A logical characterisation of qualitative coalitional games

Paul E. Dunne — **Wiebe van der Hoek** — **Michael Wooldridge**

Department of Computer Science
University of Liverpool
Liverpool L69 7ZF (UK)

{ped, wiebe, mjw}@csc.liv.ac.uk

ABSTRACT. *Qualitative coalitional games (QCGs) were introduced as abstract formal models of goal-oriented cooperative systems. A QCG is a game in which each agent is assumed to have some goal to achieve, and in which agents must typically cooperate with others in order to satisfy their goals. In this paper, we show how it is possible to reason about QCGs using Coalition Logic (CL), a formalism intended to facilitate reasoning about coalitional powers in game-like multi-agent systems. We introduce a correspondence relation between QCGs and interpretations for CL, which defines the circumstances under which a CL interpretation correctly characterises a QCG. The complexity of deciding correspondence between QCGs and interpretations for CL is shown to vary from being tractable up to Π_2^p-complete, depending on the representation chosen for the QCG and interpretation. We then show how various properties and solution concepts of QCGs can be characterised as CL formula schemes. The ideas are illustrated via a detailed worked example, in which we demonstrate how a model checker can be deployed to investigate whether a particular system has the properties in question.*

KEYWORDS: *coalitional games, cooperative games, modal logic, coalition logic.*

1. Introduction

One of the fundamental research objectives in the multi-agent systems community is to build software agents that can cooperate with other such agents in order to efficiently carry out tasks on behalf of some user or owner (Wooldridge 2002). It is widely accepted that, in order for software agents to be able to do this, they must be able to represent and reason about the multi-agent encounters in which they find themselves. This has motivated the development of knowledge representation and reasoning formalisms for multi-agent systems. Historically, such formalisms have fallen into one of two categories: those that use modal logic to represent the dynamic infor-

Journal of Applied Non-Classical Logics. Volume 17 – No. 4/2007, page 477 to 509

mational and motivational components of rational agents and their decision making processes (Wooldridge et Jennings 1995, Hoek et Wooldridge 2003c), and those that are based on a game-theoretic analysis of the scenario at hand (Kraus 1997, Sandholm 1999).

The present paper is part of a burgeoning body of work that seeks to synthesise modal logic and game theoretic approaches for reasoning about multi-agent systems. We show how a particular type of modal logic can be used to reason about a particular type of cooperative game. In more detail, we show how *Coalition Logic* (Pauly 2002b, Pauly 2002a, Pauly 2001) can be used to reason about *Qualitative Coalitional Games* (Wooldridge et Dunne 2004).

Pauly's Coalition Logic (hereafter, CL) is a formalism for reasoning about coalitional powers in game-like multi-agent systems (Pauly 2002b, Pauly 2002a, Pauly 2001). Pauly showed how the semantic structures underpinning CL could be formally understood as games of various types; he gave correspondence results between properties of the games and axioms of the logic, gave complete axiomatizations of the various resulting logics, determined the computational complexity of the satisfiability and model checking problems for his logics, and in addition, demonstrated how these logics could be applied to the formal specification and verification of social choice procedures.

Qualitative Coalitional Games (QCGs) were introduced by Wooldridge and Dunne as an abstract formal model of goal-oriented cooperative systems (Wooldridge *et al.* 2004). In a QCG, each agent is assumed to have a set of goals: an agent is "satisfied" with any outcome that accomplishes one of its goals, but is indifferent about *which* of its goals should be achieved – all are considered equally good (*cf.* individual rational). Each potential coalition is then modelled as having a set of choices available, intuitively corresponding to the different ways in which they could choose to cooperate. Associated with each choice is a set of goals, which would be achieved if the coalition chose to cooperate in this way. QCGs seem an appropriate abstract framework within which to reason about goal-oriented multi-agent systems, where numeric utility values (as in conventional coalitional games (Osborne et Rubinstein 1994, pp. 255–312)) may be either inappropriate or else impossible to derive.

Our aim in this paper is to show how CL can be used to represent and reason about QCGs. Specifically, the paper makes the following main contributions.

– First, we define a *correspondence relation*, "$\simeq$" between QCGs and interpretations for CL. The idea is that this relation characterises the circumstances under which a QCG and an interpretation for CL say the same things about coalitions, and hence are "equivalent" at this level of analysis. (Correspondence can thus be loosely understood as a kind of bisimulation between games and interpretations – *cf.* (Blackburn, de Rijke et Venema 2001, pp. 64–73).)

– Second, we investigate and characterise the computational complexity of the problem of deciding whether a QCG and an interpretation correspond to one-another for two different representations of QCGs and interpretations. We show that, with an extensive representation (in which we explicitly enumerate the components of the two structures), the problem is decidable in time polynomial in the size of the structures – although the structures are in general unfeasibly large. We also show that, for a symbolic representation (in which we represent the components of the QCG and interpretation via logical formulae), the problem is Π_2^p-complete. We argue that "practical" representations for such a problem are much closer to the symbolic representation, and therefore that the negative Π_2^p-completeness result is likely to be a better indication of the practical complexity of deciding correspondence.

– Third, we show how the properties and solution concepts of QCGs that were introduced in (Wooldridge *et al.* 2004) can be characterised as formulae of CL. That is, for each of the concepts we consider, we define a CL predicate, and we then prove that this predicate corresponds to the claimed property, in that it will be satisfied in an interpretation corresponding to a QCG iff the QCG itself has the property. This gives us a syntactic characterisation of QCG properties, that is somewhat analogous to the modal characterisation of first-order relational properties of Kripke structures in the correspondence theory of modal logic (Benthem 1984)[1].

– Fourth, and finally, we demonstrate how the ideas set out in the paper can be applied to the analysis of a concrete computational system by means of a model checking system for CL (Alur, de Alfaro, Henzinger, Krishnan, Mang, Qadeer, Rajamani et Taşiran 2000, Alur, Henzinger, Mang, Qadeer, Rajamani et Taşiran 1998).

We conclude with a brief discussion of related work, and give some pointers to future research directions.

2. Qualitative coalitional games

We give a brief introduction to Qualitative Coalitional Games (QCGs): details may be found in (Wooldridge *et al.* 2004). A QCG contains a (non-empty, finite) set $\mathcal{A} = \{1, \ldots, m\}$ of *agents*. Each agent $i \in \mathcal{A}$ is assumed to have associated with it a (finite) set $\mathcal{G}_i$ of *goals*, drawn from a set of overall possible goals $\mathcal{G}$. The intended interpretation is that the members of $\mathcal{G}_i$ represent all the individual rational outcomes for i – intuitively, the outcomes that give it "better than zero utility". That is, agent i would be happy if *any* member of $\mathcal{G}_i$ were achieved – then it has "gained something". But, in QCGs, we are not concerned with preferences over individual goals. Thus, at this level of modelling, i is *indifferent* among the members of $\mathcal{G}_i$: it will be *satisfied* if *at least one* member of $\mathcal{G}_i$ is achieved, and *unsatisfied* otherwise. Note that cases where more than one of an agent's goals are satisfied are not an issue – an agent's aim will simply be to ensure that at least one of its goals is achieved, and there is no sense of an agent i attempting to satisfy as many members of $\mathcal{G}_i$ as possible.

1. It is worth pointing out that we are unable to characterise three of the properties of QCGs that were discussed in (Wooldridge *et al.* 2004): *unattainable goal set*, *global unattainability*, and *incomplete game*. The reason for omitting these is simply that the form of quantification required to express these cannot be captured directly in Coalition Logic; possible extensions to Coalition Logic to capture them would be an interesting future avenue for research.

A *coalition*, typically denoted by C, is simply a set of agents, ***i.e.***, a subset of $\mathcal{A}$. The *grand coalition* is the set of all agents, $\mathcal{A}$. We assume that each possible coalition has available to it a set of possible *choices*, where each choice intuitively characterises the outcome of one way that the coalition could cooperate. We model the choices available to coalitions via a *characteristic function* with the signature

$$\mathcal{V} : 2^{\mathcal{A}} \to 2^{2^{\mathcal{G}}}.$$

Thus, in saying that $G \in \mathcal{V}(C)$ for some coalition $C \subseteq \mathcal{A}$, we are saying that one choice available to the coalition C is to bring about *exactly* the goals in G. At this point, the reader might expect to see some constraints placed on characteristic functions. For example, at first sight the following *monotonicity* constraint might seem natural:

$$C \subseteq C' \text{ implies } \mathcal{V}(C) \subseteq \mathcal{V}(C')$$

Although such a constraint is entirely appropriate for many scenarios, there are cases where such a constraint is not appropriate[2].

The only requirement that we put on a QCG-game Γ is that if the empty coalition $\emptyset$ is able to bring about a set of goals G, then *any* coalition can bring them about:

$$\forall G \subseteq \mathcal{G} \forall C \subseteq \mathcal{A} : G \in \mathcal{V}(\emptyset) \Rightarrow G \in \mathcal{V}(C) \tag{1}$$

Formally, a QCG Γ is an $(m+3)$-tuple (Wooldridge *et al.* 2004, p.33):

$$\Gamma = \langle \mathcal{A}, \mathcal{G}, \mathcal{G}_1, \ldots, \mathcal{G}_m, \mathcal{V} \rangle$$

where:

– $\mathcal{A} = \{1, \ldots, m\}$ is a set of *agents*;
– $\mathcal{G} = \{g_1, \ldots, g_n\}$ is a set of *possible goals*;
– $\mathcal{G}_i \subseteq \mathcal{G}$ is a set of goals for each agent $i \in \mathcal{A}$, the intended interpretation being that any of the goals in $\mathcal{G}_i$ would satisfy i – but i is indifferent between the members of $\mathcal{G}_i$;
– $\mathcal{V} : 2^{\mathcal{A}} \to 2^{2^{\mathcal{G}}}$ is a *characteristic function*, which for every coalition $C \subseteq \mathcal{A}$ determines a set $\mathcal{V}(C)$ of *choices*, the intended interpretation being that if $G \in \mathcal{V}(C)$, then one of the choices available to coalition C is to bring about *all* the goals in G simultaneously. In this paper, we will assume that in any QCG Γ, the characteristic function $\mathcal{V}$ satisfies (1).

We say a set of goals G *satisfies* agent i if $G \cap \mathcal{G}_i \neq \emptyset$; we say that G satisfies $C \subseteq \mathcal{A}$ if it satisfies every member of C. Also, we say that G is *feasible* for coalition C if $G \in \mathcal{V}(C)$.

2. For example, consider a legal scenario in which certain coalitions are forbidden by monopoly or anti-trust laws.

3. Coalition logic

The logic we use throughout the remainder of this paper is known as *Coalition Logic* (Pauly 2002b, Pauly 2002a, Pauly 2001). It was introduced by Pauly as a framework for representing and reasoning about the powers of coalitions in game-like multi-agent encounters. CL may be regarded as the “next time” fragment of the Alternating-time Temporal Logic (ATL) of Alur, Henzinger, and Kupferman (Alur, Henzinger et Kupferman 2002); see (Goranko 2001) for a discussion of the relationship between CL and ATL. In this section, we will give a complete definition of the logic, although of necessity, our presentation will be somewhat simplified and occasionally somewhat terse; see the references above for details.

Informally, CL is a propositional modal logic, containing an indexed collection of unary modal operators $[C]$, where C is a set of agents. The intended interpretation of a formula $[C]\varphi$ is that the set of agents (coalition) C are *effective* for φ. That is, the agents C could cooperate to ensure that, in the next state of the environment, φ was true. We refer to an expression of the form $[C]\varphi$ as a *coalition* or *cooperation* modality.

3.1. *Syntax and semantics*

Syntactically, formulae φ of CL are defined over a set $\mathcal{A}$ of agents and a set Φ_0 of atomic formulae by the following grammar:

$$\varphi ::= \top \mid p \mid \neg\varphi \mid \varphi \vee \varphi \mid [C]\varphi$$

where $p \in \Phi_0$ is an atomic proposition and $C \subseteq \mathcal{A}$ is a set of agents. We usually omit set brackets in coalition modalities, for example by writing $[1, 2, 3]$ instead of $[\{1, 2, 3\}]$. As usual, we use parentheses to disambiguate formulae where necessary, and define the remaining connectives of classical logic as abbreviations: $\bot \mathrel{\hat{=}} \neg\top$, $\varphi \to \psi \mathrel{\hat{=}} (\neg\varphi) \vee \psi$ and $\varphi \leftrightarrow \psi \mathrel{\hat{=}} (\varphi \to \psi) \wedge (\psi \to \varphi)$.

