}(M)$ for MCGL at all? We *do*: in the network $\mathcal{N}$, every edge has a unique label, and this will never be sufficient to get all the properties of $\mathcal{MCGL}$. For instance, if $R^s st$, then also Dst should hold. So, the network can be considered as a graph where all arcs are added only for the reason to make diamond formulas true: we did not bother about keeping track of dependencies between arcs. All types in $\mathcal{N}$ can be shown to be m.c. sets. Those types now happen to represent exactly the information that we need: for the model M that we need to demonstrate the satisfiability of φ, we take the restriction of the canonical model $\mathsf{can}(M)$ for MCGL to all the types of $\mathcal{N}$. In other words, we keep the states from $\mathcal{N}$, but replace the labelled arcs with the canonical relations $\mathsf{can}(R)$. This guarantees for instance, that if we had s and t that were only related through R_i^s in $\mathcal{N}$, in the model M, we also impose $\mathsf{can}(R_i^s)st$, but we also obtain Dst, $\mathsf{can}(R_i)st$, $\mathsf{can}(R_i^{sc})ts$ and $\mathsf{can}(R_i^c)ts$. Details are found in the proof of the following theorem.

Theorem 10 *(Completeness).* MCGL *is complete w. r. t. the models:* $\forall\varphi$: $\mathcal{MCGL} \models \varphi \Rightarrow \mathsf{MCGL} \vdash \varphi$.

Proof. A *network* is a tuple $\mathcal{N} = (N, E, d, r, \Lambda)$, where (N, E) is a finite, undirected, connected and acyclic graph, d maps each edge $\{s, t\} \in E$ to a relation in the set $\{R_i, R_i^s, R_C, R_C^s, D\colon\ i \in N, C \in \mathcal{C}\}$, r maps each edge $\{s, t\} \in E$ to either s or t, and Λ labels each node in N with a finite set of formulae in $\mathcal{L}^M(N, \Theta)$.

Given a graph and a node s, let $E(s)$ denote the set of nodes adjacent to s. Let:

$$\langle st\rangle = \begin{cases} \langle i\rangle & d(\{s,t\}) = R_i \text{ and } r(\{s,t\}) = s \\ \langle i^c\rangle & d(\{s,t\}) = R_i \text{ and } r(\{s,t\}) = t \\ \langle i^s\rangle & d(\{s,t\}) = R_i^s \text{ and } r(\{s,t\}) = s \\ \langle i^{sc}\rangle & d(\{s,t\}) = R_i^s \text{ and } r(\{s,t\}) = t \\ \langle C\rangle & d(\{s,t\}) = R_C \text{ and } r(\{s,t\}) = s \\ \langle C^c\rangle & d(\{s,t\}) = R_C \text{ and } r(\{s,t\}) = t \\ \langle C^s\rangle & d(\{s,t\}) = R_C^s \text{ and } r(\{s,t\}) = s \\ \langle C^{sc}\rangle & d(\{s,t\}) = R_C^s \text{ and } r(\{s,t\}) = t \\ \langle D\rangle & d(\{s,t\}) = D \text{ and } r(\{s,t\}) = s \\ \langle D\rangle & d(\{s,t\}) = D \text{ and } r(\{s,t\}) = t \end{cases}$$

Let:

$$\Delta(\mathcal{N},s) = \bigwedge \Lambda(s) \wedge \bigwedge_{v\in E(s)} \langle sv\rangle \Phi(\mathcal{N},v,s)$$

$$\Phi(\mathcal{N},t,s) = \bigwedge \Lambda(t) \wedge \bigwedge_{s\neq v\in E(t)} \langle tv\rangle \Phi(\mathcal{N},v,t)$$

Note that $\Delta(\mathcal{N},s)$ is a (finite) formula for any (finite and acyclic) network.

We argue that $\Delta(\mathcal{N},s)$ is consistent iff $\Delta(\mathcal{N},t)$ is consistent, for any two nodes in any network $\mathcal{N}$. We show this for any $(s,t)\in E$; the property follows for arbitrary $s,t\in N$ by induction on the length of the least path from s to t ($\mathcal{N}$ is connected). Let $(s,t)\in E$, and first assume that $d(\{s,t\}) = R_i$ and $r(\{s,t\}) = s$. We have that $\Delta(\mathcal{N},s) = \bigwedge \Lambda(s) \wedge \bigwedge_{t\neq v\in E(s)} \langle sv\rangle \Phi(\mathcal{N},v,s) \wedge \langle st\rangle \Phi(\mathcal{N},t,s) = \langle i\rangle \Phi(\mathcal{N},t,s) \wedge \Phi(\mathcal{N},s,t)$. Likewise, we have that $\Delta(\mathcal{N},t) = \bigwedge \Lambda(t) \wedge \bigwedge_{s\neq v\in E(t)} \langle tv\rangle \Phi(\mathcal{N},v,t) \wedge \langle ts\rangle \Phi(\mathcal{N},s,t) = \langle i^c\rangle \Phi(\mathcal{N},s,t) \wedge \Phi(\mathcal{N},t,s)$. Axioms *Converse*$_1(i^s)$ and *Converse*$_2(i^s)$ ensure that $\langle i\rangle\alpha \wedge \beta$ is consistent iff $\langle i^c\rangle\beta \wedge \alpha$ is consistent, for any α,β. The argument in the case that $r(\{s,t\}) = t$ is symmetric, and the other cases $d(\{s,t\}) \in \{R_i^s, R_C, R_C^s, D\}$ are similar (in the case of the $\langle D\rangle$ operator, the axiom *D1* is used instead of *Converse*$_1$ and *Converse*$_2$). Thus, $\Delta(\mathcal{N},s)$ is consistent iff $\Delta(\mathcal{N},t)$ is consistent. We say that a network $\mathcal{N}$ is *coherent* if $\Delta(\mathcal{N},s)$ is consistent for any node in the network.

Possible defects are the following:

$D1(s,\varphi)$ where s is a node and φ a formula, and $\varphi \notin \Lambda(s)$ and $\neg\varphi \notin \Lambda(s)$,
$D2(s)$ there is no formula φ such that $\varphi \wedge \neg\langle D\rangle\varphi \in \Lambda(s)$,
$D3(s,\langle\xi\rangle\varphi)$ $(\xi \in \{i, C, i^s, C^s, D\})$ where s is a node and $\langle\xi\rangle\varphi \in \Lambda(s)$ and for all $(s,t)\in E$ such that $d(\{s,t\}) = \mathsf{Rel}(\xi)$ and $r(\{s,t\}) = s$ it is the case that $\varphi \notin \Lambda(t)$,
$D4(s,\langle\xi^c\rangle\varphi)$ $(\xi \in \{i, C, i^s, C^s\})$ where s is a node and $\langle\xi^c\rangle\varphi \in \Lambda(s)$ and for all $(s,t)\in E$ such that $d(\{s,t\}) = \mathsf{Rel}(\xi)$ and $r(\{s,t\}) = t$ it is the case that $\varphi \notin \Lambda(t)$.

A network $\mathcal{N}' = (N', E', d', r', \Lambda')$ *extends* a network $\mathcal{N} = (N, E, d, r, \Lambda)$ if $N \subseteq N'$, $\Lambda(s) \subseteq \Lambda'(s)$ for each $s \in N$, and E', d', r' are E, d, r, respectively, restricted to N.

We now show that for any defect in a coherent network $\mathcal{N}$, there is a coherent network $\mathcal{N}'$ extending $\mathcal{N}$ lacking that effect. Let Y be a countably infinite set; we will use the elements of Y as states when we need to add new states to the networks in order to repair defect $D3$.

$D1(s,\varphi)$ Since $\Delta(\mathcal{N},s)$ is consistent, either $\Delta(\mathcal{N},s)\wedge\varphi$ or $\Delta(\mathcal{N},s)\wedge\neg\varphi$ is consistent. W.l.o.g. assume the first. Let $\mathcal{N}'$ be identical to $\mathcal{N}$, except that $\Lambda'(s) = \Lambda(s) \cup \{\varphi\}$ ($\Lambda'(t) = \Lambda(t)$ for $t \neq s$). Clearly, $\mathcal{N}'$ is a network, it extends $\mathcal{N}$, and it lacks the defect $D1(s,\varphi)$. It is easy to see that $\Delta(\mathcal{N}',s) = \Delta(\mathcal{N},s)\wedge\varphi$ which is consistent, so $\mathcal{N}'$ is coherent.

$D2(s)$ Let p be an atomic proposition in Θ not occurring in $\Delta(\mathcal{N},s)$ (it exists since $\Theta = \Theta' \cup \{p_C\colon C \in \mathcal{C}\}$ is infinite—Θ' was assumed to be countably infinite). The following is an alternative statement of the D-rule:

If Φ is consistent and does not contain p
$\Downarrow$
$(p \wedge \neg\langle D\rangle p) \wedge \Phi$ is consistent

Since $\Delta(\mathcal{N},s)$ is consistent, it follows that $\Delta(\mathcal{N},s) \wedge (p \wedge \neg\langle D\rangle p)$ is consistent. We define $\mathcal{N}'$ as in the $D1$ case, by setting $\Lambda'(s) = \Lambda(s) \cup \{p \wedge \neg\langle D\rangle p\}$. $\mathcal{N}'$ is a coherent network extending $\mathcal{N}$ and lacking the $D2(s)$ defect, by the same argument as in the $D1$ case.

$D3(s,\langle\xi\rangle\varphi)$ We define $\mathcal{N}'$ as follows:

- $N' = N \cup \{t\}$ for some $t \in Y \setminus N$,
- $E' = E \cup \{\{s,t\}\}$,
- $d' = d \cup \{\{s,t\} \mapsto \mathsf{Rel}(\xi)\}$,
- $r' = r \cup \{\{s,t\} \mapsto s\}$,
- $\Lambda' = \Lambda \cup \{t \mapsto \{\varphi\}\}$.

Clearly, $\mathcal{N}'$ is a network extending $\mathcal{N}$ lacking the $D4(s,\langle\xi\rangle\varphi)$ defect. It is easy to see that $\Delta(\mathcal{N}',s) = \Delta(\mathcal{N},s) \wedge \langle\xi\rangle\varphi$. But since $\langle\xi\rangle\varphi \in \Lambda(s)$, it already is a conjunct of $\Delta(\mathcal{N},s)$. Thus, $\mathcal{N}'$ is coherent.

$D4(s,\langle\xi^c\rangle\varphi)$ We define $\mathcal{N}'$ as follows:

- $N' = N \cup \{t\}$ for some $t \in Y \setminus N$,
- $E' = E \cup \{\{s,t\}\}$,
- $d' = d \cup \{\{s,t\} \mapsto \mathsf{Rel}(\xi)\}$,
- $r' = r \cup \{\{s,t\} \mapsto t\}$,
- $\Lambda' = \Lambda \cup \{t \mapsto \{\varphi\}\}$.

Clearly, $\mathcal{N}'$ is a network extending $\mathcal{N}$ lacking the $D3(s,\langle\xi^c\rangle\varphi)$ defect. It is easy to see that $\Delta(\mathcal{N}',s) = \Delta(\mathcal{N},s) \wedge \langle\xi^c\rangle\varphi$. But since $\langle\xi^c\rangle\varphi \in \Lambda(s)$, it already is a conjunct of $\Delta(\mathcal{N},s)$. Thus, $\mathcal{N}'$ is coherent.

Now, let $\hat\varphi \in \mathcal{L}^M(N,\Theta)$ be a consistent formula—we are going to show that it is satisfiable.

