Synthese
DOI 10.1007/s11229-008-9363-1

Quantified coalition logic

Thomas Ågotnes · Wiebe van der Hoek · Michael Wooldridge

Received: 10 February 2008 / Accepted: 11 June 2008

Abstract We add a limited but useful form of quantification to Coalition Logic, a popular formalism for reasoning about cooperation in game-like multi-agent systems. The basic constructs of *Quantified Coalition Logic* (QCL) allow us to express such properties as "every coalition satisfying property P can achieve φ" and "there exists a coalition C satisfying property P such that C can achieve φ". We give an axiomatisation of QCL, and show that while it is no more expressive than Coalition Logic, it is nevertheless exponentially more succinct. The complexity of QCL model checking for symbolic and explicit state representations is shown to be no worse than that of Coalition Logic, and satisfiability for QCL is shown to be no worse than satisfiability for Coalition Logic. We illustrate the formalism by showing how to succinctly specify such social choice mechanisms as majority voting, which in Coalition Logic require specifications that are exponentially long in the number of agents.

Keywords Coalition logic · Quantification · Succinctness · Model checking · Satisfiability

1 Introduction

Game theoretic models of cooperation have proved a valuable source of techniques and insights for the field of multi-agent systems, and cooperation logics such as

T. Ågotnes
Department of Computer Engineering, Bergen University College, Bergen, Norway
e-mail: tag@hib.no

W. van der Hoek (✉) · M. Wooldridge
Department of Computer Science, University of Liverpool, Liverpool, England, UK
e-mail: wiebe@csc.liv.ac.uk

M. Wooldridge
e-mail: mjw@csc.liv.ac.uk

Alternating-time Temporal Logic (ATL) (Alur et al. 2002) and Coalition Logic (CL) (Pauly 2001) have proved to be powerful and intuitive knowledge representation formalisms for such models. Many important properties of game-like cooperative scenarios require *quantification over coalitions*. However, existing cooperation logics provide no direct facility for such quantification, and expressing such properties therefore requires formulae that are exponentially long in the number of agents. Examples include expressing the notion of a *weak veto player* (Wooldridge and Dunne 2004) in CL, or solution concepts from cooperative game theory such as non-emptiness of the core in Coalitional Game Logic (Ågotnes et al. 2006). An obvious solution would be to extend, for example, ATL, with a first-order-style apparatus for quantifying over coalitions. In such a quantified ATL, one might express the fact that agent i is a necessary component of every coalition able to achieve φ by the following formula:

$$\forall C\colon \langle\langle C\rangle\rangle \Diamond\varphi \rightarrow (i \in C)$$

However, adding quantification in such a naive way leads to undecidability over infinite domains (using basic quantificational set theory we can define arithmetic), and very high computational complexity even over finite domains. The question therefore arises whether we can add quantification to cooperation logics in such a way that we can express useful properties of cooperation in games *without* making the resulting logic too computationally complex to be of practical interest. Here, we answer this question in the affirmative.

We introduce *Quantified Coalition Logic* (QCL), by modifying the existing cooperation modalities of CL in order to enable quantification. In CL, the basic cooperation constructs are $\langle\langle C\rangle\rangle\varphi$, meaning that coalition C can achieve φ; these operators are in fact modal operators with a neighbourhood semantics. In QCL, we replace these operators with expressions $\langle P\rangle\varphi$ and $[P]\varphi$; here, P is a *predicate over coalitions*, and the two sentences express the fact that *there exists a coalition C satisfying property P such that C can achieve* φ and *all coalitions satisfying property P can achieve* φ, respectively. Thus we add a limited form of quantification to CL *without* the apparatus of quantificational set theory. Our key contributions are twofold. First, we show that while QCL is equally expressive as CL, it is exponentially more succinct. And second, we show that QCL is no worse than CL with respect to the key computational problems of model checking and satisfiability.