Semantically, a *model*, $\mathcal{M}$, for CL is a quintuple:

$$\mathcal{M} = \langle \mathcal{A}, \mathcal{S}, \mathcal{E}, \Phi_0, \upsilon \rangle$$

where:

– $\mathcal{A} = \{1, \ldots, m\}$ is a fixed, finite, non-empty set of *agents* (as in QCGs);
– $\mathcal{S} = \{s_1, \ldots, s_o\}$ is a fixed, finite, non-empty set of *states*;
– $\mathcal{E} : 2^{\mathcal{A}} \times \mathcal{S} \to 2^{2^{\mathcal{S}}}$ is an *effectivity function*, where $S \in \mathcal{E}(C, s)$ is intended to mean that from state s, the coalition C can cooperate to ensure that the next state will be a member of S – note that they cannot determine *which* of the members of S will occur – they can only be sure that it will be *some* member of S;
– Φ_0 is the set of propositional variables for $\mathcal{M}$; and

– $\upsilon : \mathcal{S} \rightarrow 2^{\Phi_0}$ is a valuation function, which for every state $s \in \mathcal{S}$ gives the set $\upsilon(s)$ of propositional variables that are satisfied at s.

It is possible to define a number of constraints on effectivity functions, depending upon exactly which kinds of scenario they are intended to model (Pauly 2001, pp. 24–39). For the purposes of this paper, we shall assume just one property of effectivity functions: that the empty coalition is ineffectual. We capture this by requiring that the empty coalition has no power to do anything other than ensure that the model is closed, in the sense that the next state will be one of the defined possible states. Formally: $\mathcal{E}(\emptyset, s) = \{\mathcal{S}\}$, for all s.

An *interpretation* for CL is a pair $\mathcal{M}, s$, where $\mathcal{M}$ is a model and s is a state in $\mathcal{M}$. The satisfaction relation "$\models$" for CL holds between interpretations and formulae of CL. The satisfaction relation is defined by the following inductive rules:

$\mathcal{M}, s \models \top$
$\mathcal{M}, s \models p$ iff $p \in \upsilon(s)$ (where $p \in \Phi_0$)
$\mathcal{M}, s \models \neg\varphi$ iff $\mathcal{M}, s \not\models \varphi$
$\mathcal{M}, s \models \varphi \vee \psi$ iff $\mathcal{M}, s \models \varphi$ or $\mathcal{M}, s \models \psi$
$\mathcal{M}, s \models [C]\varphi$ iff $\exists S \in \mathcal{E}(C, s)$ such that $\forall s' \in S$, we have $\mathcal{M}, s' \models \varphi$.

Sometimes, when we fix the root of the interpretation, we also will write $(\mathcal{M}, \rho)$, in which cases it is implicitly assumed that $\rho \in \mathcal{S}$. Note that, since $\mathcal{E}(\emptyset, s) = \{\mathcal{S}\}$, the emptyset coalition modality "$[\emptyset]$" acts as a "global" or "universal" modality (Blackburn *et al.* 2001, p.367):

$$\mathcal{M}, s \models [\emptyset]\varphi \quad \text{iff} \quad \mathcal{M}, s' \models \varphi \text{ for all } s'. \tag{2}$$

3.2. *Satisfiability and model checking*

If we aim to formally reason about a particular system X (where X is, for example, a computer program), using a logic L, then there are, broadly speaking, two possible approaches we can adopt. With the *theorem proving* approach, we derive a theory $Th(X)$ using the logic L, where $Th(X)$ encodes properties of X using the language L. Checking whether X has some property φ then reduces to a proof problem in L: we simply check whether $Th(X) \vdash_L \varphi$, *i.e.*, whether φ is a theorem of $Th(X)$. In contrast, the *model checking* approach depends upon interpreting a system X as a model M_X for a logic L: checking that X has property φ (where φ is again expressed as a formula of L) then reduces to the problem of checking that M_X satisfies φ under the semantics of L, *i.e.*, that $M_X \models_L \varphi$ (Clarke, Grumberg et Peled 2000). There is an ongoing debate with respect to the relative merits of theorem proving versus model checking as an approach to the automatic verification of system properties, and it is not our intent to add to this debate here. We choose to adopt a model checking approach for two reasons. First, this approach has previously proved useful for verifying cooperation properties of multi-agent systems (Hoek et Wooldridge 2002, Hoek et Wooldridge 2003b, Hoek et Wooldridge 2003a); and second, because reliable and well-documented model checking tools exist for cooperation logics (Alur *et al.* 1998).

Given a model $\mathcal{M} = \langle \mathcal{A}, \mathcal{S}, \mathcal{E}, \Phi_0, v \rangle$, we denote the *size* of $\mathcal{M}$ by $|\mathcal{M}|$, and define this as follows (Pauly 2001, pp. 50–51):

$$|\mathcal{M}| = |\mathcal{S}| + \sum_{\{s|s\in\mathcal{S}\}} \sum_{\{C|C\subseteq\mathcal{A}\}} \sum_{\{S|S\in\mathcal{E}(C,s)\}} |S|$$

The size of a formula φ, as usual, is denoted by $|\varphi|$ and is defined as the number of sub-formulae that φ contains.

Now, the model checking problem for CL is as follows.

> MODEL CHECKING:
> *Given*: Formula φ, model $\mathcal{M}$, and state s from $\mathcal{M}$.
> *Answer*: “Yes” if $\mathcal{M}, s \models \varphi$, “no” otherwise.

PROPOSITION 1 (COMPLEXITY OF CL MODEL CHECKING (PAULY 2001, PAULY 2002B)). — *The* MODEL CHECKING *problem for Coalition Logic can be solved in time $O(|\mathcal{M}| \times |\varphi|)$, where $|\mathcal{M}|$ is the size of the model to be checked, and $|\varphi|$ is the size of formula to be checked.*

The satisfiability decision problem for CL is as follows.

> SATISFIABILITY:
> *Given*: Formula φ.
> *Answer*: “Yes” if for some $\mathcal{M}, s$ we have $\mathcal{M}, s \models \varphi$, “no” otherwise.

PROPOSITION 2 (COMPLEXITY OF CL SATISFIABILITY (PAULY 2001, PAULY 2002B)). — *In the general case, the* SATISFIABILITY *problem for Coalition Logic is* PSPACE*-complete.*

Notice that while the model checking problem for CL is no easier than the model checking problem for its more expressive counterpart ATL (Alur *et al.* 2002), there does appear to be a difference in complexity with respect to satisfiability. The ATL satisfiability problem is complete for EXPTIME, and is hence provably intractable (Drimmelen 2003). In contrast, as noted above, the satisfiability problem for CL is “only” PSPACE-complete in general (reducing to NP-complete in certain special cases).

4. Basic correspondence definitions

As was noted in (Wooldridge *et al.* 2004, p.71), there is a close relationship between the effectivity functions of Coalition Models and QCGS. In this section, we make this relationship precise. We define a *correspondence* relation, “$\simeq$”, between QCGs and interpretations. The idea is that, for a QCG Γ and an interpretation $\mathcal{M}, s$, if $\Gamma \simeq \mathcal{M}, s$, then the QCG Γ and the interpretation $\mathcal{M}, s$ are "equivalent" with respect to what they say about the way in which coalitions can cooperate. Before we define the correspondence relation, however, we need to be able to say when a QCG and an interpretation $\mathcal{M}, s$ are *comparable*: that is, the circumstances under which it is meaningful to ask whether they correspond.

Now, we say that QCG Γ and model $\mathcal{M}$ are *comparable* iff:

1) The sets of agents in both structures are the same.

2) There is a propositional variable in the model $\mathcal{M}$ for every possible goal in Γ, and $\mathcal{M}$ contains no other propositional variables. For convenience, if g is a possible goal in Γ, then we will also write g for the propositional variable in $\mathcal{M}$ corresponding to this goal.

Hence, if a model $\mathcal{M} = \langle \mathcal{A}, \mathcal{S}, \mathcal{E}, \Phi_0, \upsilon \rangle$ and a game $\Gamma = \langle \mathcal{A}', \mathcal{G}, \mathcal{G}_1, \ldots \mathcal{G}_n, \mathcal{V} \rangle$ are comparable, then $\mathcal{A} = \mathcal{A}'$ and $\Phi_0 = \mathcal{G}$. As the reader may now be able to guess, the truth of a propositional variable g in a state s will be intended to mean that the corresponding goal g is achieved in state s.

Before formally defining the correspondence relation, let us fix some notation. In what follows, G is assumed to be a set of possible goals, *i.e.*, $G \subseteq \mathcal{G}$.

$$\pi_G^- \;\hat{=}\; \bigwedge_{g \in G} \neg g \qquad \sigma_G^- \;\hat{=}\; \bigvee_{g \in G} \neg g$$

$$\pi_G^+ \;\hat{=}\; \bigwedge_{g \in G} g \qquad \sigma_G^+ \;\hat{=}\; \bigvee_{g \in G} g$$

So, if $\mathcal{M}, s \models \pi_G^-$, then this will mean that *no* goal in G is achieved in state s, whereas if $\mathcal{M}, s \models \pi_G^+$, then *every* goal in G is achieved in state s. In contrast, $\mathcal{M}, s \models \sigma_G^-$ means that *some* member of G is *not* achieved in s, while $\mathcal{M}, s \models \sigma_G^+$ will mean that some member of G *is* achieved in s.

Next, we define a formula that characterises exactly when a given set of goals is achieved in a given state – as above, it is assumed that $G \subseteq \mathcal{G}$.

$$\chi_G \;\hat{=}\; \pi_G^+ \wedge \pi_{\mathcal{G} \setminus G}^-$$

The following property is obvious. Let $\mathcal{M}$ and Γ be a comparable model and game, (so that the propositional variables in $\mathcal{M}$ are exactly the goals in Γ), and let $G \subseteq \mathcal{G}$; then:

$$\mathcal{M}, s \models \chi_G \quad \Leftrightarrow \quad \upsilon(s) = G \tag{3}$$

The following result states some useful properties of χ- and π-formulae, which are straightforwardly established.

PROPOSITION 3. — *Let Γ be a* QCG *and $\mathcal{M}$ be a comparable* CL *model. Then:*

1) If $\exists s$ in $\mathcal{M}$: $\mathcal{M}, s \models (\chi_G \wedge \pi^+_{G'})$ then $G' \subseteq G$.
2) If $G' \subseteq G$ then $\forall s$ in $\mathcal{M}$: $\mathcal{M}, s \models (\chi_G \rightarrow \pi^+_{G'})$.

We can now define the correspondence relation. Let $\Gamma = \langle \mathcal{A}, \mathcal{G}, \mathcal{G}_1, \ldots, \mathcal{G}_n, \mathcal{V} \rangle$ be a QCG game. Rather than relating it to a CL model $\mathcal{M}$, we will compare it with a CL-*interpretation*. Hence, from now on, if we write $(\mathcal{M}, \rho)$, we assume that $\mathcal{M} = \langle \mathcal{A}, \mathcal{S}, \mathcal{E}, \Phi_0, \upsilon \rangle$ with $\rho \in \mathcal{S}$. Then, we write $\Gamma \simeq (\mathcal{M}, \rho)$ iff:

1) $\mathcal{M}$ and Γ are comparable; and
2) For all $C \subseteq \mathcal{A}$ and $G \subseteq \mathcal{G}$, we have:

$$\underbrace{G \in \mathcal{V}(C)}_{\text{QCG}} \quad \Leftrightarrow \quad \underbrace{\exists S \in \mathcal{E}(C, \rho) \text{ s.t. } \forall s \in S : \upsilon(s) = G}_{\text{interpretation}}$$

The first condition essentially says that the game and model contain the same agents and goals, while the second says that a game indicates that it is possible for a coalition to get some outcome iff the interpretation indicates this also.