Fix an enumeration of possible defects of a network whose nodes are included in the set Y, i.e., an enumeration of the set $\{D1(s,\varphi), D2(s), D3(s,\langle D\rangle\varphi), D3(s,\langle\xi\rangle\varphi), D4(s,\langle\xi^c\rangle\varphi)\colon \xi\in\{i, C, i^s, C^s\colon i\in N, C\in\mathcal{C}\}, \varphi\in\mathcal{L}^M(N,\Theta), s\in Y\}$. We define a network $\mathcal{N}_i=(N_i,E_i,d_i,r_i,\Lambda_i)$ for each natural number i as follows.

- $\mathcal{N}_0$ has a single node $y\in Y$ labelled with $\{\hat{\varphi}\}$. Clearly, $\mathcal{N}_0$ is coherent.
- When $n>0$, $\mathcal{N}_{n+1}$ is the (coherent) network obtained by repairing the least (according to the enumeration) defect, by the rules given above.

Note that $\mathcal{N}_j$ extends $\mathcal{N}_i$ when $i<j$. Also observe that a repaired defect will never be reintroduced, i.e., if $\mathcal{N}_{n+1}$ is obtained from $\mathcal{N}_n$ by repairing defect D, then for every $m\geqslant n+1$ the network $\mathcal{N}_m$ lacks the defect D. Importantly, it follows that for any i, for any defect of $\mathcal{N}_i$ there is a $j>i$ such that $\mathcal{N}_j$ lacks that defect.

Let $\mathcal{N}=(N,E,d,r,\Lambda)$ be defined as follows: $N=\bigcup_{i\in\mathbb{N}}N_i$; $E=\bigcup_{i\in\mathbb{N}}E_i$; $d=\bigcup_{i\in\mathbb{N}}d_i$; $r=\bigcup_{i\in\mathbb{N}}r_i$; and $\Lambda(s)=\bigcup\{\Lambda_i(s)\colon i\in\mathbb{N}, s\in N_i\}$.

We argue that for any $s\in N$, $\Lambda(s)$ is a maximal consistent set (MCS) in the language $\mathcal{L}^M(N,\Theta)$. For maximality, assume that there is an $s\in N$ and formula $\varphi\in\mathcal{L}^M(N,\Theta)$ such that $\varphi,\neg\varphi\notin\Lambda(s)$. Let i be such that $s\in N_i$. Clearly, $\varphi,\neg\varphi\notin\Lambda_i(s)$. But then the network $\mathcal{N}_i$ has the defect $D1(s,\varphi)$. By the earlier argument, this defect will have been repaired in some $\mathcal{N}_j$ with $j>i$, i.e., $\varphi\in\Lambda_j(s)$ or $\neg\varphi\in\Lambda_j(s)$. But then $\varphi\in\Lambda(s)$ or $\neg\varphi\in\Lambda(s)$. Assume that $\Lambda(s)$ is not consistent. Then there are $\varphi_1,\dots,\varphi_k\in\Lambda(s)$ such that $\varphi_1\wedge\cdots\wedge\varphi_k$ is inconsistent. By construction, there is a j such that $\varphi_1,\dots,\varphi_k\in\Lambda_j(s)$. But then $\Delta(\mathcal{N}_k,s)$ is inconsistent, contradicting the coherency of $\mathcal{N}_k$. Thus, each $\Lambda(s)$ is an MCS. Let

$$W=\{\Lambda(s)\colon s\in N\}$$

We will now define a sub-model of the canonical model of the logic, by restricting the states to the MCSs W. Let $\mathsf{can}(M)=(\mathsf{can}(W),\{\mathsf{can}(R_i)\colon i\in N\},\{\mathsf{can}(R_i^s)\colon i\in N\},\{\mathsf{can}(R_C)\colon C\in\mathcal{C}\},\{\mathsf{can}(R_C^s)\colon C\in\mathcal{C}\},\mathsf{can}(D),\{\mathsf{can}(R_i^c)\colon i\in N\},\{\mathsf{can}(R_i^{sc})\colon i\in N\},\{\mathsf{can}(R_C^c)\colon C\in\mathcal{C}\},\{\mathsf{can}(R_C^{sc})\colon C\in\mathcal{C}\},\mathsf{can}(\pi))$ be the canonical model for MCGL: $\mathsf{can}(W)$ is the set of all MCSs over the language $\mathcal{L}^M(N,\Theta)$; $\mathsf{can}(R)wv$ iff $\psi\in v$ implies that $\Diamond\psi\in w$ for any ψ and any relation $\mathsf{can}(R)$ with corresponding diamond $\Diamond$; $\mathsf{can}(\pi)(p)=\{w\in\mathsf{can}(W)\colon p\in w\}$ for each $p\in\Theta$.[9] Let $M=(W,\{R_i\colon i\in N\},\{R_i^s\colon i\in N\},\{R_C\colon C\in\mathcal{C}\},\{R_C^s\colon C\in\mathcal{C}\},D,\{R_i^c\colon i\in N\},\{R_i^{sc}\colon i\in N\},\{R_C^c\colon C\in\mathcal{C}\},\{R_C^{sc}\colon C\in\mathcal{C}\},\pi)$ be the model obtained by replacing $\mathsf{can}(W)$ with W as defined above in $\mathsf{can}(M)$, as well as restricting each relation to W and restricting the valuation π to W.

We argue that for any $\Diamond\in\{\langle i\rangle,\langle i^s\rangle,\langle C\rangle,\langle C^s\rangle,\langle D\rangle,\langle i^c\rangle,\langle i^{sc}\rangle,\langle C^c\rangle,\langle C^{sc}\rangle\colon i\in N, C\in\mathcal{C}\}$, we have that

$$\forall(\psi\in\mathcal{L}^M(N,\Theta)\forall(\Gamma\in W)(\Diamond\psi\in\Gamma\Rightarrow\exists(\Delta\in W)\Gamma\,\mathsf{can}(R)\Delta\text{ and }\psi\in\Delta)$$

where $\mathsf{can}(R)$ is the canonical relation interpreting the diamond $\Diamond$. We show this for $\Diamond=\langle i\rangle$; the argument is analogous in the other cases (including the converses). Let $\Gamma\in W$ and let $\langle i\rangle\psi\in\Gamma$. Let $s\in N$ be such that $\Gamma=\Lambda(s)$, and let i be such that $\langle i\rangle\psi\in\Lambda_i(s)$. By the construction there is a $j\geqslant i$ and a $t\in N_j$ such that $\{s,t\}\in E_j$, $d(\{s,t\})=R_i$, $r(\{s,t\})=s$, and $\psi\in\Lambda_j(t)$. Thus, $\psi\in\Lambda(t)$. By taking $\Delta=\Lambda(t)$, it now remains to show that $\Lambda(s)\mathsf{can}(R_i)\Lambda(t)$. Assume otherwise, i.e., that there is a $\gamma\in\Lambda(t)$ such that $\langle i\rangle\gamma\notin\Lambda(s)$. Then $\neg\langle i\rangle\gamma\in\Lambda(s)$. Let k be such that $\gamma\in\Lambda_k(t)$, $\neg\langle i\rangle\gamma\in\Lambda_k(s)$, $\{s,t\}\in E_k$, $d_k(\{s,t\})=R_i$ and $r_k(\{s,t\})=s$. In the construction of $\Delta(\mathcal{N}_k,s)$, $\langle st\rangle=\langle i\rangle$, and thus includes $\langle i\rangle\gamma$ as a conjunct. But $\Delta(\mathcal{N}_k,s)$ also contains $\neg\langle i\rangle\gamma$ as a conjunct. This contradicts the coherence of $\mathcal{N}_k$, and shows that the assumption was wrong.

A truth lemma can now be shown:

$$M,\Gamma\models\psi\quad\Leftrightarrow\quad\psi\in\Gamma$$

for any $\Gamma\in W$ and any $\psi\in\mathcal{L}^M(N,\Theta)$. The proof is by induction over ψ.

Since $\hat{\varphi}\in\Lambda_0(y)$, $\hat{\varphi}$ is thus satisfied by M. It remains to be shown that M has all the properties we require of a model:

REFL,TRANS Reflexivity and transitivity of $\mathsf{can}(R_i)$ is ensured by axioms T and 4. $\mathsf{can}(R_i)$ is still reflexive and transitive when restricted to W.

CONVERSE We must show that for any $w,v\in W$, R_iwv iff R_i^cvw. Let R_iwv, and assume that $\psi\in w$. Then $\langle i^c\rangle\psi\in v$ by axiom *Converse*$_1(i)$, showing that R_i^cvw. Conversely, let R_i^cvw and assume that $\psi\in v$. Then $\langle i\rangle\in w$ by axiom *Converse*$_2(i)$, showing that R_iwv. Similar reasoning goes for the other relations and their converses.

DIFF First, let $w\neq v$ for some $w,v\in W$. We must show that $(w,v)\in\mathsf{can}(D)$, i.e., that for any $\psi\in v$, $\langle D\rangle\psi\in w$. Let $\psi\in v$. Since $w\neq v$ (and w,v are MCSs), there exists a $\gamma\in v$ such that $\neg\gamma\in w$. Thus, $\gamma\wedge\psi\in v$ and $\neg(\gamma\wedge\psi)\in w$. Since $w,v\in W$ and $\mathcal{N}$ is connected, there is a path of nodes $r_0\cdots r_k$ ($k\geqslant1$) such that $r_0,\dots,r_k\in N$ and $(r_i,r_{i+1})\in E$ for each $i\in[0,k-1]$. It follows that $\langle r_0r_1\rangle\cdots\langle r_{k-1}r_k\rangle(\gamma\wedge\psi)\in w$, because otherwise $\neg\langle r_0r_1\rangle\cdots\langle r_{k-1}r_k\rangle(\gamma\wedge\psi)\in w$ and $\Delta(\mathcal{N},w)$ would be inconsistent and $\mathcal{N}$ incoherent. By axiom D_2 it follows that $\langle D\rangle(\gamma\wedge\psi)\in w$, and thus that $\langle D\rangle\psi\in w$.

Second, let $(w,v)\in\mathsf{can}(D)$ for $w,v\in W$. We must show that $w\neq v$. Assume otherwise; $w=v$. By the elimination of the $D2(w)$-defect, there is a formula φ such that $\varphi\wedge\neg\langle D\rangle\varphi\in w$. But by the fact that $(w,w)\in\mathsf{can}(D)$ the fact that $\varphi\in w$ implies that $\langle D\rangle\varphi\in w$, which is a contradiction.

---

[9] Note that MCGL is a normal modal logic.

COMPL It follows immediately from DIFF that M is *named*, i.e., that for every state w in M there is a formula φ_w such that $M, w \models \varphi_w$, and for any $w \neq v$, $M, v \not\models \varphi$: since $\mathcal{N}$ does not have any $D2(w)$ defect, every state $w \in W$ contains a formula of the form $\varphi_w \wedge \neg\langle D\rangle\varphi_w$. By DIFF, no other state contains the same formula.

Assume that COMPL does not hold, i.e., that there are states $w, u \in W$ such that $w \neq u$, $\neg R_i uw$ and $\neg R_i wu$. Suppose $M, u \models \langle i^c\rangle\varphi_w$. Then there exists a v such that $R_i^c uv$ and $M, v \models \varphi_w$. But that means that $v = w$, and $R_i^c uw$. By CONVERSE, that means that $R_i wu$, but that contradicts the assumption. Thus, $M, u \not\models \langle i^c\rangle\varphi_w$. We then have that $M, u \not\models (\varphi_w \vee \langle i^c\rangle\varphi_w \vee \langle i\rangle\varphi_w)$ and, since $w \neq u$, that $M, w \not\models [D]((\varphi_w \vee \langle i^c\rangle\varphi_w \vee \langle i\rangle\varphi_w)$. We also have that $M, w \models (\varphi_w \wedge [i]\langle i^c\rangle\varphi_w)$—but that contradicts the Trichotomy axiom. Thus, COMPL must hold.

STRICT As in the COMPL case, we use the fact that M is named and let φ_w be the formula uniquely true in w.