The remainder of the paper is structured as follows. After briefly reviewing the main concepts of Pauly's Coalition Logic, we introduce a language for expressing coalition predicates, and show that the satisfiability problem for this language is NP-complete. We then introduce QCL itself, and give a complete axiomatisation. We show that while QCL is no more expressive than Coalition Logic (i.e., there are no properties expressible in QCL that can not be expressed in Coalition Logic), it is nevertheless exponentially more succinct, in a precise and formal sense. We then study two computational properties of the logic. First, we show that the complexity of the model checking problem is no worse than that of Coalition Logic, assuming an explicit representation of models. We also study the problem under a more realistic representation. Second, we show that the complexity of the satisfiability problem is also no worse than that of Coalition Logic. We then extend the language of coalition predicates in order to be able to succinctly express properties related to the cardinality of coalitions, and show completeness results for the extended logic. We illustrate QCL by showing how it can be used to succinctly specify such social choice mechanisms as majority voting, which in Coalition Logic require specifications that are exponentially long in the number of agents.

2 Background: coalition logic

Since QCL is based on Pauly's Coalition Logic CL (Pauly 2001), we first briefly introduce the latter. CL is a propositional modal logic, containing an indexed collection of unary modal operators $\langle\langle C\rangle\rangle$, where C is a coalition, i.e., a subset of a given set of agents Ag. The intended interpretation of $\langle\langle C\rangle\rangle\varphi$ is that 'C *can achieve* φ', or, that 'C *is effective for* φ', or that 'C has a choice such that φ'. One warning regarding notation is in place, here. In Pauly (2001), the notation for 'C has a choice such that φ' is in fact $[C]\varphi$. Since we will later use $\langle\cdot\rangle$ and $[\cdot]$ for an existential and universal statement over kinds of coalitions, and the construct $[C]\varphi$ of Pauly (2001) will turn out to be a special case of our $\langle\cdot\rangle\varphi$, in this paper we use $\langle\langle C\rangle\rangle\varphi$ where Pauly (2001) uses $[C]\varphi$.

Formulae of CL are defined by the following grammar (with respect to a set Φ_0 of Boolean variables, and a fixed set Ag of agents):

$$\varphi ::= \top \mid p \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle\langle C\rangle\rangle\varphi$$

where $p \in \Phi_0$ is an atomic proposition and C a subset of Ag. As usual, we use parentheses to disambiguate formulae where necessary, and define the remaining connectives of classical logic as abbreviations: $\bot \equiv \neg\top$, $\varphi \wedge \psi \equiv \neg(\neg\varphi \vee \neg\psi)$, $\varphi \rightarrow \psi \equiv (\neg\varphi) \vee \psi$ and, finally, $\varphi \leftrightarrow \psi \equiv (\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi)$.

A model $\mathcal{M}$ for (over Φ_0, Ag) is a triple $\mathcal{M} = \langle S, \mathcal{E}, \pi\rangle$ where

- $S = \{s_1, \ldots, s_o\}$ is a finite non-empty set of *states*;
- $\mathcal{E}\colon 2^{Ag} \times S \rightarrow 2^{2^S}$ is an *effectivity function*, where $T \in \mathcal{E}(C, s)$ is intended to mean that from state s, the coalition C can cooperate to ensure that the next state will be a member of T—note that they cannot determine *which* of the members of T will occur—they can only be sure that it will be *some* member of T; and
- $\pi\colon S \rightarrow 2^{\Phi_0}$ is a valuation function, which for every state $s \in \mathcal{S}$ gives the set $\pi(s)$ of Boolean 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). Throughout this paper, we assume that models $\mathcal{M}$ are *weak playability* models $\mathcal{M} \in \mathcal{WP}$, in which effectivity functions have the following properties (Pauly 2001, p. 30):

- *Outcome monotonicity*: $\forall X \subseteq X' \subseteq S, C \subseteq Ag$, if $X \in \mathcal{E}(C, s)$ then $X' \in \mathcal{E}(C, s)$
- $\emptyset \notin \mathcal{E}(Ag, s)$
- $\forall C' \subseteq C \subseteq Ag$ if $\emptyset \in \mathcal{E}(C, s)$ then $\emptyset \in \mathcal{E}(C', s)$
- $\forall C \subseteq Ag$ if $\emptyset \notin \mathcal{E}(\emptyset, s)$ then $S \in \mathcal{E}(C, s)$
- *Ag-maximality*: $\forall X \subseteq S\ ((S \setminus X) \notin \mathcal{E}(\emptyset, s) \Rightarrow X \in \mathcal{E}(Ag, s))$

– *Superadditivity*: $\forall X_1, X_2 \subseteq S, C_1, C_2 \subseteq Ag\ (C_1 \cap C_2 = \emptyset), (X_1 \in \mathcal{E}(C_1, s)), (X_2 \in \mathcal{E}(C_2, s)) \Rightarrow X_1 \cap X_2 \in \mathcal{E}(C_1 \cup C_2, s)$

For a brief explanation of these assumptions, see below. By "model" we henceforth mean this general "weak playability model". For more restricted classes of models, see Pauly (2001).