4.1. *The structure of our correspondence results*

Consider the CL predicate $\mathsf{FEAS}(\cdots)$, defined as follows:

$$\mathsf{FEAS}(G, C) \hat{=} [C]\chi_G$$

The following property follows from (3) and the semantics of the coalition modality "[_]". Let $\mathcal{M}$ and Γ be a comparable model and game; then:

$$(\mathcal{M}, \rho) \models \mathsf{FEAS}(G, C) \quad \Leftrightarrow \quad \exists S \in \mathcal{E}(C, s) \text{ s.t. } \forall s \in S : \upsilon(s) = G \tag{4}$$

Now, it is not hard to see from this that if $\mathcal{M}, s$ is a CL interpretation that corresponds to some QCG Γ, then $\mathcal{M}, s \models \mathsf{FEAS}(G, C)$ iff the goal set G represents a feasible choice for the coalition C in Γ. In this way, the predicate $\mathsf{FEAS}(\cdots)$ can be said to *characterise* feasible choices in corresponding QCGs. The motivation for this terminology should be clear to readers who are familiar with modal logic (Chellas 1980, Blackburn *et al.* 2001). Our correspondence results give us a CL-based syntactic characterisation of QCG properties, in much the same way that the correspondence theory of conventional modal logic provides a syntactic characterisation of first-order relational properties of Kripke structures (Benthem 1984).

In the remainder of this paper, we define a number of similar predicates, and show that the truth of such a predicate in an interpretation corresponding to a particular QCG characterises a certain property of the QCG. In this section, we present some notation that is intended to render these results somewhat more intelligible.

Where $\mathsf{D}(\alpha_1, \ldots, \alpha_k)$ is a CL predicate parameterised by k arguments, $\alpha_1, \ldots, \alpha_k$, and $P(\alpha_1, \ldots, \alpha_k)$ is a property of QCGs parameterised in the same way, then we write

$$\mathsf{D}(\alpha_1, \ldots, \alpha_k) \equiv P(\alpha_1, \ldots, \alpha_k)$$

as a general abbreviation for the following:

> Let Γ be a QCG, and let $(\mathcal{M}, \rho)$ be an interpretation such that $\Gamma \simeq (\mathcal{M}, \rho)$.
> Then $(\mathcal{M}, \rho) \models \mathsf{D}(\alpha_1, \ldots, \alpha_k)$ iff $P(\alpha_1, \ldots, \alpha_k)$ is true of Γ.

In the text, we refer to results of the form $\mathsf{D}(\alpha_1, \ldots, \alpha_k) \equiv P(\alpha_1, \ldots, \alpha_k)$ as *correspondence results*, and say that $\mathsf{D}(\alpha_1, \ldots, \alpha_k)$ *characterises* $P(\alpha_1, \ldots, \alpha_k)$.

Expressed using this terminology, the key property of the $\mathsf{FEAS}(\cdots)$ predicate, which we defined above, is as follows.

PROPOSITION 4. — $\mathsf{FEAS}(G, C) \equiv$ *goal set G is feasible for C*, i.e., $G \in \mathcal{V}(C)$.

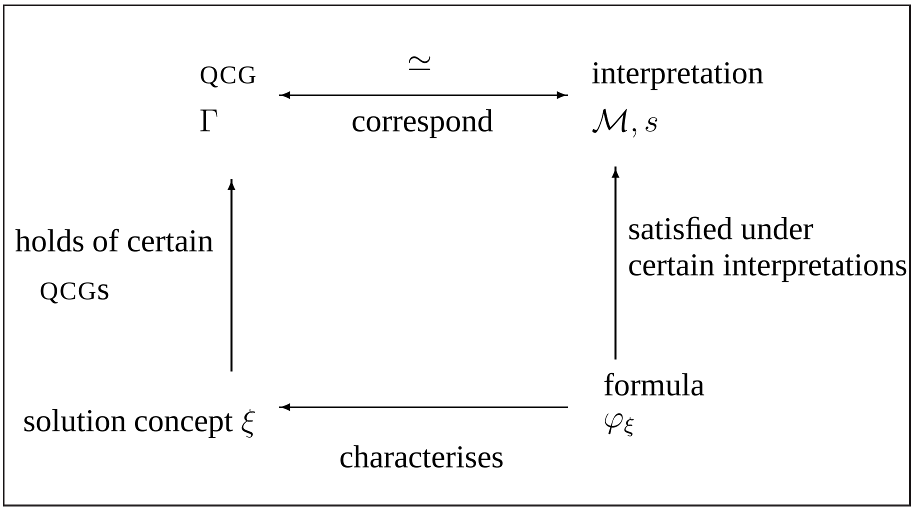

Figure 1. *The overall structure of our correspondence results*

The overall structure of our correspondence results is illustrated in Figure 1.

4.2. *The complexity of deciding correspondence*

There is an obvious decision problem associated with the correspondence relation "$\simeq$".

> CORRESPONDENCE:
> *Given*: QCG Γ and comparable interpretation $(\mathcal{M}, \rho)$.
> *Answer*: "Yes" if $\Gamma \simeq (\mathcal{M}, \rho)$, "no" otherwise.

The complexity of this problem depends on the *representation* that is assumed for the QCG and interpretation, and in particular, the representation that is chosen for the characteristic function $\mathcal{V}$ in the QCG, and the effectivity function $\mathcal{E}$ in the model.

Perhaps the simplest representation for any given function f is the obvious set-theoretic one, in which we explicitly enumerate it as a set of ordered pairs $\{(x, y) : y = f(x)\}$. We refer to such a representation as *extensive*. There is an obvious problem with extensive representations. Consider the characteristic function $\mathcal{V}$: the size of an extensive representation of $\mathcal{V}$ will clearly be $O(2^{|\mathcal{A} \cup \mathcal{G}|})$. It was argued in (Wooldridge *et al.* 2004, p.34) that such a representation is (*i*) *utterly* infeasible in practice; and (*ii*) so large that it renders comparisons to this input size meaningless, since stating that we have an algorithm that runs in (say) time linear in the size of such a representation only actually means that it runs in time exponential in the size of $\mathcal{A} \cup \mathcal{G}$.

In (Wooldridge *et al.* 2004, p.35), an alternative representation was suggested, whereby the function is characterised as a formula of propositional logic. The idea is that any finite function $f : X \to Y$ can be represented as a formula Ψ_f of propositional logic, over propositional variables corresponding to the input and output sets X and Y: for Ψ_f to correctly capture f, we simply require that $\Psi_f[x, y] = \top$ iff $y = f(x)$.

So, given a characteristic function $\mathcal{V}$, we represent it as a formula $\Psi_\mathcal{V}$ whose vocabulary of propositional variables is $\mathcal{A} \cup \mathcal{G}$ (*i.e.*, there is a propositional variable for each agent and each possible goal). For a formula $\Psi_\mathcal{V}$ (over variables $\mathcal{A}$, $\mathcal{G}$) to represent a characteristic function $\mathcal{V}$, we require that for all $C \subseteq \mathcal{A}$ and $G \subseteq \mathcal{G}$ we have:

$$\Psi_\mathcal{V}[C, G] = \top \quad \Leftrightarrow \quad G \in \mathcal{V}(C).$$

There are two observations to make about this representation. The first is that given any $\Psi_\mathcal{V}$ (representing a characteristic function $\mathcal{V}$), $C \subseteq \mathcal{A}$, and $G \subseteq \mathcal{G}$, determining whether $\Psi_\mathcal{V}[C, G] = \top$ (and hence whether $G \in \mathcal{V}(C)$) can be done in deterministic polynomial time (it just requires evaluating the truth of a propositional logic formula under a given valuation). The second observation is that, while it is possible to prove that there exist "pathological" characteristic functions $\mathcal{V}$, which require exponential length formulas $\Psi_\mathcal{V}$ to characterise them (Wooldridge *et al.* 2004, p.37), the representation is nevertheless extremely succinct in many (arguably, most) naturally arising cases. And this is, of course, exactly why propositional logic is such a widely used representational formalism in artificial intelligence and computer science generally. We refer to the propositional logic representation of functions as *symbolic*, following the usage of this term in the model checking literature (Clarke *et al.* 2000, pp. 61–95).

For a symbolic representation of effectivity functions, we can use the same idea, although we need to work a little harder. We use a formula of propositional logic, in much the same way, but since we have states appearing as both inputs *and* outputs to the function, we need extra notation. If $\mathcal{S} = \{s_1, \ldots, s_n\}$ is the set of states in a model $\mathcal{M}$, then we will denote by $\hat{\mathcal{S}} = \{\hat{s_1}, \ldots, \hat{s_n}\}$ a new set containing a member

$\hat{s}$ for every member s of $\mathcal{S}$ (note that we intend s and $\hat{s}$ to correspond directly to one-another). A formula $\Psi_{\mathcal{E}}$ capturing an effectivity function $\mathcal{E}$ then takes as its vocabulary of propositional variables the sets $\mathcal{A}$, $\mathcal{S}$, and $\hat{\mathcal{S}}$. For a formula $\Psi_{\mathcal{E}}$ (over variables $\mathcal{A}$, $\mathcal{S}$, $\hat{\mathcal{S}}$) to represent an effectivity function $\mathcal{E}$, we require that for all $C \subseteq \mathcal{A}$, $s \in \mathcal{S}$, and $S \subseteq \mathcal{S}$, we have:

$$\Psi_{\mathcal{E}}[C, \hat{s}, S] = \top \quad \Leftrightarrow \quad S \in \mathcal{E}(C, s).$$

As above, the key point with this representation is that determining whether $S \in \mathcal{E}(C, s)$ for any given $S \subseteq \mathcal{S}$, $C \subseteq \mathcal{A}$, and $s \in \mathcal{S}$ can be done in polynomial time: it requires evaluating whether $\Psi_{\mathcal{E}}[C, \hat{s}, S] = \top$.

In sum, this gives us two alternative representations for the CORRESPONDENCE problem: an extensive one (in which we explicitly enumerate the functional components of the structures), and a symbolic one (in which we represent the characteristic function and effectivity function as propositional logic formulae). Given these two representations, we can now ask how hard the correspondence decision problem is. It is trivial to see that assuming an extensive representation, correspondence can be checked in polynomial time by exhaustive search; but of course this is not terribly useful, since, as we already argued, extensive representations are infeasibly large. It is no surprise that CORRESPONDENCE is harder under the assumption of a symbolic representation for effectivity functions. Before we investigate exactly how much harder it is, consider the (simpler) problem of checking whether a particular goal set is feasible for a given coalition.

FEASIBILITY:
Given: Interpretation $(\mathcal{M}, \rho)$, goal set G, and coalition $C \subseteq \mathcal{A}$.
Answer: "Yes" if $(\mathcal{M}, \rho) \models \mathsf{FEAS}(G, C)$, "no" otherwise.

It turns out that even this problem – which is trivially seen to be tractable for the extensive representation – is hard for symbolic representations.

PROPOSITION 5. — *For the symbolic representation,* FEASIBILITY *is* NP*-complete.*

PROOF 6. — By the semantics of CL and the definition of $\mathsf{FEAS}(\cdots)$, checking that $(\mathcal{M}, \rho) \models \mathsf{FEAS}(G, C)$ amounts to checking whether the following holds:

$$\exists S \subseteq \mathcal{S} : \underbrace{\Psi_{\mathcal{E}}[C, \hat{\rho}, S] = \top \;\&\; \forall s \in S, \upsilon(s) = G}_{(*)}.$$

Guessing a subset S of $\mathcal{S}$ can clearly be done in polynomial time, and the condition $(*)$ can easily by checked in polynomial time.

For NP-hardness, we reduce from SAT (Papadimitriou 1994, p.171). Given an instance $\Phi(x_1, \ldots, x_k)$ of SAT, we must show how to construct – in polynomial time – an interpretation $\mathcal{M}_\Phi, \rho$, goal set $\mathcal{G}$, and coalition $C \subseteq \mathcal{A}$ such that $\mathcal{M}_\Phi, \rho \models \mathsf{FEAS}(G, C)$ iff $\Phi(x_1, \ldots, x_k)$ is satisfiable. To construct the interpretation $\mathcal{M}_\Phi, \rho$, we define a single agent $\mathcal{A}_\Phi = \{a_1\}$, define a state for each propositional variable, *i.e.*, $\mathcal{S}_\Phi = \{x_1, \ldots, x_k\}$, and define the interpretation state ρ to be any member of $\mathcal{S}_\Phi$. We define a single goal, $\mathcal{G}_\Phi = \{g_1\}$, with the formula for the effectivity function be given by the propositional formula

$$\Psi_{\mathcal{E}_\Phi} \;=\; (\neg a_1 \wedge \bigwedge_{i=1}^{k} x_i) \quad \vee \quad (a_1 \wedge (\; \Phi(x_1, \ldots, x_k) \vee \Phi(\neg x_1, \ldots, \neg x_k)\;))$$

The first clause in this definition captures the case where a coalition is empty (the empty coalition cannot choose between any of the states in the system), while the second deals with non-empty coalitions.