Assume that STRICT does not hold. This means that there are w and u for which either (i) not $R_i^s wu$ and $(R_i wu \wedge \neg R_i uw)$ or (ii) $R_i^s wu$ but not $(R_i wu \wedge \neg R_i uw)$. In the first case, $w \neq u$. We have that $M, w \models (\varphi_w \wedge \langle i\rangle(\varphi_u \wedge [i]\neg\varphi_w))$. But we also have that $M, w \not\models \langle i^s\rangle\varphi_u$, contradicting axiom *Strict*$_1$. Suppose we are in case (ii): we have $R_i^s wu$ and also (not $R_i wu$ or $R_i uw$). First, suppose $w = u$. $M, w \models \langle i^s\rangle\varphi_w$, but $M, w \not\models \langle D\rangle\varphi_w$, contradicting axiom *Strict*$_3$. Thus, it must be the case that $w \neq u$. We have either (a) $R_i^s wu$ and not $R_i wu$, or (b) $R_i^s wu$ and $R_i wu$ and $R_i uw$. In both cases, uniformly substituting φ_w for p, and φ_u for q, in axiom *Strict*$_2$, shows that *Strict*$_2$ does not hold in w. Thus, STRICT must hold.

INTERSECTION As in the COMPL case, we use the fact that M is named and let φ_w be the formula uniquely true in w.

Assume that INTERSECTION does not hold. This means that there are w and u for which either (i) $R_C wu$ and not $(\bigcap_{i \in C} R_i)wu$, or (ii) not $R_C wu$ and $(\bigcap_{i \in C} R_i)wu$. Consider case (i). Then there is a $i \in C$ such that $\neg R_i wu$, and thus we have that $M, w \models \langle C\rangle\varphi_u$ but $M, w \not\models \langle i\rangle\varphi_u$, contradicting axiom *Intersect*$_3$. Consider case (ii). Note that $((p \wedge [D]\neg p) \vee \langle D\rangle(p \wedge [D]\neg p))$ is true in a world w if and only if there is a unique world in which p is true. Let us abbreviate it to $\langle\exists!\rangle p$. We have that $M, w \models \langle\exists!\rangle\varphi_u$, that $M, w \models \bigwedge_{i \in C}\langle i\rangle\varphi_u$, but $M, w \not\models \langle C\rangle\varphi_u$, contradicting axiom *Intersect*$_1$.

INTERSECTION-STRICT As the INTERSECTION case, using axioms *Intersect*$_2$ and *Intersect*$_4$. □

Corollary 3. MCGL *is sound and complete with respect to the class of all coalitional games.*

Proof. First we define mappings between models and games. Given a game $\Gamma = (N, \Omega, V, \sqsupseteq_1, \ldots, \sqsupseteq_m)$ and an interpretation π' of Θ' in Ω, we define a model $f(\Gamma, \pi) = M = (W, \{R_i\colon i \in N\}, \{R_i^s\colon i \in N\}, \{R_C\colon C \in \mathcal{C}\}, \{R_C^s\colon C \in \mathcal{C}\}, D, \pi)$ by taking $W = \Omega$, $R_i = \sqsupseteq_i$, defining $w \in \pi(p_C)$ iff $w \in V(C)$ and $\pi(p) = \pi'(p)$ when $p \in \Theta'$, and letting the remaining relations in M be defined by the semantic restrictions on models. Conversely, given a model M, we define $g(M) = (\Gamma, \pi')$ as follows: $\Omega = W$, $w \in V(C)$ iff $w \in \pi(p_C)$, $\sqsupseteq_i = R_i$ and $w \in \pi'(p)$ iff $w \in \pi(p)$. Clearly, $\Gamma, \pi, w \models \varphi$ iff $f(\Gamma, \pi'), w \models \varphi$, and $g(M), w \models \varphi$ iff $M, w \models \varphi$, for any formula φ.

For soundness, let $\vdash \varphi$. By soundness w. r. t. models (Theorem 9), $f(\Gamma, \pi'), w \models \varphi$ for any Γ, π', so $\Gamma, \pi', w \models \varphi$. For completeness, assume that φ is consistent. By completeness w.r.t. models (Theorem 10), $M, w \models \varphi$ for some M, w. Then also $g(M), w \models \varphi$. □

4.3. Expressing game properties

The fact that an outcome w of a coalitional game Γ is in the core of Γ, can be expressed as follows.

$$MCM \equiv p_N \wedge \bigwedge_{C \subseteq N} [C^s]\neg p_C$$

Theorem 11. $(\Gamma, \omega) \models MCM$ *iff* ω *is in the core of* Γ.

Proof. $(\Gamma, \omega) \models MCM$ iff $\omega \in V(N)$ and for every coalition C and every outcome ω' such that $(\omega, \omega') \in \sqsupset_i$ for every $i \in C$, $\omega' \notin V(C)$. This holds iff ω is in the core of Γ. □

We can now express the fact that the core is non-empty:

$$MCNE \equiv MCM \vee \langle D\rangle MCM$$

Theorem 12. $(\Gamma, \omega) \models MCNE$ *iff* $\Gamma \models MCNE$ *iff the core of* Γ *is non-empty.*

Proof. The core of Γ is non-empty iff either ω is in the core, or v is in the core for some $v \neq \omega$. In the first case, $(\Gamma, \omega) \models MCM$; in the second case $(\Gamma, \omega) \models \langle D\rangle MCM$. □

Imputations can be characterised as follows.

$$MIMP \equiv p_N \wedge \bigwedge_{i \in N} [C^s]\neg p_i$$

Proposition 1. *$(\Gamma, \omega) \models MIMP$ iff ω is an imputation in Γ.*

Compare *MCM* and *MIMP*—the logical characterisation highlights the similarities and differences between the core and imputation concepts.

Moving on to stable sets, we first characterise the existence of objections to some imputation. The following formula expresses a more general property, namely that there exists a C-s-objection which satisfies some formula α.

$$MOBJ(C,\alpha) \equiv \mathrm{MIMP} \wedge \langle C^s \rangle (\mathrm{MIMP} \wedge \alpha \wedge \langle C \rangle p_C)$$

Proposition 2. *Let C be a coalition and α a formula. $(\Gamma, \omega) \models MOBJ(C,\alpha)$ iff ω is an imputation and there exists a C-s-objection ω' to ω such that $(\Gamma, \omega') \models \alpha$.*

Proof. Assume that ω is an imputation (the other case is trivial). $(\Gamma, \omega) \models MOBJ(C,\alpha)$ iff there is an imputation ω' such that every $i \in C$ strictly prefers ω' over ω, and there is an outcome ω'' such that every $i \in C$ weakly prefers ω'' over ω', $(\Gamma, \omega'') \models \alpha$ and $\omega'' \in V(C)$. □

Of course, now $MOBJ(C, \top)$ (for some tautology $\top$) is true in imputation ω iff there exists a C-s-objection to ω.

Characterising stable sets and the bargaining set in MCGL is not as straightforward. Formulae are interpreted in single outcomes, and unlike in CGL we cannot name outcomes in formulae. But a set of outcomes is precisely what is *denoted* by a formula: let $\varphi^\Gamma = \{\omega\colon\ \Gamma, \omega \models \varphi\}$ be the extension of formula φ in game Γ. For example, we have that MCM^Γ is the core of Γ. We can characterise stable sets in the following sense: we can describe exactly the formulae whose extensions are stable sets.

Theorem 13. *Let γ be a formula.*

$$\Gamma \models (\gamma \to MIMP) \wedge \left(\gamma \to \neg \bigvee_{C \in \mathcal{C}} MOBJ(C,\gamma)\right) \wedge \left(\neg\gamma \to \bigvee_{C \in \mathcal{C}} MOBJ(C,\gamma)\right)$$

iff γ^Γ is a stable set in Γ.

Proof. Let ξ be the formula on the right hand side of $\models$. $\Gamma \models \xi$ iff for every ω: (i) if $\omega \in \gamma^\Gamma$ then ω is an imputation; (ii) if $\omega \in \gamma^\Gamma$ then there is no C-s-objection to ω for any C, which is a member of γ^Γ and (iii) if $\omega \notin \gamma^\Gamma$ then there is a C-s-objection to ω for some C, which is a member of γ^Γ. This holds iff γ^Γ is a stable set. □

However, note that the theorem above does not guarantee that given a stable set Y there necessarily exists a formula γ such that $\gamma^\Gamma = Y$.

4.4. Proof examples

We here illustrate the proof theory of MCGL. Before we formally prove some well known properties of coalitional games in Example 6, we discuss some general proof theoretic principles.

Definition 6. Given the operator $\langle D \rangle$, we can define universal 'everywhere' and 'somewhere' operators A and E, respectively:

1. $A\varphi := \varphi \wedge [D]\varphi$,
2. $E\varphi := \varphi \vee \langle D \rangle \varphi$.

Clearly, $A\varphi$ is equivalent to $\neg E \neg \varphi$, and $E\varphi$ equals $\neg A \neg \varphi$.

Lemma 7. *The following is a useful modal principle.*

$$\vdash \langle i \rangle (\langle i \rangle p \wedge [i]q) \to \langle i \rangle (p \wedge [i]q)$$

Proof. Let ML denote basic Modal Logic properties.

$$\langle i \rangle (\langle i \rangle p \wedge [i]q) \Rightarrow (\text{Axiom 4})$$

$$\langle i \rangle (\langle i \rangle p \wedge [i][i]q) \Rightarrow (\text{ML: } (\Diamond r \wedge \Box s) \to \Diamond (r \wedge s))$$

$$\langle i \rangle (\langle i \rangle (p \wedge [i]q)) \Rightarrow (\text{Dual Axiom 4: } (\langle i \rangle \langle i \rangle s \to \langle i \rangle s))$$

$$\langle i \rangle (p \wedge [i]q) \qquad \Box$$

The following lemma summarises some useful MCGL properties. Lemma 8.1 expresses that if agent i strictly prefers outcome y over x, while he also prefers z over y, then he strictly prefers z over x. To prove this, we need the axioms $Strict_1$ and $Strict_2$. Then item 3 of the same lemma says that a similar kind of transitivity holds for the coalitional preferences. But the reader should be aware that the relation for C^s is *not* the strict version of the relations for C; rather, it is the intersection of the strict relations for i, with $i \in C$. It is in fact not hard to show that the axiom corresponding to $Strict_1$ is not valid for C^s: we do this in item 2. Items 4, 5 and 6 are easy consequences of the definition of E, A and $MOBJ(C, \cdot)$.

Lemma 8. *The following are some properties of* MCGL.

(1) $\vdash \langle i^s\rangle\langle i\rangle r \to \langle i^s\rangle r$,
(2) $\not\models p \wedge \langle C\rangle(q \wedge [C]\neg p) \to \langle C^s\rangle q$,
(3) $\vdash \langle C^s\rangle\langle C\rangle r \to \langle C^s\rangle r$,
(4) $\vdash A(p \to q) \to (\Diamond p \to \Diamond q)$, *with* $\Diamond \in Diamonds$,
(5) $\vdash A(p \to q) \to (MOBJ(C, p) \to MOBJ(C, q))$,
(6) *A is a box operator, in particular, it satisfies Nec and K.*

Proof.

(1) A derivation is the following.