We observe that $\Psi_{\mathcal{E}_\Phi}[\emptyset, \hat{\rho}, S] = \top$ if and only if $S = \mathcal{S}$. We then define υ_Φ so that $\upsilon_\Phi(s) = \mathcal{G}_\Phi$ for all $s \in \mathcal{S}_\Phi$. Finally, we must exhibit a coalition and goal set to check against: we define $C = \{a_1\}$ and $G = \mathcal{G}_\Phi$. We now claim that $\mathcal{M}_\Phi, \rho \models \mathsf{FEAS}(G, C)$ iff $\Phi(x_1, \ldots, x_n)$ is satisfiable.

($\Rightarrow$) Assume $\mathcal{M}_\Phi, \rho \models \mathsf{FEAS}(G, C)$. Then $\exists S \subseteq \mathcal{S}$ such that $\Psi_{\mathcal{E}_\Phi}[C, \hat{\rho}, S] = \top$. But since by construction $\Psi_{\mathcal{E}_\Phi} = \Phi(x_1, \ldots, x_k) \vee \Phi(\neg x_1, \ldots, \neg x_k)$, then $\Phi(x_1, \ldots, x_k)$ is satisfiable.
($\Leftarrow$) Assume $\Phi(x_1, \ldots, x_k)$ is satisfiable. Since by construction $\Psi_{\mathcal{E}_\Phi} = \Phi(x_1, \ldots, x_k) \vee \Phi(\neg x_1, \ldots, \neg x_k)$, then this implies $\exists S \subseteq \mathcal{S}$ such that $S \neq \emptyset$ and $\Psi_{\mathcal{E}_\Phi}[C, \hat{\rho}, S] = \top$. Moreover, since by construction $\upsilon_\Phi(s) = \mathcal{G}_\Phi$ for all $s \in \mathcal{S}$, then by definition we have $\mathcal{M}_\Phi, \rho \models \mathsf{FEAS}(G, C)$.

Since the construction can clearly be done in time polynomial in the size of the input instance, it follows that FEASIBILITY for symbolically represented games and interpretations is NP-hard. ■

This result suggests that checking correspondence under the assumption of a symbolic representation is going to be rather hard, and this is indeed the case[3].

PROPOSITION 7. — *For the symbolic representation,* CORRESPONDENCE *is* Π_2^p-*complete.*

PROOF 8. — For membership of Π_2^p, first note that checking whether $\Gamma \simeq (\mathcal{M}, \rho)$ amounts to checking the following.

$$\forall G \subseteq \mathcal{G}, \forall C \subseteq \mathcal{A}, \underbrace{(\Psi_{\mathcal{V}}[G, C] = \top \Leftrightarrow (\mathcal{M}, \rho) \models \mathsf{FEAS}(G, C))}_{(**)}.$$

We can determine the truth of this expression by universally selecting each $G \subseteq \mathcal{G}$ and $C \subseteq \mathcal{A}$, and then checking that condition $(**)$ holds. (We make use of an NP-oracle when checking whether $(\mathcal{M}, \rho) \models \mathsf{FEAS}(G, C)$ – see Proposition 5.) Hence the problem is in Π_2^p.

3. Recall that Π_2^p is the class of languages/problems that are in co-NP assuming the availability of an oracle for languages/problems in NP (Papadimitriou 1994, p.426).

To show that the problem is Π_2^p hard, we reduce QSAT$_{2,\forall}$, the quintessential Π_2^p complete problem (Johnson 1990, p.96). An instance of QSAT$_{2,\forall}$ is given by a quantified Boolean formula ξ with the following structure:

$$\xi = \forall \bar{x}\ \exists \bar{y}\ \varphi(\bar{x}, \bar{y})$$

where $\bar{x} = x_1, \ldots, x_r$ and $\bar{y} = y_1, \ldots, y_s$ are sets of propositional variables. The formula ξ is true iff for all assignments that we can give to Boolean variables $\bar{x}$, there is some assignment we can give to Boolean variables $\bar{y}$ such that $\varphi(\bar{x}, \bar{y})$ is true. Here is a concrete example of such a formula.

$$\forall x_1 \exists x_2 [(x_1 \vee x_2) \wedge (x_1 \vee \neg x_2)] \tag{5}$$

Formula (5) in fact evaluates to false. (If x_1 is false, there is no value we can give to x_2 that will make the body of the formula true.)

To reduce an instance ξ of QSAT$_{2,\forall}$ to CORRESPONDENCE, we must exhibit a QCG Γ_ξ and an interpretation $\mathcal{M}_\xi, \rho_\xi$ such that $\Gamma_\xi \simeq \mathcal{M}_\xi, \rho_\xi$ iff ξ is true. Consider first the QCG Γ_ξ. We define an agent for each propositional variable in $\bar{x}$ and one "dummy" agent x_{r+1}, so $\mathcal{A}_\xi = \{x_1, \ldots, x_r, x_{r+1}\}$. We define a single goal $\mathcal{G}_\xi = \{g\}$. For each agent $1 \leq i \leq r+1$, we define $\mathcal{G}_{\xi_i} = \mathcal{G}_\xi$ (although in fact, the $\mathcal{G}_{\xi_i}$ components play no subsequent part in the proof, and could be any subset of $\mathcal{G}_\xi$). Finally, we define $\Psi_\xi = \top$. With respect to the interpretation $\mathcal{M}_\xi, s$, the agents and propositional variables are the same. We define a state for each propositional variable in $\bar{y}$, so $\mathcal{S} = \{y_1, \ldots, y_s\}$, and define $\upsilon_\xi(s) = \mathcal{G}_\xi$ for all $s \in \mathcal{S}$. The formula $\Psi_{\mathcal{E}_\xi}$ for the effectivity function is given by

$$\Psi_{\mathcal{E}_\xi} \;=\; (x_{r+1} \wedge \varphi(\bar{x}, \bar{y}))\ \vee\ (\neg x_{r+1} \wedge \bigwedge_{i=1}^{s} y_i)$$

As before we observe that $\Psi_{\mathcal{E}_\xi}[\emptyset, \hat{s}, S] = \top$ if and only if $S = \mathcal{S}$. Finally, the interpretation state ρ_ξ is defined to be any member of $\mathcal{S}$. Now, we claim that $\Gamma_\xi \simeq \mathcal{M}_\xi, \rho_\xi$ iff ξ is true. To see this, observe that $\Gamma_\xi \simeq \mathcal{M}_\xi, \rho_\xi$ is equivalent to the following:

$$\forall G \subseteq \mathcal{G}_\xi, \forall C \subseteq \mathcal{A}_\xi :\ (\Psi_\xi[C, G] = \top \Leftrightarrow \exists S \subseteq \mathcal{S}_\xi :\ \Psi_{\mathcal{E}_\xi}[C, \hat{\rho}, S] = \top).$$

Since by construction $\Psi_\xi = \top$, this simplifies to the following.

$$\forall G \subseteq \mathcal{G}_\xi, \forall C \subseteq \mathcal{A}_\xi, \exists S \subseteq \mathcal{S}_\xi :\ (\Psi_{\mathcal{E}_\xi}[C, \hat{\rho}, S] = \top).$$

Moreover, since no member of $\mathcal{G}_\xi$ appears in $\Psi_{\mathcal{E}_\xi}$, this further simplifies to the following.

$$\forall C \subseteq \mathcal{A}_\xi, \exists S \subseteq \mathcal{S}_\xi :\ \Psi_{\mathcal{E}_\xi}[C, S] = \top.$$

Noting that this is satisfied by any $C \subseteq \mathcal{A}_\xi$ for which $x_{r+1} \notin C$, without loss of generality, we need consider only those C with $x_{r+1} \in C$. Now, since $\mathcal{A}_\xi \setminus x_{r+1} = \bar{x}$, $S_\xi = \bar{y}$, from the definition of $\Psi_{\mathcal{E}_\xi}$, this further reduces to the following.

$$\forall X \subseteq \bar{x}, \exists Y \subseteq \bar{y} : \ \Phi(\bar{x}, \bar{y})[X, Y] = \top.$$

And this is exactly the condition for the truth of the $\text{QSAT}_{2,\forall}$ formula ξ.

Since the construction can clearly be done in time polynomial in the size of the input instance, it follows that CORRESPONDENCE for symbolically represented games and interpretations is Π_2^p-hard. ■

It is worth remarking that implemented software tools for reasoning about coalitional games (such as the MOCHA model checking system for ATL (Alur *et al.* 2000, Alur *et al.* 1998)) do not use an extensive representation (and indeed, no practical tool could use such a representation). Instead, they use a concise representation for games/models, that is much more akin to our symbolic representation. In the MOCHA example, for instance, a structured, compositional language called REACTIVE MODULES is used to specify games/models, which makes for extremely concise representations of large state spaces (Alur et Henzinger 1999). This suggests (to us at least) that the Π_2^p-completeness result of Proposition 7 is a more “realistic” measure of the cost of checking correspondence.

5. Characterising qualitative coalitional games

In this section, we present our correspondence results. First, we define a formula γ_C^+ (where $C \subseteq \mathcal{A}$) such that γ_C^+ will be satisfied in a state s if *every* agent is satisfied in that state, *i.e.*, if every agent has at least one of its goals satisfied in s. Similarly, γ_C^- will mean that *no* member of C is satisfied.

$$\gamma_C^+ \ \hat{=} \ \bigwedge_{i \in C} \sigma_{\mathcal{G}_i}^+ \qquad \gamma_C^- \ \hat{=} \ \bigwedge_{i \in C} \pi_{\mathcal{G}_i}^-$$

The following properties of γ-formulae are useful subsequently. (They are straightforward to establish.)

PROPOSITION 9. — *Let Γ be a* QCG *and $\mathcal{M}$ be a comparable* CL *model. Then:*

1) If $G \subseteq \mathcal{G}$ satisfies $C \subseteq \mathcal{A}$ then $\forall s$ in $\mathcal{M}$: $\mathcal{M}, s \models (\chi_G \to \gamma_C^+)$.
2) If $\exists s$ in $\mathcal{M}$: $\mathcal{M}, s \models (\chi_G \wedge \gamma_C^+)$ then $G \subseteq \mathcal{G}$ satisfies $C \subseteq \mathcal{A}$.

5.1. *Empty sets of goals*

Let us first look at some extreme cases regarding the goals sets in QGL, *i.e.*, those in which a set of goals can be empty. We have the following possibilities:

(i) $\mathcal{V}(C) = \emptyset$

(ii) $\emptyset \in \mathcal{V}(C)$

(iii) $\mathcal{V}(C) = \{\emptyset\}$

In the first case, coalition C has no choice, in the second, C can make a choice such that no goal (of anybody, whatsoever) is achieved, and, finally, in case (iii), the only available choice of C leads to "nothing": no goal is satisfied.

How do these cases find their corresponding counterparts in CL? We provide them in Table 1, together with their characterising CL-properties. In other words, for every row in Table 1, the entry in the second column corresponds to the entry in the fourth column.

Table 1. *Empty goals: extreme cases*

	Property of QCG structure	Property of CL interpretation	Characteristic CL-formula
(i)	$\mathcal{V}(C) = \emptyset$	$\mathcal{E}(C, s) = \emptyset$	$\neg[C]\top$
(ii)	$\emptyset \in \mathcal{V}(C)$	$\exists S \in \mathcal{E}(C, s) s.t. \forall s \in S : \upsilon(s) = \emptyset$	$[C] \bigwedge_{g \in G} \neg g$
(iii)	$\mathcal{V}(C) = \{\emptyset\}$	$\forall S \in \mathcal{E}(C, s) \forall s \in S : \upsilon(s) = \emptyset$	$[C]\top \wedge \neg \bigvee_{g \in G}[C]g$

5.2. *Successful coalitions*

In many ways, the idea of a successful coalition incorporates the most basic question that is of interest with respect to any given QCG (Wooldridge *et al.* 2004, p.47). A coalition is *successful* if that coalition has a feasible choice satisfying all members of the coalition. Formally, given a QCG $\Gamma = \langle \mathcal{A}, \mathcal{G}, \mathcal{G}_1, \ldots, \mathcal{G}_n, \mathcal{V} \rangle$ and a coalition $C \subseteq \mathcal{A}$, we say that C is successful iff:

$$\exists G \in \mathcal{V}(C) \text{ s.t. } \forall i \in C, \text{ we have } G \cap \mathcal{G}_i \neq \emptyset.$$

Given that a particular coalition is successful in this sense, we cannot be certain that this coalition *will* form; but we *can* be certain that an *unsuccessful* coalition will *not* form – because, by definition, the formation of such a coalition would leave at least one member unsatisfied. In this sense, success is a necessary, but not sufficient condition for coalition formation in QCGs. We can easily characterise successful coalitions, via the defined predicate $\mathsf{SC}(\cdots)$.

$$\mathsf{SC}(C) \hat{=} \bigvee_{G \subseteq \mathcal{G}} [C](\chi_G \wedge \gamma_C^+)$$

PROPOSITION 10. — $\mathsf{SC}(C) \equiv$ *coalition C is successful.*

PROOF 11. —

($\Rightarrow$) Assume $\mathcal{M}, s \models \mathsf{SC}(C)$: we need to show that this implies there is a goal set that is both feasible for and satisfies C. Since $\mathcal{M}, s \models \mathsf{SC}(C)$, then by definition, $\exists G \subseteq \mathcal{G}$ such that $\mathcal{M}, s \models [C](\chi_G \wedge \gamma_C^+)$. Since $\mathcal{M}, s \models [C]\chi_G$, then by the definition of $\simeq$, G is feasible for C, *i.e.*, $G \in \mathcal{V}(C)$. Moreover, since $\mathcal{M}, s \models [C](\chi_G \wedge \gamma_C^+)$, then $\exists S \in \mathcal{E}(C, s)$ such that $\forall s' \in S$, $\mathcal{M}, s' \models (\chi_G \wedge \gamma_C^+)$. Then by Proposition 9(2), G satisfies C.
($\Leftarrow$) Assume that C is successful. Then $\exists G \subseteq \mathcal{G}$ such that (*i*) G is feasible for C, and (*ii*) G satisfies every member of C. From (*i*), we have $\mathcal{M}, s \models [C]\chi_G$ and from (*ii*) by Proposition 9(1), we have $\mathcal{M}, s' \models \chi_G \rightarrow \gamma_C^+$ for all s'. Hence $\exists G \subseteq \mathcal{G}$ such that $\mathcal{M}, s \models [C](\chi_G \wedge \gamma_C^+)$, and so by definition $\mathcal{M}, s \models \mathsf{SC}(C)$. ■

At first sight, the reader may suspect that the definition of $\mathsf{SC}(\cdots)$ is over engineered: would the following, simpler definition not suffice to characterise successful coalitions?

$$\mathsf{SC?}(C) \hat{=} [C]\gamma_C^+$$

The answer is no. To see why, consider the fragment of model $\mathcal{M}$ illustrated in Figure 2, where $\Gamma \simeq \mathcal{M}, s_1$ for some Γ. In this model fragment, states in the model are drawn as circles, and the arrows labelled with i indicate the choices available to agent i according to the effectivity function $\mathcal{E}$; hence one choice for i is $\{s_2, s_3, s_4\}$. Inside each state, we write the propositions that are satisfied in this state. Now, suppose that agent i's goal set is $\mathcal{G}_i = \{g_5, g_9, g_{10}\}$. Then clearly, according to the definition of $\mathsf{SC?}(\cdots)$, we would have that i is successful, since $\mathcal{M}, s_1 \models [i]\gamma_i^+$. (To see this, simply note that $\{s_2, s_3, s_4\} \in \mathcal{E}(\{i\}, s_1)$ and $\forall s' \in \{s_2, s_3, s_4\}$, we have $\mathcal{M}, s' \models \gamma_i^+$.) But this does not imply that any non-empty subset of $\{g_5, g_9, g_{10}\}$ represents a feasible choice for i in Γ. This is because there is no non-empty goal set G such that $\mathcal{M}, s' \models \chi_G$ for all $s' \in \{s_2, s_3, s_4\}$, and hence this set of states does not characterise a set of goals that is feasible for and satisfies agent i.

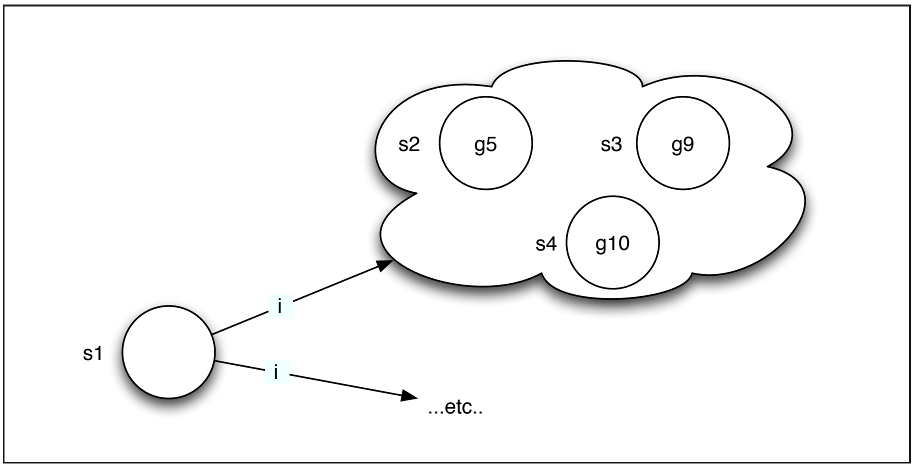

Figure 2. *Why the simpler definition of* $\mathsf{SC}(\cdots)$ *does not work*

A few words about this counterexample are in place here. Whether such an example tells us something interesting about what coalitions can achieve, depends on the object language under consideration. If our main emphasis is on $[C]\chi_G$ formulas, describing what a coalition can *exactly* bring about, the example of Figure 2 is in some sense harmless: one can show that the balloon consisting of states s_2, s_3 and s_4 can be removed without affecting the truth of any $[C]\chi_G$-like property (in s_1). However, on a finer level of granularity, the situation in the counterexample demonstrates an interesting difference between QCG-games and CL-interpretations, *i.e.*, that in the latter, is it possible to express that a coalition can achieve a goal, *without* having to specify which set of goals it *exactly* can bring about. In this paper, we choose to not analyse the subtleties that different object languages can express, but rather to directly relate the notion of a QCG-game Γ with that of a CL-interpretation $(\mathcal{M}, \rho)$, thereby respecting the reading of the characteristic function $\mathcal{V}$ as given in Section 2 ("C can bring about *exactly* ...").

5.3. *Selfish successful coalitions*

The fact that C are successful does not preclude agents outside C having goals satisfied by a subset G attesting to the success of C. This suggests the notion of a successful *selfish* coalition, as a coalition C for which there is some $G \in \mathcal{V}(C)$ that satisfies *only* the members of C (Wooldridge *et al.* 2004, pp. 48–49). Formally, coalition C is a selfish successful coalition if $\exists G \subseteq \mathcal{G}$ s.t. $G \in \mathcal{V}(C)$ and for which $\forall i \in \mathcal{A}$, $\mathcal{G}_i \cap G \neq \emptyset$ if and only if $i \in C$. Of course, assuming that an agent's (principal) aim is to enlist in a coalition with whose support it can realise a goal it wishes to be satisfied, it may not necessarily be concerned with the status of agents outside the coalition, and in particular, whether such might be satisfied with a particular feasible goal set. In many scenarios, an agent will be indifferent to the level of satisfaction achieved by non-members: our contention, however, is that such scenarios do not encompass all settings that might usefully be modelled within a QCG environment – see (Wooldridge *et al.* 2004, pp. 48–49) for further discussion. (Note that we do not mean to say that a selfishly successful coalition will choose to be selfish in practice: it simply means that the possibility is there.)

We characterise selfish successful coalitions via the predicate $\mathsf{SSC}(\cdots)$, as follows.

$$\mathsf{SSC}(C) \hat{=} \bigvee_{G \subseteq \mathcal{G}} [C](\chi_G \wedge \gamma^+_C \wedge \gamma^-_{\mathcal{A}\setminus C})$$

PROPOSITION 12. — $\mathsf{SSC}(C) \equiv$ *coalition C is selfishly successful.*

PROOF 13. — Similar to that of Proposition 10: the point to note is that if $(\mathcal{M}, \rho) \models \chi_G \wedge \gamma^+_C \wedge \gamma^-_{\mathcal{A}\setminus C}$, then G must satisfy an agent i iff $i \in C$. ∎

5.4. *Goal realisability*

We say a set of goals G is realisable if there is *any* coalition for which G is both feasible and satisfies every member (Wooldridge *et al.* 2004, p.50). Thus, the fact that a set of goals is realisable implies that there is at least some chance of this goal set being achieved, as it would satisfy at least one coalition. Of course, it does not imply that this goal set will be the *actual* choice of any coalition. Thus realisability is a necessary condition for the achievement of any set of goals – although it is of course not sufficient.

We characterise realisability via the predicate $\mathsf{GR}(\cdots)$.

$$\mathsf{GR}(G) \hat{=} \bigvee_{C \subseteq \mathcal{A}} \bigvee_{G' \subseteq \mathcal{G}} [C](\chi_{G'} \wedge \pi_G^+ \wedge \gamma_C^+)$$

We have:

PROPOSITION 14. — $\mathsf{GR}(G) \equiv$ *goal set G is realisable.*

PROOF 15. —

$(\Rightarrow)$ $(\mathcal{M}, \rho) \models \mathsf{GR}(G)$. Then $\exists C \subseteq \mathcal{A}$, $\exists G' \subseteq \mathcal{G}$ such that $(\mathcal{M}, \rho) \models [C](\chi_{G'} \wedge \pi_G^+ \wedge \gamma_C^+)$. Since $(\mathcal{M}, \rho) \models [C]\chi_{G'}$, then by the definition of $\simeq$, G' is feasible for C, *i.e.*, $G' \in \mathcal{V}(C)$. From $(\mathcal{M}, \rho) \models [C]\chi_{G'} \wedge \gamma_C^+$, and Proposition 9, we know that G' satisfies C. From Proposition 3(1), and the fact that $(\mathcal{M}, \rho) \models [C](\chi_{G'} \wedge \pi_G^+)$, we know that $G \subseteq G'$. Hence G is both feasible for and satisfies C, and so G is realisable.
$(\Leftarrow)$ Assume G is realisable. Then $\exists C \subseteq \mathcal{A}$ and $G' \subseteq \mathcal{G}$ such that G is both feasible for and satisfies C, and $G \subseteq G'$. From Proposition 10 and the fact that G' is both feasible for and satisfies C, we have $(\mathcal{M}, \rho) \models [C](\chi_{G'} \wedge \gamma_C^+)$, and from this and the fact that $G \subseteq G'$ together with Proposition 3(2), we then have $(\mathcal{M}, \rho) \models [C](\chi_{G'} \wedge \gamma_C^+ \wedge \pi_G^+)$. Hence $(\mathcal{M}, \rho) \models \mathsf{GR}(G)$ by definition. ■

5.5. *Minimal coalitions*

We say a coalition is *minimal* if no strict subset of this coalition is successful. The notion of minimality is important because it implies a kind of *internal stability* for a coalition (*cf.* (Osborne *et al.* 1994, p.281)). That is, in a minimal coalition, there is no incentive for subsets of the coalition to defect away from the coalition, as, by definition, such sub-coalitions cannot be successful. Formally, a coalition C is minimal iff $\forall C' \subset C$, $\forall G \subseteq \mathcal{G}$, if $\forall i \in C'$, $G \cap \mathcal{G}_i \neq \emptyset$, then $G \notin \mathcal{V}(C')$. Minimality is easily captured in the predicate $\mathsf{MC}(\cdots)$.

$$\mathsf{MC}(C) \hat{=} \bigwedge_{C' \subset C} \neg \mathsf{SC}(C')$$

The correspondence result is obvious, and we therefore omit the proof.