1	$(p \wedge [D]\neg p) \to (\langle i^s\rangle\langle i\rangle r \to \langle i\rangle(\langle i\rangle r \wedge [i]\neg p))$	$Strict_2$
2	$(p \wedge [D]\neg p) \to (\langle i^s\rangle\langle i\rangle r \to \langle i\rangle(r \wedge [i]\neg p))$	1, *Lemma* 7
3	$(p \wedge [D]\neg p) \to (\langle i^s\rangle\langle i\rangle r \to (p \wedge \langle i\rangle(r \wedge [i]\neg p)))$	2, *Taut*
4	$(p \wedge [D]\neg p) \to (\langle i^s\rangle\langle i\rangle r \to \langle i^s\rangle r)$	3, $Strict_1$
5	$\langle i^s\rangle\langle i\rangle r \to \langle i^s\rangle r$	4, *D-rule*

(2) Let R_1, the preferences of 1, be the reflexive transitive closure of $\{(x, y), (y, z)\}$ and similarly for $R_2 \supseteq \{(x, y), (y, x), (y, z)\}$. Let p be true in x only, and q true in y only. Let $C = \{1, 2\}$. Let this be the full description of model M, in particular, x, y and z are all alternatives. Then we have that R_C is the reflexive transitive closure of $\{(x, y), (y, z)\}$ whereas the relation for C^s is $\{(y, z)\}$. Then $M, x \models p \wedge \langle C\rangle(q \wedge [C]\neg p)$ while at the same time $M, x \models \neg\langle C^s\rangle q$. Note that indeed, the strict version of $R_C = \{(x, y), (y, z)\}$ is not the same as the intersection of the individual strict relations $R^s_C = \{(y, z)\}$.
(3) We use the following result for logics which contain $\langle D\rangle$ and a number of other modalities with their converse [11, Lemma 3.3.31]. Let Op ('only here, p') be defined as $p \wedge [D]\neg p$. The D-rule says that if one can derive θ, which does not involve p, from the assumption that Op, then one can derive θ. The following result generalises this by lifting the assumption Op *here, locally* to assuming Op in an arbitrary state, or outcome. □

Definition 7. (See [11, pp. 31, 37].) We write $\psi \trianglelefteq \varphi$ for 'ψ occurs as a sub-formula in φ'. We do not identify different occurrences of ψ in φ. Define the function $\mathsf{Paste}(\nu, \psi, \varphi)$ (paste ν (the name) next to the occurrence of ψ in φ) by induction on φ, treating ψ as an atomic symbol in φ. Let $\langle X\rangle$ be a modal diamond operator.

$$\mathsf{Paste}(\nu, \psi, p) = p \text{ if } \psi \neq p$$

$$\mathsf{Paste}(\nu, \psi, \psi) = \nu \wedge \psi$$

$$\mathsf{Paste}(\nu, \psi, \neg\varphi) = \neg\varphi$$

$$\mathsf{Paste}(\nu, \psi, \varphi \wedge \chi) = \mathsf{Paste}(\nu, \psi, \varphi) \wedge \mathsf{Paste}(\nu, \psi, \chi)$$

$$\mathsf{Paste}(\nu, \psi, \langle X\rangle\varphi) = \langle X\rangle\mathsf{Paste}(\nu, \psi, \varphi)$$

As an example, $\mathsf{Paste}(Op, r \wedge \neg q, r \wedge \langle i\rangle(r \wedge \neg q)) = \mathsf{Paste}(Op, r \wedge \neg q, r) \wedge \mathsf{Paste}(Op, r \wedge \neg q, \langle i\rangle(r \wedge \neg q)) = r \wedge \langle i\rangle\mathsf{Paste}(Op, r \wedge \neg q, (r \wedge \neg q) = r \wedge \langle i\rangle(Op \wedge r \wedge \neg q)$.

Before commenting on this definition, we formulate a lemma.

Lemma 9 *(Pasting Lemma). Suppose we have a logic with the $\langle D\rangle$ operator, and for which for every operator, the converse is also present. Assume Op has no proposition letters in common with φ and θ. For any sub-formula occurrence $\psi \trianglelefteq \varphi$ we have*

If $\vdash \mathsf{Paste}(Op, \psi, \varphi) \to \theta$ *then* $\vdash \varphi \to \theta$

Proof. See [11, p. 37], where the only operators used indeed have a converse. □

1	$\langle C^s\rangle\langle C\rangle(p\wedge[D]\neg p\wedge r)\to\bigwedge_{i\in C}\langle i^s\rangle\langle i\rangle(p\wedge[D]\neg p\wedge r)$	*Intersect*$_3$, *Intersect*$_4$
2	$\bigwedge_{i\in C}\langle i^s\rangle\langle i\rangle(p\wedge[D]\neg p\wedge r)\to\bigwedge_{i\in C}\langle i^s\rangle(p\wedge[D]\neg p\wedge r)$	*Lemma* 8.1
3	$(p\wedge[D]\neg p\wedge r)\to((p\wedge r)\wedge[D]\neg(p\wedge r))$	$Nec, K, \vdash\neg p\to\neg(p\wedge q)$
4	$\langle C^s\rangle\langle C\rangle(p\wedge[D]\neg p\wedge r)\to\bigwedge_{i\in C}\langle i^s\rangle((p\wedge r)\wedge[D]\neg(p\wedge r))$	1, 2, 3, *ML*
5	$\langle i^s\rangle((p\wedge r)\wedge[D]\neg(p\wedge r))\to\langle D\rangle((p\wedge r)\wedge[D]\neg(p\wedge r))$	*Strict*$_3$
6	$\langle D\rangle((p\wedge r)\wedge[D]\neg(p\wedge r))\to\bigwedge_{i\in C}\langle i^s\rangle(p\wedge r)\to\langle C^s\rangle(p\wedge r)$	*Intersect*$_2$
7	$\langle C^s\rangle\langle C\rangle(p\wedge[D]\neg p\wedge r)\to\langle C^s\rangle(p\wedge r)$	4, 5, 6
8	$\langle C^s\rangle((p\wedge r)\wedge[D]\neg(p\wedge r))\to\langle C^s\rangle r$	*ML*
9	$\langle C^s\rangle\langle C\rangle(p\wedge[D]\neg p\wedge r)\to\langle C^s\rangle r$	7, 8

Fig. 3. A deductive proof of (1).

A few remarks are in place here. First of all, note that $\mathsf{Paste}(\cdot,\cdot,\cdot)$ is only defined for atoms, conjunction, negation and Diamond formulas, and for cases where the sub-formula occurrence ψ is exactly φ. Also note, that in the latter case, the Pasting Lemma is just the D-rule. For Diamond formulas of the form $\langle X\rangle\varphi$, the lemma basically lets one assume $\langle X\rangle(\nu\wedge\varphi)$, which has the role of 'assume there is an accessible state in which φ holds, and assume p is only true there'. This is like giving a temporary name for that state. If we can derive a 'p-free' θ from that assumption, then θ was derivable from $\langle X\rangle\varphi$ already.

Now we can also explain why the definition of $\mathsf{Paste}(\nu,\psi,\neg\varphi)$ should not enter a recursive call: suppose we would have defined a function $\mathsf{Paste}^*(\cdot,\cdot,\cdot)$ and define $\mathsf{Paste}^*(\nu,\psi,\neg\varphi)=\neg\mathsf{Paste}^*(\nu,\psi,\varphi)$. Then one easily shows that $\mathsf{Paste}^*(Op,q,[X]q)$, or, equivalently $\mathsf{Paste}^*(Op,q,\neg\langle X\rangle\neg q)$ would be equal to $[X](Op\wedge q)$ (where $\langle X\rangle$ is some Diamond operator). But the latter says: Suppose that all X-successors satisfy q *and that they all have the same unique name* p. From that, it would of course follow that there can be at most one such a successor, in other words, we have $\vdash\mathsf{Paste}^*(Op,q,[X]q)\to(\langle X\rangle r\to[X]r)$, but we do not have $\vdash\langle X\rangle r\to[X]r$, i.e., the Pasting Lemma would not hold for this $\mathsf{Paste}^*(\cdot,\cdot,\cdot)$.

Finally note, that the Pasting Lemma only lets us replace *one* occurrence of ψ by ν: this is also sensible, since without this constraint we would obtain $\mathsf{Paste}^*(Op,q,\langle X\rangle(q\wedge r)\wedge\langle X\rangle(q\wedge s))=\langle X\rangle(Op\wedge q\wedge r)\wedge\langle X\rangle(Op\wedge q\wedge s)$, and this again, would identify two successor states (by saying they have the same unique name): we have $\vdash\mathsf{Paste}^*(Op,q,\langle X\rangle(q\wedge r)\wedge\langle X\rangle(q\wedge s))\to\langle X\rangle(s\wedge r)$, and would the Pasting Lemma hold for $\mathsf{Paste}^*(\cdot,\cdot,\cdot)$, we would conclude $\vdash(\langle X\rangle(q\wedge r)\wedge\langle X\rangle(q\wedge s)\to\langle X\rangle(r\wedge s)$, which is obviously undesirable.

Note that in our logic MCGL every operator has a converse (D is its own converse). In other words, we can apply the pasting lemma. To do this in order to prove $\vdash\langle C^s\rangle\langle C\rangle r\to\langle C^s\rangle r$, fix the following parameters in the pasting lemma:

$$Op=Op$$

$$\psi=r$$

$$\varphi=\langle C^s\rangle\langle C\rangle r$$

$$\theta=\langle C^s\rangle r$$

According to the lemma, in order to prove $\vdash\langle C^s\rangle\langle C\rangle r\to\langle C^s\rangle r$, it is sufficient to prove

$$\vdash\mathsf{Paste}(Op,r,\langle C^s\rangle\langle C\rangle r)\to\langle C^s\rangle r)$$

Using the definition of $\mathsf{Paste}(\cdot,\cdot,\cdot)$ and of Op, this boils down to proving

$$\vdash\langle C^s\rangle\langle C\rangle(p\wedge[D]\neg p\wedge r)\to\langle C^s\rangle r \qquad (1)$$

A proof is given in Fig. 3.

(4) It is a general modal principle that $\Box(p\to q)\to(\Diamond p\to\Diamond q)$, so it suffices to prove that $\vdash A(p\to q)\to\Box(p\to q)$.

1	$\neg\Box(p\to q)\to\Diamond(p\wedge\neg q)$	*ML*
2	$\Diamond(p\wedge\neg q)\to((p\wedge\neg q)\vee\langle D\rangle(p\wedge\neg q))$	D_2
3	$((p\wedge\neg q)\vee\langle D\rangle(p\wedge\neg q))\to E\neg(p\to q)$	*Taut*, *Def E*
4	$E\neg(p\to q)\to\neg A(p\to q)$	$A\varphi=\neg E\neg\varphi$
5	$A(p\to q)\to\Box(p\to q)$	1–4, *Taut*

(5) This follows immediately from the definition of $MOBJ(C,\alpha)$ and item 4.

(6) Immediate from the definition of A and the fact that $[D]$ is a normal box operator. □

We now illustrate the formal derivation of some well-known properties of coalitional games.

Let $ST(p)$ denote that p marks a stable set: $ST(p)=A(st(p))$ where

$$st(p):=(p\to MIMP)\wedge\left(p\to\bigwedge_{C\in\mathcal{C}}\neg MOBJ(C,p)\right)\wedge\left(\neg p\to\bigvee_{C\in\mathcal{C}}MOBJ(C,p)\right)$$

1a	$(p \to MIMP)$	Assumption $st(p)$
1b	$(p \to \bigwedge_{C \in \mathcal{C}} \neg MOBJ(C, p))$	Assumption $st(p)$
1c	$(\neg p \to \bigvee_{C \in \mathcal{C}} MOBJ(C, p))$	Assumption $st(p)$
2	$p_N \wedge \bigwedge_{C \subseteq N} [C^s] \neg p_C$	Assumption MCM
3	$MOBJ(C, p) \to \langle C^s \rangle (MIMP \wedge p \wedge \langle C \rangle p_C)$	Definition $MOBJ(C, p)$
4	$MOBJ(C, p) \to \langle C^s \rangle \langle C \rangle p_C$	3, ML
5	$MOBJ(C, p) \to \langle C^s \rangle p_C$	4, Lemma 8.1
6	$MOBJ(C, p) \to (\langle C^s \rangle p_C \wedge [C^s] \neg p_C)$	2, 5
7	$(\langle C^s \rangle p_C \wedge [C^s] \neg p_C) \to \bot$	$ML : \vdash (\Diamond \varphi \wedge \Box \neg \varphi) \to \bot$
8	$\neg p \to \bot$	1c, 6, 7
9	p	8, $Taut$

Fig. 4. Deductive proof for Example 6.1.