PROPOSITION 16. — $\mathsf{MC}(C) \equiv$ *coalition C is minimal.*

5.6. ***Core membership and core non-emptiness***

Perhaps the most widely studied issue in cooperative game theory is that of coalitional stability, and the tool used most widely to analyse this issue is the core (Osborne *et al.* 1994, pp. 257–274). Intuitively, the core of a coalition is the set of feasible choices for that coalition from which the members of that coalition have no incentive to deviate. In the QCG setting, a parallel notion was introduced in (Wooldridge *et al.* 2004, p.54). Formally, we say a set of goals G is in the core of a coalition C iff: (*i*) C is minimal; (*ii*) G is feasible for C; and in addition (*iii*) G satisfies every member of C. Formally, G is in the core if (*i*) $G \in \mathcal{V}(C)$; (*ii*) $\forall C' \subset C$, $\forall G' \subseteq \mathcal{G}$ if $\forall i \in C'$, $\mathcal{G}_i \cap G' \neq \emptyset$ then $G' \notin \mathcal{V}(C)$; and (*iii*) $\forall i \in C\ \mathcal{G}_i \cap G \neq \emptyset$. We define the predicate $\mathsf{CNE}(\cdots)$ to capture core membership.

$$\mathsf{CM}(G, C) \hat{=} \mathsf{MC}(C) \wedge [C](\chi_G \wedge \gamma_C^+)$$

The correspondence result is now obvious.

PROPOSITION 17. — $\mathsf{CM}(G, C) \equiv$ *goal set G is in the core of C.*

The core of a coalition will thus be non-empty if that coalition is both minimal and successful, which easily leads to the following predicate definition.

$$\mathsf{CNE}(C) \hat{=} \mathsf{MC}(C) \wedge \mathsf{SC}(C)$$

PROPOSITION 18. — $\mathsf{CNE}(C) \equiv$ *the core of C is non-empty.*

Observe that, by this definition, $\mathsf{CNE}(C) \leftrightarrow \bigvee_{G \subseteq \mathcal{G}} \mathsf{CM}(G, C)$.

Notice that this definition of the core, when applied to the grand coalition, implies that the grand coalition is the *uniquely* successful coalition, and thus is the only coalition that a rational agent would choose to join.

5.7. ***Veto players***

The notion of a veto player is generally defined in cooperative game theory with respect to *simple* coalitional games: those where every coalition simply either wins or loses. A veto player is said to be one that is a member of every winning coalition. Veto players are important because their cooperation is essential for every coalition that aspires to win: by definition, without their support, a coalition cannot win. In our framework, we use a related notion of veto power, in which we talk of an agent as being a veto player *for some particular state of affairs*. We say i is a veto player for φ (where φ is a formula which characterises some state of affairs) if i is a member of every coalition that can choose φ.

$$\mathsf{VETO}(i, \varphi) \hat{=} \bigwedge_{C \subseteq \mathcal{A}} (([C]\varphi) \rightarrow \neg[C \setminus \{i\}]\varphi)$$

Note that i being a veto player for φ does *not* imply that i can bring about φ, and thus $\mathsf{VETO}(i, \varphi) \rightarrow [i]\varphi$ is not a valid formula scheme.

Let us now return to QCGs. In (Wooldridge *et al.* 2004, p56–57), a notion of veto player was defined that generalised that of conventional coalitional games (Osborne *et al.* 1994, p.261). This definition related to the circumstances under which one agent is a veto player for another agent: that is, whether one agent i is a member of every coalition that is capable of satisfying j.

Formally, i is a veto player for j iff for all $C \subseteq \mathcal{A}$ and $G \in \mathcal{V}(C)$, if $G \cap \mathcal{G}_j \neq \emptyset$ then $i \in C$. It should be noted that j need not be a member of C.

$$\mathsf{VP}(i, j) \hat{=} \bigwedge_{C \subseteq \mathcal{A}} \bigwedge_{G \subseteq \mathcal{G}} (([C](\chi_G \wedge \gamma_j^+)) \rightarrow \neg[C \setminus \{i\}]\chi_G)$$

PROPOSITION 19. — $\mathsf{VP}(i, j) \equiv$ *agent i is a veto player for agent j.*

PROOF 20. —

($\Rightarrow$) Suppose for purposes of contradiction that $(\mathcal{M}, \rho) \models \mathsf{VP}(i, j)$ but that i is not a veto player for j. Then $\exists C \subseteq \mathcal{A}$ such that $i \notin C$ and for some $G \subseteq \mathcal{G}$: $(\mathcal{M}, \rho) \models [C](\chi_G \wedge \gamma_j^+)$. Since $i \notin C$, then $C \setminus \{i\} = C$, hence $(\mathcal{M}, \rho) \models [C \setminus \{i\}](\chi_G \wedge \gamma_j^+)$. But by the definition of $\mathsf{VP}(\cdots)$, we know that $\forall C \subseteq \mathcal{A}$, $\forall G \subseteq \mathcal{G}$, if $(\mathcal{M}, \rho) \models [C](\chi_G \wedge \gamma_j^+)$ then $(\mathcal{M}, \rho) \not\models [C \setminus \{i\}]\chi_G$, and in particular, $(\mathcal{M}, \rho) \not\models [C \setminus \{i\}](\chi_G \wedge \gamma_j^+)$. Contradiction.

($\Leftarrow$) Suppose i is a veto player for j. Then $\forall C \subseteq \mathcal{A}$, $\forall G \subseteq \mathcal{G}$, if $G \in \mathcal{V}(C)$ and $G \cap \mathcal{G}_j \neq \emptyset$ then $i \in C$. Hence $\forall C \subseteq \mathcal{A}$, $\forall G \subseteq \mathcal{G}$, if $(\mathcal{M}, \rho) \models [C](\chi_G \wedge \gamma_j^+)$ then $i \in C$. We now claim that this implies $(\mathcal{M}, \rho) \not\models [C \setminus \{i\}]\chi_G$. For suppose that $(\mathcal{M}, \rho) \models [C \setminus \{i\}]\chi_G$: then since i is a veto player for j, we have $i \in (C \setminus \{i\})$, which is a contradiction. So if $(\mathcal{M}, \rho) \models [C](\chi_G \wedge \gamma_j^+)$ then $(\mathcal{M}, \rho) \models \neg[C \setminus \{i\}]\chi_G$, and so by definition $(\mathcal{M}, \rho) \models \mathsf{VP}(i, j)$. ■

5.8. *Necessary goal sets*

The notion of a *necessary goal* was introduced in (Wooldridge *et al.* 2004, p.57) as a natural counterpart to that of veto players. The idea is that a goal set is necessary if this goal set is a "side effect" of any coalition achieving their goals. More formally, G is necessary if for every coalition $C \subseteq \mathcal{A}$ and goal set $G' \subseteq \mathcal{G}$: if $\forall i \in C$, $\mathcal{G}_i \cap G' \neq \emptyset$ and $G' \in \mathcal{V}(C)$ then $G \subseteq G'$. We characterise necessary goal sets via the predicate $\mathsf{NG}(\cdots)$.

$$\mathsf{NG}(G) \hat{=} \bigwedge_{C \subseteq \mathcal{A}} \bigwedge_{G' \subseteq \mathcal{G}} (([C](\chi_{G'} \wedge \gamma_C^+)) \rightarrow [\emptyset](\chi_{G'} \rightarrow \pi_G^+))$$

PROPOSITION 21. — $\mathsf{NG}(G) \equiv$ *goal set G is necessary.*

PROOF 22. —

($\Rightarrow$) Assume $(\mathcal{M}, \rho) \models \mathsf{NG}(G)$ and $(\mathcal{M}, \rho) \models [C](\chi_{G'} \wedge \gamma_C^+)$ for some $C \subseteq \mathcal{A}$ and $G' \subseteq \mathcal{G}$. Thus G' is both feasible for and satisfies C. We need to show that this implies $G \subseteq G'$. By the definition of $\mathsf{NG}(\cdots)$, we have that if $(\mathcal{M}, \rho) \models [C](\chi_{G'} \wedge \gamma_C^+)$ then $(\mathcal{M}, \rho) \models [\emptyset](\chi_{G'} \rightarrow \pi_G^+)$. So $(\mathcal{M}, \rho)' \models (\chi_{G'} \wedge \pi_G^+)$ for some state s' in $\mathcal{M}$. By Proposition 3(1), therefore, $G \subseteq G'$.
($\Leftarrow$) Assume G is a necessary goal set. Then $\forall C \subseteq \mathcal{A}$ and $\forall G' \subseteq \mathcal{G}$: if $\forall i \in C$, $\mathcal{G}_i \cap G' \neq \emptyset$ and $G' \in \mathcal{V}(C)$ then $G \subseteq G'$. We need to show that this implies that if $(\mathcal{M}, \rho) \models [C](\chi_{G'} \wedge \gamma_C^+)$ then $(\mathcal{M}, \rho) \models [\emptyset](\chi_{G'} \rightarrow \pi_G^+)$. So, assume $(\mathcal{M}, \rho) \models [C](\chi_{G'} \wedge \gamma_C^+)$; then G' both satisfies and is feasible for C, and since G is a necessary goal set, $G \subseteq G'$. Thus by Proposition 3(2), $\forall s'$ in $\mathcal{M}$: $(\mathcal{M}, \rho)' \models (\chi_{G'} \rightarrow \pi_G^+)$, and so $(\mathcal{M}, \rho) \models [\emptyset](\chi_{G'} \rightarrow \pi_{G'}^+)$, and we are done. ■

5.9. *Mutual dependence*

If you are a veto player for me, then this might appear to put me in a weak position with respect to you – because I am absolutely reliant upon you for the satisfaction of my goals. Unless, of course, the situation is reciprocal: that is, unless *you* are also dependent upon *me* in return. This consideration gave rise in (Wooldridge *et al.* 2004, p.59) to the notion of a *mutually dependent* coalition, in which everybody is dependent upon everybody else. Formally, coalition C will be mutually dependent if:

$$\forall C' \subseteq \mathcal{A}, \forall G \in \mathcal{V}(C') \text{ if } G \text{ satisfies at least one member of } C \text{ then } C \subseteq C'$$

Notice that saying that C are mutually dependent implies that C are a necessary component of any coalition to achieve their goals; but it does not say that they are a *successful* coalition. They may not be able to cooperate so as to satisfy their goals jointly, or they may require the cooperation of other agents to achieve all their goals.

$$\mathsf{MD}(C) \mathrel{\hat{=}} \bigwedge_{i \in C} \bigwedge_{j \in C} \mathsf{VP}(i, j)$$

PROPOSITION 23. — $\mathsf{MD}(C) \equiv$ *coalition C are mutually dependent.*

PROOF 24. — By definition, C are mutually dependent iff $\forall C' \subseteq \mathcal{A}, \forall G \in \mathcal{V}(C')$ if G satisfies at least one member of C then $C \subseteq C'$. It suffices to note that this is equivalent to saying that, $\forall i, j \in C$, i is a veto player for j and vice versa. ■

5.10. *Empty and trivial games*

We conclude with by characterising two general properties of QCGs. We say a QCG is *empty* if no coalition is successful, and *trivial* if every coalition is successful (Wooldridge *et al.* 2004, pp. 59–60). Although these two types of games represent extremes in the spectrum of possible QCGs it seems inevitable that coalitions will not form in either type of QCG. Coalitions will not form in the first type because to do so would serve no rational purpose: as we argued above, success seems a necessary condition for rational agents to form a coalition, and in an empty game, no coalition is successful. There is also no point in coalitions forming in trivial games, but this time because every agent can achieve its goals in isolation. Note that in this sense, the concept of a trivial game is rather like that of an *inessential game* in cooperative game theory: recall that a conventional coalitional game $\langle \mathcal{A}, \nu : 2^{\mathcal{A}} \to \mathbb{R} \rangle$ is inessential if for all $C \subseteq \mathcal{A}$, we have $\nu(C) = \sum_{i \in C} \nu(\{i\})$.

The nullary predicates EG and TG may be easily seen to characterise empty and trivial games, respectively. (We will not give formal proofs, as the results are obvious.)

$$\mathsf{EG} \quad \hat{=} \quad \bigwedge_{C \subseteq \mathcal{A}} \neg \mathsf{SC}(C) \qquad\qquad \mathsf{TG} \quad \hat{=} \quad \bigwedge_{C \subseteq \mathcal{A}} \mathsf{SC}(C)$$

6. A case study: political voting games

We now present a case study, to illustrate how the ideas and definitions introduced above might be applied in practice to reasoning about multi-agent systems. More precisely, we proceed as follows. We represent a system whose properties we wish to check using the REACTIVE MODULES language of Alur and Henzinger (Alur *et al.* 1999). This language is well-suited to expressing the structure and behaviour of game-like multi-agent systems. We then express the properties of the system that we wish to check (e.g., whether one player is a veto player for another) using ATL, the temporal extension of CL developed by Alur, Henzinger, and Kupferman (Alur *et al.* 2002). In order to do this, we express each agent i's goals $\mathcal{G}_i$ as a formula γ_i of ATL, the idea being that γ_i is satisfied in a state s of the system iff one of the goals in $\mathcal{G}_i$ is achieved in state s. We can then use MOCHA, a model checker for ATL, to check properties of the system in question. We begin with an introduction to the scenario we will study.