1a	$A(p \to MIMP)$	Assumption $ST(p)$
1b	$A(p \to \neg \bigvee_{C \in \mathcal{C}} MOBJ(C, p))$	Assumption $ST(p)$
1c	$A(\neg p \to \bigvee_{C \in \mathcal{C}} MOBJ(C, p))$	Assumption $ST(p)$
2a	$A(q \to MIMP)$	Assumption $ST(q)$
2b	$A(q \to \neg \bigvee_{C \in \mathcal{C}} MOBJ(C, q))$	Assumption $ST(q)$
2c	$A(\neg q \to \bigvee_{C \in \mathcal{C}} MOBJ(C, q))$	Assumption $ST(q)$
3	$A(p \to q)$	Assumption
4	$E(q \wedge \neg p) \to (q \wedge \neg p) \vee \langle D \rangle (q \wedge \neg p)$	$Def E$
5	$(q \wedge \neg p) \to (\bigvee_{C \subseteq N} MOBJ(C, p) \wedge \neg \bigvee_{C \subseteq N} MOBJ(C, q))$	1c, 2b
6	$(q \wedge \neg p) \to (\bigvee_{C \subseteq N} MOBJ(C, q) \wedge \neg \bigvee_{C \subseteq N} MOBJ(C, q))$	3, 5, Lemma 8.5
7	$(q \wedge \neg p) \to \bot$	6, ML
8	$\langle D \rangle (q \wedge \neg p) \to \langle D \rangle (\bigvee_{C \subseteq N} MOBJ(C, p) \wedge \neg \bigvee_{C \subseteq N} MOBJ(C, q))$	1c, 2b, Lemma 8.4
9	$\langle D \rangle (q \wedge \neg p) \to \langle D \rangle (\bigvee_{C \subseteq N} MOBJ(C, q) \wedge \neg \bigvee_{C \subseteq N} MOBJ(C, q))$	3, 8, Lemma 8.5
10	$\langle D \rangle (q \wedge \neg p) \to \bot$	9, ML
11	$\neg E(q \wedge \neg p)$	4, 7, 9
12	$A(q \to p)$	11, $DefnA, E$

Fig. 5. Deductive proof for Example 6.2.

1	$ST(MCM) \to (ST(p) \to A(MCM \to p))$	item 1 of this lemma
2	$ST(MCM) \to (ST(p) \to (A(MCM \to p) \to A(p \to MCM)))$	item 2 of this lemma
3	$ST(MCM) \to (ST(p) \to (A(MCM \to p) \wedge A(p \to MCM)))$	1, 2, $Taut$, MP
4	$A(MCM \to p) \wedge A(p \to MCM) \leftrightarrow A(MCM \leftrightarrow p)$	Lemma 8.6
5	$ST(MCM) \to (ST(p) \to A(MCM \leftrightarrow p))$	3, 4, ML

Fig. 6. Deductive proof for Example 6.3.

Example 6. The following properties concerning the core and stable sets are now derivable:

(1) The core is a subset of any stable set. In our object language, this is written as:

$$ST(p) \to A(MCM \to p)$$

(2) No stable set is a proper subset of any other. This is represented as:

$$\big(ST(p) \wedge ST(q)\big) \to \big(A(p \to q) \to A(q \to p)\big)$$

(3) If the core is a stable set then it is the only stable set. We represent this as:

$$ST(MCM) \to \big(ST(p) \to A(MCM \leftrightarrow p)\big)$$

Proof.

(1) We will show $\vdash st(p) \to (MCM \to p)$; we then can use Lemma 8.6 to conclude $\vdash ST(p) \to A(MCM \to p)$. We prove implications $\varphi \to \psi$ by using assumptions, proving ψ from the assumption φ. Note that such assumptions do not state that $A\varphi$ is *derivable*, and hence *Nec* cannot be applied to them. A proof is given in Fig. 4. In the proof, note that 1a, 1b, and 1c together directly follow from assumption $st(p)$, and 2 denotes MCM. The goal is to derive p from them.
(2) See Fig. 5. Again, we use assumptions. (Note that $A(\varphi \wedge \psi)$ is equivalent to $A\varphi \wedge A\psi$.)
(3) A proof is given in Fig. 6. □

5. Logical comparisons

In this section we compare the two logics CGL and MCGL introduced in Sections 3 and 4 respectively, first to each other and then to Coalition Logic.

5.1. CGL vs. MCGL

We have introduced two logics, CGL and MCGL, both interpreted in coalitional games. We know that the former is very expressive when it comes to *finite* games, while the latter can express many interesting properties also of games with infinitely many outcomes. In this section, we make the relationship between the two logics more precise.

First, observe that, while formulae of both logics express properties of coalitional games, an MCGL formula is interpreted as a property of a given *outcome* of a game. A CGL formula, on the other hand, is interpreted as a general property of the game. Of course the CGL *outcome* language $\mathcal{L}_o$ is also interpreted "locally" in a given outcome, but it is not so interesting to compare $\mathcal{L}_o$ to the MCGL language because the former is not very expressive. However, the main CGL (cooperation) language $\mathcal{L}_c$ allows us to refer to particular outcomes in a formula, so a property of a given outcome can be written in CGL as a property of a game by referring to the outcome in the formula. CGL can express any property of finite games (Theorem 1), so it must be able to express any property MCGL can express about a particular outcome in such a game. While we already know that this holds, an actual mapping from the MCGL language to the CGL language might be of interest. We now provide such a mapping. Assume that Γ is a finite coalitional game. Given an MCGL formula φ and an outcome ω of Γ, we define a CGL formula $f_\Gamma(\varphi,\omega)$. The idea is that ω has the (MCGL) property φ in the game Γ iff Γ has the (CGL) property $f_\Gamma(\varphi,\omega)$. Let:

- $f_\Gamma(p,\omega)=\bot$ (when $p\in\Theta'$);
- $f_\Gamma(p_C,\omega)=\langle C\rangle\omega$;
- $f_\Gamma(\langle i\rangle\varphi,\omega)=\bigvee_{\omega'\in\Omega}(\omega'\succeq_i\omega\wedge f_\Gamma(\varphi,\omega'))$;
- $f_\Gamma(\langle i^s\rangle\varphi,\omega)=\bigvee_{\omega'\in\Omega}(\omega'\succeq_i\omega\wedge\neg(\omega\succeq_i\omega')\wedge f_\Gamma(\varphi,\omega'))$;
- $f_\Gamma(\langle C\rangle\varphi,\omega)=\bigvee_{\omega'\in\Omega}(\bigwedge_{i\in C}(\omega'\succeq_i\omega)\wedge f_\Gamma(\varphi,\omega'))$;
- $f_\Gamma(\langle C^s\rangle\varphi,\omega)=\bigvee_{\omega'\in\Omega}(\bigwedge_{i\in C}(\omega'\succeq_i\omega\wedge\neg(\omega\succeq_i\omega'))\wedge f_\Gamma(\varphi,\omega'))$;
- $f_\Gamma(\langle D\rangle\varphi,\omega)=\bigvee_{\omega'\in\Omega\setminus\{\omega\}}f_\Gamma(\varphi,\omega')$;
- $f_\Gamma(\langle i^c\rangle\varphi,\omega)=\bigvee_{\omega'\in\Omega}(\omega\succeq_i\omega'\wedge f_\Gamma(\varphi,\omega'))$;
- $f_\Gamma(\langle i^{sc}\rangle\varphi,\omega)=\bigvee_{\omega'\in\Omega}(\omega\succeq_i\omega'\wedge\neg(\omega'\succeq_i\omega)\wedge f_\Gamma(\varphi,\omega'))$;
- $f_\Gamma(\langle C^c\rangle\varphi,\omega)=\bigvee_{\omega'\in\Omega}(\bigwedge_{i\in C}(\omega\succeq_i\omega')\wedge f_\Gamma(\varphi,\omega'))$;
- $f_\Gamma(\langle C^{sc}\rangle\varphi,\omega)=\bigvee_{\omega'\in\Omega}(\bigwedge_{i\in C}(\omega\succeq_i\omega'\wedge\neg(\omega'\succeq_i\omega))\wedge f_\Gamma(\varphi,\omega'))$;
- $f(\neg\varphi,\omega)=\neg f(\varphi,\omega)$;
- $f(\varphi_1\wedge\varphi_2,\omega)=f(\varphi_1,\omega)\wedge f(\varphi_2,\omega)$.

We will henceforth sometimes use $\models_{cgl}$ and $\models_{mcgl}$ to denote the $\models$ relation in CGL and MCGL, respectively, in case confusion can occur.

Lemma 10. *For any finite coalitional game Γ, outcome ω of Γ and* MCGL *formula φ:*

$$\Gamma,\omega\models_{mcgl}\varphi \quad \textit{iff} \quad \Gamma\models_{cgl}f_\Gamma(\varphi,\omega)$$

Proof. The proof is by induction on the structure of φ. For the first base case, $\Gamma,\omega\models_{mcgl}p$ iff $\Gamma,\pi,\omega\models_{mcgl}p$ for every π, which is never true. It is also not true that $\Gamma\models_{cgl}\bot$. For the second base case, $\Gamma,\omega\models_{mcgl}p_C$ iff $\omega\in V(C)$ iff $\Gamma\models_{cgl}\langle C\rangle\omega$. Consider the inductive step. $\Gamma,\omega\models_{mcgl}\langle i\rangle\varphi$ iff there is a ω' such that $\omega'\sqsupseteq_i w$ and $\Gamma,\omega'\models_{mcgl}\varphi$ iff, by the induction hypothesis, there is a ω' such that ($\Gamma\models_{cgl}\omega'\succeq_i\omega$ and $\Gamma\models_{cgl}f_\Gamma(\varphi,\omega')$) iff $\Gamma\models_{cgl}\bigvee_{\omega'\in\Omega}(\omega'\succeq_i\omega\wedge f_\Gamma(\varphi,\omega'))$. The argument is similar for the other cases. □

As an example, let ω be an outcome of a finite coalitional game Γ, and consider the property "ω is in the core of Γ". We have seen that this property can be expressed in MCGL by the formula *MCM* (p. 68). We leave it to the reader to check that $f_\Gamma(MCM,\omega)$ is equal to the CGL formula $CM(\omega)$ (p. 59). Furthermore, $f_\Gamma(MCNE,\omega)$ is equivalent to *CNE*, both expressing the fact that the core of Γ is non-empty (note that $f_\Gamma(MCNE,\omega)$ does not depend on ω).

It is obvious that, for finite games, CGL is strictly more expressive than MCGL. For example, truth of a MCGL formula is invariant under "renaming" of outcomes, i.e., changing the names of the outcomes in Ω does not change the truth value of a formula in that game, but this is not the case for CGL.

Moving on to infinite games, it is no longer the case that CGL is strictly more expressive than MCGL.

Proposition 3. *There is an* MCGL *formula φ', such that for any* CGL *formula φ there exist coalitional games Γ,Γ', both having an outcome ω, such that*

$$\Gamma,\omega\models_{mcgl}\varphi' \quad \textit{and} \quad \Gamma',\omega\not\models_{mcgl}\varphi'$$

but

$$\Gamma \models_{cgl} \varphi \quad \textit{iff} \quad \Gamma' \models_{cgl} \varphi$$

Proof. Take

$$\varphi' = \langle\{i\}\rangle p_{\{i\}}$$

i.e., the property of a given outcome ω that there exist a $\omega' \in V(\{i\})$ such that $\omega' \sqsupset_i \omega$. This property is expressible for *finite* games by the CGL formula $\bigvee_{\omega' \in \Omega}(\langle\{i\}\rangle\omega' \wedge \omega' \succeq_i \omega)$. However, given an arbitrary CGL formula we now construct two *infinite* games, one which has the property and one which does not, which are indiscernible by the CGL formula. Note that in CGL (unlike in MCGL), the logical language is parameterised by a set of outcomes, and formulae are interpreted by games over the same set of outcomes. Thus, we fix a set of outcomes Ω, and we assume that Ω is infinite. We assume that Ω contains an outcome ω. Let φ be a CGL formula over Ω. Let $\omega' \neq \omega$ be an outcome in Ω not mentioned (as a symbol in the outcome language) in φ. Such an outcome exists, since there are assumed to be infinitely many outcomes, and only finitely many can occur in φ. Let $\Gamma' = \langle N', \Omega, V', \sqsupseteq'_1, \dots, \sqsupseteq'_m\rangle$ be a game such that $V'(\{i\}) = \{\omega'\}$ ($V'(C)$ arbitrary for other C) and $\sqsupseteq'_i$ be such that ($\sqsupseteq_j$ arbitrary for $j \neq i$):

(a) $\omega \sqsupset'_i \omega'$ (ω is strictly better than ω' for i),
(b) if $\omega \sqsupseteq'_i \omega''$ then $\omega'' = \omega$ or $\omega'' = \omega'$ (for i, other than ω' there are no outcomes less than or equal to ω).

Clearly, such a preference relation exists. Observe that $\Gamma', \omega \not\models_{mcgl} \varphi'$. We now define Γ from Γ' by making i indifferent between ω and ω'. Let $\Gamma = \langle N, \Omega, V, \sqsupseteq_1, \dots, \sqsupseteq_m\rangle$ such that $N' = N$, $V = V'$, $\sqsupseteq_j = \sqsupseteq'_j$ for $j \neq i$ and

$$\sqsupseteq_i = \sqsupseteq'_i \cup \{(\omega', \omega)\}$$

It must be shown that $\sqsupseteq_i$ is indeed a preference relation. Reflexivity and completeness are straightforward. For transitivity, assume that $(\omega_1, \omega_2), (\omega_2, \omega_3) \in \sqsupseteq_i$; we must show that $(\omega_1, \omega_3) \in \sqsupseteq_i$. First, consider the case when $(\omega_1, \omega_2) = (\omega', \omega)$ (i.e., the new pair we added to the preference relation). We then have that $(\omega, \omega_3) \in \sqsupseteq_i$. Since $\omega \neq \omega'$, we have that $(\omega, \omega_3) \neq (\omega', \omega)$ and thus that $(\omega, \omega_3) \in \sqsupseteq'_i$. By (b) we have that either $\omega_3 = \omega$, in which case $(\omega_1, \omega_3) = (\omega', \omega)$ is in $\sqsupseteq_i$ by construction, or $\omega_3 = \omega'$ in which case $(\omega_1, \omega_3) = (\omega', \omega')$ is in $\sqsupseteq_i$ by reflexivity. Second, consider the case when $(\omega_2, \omega_3) = (\omega', \omega)$. Then $\omega_1 \sqsupseteq_i \omega'$. In the case that $\omega_1 = \omega$, $(\omega_1, \omega_3) = (\omega, \omega)$ and we are done due to reflexivity, so assume that $\omega_1 \neq \omega$. If it were not the case that $\omega_1 \sqsupseteq_i \omega_3$, then $\omega_3 \sqsupseteq_i \omega_1$ by completeness, and thus $\omega \sqsupseteq_i \omega_1$. Since $\omega_3 = \omega \neq \omega'$, we have that $(\omega_3, \omega_1) \neq (\omega', \omega)$, so $(\omega_3, \omega_1) \in \sqsupseteq'_i$. By (b), $\omega_1 = \omega'$, but then we have that $\omega_1 \sqsupseteq_i \omega_3$ by construction, contradicting the assumption. Finally, consider the case when neither $(\omega_1, \omega_2) = (\omega', \omega)$ nor $(\omega_2, \omega_3) = (\omega', \omega)$. In this case, $\omega_1 \sqsupseteq'_i \omega_2$ and $\omega_2 \sqsupseteq'_i \omega_3$, $\omega_1 \sqsupseteq'_i \omega_3$ by transitivity of $\sqsupseteq'_i$, and thus $\omega_1 \sqsupseteq_i \omega_3$. Thus, $\sqsupseteq_i$ is transitive, and a preference relation. Observe that $\Gamma, \omega \models_{mcgl} \varphi'$.

It remains to be shown that $\Gamma \models_{cgl} \varphi$ iff $\Gamma' \models_{cgl} \varphi$. First, we show that for any $\omega'' \in \Omega$, $\Gamma, \omega'' \models_{cgl} \psi$ iff $\Gamma', \omega'' \models_{cgl} \psi$, for any outcome language sub-formula ψ of φ. For the base case $\psi = \omega'''$. $\Gamma, \omega'' \models_{cgl} \omega'''$ iff $\omega'' = \omega'''$ iff $\Gamma', \omega'' \models_{cgl} \omega'''$. The inductive step is straightforward. We now show that for every sub-formula ψ of φ, we have that $\Gamma \models_{cgl} \psi$ iff $\Gamma' \models_{cgl} \psi$. For the base case, $\psi = \omega_1 \succeq_j \omega_2$. For $j \neq i$ we are done immediately, so let $j = i$. $\Gamma' \models_{cgl} \psi$ iff $\omega_1 \sqsupseteq'_i \omega_2$ iff, since $(\omega_1, \omega_2) \neq (\omega', \omega)$ (it was assumed that ω' is not mentioned in φ and ψ is a sub-formula of φ), $\omega_1 \sqsupseteq_i \omega_2$ iff $\Gamma \models_{cgl} \psi$. For the inductive step, let $\psi = \langle C\rangle\gamma$. $\Gamma' \models_{cgl} \psi$ iff there is an $\omega'' \in V'(C)$ such that $\Gamma', \omega'' \models_{cgl} \gamma$ iff there is an $\omega'' \in V(C)$ such that $\Gamma, \omega'' \models_{cgl} \gamma$ iff $\Gamma \models_{cgl} \psi$. The other cases in the inductive step are straightforward. □

Proposition 3 shows that MCGL can express properties of infinite games which cannot be expressed in CGL. An example of such a property (used in the proof of the proposition) is that for a given outcome ω there exist an $\omega' \in V(\{i\})$ such that $\omega' \sqsupset_i \omega$. Another, related, example is the expression of the property of ω being in the core. The CGL formula $CM(\omega)$ only expresses this property under the assumption of a finite set of outcomes; the MCGL formula *MCM* is true in an outcome of a game iff the outcome is in the core, regardless of whether the set of outcomes is finite or not. Of course, MCGL is not *strictly* more expressive, since, e.g., CGL (still) can refer to particular outcomes directly.

In addition to the ability to express properties such as core membership and core emptiness of infinite games, MCGL has another main advantage over CGL: *succinctness*. While MCGL cannot express everything CGL can express, the properties it *can* express can often be expressed much more succinctly (even for finite games). Again, consider the expressions of non-emptiness of the core, *CNE* and *MCNE* for CGL and MCGL, respectively. *CNE* will typically be a much longer expression, and its size increases quadratically with the number of outcomes in the game while the size of *MCNE* does not depend on the number of outcomes at all.

5.2. Coalition logic

As we noted in Section 2.1, it may be rather tempting to believe that the outcomes of coalitional games can be interpreted as states, and that the characteristic function can be interpreted as an effectivity function, and that as a consequence Coalition Logic (CL) [27] could be interpreted directly in coalitional games. In this section, we compare the semantics of CL on the one hand and CGL on the other, and show that in fact there is a fundamental difference between the two approaches. We also do the same for MCGL.

We first give a very brief review of some of the concepts of CL. A *coalition model* for agents N over a set of atomic propositions Φ_0 is a triple $M = (S, E, \pi)$, where S is a non-empty set of states, $\pi : \Phi_0 \to 2^S$ an assignment and E gives a function of type $2^N \to 2^{2^S}$ for each state $s \in S$. It is required that each $E(s)$ is a *playable effectivity function*, satisfying the following conditions:

(i) $\forall_{C \subseteq N} \emptyset \notin E(s)(C)$,
(ii) $\forall_{C \subseteq N} S \in E(s)(C)$,
(iii) for any X, if $S \setminus X \notin E(s)(N \setminus C)$ then $X \in E(s)(C)$ (*N-maximality*),
(iv) for all $X \subseteq X' \subseteq S$ and all C, if $X \in E(s)(C)$ then $X' \in E(s)(C)$ (*outcome monotonicity*),
(v) if $C_1 \cap C_2 = \emptyset$, $X_1 \in E(s)(C_1)$ and $X_2 \in E(s)(C_2)$, then $X_1 \cap X_2 \in E(s)(C_1 \cup C_2)$ (*superadditivity*).

Formulae of coalition logic, and their satisfaction in states s of coalition models M, are defined as follows[10]:

$$\begin{array}{lll} M, s \models p & \text{iff} & p \in \Phi_0 \text{ and } s \in \pi(p) \\ M, s \models \neg\psi & \text{iff} & M, s \not\models \psi \\ M, s \models \psi_1 \vee \psi_2 & \text{iff} & M, s \models \psi_1 \text{ or } M, s \models \psi_2 \\ M, s \models \langle C \rangle \psi & \text{iff} & \psi^M \in E(s)(C) \end{array}$$

where

$$\psi^M = \{s \in S: \ M, s \models \psi\}$$

A formula φ is valid, $\models \varphi$, if it satisfied by every state in every coalition model.

5.2.1. CGL *vs.* CL

If we take the set of atomic propositions to be $\Phi_0 = \Omega \cup \{\omega \succeq_i \omega' : \ \omega, \omega' \in \Omega, i \in N\}$, then we can read every formula in $\mathcal{L}_o \cup \mathcal{L}_c$ as a formula of coalition logic. Thus we can interpret $\mathcal{L}_c$ formulae in both a game Γ and in a pointed coalition model (M, s), and a $\mathcal{L}_o$ formula in both a pointed game (Γ, ω) and in a pointed model (M, s). Coalitional games and coalition models have many similarities. The former have "outcomes" while the latter have "states". An interesting question is: given a coalitional game Γ, does there exist an equivalent coalition model M with states corresponding to the outcomes of Γ, maybe in addition to a designated "initial" state t? Equivalence here means that Γ and the pointed coalition model (M, t) agree on $\mathcal{L}_c$ formulae and (Γ, ω) and (M, ω) agree on $\mathcal{L}_o$ formulae for any outcome ω. We can say that Γ and M then are *outcome-equivalent*.

In other words, a coalitional game Γ and a coalition model M, defined over atomic propositions Φ_0 above and having states $\Omega \cup \{t\}$, are outcome-equivalent iff for any $\varphi_1 \in \mathcal{L}_c$, any $\varphi_0 \in \mathcal{L}_o$ and any $\omega \in \Omega$, we have both:

(a) $\Gamma \models \varphi_1$ iff $M, t \models \varphi_1$, and
(b) $\Gamma, \omega \models \varphi_0$ iff $M, \omega \models \varphi_0$.

A natural question then is: given a game, does there exist an outcome-equivalent coalition model?

The answer, given by the following theorem, is "no", except for certain very special classes of games. The latter is the class of games where $V(C) = \{\omega\}$ for all coalitions $C \neq N$, for some fixed outcome $\omega \in \Omega$. To give them a name, we will call such games *limited games*, since, first, most games are not of this kind and, second, they are not very interesting. The only coalition in a limited game which possibly can select an outcome different from the fixed outcome ω is the grand coalition.