Our case study is adapted from (Pauly 2001, pp. 99–101) and (Pauly et Wooldridge 2003), and is concerned with the following voting game:

> *A political body $\mathcal{A} = \{1, 2, 3, 4\}$ has to decide on passing a new law. There are two versions of the law,* `law1` *and* `law2`*, and the process begins by a single agent, agent 2, proposing which of these versions should be adopted. Once agent 2 has selected a version, the entire body votes on whether to accept the proposal; if there is a majority in favour of acceptance, then the proposed version is accepted; if there is a majority against, then there is deadlock, and the process begins again, with agent 2 selecting a version of the law to propose; if there is no majority one way or the other, then the vote of the chairman, agent 1, is decisive, in either accepting the proposed law or returning it to agent 2.*

```
type law : {law1, law2, deadlock}
module Agent1
   external scchosen : law
   interface agree1 : bool
   atom controls agree1 reads scchosen
   update
        [] true -> agree1' := true
        [] true -> agree1' := false
   endatom
endmodule - Agent1
module Agent2
   external subCommittee : bool
   interface agree2 : bool;
        scchosen : law
   atom controls scchosen reads subCommittee
   update
        [] subCommittee -> scchosen' := law1
        [] subCommittee -> scchosen' := law2
   endatom
   atom controls agree2 reads scchosen
   update
        [] true -> agree2' := true
        [] true -> agree2' := false
   endatom
endmodule - Agent2
...
module Outcome
   external
        scchosen : law;
        agree1, agree2, agree3, agree4, subCommittee : bool
   interface
        outcome : law;
        subCommittee : bool
   atom controls subCommittee reads outcome awaits outcome
   init
        [] true -> subCommittee' := true
   update
        [] outcome' = deadlock -> subCommittee' := true
        [] ~(outcome' = deadlock) -> subCommittee' := false
   endatom
   atom controls outcome
        reads scchosen, agree1, agree2, agree3, agree4
        awaits scchosen, agree1, agree2, agree3, agree4
   init
        [] true -> outcome' := deadlock
   update
        [] NoConsensusFmla -> outcome' :=
           if (agree1') then scchosen' else deadlock fi
        [] MajorityInFavourFmla -> outcome' := scchosen'
        [] MajorityAgainstFmla -> outcome' := deadlock
   endatom
endmodule - Outcome
System := (Agent1 || Agent2 || Agent3 || Agent4 || Outcome)
```

Figure 3. *The voting scenario, represented in* REACTIVE MODULES, *the* MOCHA *modelling language.*

We begin by modelling the game in a form suitable for MOCHA — the code is given in Figure 3 (a brief introduction to REACTIVE MODULES, the language used to express the case study, is presented in Appendix A).

– `Agent1` is the “generic” agent in this scenario, which simply has to decide whether to accept or reject the proposed law. The vote of `Agent1` is held in a variable `agree1`. (Agents 3 and 4 are essentially identical, and for this reason we do not give their code.)

– `Agent2` is composed of two separate “update threads”. The first of these is responsible for proposing a law when the mechanism is in the “subcommittee” phase (*i.e.*, at the start, and whenever a proposal has been rejected by the whole political body). The variable `subCommittee` keeps track of when the mechanism is in subcommittee phase. The proposal made by agent 2 is carried in the variable `scchosen`. The second update thread is responsible for deciding whether to accept a proposal, as with the other agents in the system. (Of course, it would in some sense be nonsensical for agent 2 to vote against accepting a proposal that it had put forward, but the mechanism does not preclude it.)

– The `Outcome` module is responsible for determining the outcome, based on the votes of the agents in the system. It is also composed of two update threads. The first is responsible for keeping track of whether the mechanism is in sub-committee phase (this will be initially, and whenever the overall outcome is deadlock). The second update thread decides what the outcome is: the three rules defining this thread correspond to (a) whether there is an equal number of agents for and against accepting the proposal, in which case the mechanism looks to the vote of agent 1, and if this was positive (*i.e.*, agent 1 voted to accept), then the proposal is accepted; otherwise it is rejected; (b) a majority agree on accepting the proposal, in which case the overall outcome is that in the variable `scchosen`; and (c) there is a majority against accepting the proposal, in which case the outcome is `deadlock`. Note that the overall outcome (*i.e.*, `law1`, `law2`, or `deadlock`) is recorded in variable `outcome`. Note that:

- `MajorityInFavourFmla`,
- `MajorityAgainstFmla`, and
- `NoConsensusFmla`

are abbreviations we are using here for MOCHA expressions which capture whether or not there is a majority for or against acceptance, or whether there is a deadlock, respectively.

Let us call the mechanism, as defined by the REACTIVE MODULES code above, $\mathcal{M}_1$.

For each agent i, there are three relevant goal formulae that we might consider for this scenario:

$$\begin{array}{rcl} \gamma_i & = & (outcome = law_1) \\ \gamma_i & = & (outcome = law_2) \\ \gamma_i & = & (outcome = deadlock) \end{array}$$

Thus, an agent might have a goal that the outcome is that law_1 is chosen, that law_2 is chosen, or that no law is chosen (*i.e.*, there is a deadlock). In the remainder of this section, we shall assume that each agent is assigned one of these three goals.

We can now start to investigate some properties of the mechanism. First, note that for any coalition C, if $\mathsf{SC}(C)$ then $\forall i, j \in C$, $\gamma_i = \gamma_j$: that is, a coalition can only be successful in this scenario if they share a common goal. (This is straightforward to see from the fact that the three relevant goals are mutually exclusive.)

The first property we can check is that, if they share a common goal, the coalition $\{1, 2\}$ is all-powerful: this coalition can bring about any outcome, including deadlock.

PROPOSITION 25. — *If* $\gamma_1 = \gamma_2$*, then* $\mathcal{M}_1 \models \mathsf{SC}(\{1, 2\})$.

This property can be checked directly. In contrast to the coalition $\{1, 2\}$, the coalition $\{3, 4\}$ is unsuccessful:

PROPOSITION 26. — $\mathcal{M}_1 \not\models \mathsf{SC}(\{3, 4\})$*, irrespective of the goals of agents* 3 *or* 4.

Again, this property is trivial to check using MOCHA. With respect to the success (or failure) of other coalitions, first note that any coalition of cardinality three with a shared goal containing agent 2 is successful (because agent 2 can propose the law, and the coalition being of majority size can then enforce it).

PROPOSITION 27. — *For all* $C \subseteq \mathcal{A}$*, we have* $\mathcal{M}_1 \models \mathsf{SC}(C)$ *if all of (1) - (3) hold:*

(1) $2 \in C$*;*
(2) $|C| \geq 3$*; and*
(3) $\forall i, j \in C, \gamma_i = \gamma_j$.

Turning to minimal coalitions and the core, we have the following.

PROPOSITION 28. —

1) $\mathcal{M}_1 \models \mathsf{MC}(\{1, 2\})$*, and hence if* $\gamma_1 = \gamma_2$*, then* $\mathcal{M}_1 \models \mathsf{CNE}(\{1, 2\})$.

2) For all $C \subseteq \mathcal{A}$*, we have* $\mathcal{M}_1 \models \mathsf{CNE}(C)$ *(and hence* $\mathcal{M}_1 \models \mathsf{MC}(C)$*) if the following four conditions simultaneously hold:*

a) $2 \in C$*;*
b) $|C| = 3$*;*
c) $\forall i, j \in C, \gamma_i = \gamma_j$*; and*
d) $1 \notin C$.

To see why the second result cannot be strengthened to an "iff", observe that adding an agent would mean that the coalition was no longer minimal, and hence no longer be stable (because a strict subset would be successful).

Let us now consider all possible goals: we can easily check that all goals are individually achievable but mutually exclusive. Goal realisability in the voting setting maps more or less directly to strategic ability in the sense of the cooperation modality $[C]\varphi$, and so (with a small abuse of notation), we have the following.

PROPOSITION 29. —

1) $\mathcal{M}_1 \models \mathsf{GR}(\gamma)$ *for each of the three relevant goals* γ *listed above; but*

2) For any combination $\gamma \wedge \gamma'$ *of the three goals listed above, (*$\gamma \neq \gamma'$*), we have* $\mathcal{M}_1 \not\models \mathsf{GR}(\gamma \wedge \gamma')$.

With respect to veto players, since agent 2 always proposes the law upon which the committee votes, we have the following.

PROPOSITION 30. — *For any allocations of the three relevant goals to agents, we have:*

1) $\mathcal{M}_1 \models \mathsf{VETO}(2, (outcome = law_1)$*;*
2) $\mathcal{M}_1 \models \mathsf{VETO}(2, (outcome = law_2)$*; and*
3) $\mathcal{M}_1 \not\models \mathsf{VETO}(2, (outcome = deadlock))$.

There is also a veto player that can avoid deadlock:

PROPOSITION 31. — *For any allocations of the three relevant goals to agents, we have:*

1) $\mathcal{M}_1 \models \mathsf{VETO}(1, (outcome = deadlock)$*;*

Finally, note that there are no necessary goals in this scenario; and moreover, the scenario is neither trivial nor empty (and is hence "typical").

7. Related work

Although logical formalisms for knowledge representation in multi-agent systems have been studied for decades, until recently the dominant approach was to adopt a "mentalistic" stance, whereby logics — usually, modal logics — were used to characterise the mental states (beliefs, desires, intentions, and the like) of agents: see, e.g., the work of Halpern et al (Halpern et Moses 1992, Fagin, Halpern, Moses et Vardi 1995), Shoham (Shoham 1993), Cohen and Levesque (Cohen et Levesque 1990), Meyer and colleagues (Meyer, Hoek et Linder 1999) and Rao and Georgeff (Rao et Georgeff 1998, Wooldridge 2000) (see (Wooldridge *et al.* 1995, Hoek *et al.* 2003c) for surveys of this work).

More recently, however, game theory has come to be seen as an alternative foundation upon which to develop logic-based knowledge representation formalisms for multi-agent systems. It has been recognised for several decades that there are close links between modal logics of rational agency and the formal theory of games: see, for example, Ladner and Reif's Church/Turing-like thesis for distributed computing, and the conclusions they draw from this (Ladner et Reif 1986, pp. 208–209). Recently, a number of formalisms have been proposed which attempt to synthesise logical and game-theoretic approaches in a single system, in which the links between the game and the logic are explicitly defined. One example is the work of Harrenstein et al (Harrenstein, van der Hoek, Meyer et Witteveen 2002), in which a dynamic logic is used to give a modal characterisation of Nash equilibria – arguably the most important and powerful analytical weapon in the game theory arsenal (Osborne *et al.* 1994, pp. 14–15). Those characterisation results of (Harrenstein *et al.* 2002) are phrased in a format not unlike our correspondence results: on the one hand, there is a solution concept in games, which, on the other hand, are in a formal way connected to truths or validities in an interpretation. In a similar spirit, but more generally, de Bruin (de Bruin 2004) has developed a logical framework in which many solution concepts in games, especially those involving an epistemic component, can be explained, compared or justified.

Other examples of frameworks that combine logical and game-theoretic ideas include the closely related work of Baltag (Baltag 2002), and of course Pauly's Coalition Logic, which we have used in this paper (Pauly 2002b, Pauly 2002a, Pauly 2001). One of Pauly's main concerns was to explore the links between modal logics (in particular, modal logics with a "neighbourhood" semantics) and formal games. Similar explorations have been undertaken by van Benthem, whose starting point is that the labelled transition systems/Kripke structures, which are canonically used to give a semantics to modal logics, can be interpreted as extensive form games, and that as a consequence modal operators of various kinds can be used to express properties of games (Benthem 2001, Benthem 2002). Agotnes, van der Hoek et Wooldridge (2006a) develop a logic specifically intended for reasoning about coalitional games, and Agotnes, van der Hoek et Wooldridge (2006b) investigate the use of temporal logics for representing iterated coalitional games. However, in both of these works, a special purpose logic is developed; in the present paper, we specifically focus on using a general purpose logic for characterising games.