Theorem 14. *No non-limited coalitional game with more than one player has an outcome-equivalent coalition model.*

Proof. Let $\Gamma = \langle N, \Omega, V, \sqsupseteq_1, \ldots, \sqsupseteq_m \rangle$ be a coalitional game with more than one player, and assume that $M = (S, E, \pi)$ with $S = \Omega \cup \{t\}$ is a coalition model outcome-equivalent to Γ. We argue, by using the properties (i)–(v) of a playable effectivity function given above, that Γ must be limited. First, observe that for any $\mathcal{L}_o$ formula φ and coalition $C \neq \emptyset$, $\Gamma \models \langle C \rangle \varphi$ iff, by (a), $M, t \models \langle C \rangle \varphi$, i.e.

$$\varphi^M \in E(t)(C) \quad \Leftrightarrow \quad \exists_{\omega \in V(C)} \Gamma, \omega \models \varphi \tag{2}$$

for any $\varphi \in \mathcal{L}_0$, $C \subseteq N$, $C \neq \emptyset$.

Observe that $V(C) \neq \emptyset$ for any coalition $C \neq \emptyset$. This follows from (ii) and (2): $S = \top^M \in E(t)(C)$, where $\top$ is some tautology in the $\mathcal{L}_0$ language (e.g., $\omega' \vee \neg\omega'$), so $\exists_{\omega \in V(C)} \Gamma, \omega \models \top$, which ensures that $V(C)$ is non-empty.

[10] Pauly [27] uses $[C]$ where we use $\langle C \rangle$; here we use the latter notation for easier comparison.

We show that for any coalition $C \neq \emptyset$ and any $\omega \in \Omega$

$$\omega \in V(C) \quad \Leftrightarrow \quad \{\omega\} \in E(t)(C) \tag{3}$$

For the direction to the right, assume that $\omega \in V(C)$. $\Gamma, \omega \models \omega$ (note the dual role of ω as both an outcome and a formula), so $\omega^M \in E(t)(C)$ by (2). By (b), $M, \omega \models \omega$, so $\omega \in \omega^M$. If $\omega' \in \omega^M$, then $M, \omega' \models \omega$, $\Gamma, \omega' \models \omega$ by (b) and $\omega' = \omega$. Thus, $\{\omega\} = \omega^M \in E(t)(C)$. For the direction to the left, let $\{\omega\} \in E(t)(C)$. Since, again, $\omega^M = \{\omega\}$, by (2) there is a $\omega' \in V(C)$ such that $\Gamma, \omega' \models \omega$. This is only the case when $\omega = \omega' \in V(C)$.

For any non-empty disjoint coalitions C_1 and C_2:

$$\big(\omega_1 \in V(C_1) \text{ and } \omega_2 \in V(C_2)\big) \quad \Rightarrow \quad \omega_1 = \omega_2 \tag{4}$$

$$V(C_1) = V(C_2) \tag{5}$$

$$\big|V(C_1)\big| = 1 \tag{6}$$

We prove (4)–(6):

(4) Assume otherwise, that $\omega_1 \in V(C_1)$ and $\omega_2 \in V(C_2)$ and $\omega_1 \neq \omega_2$. By (3), $\{\omega_1\} \in E(t)(C_1)$ and $\{\omega_2\} \in E(t)(C_2)$, and by superadditivity it must be the case that $\emptyset = \{\omega_1\} \cap \{\omega_2\} \in E(t)(C_1 \cup C_2)$, but this contradicts (i). Thus, (4) must hold.
(5) Assume that $\omega \in V(C_1)$; we show that $\omega \in V(C_2)$. Since $V(C_2)$ is non-empty, let $\omega' \in V(C_2)$. By (4), $\omega' = \omega$. Thus $\omega \in V(C_2)$. By a symmetric argument, $\omega \in V(C_2)$ implies that $\omega \in V(C_1)$.
(6) Since $V(C_1)$ is non-empty, there is an $\omega_1 \in V(C_1)$. If $\omega_2 \in V(C_1)$, then $\omega_2 \in V(C_2)$ by (5) and $\omega_1 = \omega_2$ by (4). Thus, $V(C_1) = \{\omega_1\}$.

Let $a, b \in N$ such that $a \neq b$ (existence is ensured by the assumption of more than one player). By (5) and (6) there is an ω_1 such that $V(\{a\}) = V(\{b\}) = \{\omega_1\}$. For any $d \in N$ such that $d \neq a$ and $d \neq b$, $\{a\}$ and $\{d\}$ are disjoint and we again get that $V(\{d\}) = V(\{a\}) = \{\omega_1\}$. Thus, $V(\{d\}) = \{\omega_1\}$ for any $d \in N$. Let $C \subset N$ be a coalition different from the grand coalition. There is a $d \in N$ such that C and $\{d\}$ are disjoint, so by (5) $V(C) = V(\{d\}) = \{\omega_1\}$, which shows that Γ is limited. □

Thus, in general, a coalitional game is *not* simply a coalition model with outcomes as states. Even though the language of Coalition Logic is similar to the language of CGL, it follows from Theorem 14 that we cannot use the semantic rules of Coalition Logic directly to say whether a formula is true or not in a coalitional game. The main reason is that a difference between outcomes in coalitional games and states in coalition models is that an outcome is *local* to the coalition which chooses it, while states are *global*. As a consequence, while it is perfectly possible in a coalitional game that both a coalition C can choose outcome ω ($\omega \in V(C)$) *and* a coalition C', C' and C disjoint, can choose outcome ω' ($\omega' \in V(C')$) when $\omega' \neq \omega$, it is not possible in a coalition model that both C is effective for $\{\omega\}$ and C' is effective for $\{\omega'\}$. The proof of Theorem 14 shows that in general there is no playable effectivity function corresponding, in the sense of (3), to a characteristic function.

When it comes to logical properties of the two logics, in the form of valid formulae, it is straightforward to see that the logics differ. To compare validities, we must take the set of atomic propositions Φ_0 for CL as defined above, and it is the formulae in $\mathcal{L}_c$ which are relevant since they are formulae of both logics ($\mathcal{L}_o$ are formulae of CL in this case, but not of CGL).

Theorem 15. *There are formulae $\varphi, \psi \in \mathcal{L}_c$ such that (here we use subscripts on the satisfiability relations with the obvious meaning):*

(1) $\models_{CL} \varphi$ *but* $\not\models_{CGL} \varphi$,
(2) $\models_{CGL} \psi$ *but* $\not\models_{CL} \psi$.

Proof. Let $C_1 \cap C_2 = \emptyset$ and $\omega_1 \neq \omega_2 \in \Omega$. Take $\varphi = (\langle C_1\rangle\omega_1 \wedge \langle C_2\rangle\omega_2) \rightarrow (\omega_1 \wedge \omega_2)$. This formula (superadditivity) is valid in CL, but not in CGL. To see the latter, observe that in fact $\neg\langle C\rangle(\omega_1 \wedge \omega_2)$ is valid in CGL for any C when $\omega_1 \neq \omega_2$ (and there are obviously games where the antecedent of φ holds). This immediately gives us the second claim: take $\psi = \neg\langle C\rangle(\omega_1 \wedge \omega_2)$. ψ is not valid in CL. Another example is to take $\varphi = \langle C_1\rangle\omega \rightarrow \langle C_2\rangle\omega$, where $C_1 \subseteq C_2$ (coalition monotonicity). □

In other words, the axiomatisation of CGL is indeed different from that of CL.

5.2.2. MCGL *vs.* CL

The languages of CL and MCGL are similar, too. Under the surface of the syntactic similarities, however, is a bigger conceptual difference than between CL and CGL. In the two latter logics, the meaning of an expression of the form $\langle C\rangle\varphi$ is related to what the coalition C can *achieve* or *make come about*. The intended meaning of $\langle C\rangle\varphi$ in MCGL is fundamentally different: it means that C *prefers* φ. One could nevertheless ask whether a coalition model could "emulate" a coalition model under this interpretation, and we now discuss that question.

The language of CL can be seen as a subset of the MCGL language. To compare the two types of models we need the same language. Thus, we fix the set of agents N, and we consider the restricted MCGL language $\mathcal{L}^-$ defined by the following grammar: $\varphi ::= p_C \mid \langle C\rangle\varphi \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2$, $C \in \mathcal{C}$. By taking $\Phi_0 = \{p_C\colon C \in \mathcal{C}\}$ in CL and $\Theta' = \emptyset$ in MCGL, $\mathcal{L}^-$ is the same as the language of CL when the latter is restricted such that expressions of the form $\langle\emptyset\rangle\varphi$ are not allowed. In a sense, $\mathcal{L}^-$ is the least common denominator between the two respective languages.

We thus say that a coalitional game Γ and a coalition model M are *equivalent* if they are defined over the same set N of agents, $\Phi_0 = \{p_C\colon C \in \mathcal{C}\}$, $\Theta' = \emptyset$, $S = \Omega$, and for every $\varphi \in \mathcal{L}^-$,

$$\Gamma, \omega \models \varphi \quad \text{iff} \quad M, \omega \models \varphi$$

for every $\omega \in \Omega$.

The following Theorem 16 shows, in a similar way to Theorem 14, that except for very trivial games, MCGL properties of coalitional games do not correspond to CL properties of coalition models.

Two coalitional games Γ_1, Γ_2 are *logically equivalent* (w.r.t. $\mathcal{L}^-$) if they have the same set of agents and the same set of outcomes Ω and it is the case that for any formula $\varphi \in \mathcal{L}^-$, $\Gamma_1, \omega \models \varphi$ iff $\Gamma_2, \omega \models \varphi$.

Theorem 16. *If a coalitional game Γ is equivalent to a coalition model M, then Γ is logically equivalent to a game Γ' in which all agents have the same preference relation.*

Proof. Assume that $\Gamma, \omega \models \varphi$ iff $M, \omega \models \varphi$, for any ω and $\varphi \in \mathcal{L}^-$. Let ω be an outcome. Let $C' \in \mathcal{C}$, and let $C \subseteq C'$. Playable effectivity functions are *coalition-monotone*: we have that $E(\omega)(C) \subseteq E(\omega)(C')$ ([27], Lemma 3.1). This means that $M, \omega \models \langle C\rangle\varphi \to \langle C'\rangle\varphi$, and thus that $\Gamma, \omega \models \langle C\rangle\varphi \to \langle C'\rangle\varphi$, for any φ. It is easy to see that $\Gamma, \omega \models \langle C'\rangle\varphi \to \langle C\rangle\varphi$ for any φ, from the semantics of MCGL. Thus, we have that

$$\Gamma, \omega \models \langle C\rangle\varphi \quad \text{iff} \quad \Gamma, \omega \models \langle C'\rangle\varphi \tag{7}$$

for any ω, any φ, and any $C \subseteq C'$.

Let Γ' be like Γ, except that $\sqsupseteq'_i = \sqsupseteq_1$ for every $i \in N$ (all agents have the same preferences). We argue that Γ and Γ' are logically equivalent w.r.t. $\mathcal{L}^-$. This can be shown by structural induction over φ: the $p_C, \neg, \wedge$ cases are straightforward, so it suffices to show that

$$\Gamma, \omega \models \langle C\rangle\psi \quad \text{iff} \quad \Gamma', \omega \models \langle C\rangle\psi$$

for any C and ω and ψ under the induction hypothesis. $\Gamma, \omega \models \langle C\rangle\psi$ iff, by (7), $\Gamma, \omega \models \langle N\rangle\psi$ iff, again by (7), $\Gamma, \omega \models \langle\{1\}\rangle\psi$ iff there is a $\omega' \in \Omega$ such that $\omega' \sqsupseteq_1 \omega$ and $\Gamma, \omega' \models \psi$. By the induction hypothesis and the facts that $\omega' \sqsupseteq_1 \omega$ iff $\omega' \sqsupseteq'_i \omega$ for any i and $\sqsupseteq'_i = \sqsupseteq'_j$ for any i, j, this holds iff $\Gamma', \omega \models \langle C\rangle\psi$. Thus, Γ is logically equivalent to Γ'. □

Again, it is easy to see that the two logics differ on the level of validities (in the language $\mathcal{L}^-$).

Theorem 17. *There are formulae $\varphi, \psi \in \mathcal{L}^-$ such that (here we use subscripts on the satisfiability relations with the obvious meaning):*

(1) $\models_{CL} \varphi$ *but* $\not\models_{MCGL} \varphi$,
(2) $\models_{MCGL} \psi$ *but* $\not\models_{CL} \psi$.

Proof. Let $C \subseteq C'$. Take $\psi = \langle C'\rangle\gamma \to \langle C\rangle\gamma$ for some arbitrary γ. It is easy to see that ψ is valid in MCGL (if everyone in C' prefers γ then everyone in C prefers γ) but not in CL (a coalition can typically achieve *more* than its proper subsets). Take φ to be the converse; $\varphi = \langle C\rangle\gamma \to \langle C'\rangle\gamma$. φ is valid in CL (coalition monotonicity), but it is easy to see that it is not valid in MCGL. □

6. Discussion and future work

In summary, we have introduced two different logics for representing and reasoning about coalitional games without transferable payoffs. We presented Coalitional Game Logic (CGL), gave a complete axiomatisation for it, showed it was expressively complete with respect to finite coalitional games, showed that the satisfiability problem was NP-complete, gave a decision procedure for the logic, showed how the logic could be used to capture a range of solution concepts for finite coalitional games, and finally, showed formally why the logic was fundamentally different to existing cooperation logics. We presented Modal Coalitional Game Logic (MCGL), used it to characterise solution concepts for general (not necessarily finite) coalitional games, and gave a complete axiomatisation of it. CGL is formally more expressive than MCGL, but can only be used to express properties such as non-emptiness of the core for games with finitely many outcomes. MCGL can on the other hand express such properties also of games with infinitely many outcomes, and in addition more succinctly.

Logics for reasoning about properties of coalitions in general and cooperation between agents in particular have received much attention lately, the most prominent frameworks being Pauly's Coalition Logic (CL) and Alur, Henzinger and Kupferman's ATL. By using standard tools and techniques developed in AI and computer science, we can employ such logics for, e.g., model checking, automated theorem proving and automated synthesis of models. The key construct of CL/ATL is of the form $\langle C\rangle\varphi$, where C is a coalition, with the intended meaning that C can achieve φ. Syntactically, such logics resemble CGL in particular, and so it is natural to ask whether we could just use CL/ATL to represent properties of coalitional games. What are the practical differences between the logics we have developed in this paper and CL—what, if anything, can be done with the former that cannot be done with the latter? For example, can we express properties such as non-emptiness of the core in CL? Let us give some answers. The results of Section 5.2 imply that we cannot directly use CL/ATL for this purpose and, more importantly, that CL/ATL and CGL are *fundamentally different logics*, with quite different validities. In contrast, CGL *can* be directly and transparently used for this purpose, since it is interpreted over coalitional games. We showed that, MCGL, too, is fundamentally different from CL/ATL.

In future work, the computational complexity of MCGL should be studied. In our complexity analysis for CGL we assumed an explicit representation for games. In some circumstances this assumption might not be practical, and more succinct representations of games must be used. Several succinct representations of coalitional games have been studied in the literature [8,9,20], and in future work the complexity of model checking against such representations should be studied. This is particularly relevant for games with an infinite set of outcomes, which obviously must be finitely represented. One particular class of games that have an infinite set of outcomes when modelled as games without transferable utilities is of course the class of games *with* transferable utilities. Model checking CGL and MCGL directly against succinct representations of TU games (such as marginal contribution nets [20]) is one particularly interesting possibility. Also very interesting for future work is to find restricted classes of games that are tractable with respect to satisfiability checking (for both CGL and MCGL).

It would also be interesting to consider the logical characterisation of other solution concepts for coalitional games [24]. The Shapley value, in particular, would be interesting to analyse. A key issue would be how to capture the additivity axiom of the Shapley value, since this seems to require quantifying over games. Perhaps the obvious line of attack would be to consider NTU versions of the Shapley value—several have been proposed in the literature.

Acknowledgements

The research reported in this paper started when the first author was visiting the Department of Computer Science, University of Liverpool, supported by grant 166525/V30 from the Research Council of Norway. We thank the reviewers for their helpful suggestions, which have enabled us to substantially improve the paper. Note that parts of Section 3 were previously presented in the conference paper [2].

References

[1] J. Abdou, H. Keiding, Effectivity Functions in Social Choice Theory, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1991.
[2] T. Ågotnes, W. van der Hoek, M. Wooldridge, On the logic of coalitional games, in: Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2006), Hakodate, Japan, 2006.
[3] R. Alur, T.A. Henzinger, O. Kupferman, Alternating-time temporal logic, Journal of the ACM 49 (5) (September 2002) 672–713.
[4] R.J. Aumann, M. Maschler, The bargaining set for cooperative games, in: Advances in Game Theory, Annals of Mathematics Studies 52 (1964) 443–467.
[5] J. Bilbao, J. Fernández, J. López, Complexity in cooperative game theory, Manuscript.
[6] P. Blackburn, M. de Rijke, Y. Venema, Modal Logic, Cambridge University Press, Cambridge, England, 2001.
[7] E.M. Clarke, O. Grumberg, D.A. Peled, Model Checking, The MIT Press, Cambridge, MA, 2000.
[8] V. Conitzer, T. Sandholm, Complexity of determining nonemptiness of the core, in: Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI-03), Acapulco, Mexico, 2003, pp. 613–618.
[9] V. Conitzer, T. Sandholm, Computing Shapley values, manipulating value division schemes, and checking core membership in multi-issue domains. in: Proceedings of the Nineteenth National Conference on Artificial Intelligence (AAAI-2004), San Jose, CA, 2004, pp. 219–225.
[10] V. Conitzer, T. Sandholm, Complexity of constructing solutions in the core based on synergies among coalitions, Artificial Intelligence 170 (2006) 607–619.
[11] M. de Rijke, Extended modal logic, PhD thesis, University of Amsterdam, 1993.
[12] X. Deng, C.H. Papadimitriou, On the complexity of cooperative solution concepts, Mathematics of Operations Research 19 (2) (1994) 257–266.
[13] E. Elkind, L. Goldberg, P. Goldberg, M. Wooldridge, Computational complexity of weighted threshold games, in: Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence (AAAI-2007), Vancouver, British Columbia, Canada, 2007.
[14] E.A. Emerson, Temporal and modal logic, in: J. van Leeuwen (Ed.), Handbook of Theoretical Computer Science, vol. B: Formal Models and Semantics, Elsevier Science Publishers B.V., Amsterdam, The Netherlands, 1990, pp. 996–1072.
[15] D.B. Gillies, Solutions to general non-zero-sum games, in: Contribution to the Theory of Games, IV, Annals of Mathematics Studies 40 (1959) 47–85.
[16] P. Harrenstein, Logic in conflict, PhD thesis, Utrecht University, 2004.
[17] P. Harrenstein, W. van der Hoek, J.-J. Meyer, C. Witteven, A modal characterization of Nash equilibrium, Fundamenta Informaticae 57 (2–4) (2003) 281–321.
[18] I. Horrocks, U. Hustadt, U. Sattler, R. Schmidt, Computational modal logic, in: P. Blackburn, J. van Benthem, F. Wolter (Eds.), Handbook of Modal Logic, Elsevier Science Publishers B.V., Amsterdam, The Netherlands, 2007, pp. 181–245.
[19] G.E. Hughes, M.J. Cresswell, Introduction to Modal Logic, Methuen and Co., Ltd., 1968.
[20] S. Ieong, 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.
[21] H.J. Levesque, All I know: a study in autoepistemic logic, Artificial Intelligence 42 (2–3) (March 1990) 263–309.
[22] J. Von Neumann, O. Morgenstern, Theory of Games and Economic Behaviour, Princeton University Press, Princeton, NJ, 1944.
[23] N. Ohta, A. Iwasaki, M. Yokoo, K. Maruono, V. Conitzer, T. Sandholm, A compact representation scheme for coalitional games in open anonymous environments, in: Proceedings of the Twenty-First National Conference on Artificial Intelligence (AAAI-2006), Boston, MA, 2006.
[24] M.J. Osborne, A. Rubinstein, A Course in Game Theory, The MIT Press, Cambridge, MA, 1994.
[25] C.H. Papadimitriou, Computational Complexity, Addison-Wesley, Reading, MA, 1994.
[26] M. Pauly, Logic for social software, PhD thesis, University of Amsterdam, 2001. ILLC Dissertation Series 2001-10.
[27] M. Pauly, A modal logic for coalitional power in games, Journal of Logic and Computation 12 (1) (2002) 149–166.

[28] M. Pauly, M. Wooldridge, Logic for mechanism design—a manifesto, in: Proceedings of the 2003 Workshop on Game Theory and Decision Theory in Agent Systems (GTDT-2003), Melbourne, Australia, 2003.
[29] T. Sandholm, Distributed rational decision making, in: G. Weiß (Ed.), Multiagent Systems, The MIT Press, Cambridge, MA, 1999, pp. 201–258.
[30] O. Shehory, S. Kraus, Coalition formation among autonomous agents: Strategies and complexity, in: C. Castelfranchi, J.-P. Müller (Eds.), From Reaction to Cognition—Fifth European Workshop on Modelling Autonomous Agents in a Multi-Agent World, MAAMAW-93, in: LNAI, vol. 957, Springer-Verlag, Berlin, Germany, 1995, pp. 56–72.
[31] O. Shehory, S. Kraus, Task allocation via coalition formation among autonomous agents, in: Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence (IJCAI-95), Montréal, Québec, Canada, August 1995, pp. 655–661.
[32] O. Shehory, S. Kraus, Methods for task allocation via agent coalition formation, Artificial Intelligence 101 (1–2) (1998) 165–200.
[33] R.M. Smullyan, First-Order Logic, Springer-Verlag, Berlin, Germany, 1968.
[34] W. van der Hoek, W. Jamroga, M. Wooldridge, A logic for strategic reasoning, in: Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2005), Utrecht, The Netherlands, 2005, pp. 157–164.
[35] W. van der Hoek, J.-J. Meyer, Making some issues of implicit knowledge explicit, International Journal of Foundations of Computer Science 3 (2) (1992) 193–224.
[36] W. van der Hoek, J.-J. Meyer, A complete epistemic logic for multiple agents—combining distributed and common knowledge, in: Epistemic Logic and the Theory of Games and Decisions, Kluwer Academic Publishers, Dordrecht, The Netherlands, 1997, pp. 35–68.
[37] W. van der Hoek, M. Pauly, Modal logic for games and information, in: P. Blackburn, J. van Benthem, F. Wolter (Eds.), Handbook of Modal Logic, Elsevier Science Publishers B.V., Amsterdam, The Netherlands, 2006, pp. 1077–1148.
[38] F. van Harmelen, V. Lifschitz, B. Porter, Handbook of Knowledge Representation, Elsevier Science Publishers B.V., Amsterdam, The Netherlands, 2007.
[39] M. Wooldridge, T. Ågotnes, P.E. Dunne, W. van der Hoek, Logic for automated mechanism design—a progress report, in: Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence (AAAI-2007), Vancouver, British Columbia, Canada, 2007.
[40] M. Wooldridge, P.E. Dunne, On the computational complexity of qualitative coalitional games, Artificial Intelligence 158 (1) (2004) 27–73.
[41] M. Wooldridge, P.E. Dunne, On the computational complexity of coalitional resource games, Artificial Intelligence 170 (10) (2006) 853–871.