Finally, one issue that our work raises relates to the *size* of formulae that are necessary to characterise properties of QCGs. We have used conjunctions to represent universal quantification and disjunctions to represent existential quantification. Since many properties require quantifying over goals and/or coalitions, we have as a consequence some formulae that are exponential in the number of agents and/or goals. Such formulae are clearly not realistic in practice. The obvious solution to this problem is to extend Coalition Logic with quantification over both agents and coalitions. However, such quantification is clearly dangerous from a computational point of view. Recently, Agotnes, van der Hoek et Wooldridge (2007) proposed a form of limited quantification over coalitions via modal quantifiers. It was shown that this *Quantified Coalition Logic* was exponentially more succinct than Coalition Logic, while being no more complex with respect to the key problem of model checking. It would be interesting to investigate the class of solution concepts that might be succinctly expressed using Quantified Coalition Logic.

8. Conclusions

We have given a systematic logical characterisation of solution concepts from Qualitative Coalitional Games (QCGs), using as our language of expression Pauly's Coalition Logic – the next-time fragment of the Alternating-time Temporal Logic of Alur, Henzinger, and Kupferman. Starting from a notion of when a QCG and a model for Coalition Logic could be said to "correspond" (*i.e.*, when they agreed on what coalitions could bring about), we went on to investigate formally the computational complexity of deciding correspondence between a QCG and a model for Coalition Logic, and then systematically showed how the QCG solution concepts developed in (Wooldridge *et al.* 2004) could be characterised via formula schemes of CL. We then showed how model checkers for ATL could be used to check whether these solution concepts held of particular games; to the best of our knowledge, this is the first time that model checking has actually been used to check such properties of games.

There are, as ever, many possible avenues for future research. One is to investigate the effect that *imperfect information* (Fagin *et al.* 1995) might have in QCGs, and how epistemic extensions to CL and ATL might then be used to express appropriate solution concepts (Hoek *et al.* 2003b). Another possible avenue for further work is to consider temporally extended (*i.e.*, iterated) QCGs and cooperative games, and to investigate appropriate solution concepts for such games. Here, the "full" temporal language of ATL might be appropriate, rather than just the next-time fragment represented by CL.

Acknowledgements

This research was supported by the EPSRC under grants GR/R60836/01 and GR/S62727/01.

9. References

Agotnes T., van der Hoek W., Wooldridge M., "On the Logic of Coalitional Games", *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2006)*, Hakodate, Japan, 2006a.

Agotnes T., van der Hoek W., Wooldridge M., "Temporal Qualitative Coalitional Games", *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2006)*, Hakodate, Japan, 2006b.

Agotnes T., van der Hoek W., Wooldridge M., "Quantified Coalition Logic", *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI-07)*, Hyderabad, India, 2007.

Alur R., de Alfaro L., Henzinger T. A., Krishnan S. C., Mang F. Y. C., Qadeer S., Rajamani S. K., Taşiran S., "MOCHA User Manual", University of Berkeley Report, 2000.

Alur R., Henzinger T. A., "Reactive Modules", *Formal Methods in System Design*, vol. 15, num. 11, pp. 7-48, July, 1999.

Alur R., Henzinger T. A., Kupferman O., "Alternating-time temporal logic", *Journal of the ACM*, vol. 49, num. 5, pp. 672–713, September, 2002.

Alur R., Henzinger T. A., Mang F. Y. C., Qadeer S., Rajamani S. K., Taşiran S., "Mocha: Modularity in Model Checking", *CAV 1998: Tenth International Conference on Computer-aided Verification, (LNCS Volume 1427)*, Springer-Verlag: Berlin, Germany, pp. 521-525, 1998.

Baltag A., "A Logic for Suspicious Players", *Bulletin of Economic Research*, vol. 54, num. 1, pp. 1–46, 2002.

Benthem J., "Correspondence theory", *in* D. Gabbay, F. Guenthner (eds), *Handbook of Philosophical Logic, Volume II: Extensions of Classical Logic*, D. Reidel Publ. Co.: Dordrecht, pp. 167–247, 1984.

Benthem J., "Games in dynamic-epistemic logic", *Bulletin of Economic Research*, vol. 53, num. 4, pp. 219–248, 2001.

Benthem J., "Extensive Games as Process Models", *Journal of Logic, Language, and Information*, vol. 11, num. 3, pp. 289-313, 2002.

Blackburn P., de Rijke M., Venema Y., *Modal Logic*, Cambridge University Press: Cambridge, England, 2001.

Chellas B., *Modal Logic: An Introduction*, Cambridge University Press: Cambridge, England, 1980.

Clarke E. M., Grumberg O., Peled D. A., *Model Checking*, The MIT Press: Cambridge, MA, 2000.

Cohen P. R., Levesque H. J., "Intention is Choice with Commitment", *Artificial Intelligence*, vol. 42, pp. 213–261, 1990.

de Bruin B., Explaining Games – on the logic of game theoretic explanations, PhD thesis, ILLC, University of Amsterdam, 2004.

Drimmelen G., "Satisfiability in Alternating-time Temporal Logic", *Eighteenth Annual IEEE Symposium on Logic in Computer Science (LICS 2003)*, Ottawa, Canada, pp. 208–217, 2003.

Fagin R., Halpern J. Y., Moses Y., Vardi M. Y., *Reasoning About Knowledge*, The MIT Press: Cambridge, MA, 1995.

Goranko V., "Coalition Games and Alternating Temporal Logics", *in* J. van Benthem (ed.), *Proceeding of the Eighth Conference on Theoretical Aspects of Rationality and Knowledge (TARK VIII)*, Siena, Italy, pp. 259-272, 2001.

Halpern J. Y., Moses Y., "A Guide to Completeness and Complexity for Modal Logics of Knowledge and Belief", *Artificial Intelligence*, vol. 54, pp. 319–379, 1992.

Harrenstein P., van der Hoek W., Meyer J.-J., Witteveen C., "On Modal Logic Interpretations for Games", *Proceedings of the Fifteenth European Conference on Artificial Intelligence (ECAI-2002)*, Lyon, France, pp. 28–32, 2002.

Hoek W., Wooldridge M., "Tractable Multiagent Planning for Epistemic Goals", *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2002)*, Bologna, Italy, pp. 1167–1174, 2002.

Hoek W., Wooldridge M., "Model Checking Cooperation, Knowledge, and Time — A Case Study", *Research in Economics*, vol. 57, num. 3, pp. 235–265, September, 2003a.

Hoek W., Wooldridge M., "Time, Knowledge, and Cooperation: Alternating-time Temporal Epistemic Logic and Its Applications", *Studia Logica*, vol. 75, num. 1, pp. 125–157, 2003b.

Hoek W., Wooldridge M., "Towards a Logic of Rational Agency", *Logic Journal of the IGPL*, vol. 11, num. 2, pp. 135–159, 2003c.

Johnson D. S., “A Catalog of Complexity Classes”, *in* J. van Leeuwen (ed.), *Handbook of Theoretical Computer Science Volume A: Algorithms and Complexity*, Elsevier Science Publishers B.V.: Amsterdam, The Netherlands, pp. 67–161, 1990.

Kraus S., “Negotiation and Cooperation in Multi-Agent Environments”, *Artificial Intelligence*, vol. 94, num. 1-2, pp. 79–98, July, 1997.

Ladner R. E., Reif J. H., “The Logic of Distributed Protocols: preliminary report”, *Proceedings of the 1986 Conference on Theoretical Aspects of Reasoning About Knowledge*, Morgan Kaufmann Publishers: San Mateo, CA, pp. 207–222, 1986.

Meyer J., Hoek W., Linder B., “A logical approach to the dynamics of commitments”, *Artificial Intelligence*, vol. 113, pp. 1–40, 1999.

Osborne M. J., Rubinstein A., *A Course in Game Theory*, The MIT Press: Cambridge, MA, 1994.

Papadimitriou C. H., *Computational Complexity*, Addison-Wesley: Reading, MA, 1994.

Pauly M., Logic for Social Software, PhD thesis, University of Amsterdam, 2001. ILLC Dissertation Series 2001-10.

Pauly M., “A Logical Framework for Coalitional Effectivity in Dynamic Procedures”, *Bulletin of Economic Research*, vol. 53, num. 4, pp. 305–324, 2002a.

Pauly M., “A Modal Logic for Coalitional Power in Games”, *Journal of Logic and Computation*, vol. 12, num. 1, pp. 149-166, 2002b.

Pauly M., Wooldridge M., “Logic for Mechanism Design — A Manifesto”, *Proceedings of the 2003 Workshop on Game Theory and Decision Theory in Agent Systems (GTDT-2003)*, Melbourne, Australia, 2003.

Rao A. S., Georgeff M., “Decision Procedures for BDI Logics”, *Journal of Logic and Computation*, vol. 8, num. 3, pp. 293–344, 1998.

Sandholm T., “Distributed Rational Decision Making”, *in* G. Weiß (ed.), *Multiagent Systems*, The MIT Press: Cambridge, MA, pp. 201–258, 1999.

Shoham Y., “Agent-Oriented Programming”, *Artificial Intelligence*, vol. 60, num. 1, pp. 51–92, 1993.

Wooldridge M., *Reasoning about Rational Agents*, The MIT Press: Cambridge, MA, 2000.

Wooldridge M., *An Introduction to Multiagent Systems*, John Wiley & Sons, 2002.

Wooldridge M., Dunne P. E., “On the Computational Complexity of Qualitative Coalitional Games”, *Artificial Intelligence*, vol. 158, num. 1, pp. 27–73, 2004.

Wooldridge M., Jennings N. R., “Intelligent Agents: Theory and Practice”, *The Knowledge Engineering Review*, vol. 10, num. 2, pp. 115–152, 1995.

A. A very brief introduction to reactive modules

Agents in MOCHA are known as *modules*. Each module consist of an *interface* and a number of `atoms`, which define the behaviour of the corresponding agent. The interface part of a module defines two key classes of variables[4]:

– `interface` variables are those variables which the module "owns" and makes use of, but which may be observed by other modules;

– `external` variables are those which the module sees, but which are "owned" by another module (thus a variable that is declared as `external` in one module must be declared as an `interface` variable in another module).

Within each module are a number of `atoms`, which may be thought of as *update threads*. Atoms are rather like "micro modules" — in much the same way to regular modules, they have an interface part, and an action component, which defines their behaviour. The interface of an atom defines the variables it `reads` (*i.e.*, the variables it observes and makes use of in deciding what to do), and those it `controls`. If a variable is controlled by an atom, then no other atom is permitted to make any changes to this variable. A further class of variables may be accessed by an atom are those it `awaits`. An "awaited" variable is one whose value is read by the atom *after it has been defined for the next state*. In short, an atom may view the value of an awaited variable after it has been set, before it needs to make its choices. Clearly, care needs to be taken to ensure that circular chains of awaiting are not established!

The behaviour of an atom is defined by a number of *guarded commands*, which have the following syntactic form.

```
[] guard -> command
```

Here, `guard` is a predicate over the variables accessible to the atom (those it `reads` and `controls`), and `command` is a list of assignment statements. The idea is that if the guard part evaluates to true, then the corresponding command part is "enabled for execution". At any given time, a number of guards in an atom may be true, and so a number of commands within the atom may be enabled for execution. However, only one command within an atom may actually be selected for execution: the various alternatives represent the *choices* available to the agent.

Assignment statements have the following syntactic form.

```
var' := expression
```

On the right-hand side of the assignment statement, `expression` is an expression over the variables that the atom observes. On the left-hand side is a variable which must

4. MOCHA also allows an additional class of `private` (local) variables, which are internal to a module, and which may not be seen by other modules. However, we make no use of these in our model.

be controlled by the atom. The prime symbol on this variable name means "the value of the variable *after* this set of updates are carried out"; an un-primed variable means "the value of this variable *before* updating is carried out". Thus the prime symbol is used to distinguish between the value of variables before and after updating has been carried out.

There are two classes of guarded commands that may be declared in an atom: `init` and `update`. An `init` guarded command is only used in the first round of updating, and as the name suggests, these commands are thus used to initialise the values of variables.