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 "$\models_{CL}$" for CL holds between interpretations and formulae of CL. We say that coalition C *can enforce* φ in s if for some $T \in \mathcal{E}(C, s)$, φ is true in all $t \in T$. That is, C can make a choice such that, irrespective of the others' choices, φ will hold. Formally, the satisfaction relation is defined as follows:

– $\mathcal{M}, s \models_{CL} \top$
– $\mathcal{M}, s \models_{CL} p$ iff $p \in \pi(s)$ (where $p \in \Phi_0$)
– $\mathcal{M}, s \models_{CL} \neg\varphi$ iff $\mathcal{M}, s \not\models_{CL} \varphi$
– $\mathcal{M}, s \models_{CL} \varphi \vee \psi$ iff $\mathcal{M}, s \models_{CL} \varphi$ or $\mathcal{M}, s \models_{CL} \psi$
– $\mathcal{M}, s \models_{CL} \langle\langle C\rangle\rangle\varphi$ iff $\exists T \in \mathcal{E}(C, s)$ such that $\forall t \in T$, we have $\mathcal{M}, t \models_{CL} \varphi$.

Coming back to the constraints on weak playability models, note that the smaller a set X is that a coalition C can enforce, the more specific and stronger the property that they can ensure. *Outcome monotonicity* says that if a coalition C can enforce an outcome in a set, it can enforce an outcome in any bigger set. (Or: if C can enforce something, it also can enforce all its consequences). The second assumption says that the grand coalition Ag cannot achieve the ultimate strong property $\bot$ (if $\emptyset \in \mathcal{E}(Ag, s)$, we would have $\mathcal{M}, s \models_{CL} \bot$). Given this constraint, the third assumption ensures that in fact *no* coalition can achieve $\bot$. Note that a generalisation of this third constraint, i.e., $\forall C' \subseteq C \subseteq Ag, \forall X \subseteq S$ if $X \in \mathcal{E}(C, s)$ then $X \in \mathcal{E}(C', s)$ (*Coalition monotonicity*) is not required here. Under the first three assumptions, the fourth one says that any coalition C can achieve *something*: It is equivalent to saying that there must be some $X \in \mathcal{E}(C, s)$, for every C and s. *Supperaditivity* explains how coalitions can join forces. If C_1 can enforce X_1 (satisfying φ_1, for example) and C_2 can enforce X_2 (satisfying φ_2), then, given those coalitions are disjoint, they can both exercise their ability to enforce $X_1 \cap X_2$ (they can guarantee $\varphi_1 \wedge \varphi_2$). This requirement implies $X \in \mathcal{E}(Ag, s) \Rightarrow (S \setminus X) \notin \mathcal{E}(\emptyset, s)$. Now *Ag-maximality* is the converse of the latter. Its contraposition reads: If Ag cannot enforce X, it implies that $(S \setminus X)$ is already enforced, even by the coalition that takes no agents.

For a formula φ and model $\mathcal{M}$, let $\varphi^{\mathcal{M}}$ denote the set of states $\{s \in S : \mathcal{M}, s \models \varphi\}$. Observe that $\mathcal{M}, s \models \langle\langle C\rangle\rangle\psi$ iff $\psi^{\mathcal{M}} \in \mathcal{E}(C, s)$.

The notions of truth of φ in a model ($\mathcal{M} \models_{CL} \varphi$) and validity in a class of models $\mathcal{C}$ ($\mathcal{C} \models_{CL} \varphi$) are defined as usual. The inference relation $\vdash_{CL}$ for CL is given in Table 1 (taken from Pauly 2001, but adapted to our notation): it is sound and complete with respect to the class of weak playability models $\mathcal{WP}$ (Pauly 2001, p. 55).

3 Quantified coalition logic

If we have n agents in Ag, and one wants to express that *some* coalition can enforce some atomic property p, one needs to enumerate 2^n disjunctions of the form $\langle\langle C\rangle\rangle p$.

Table 1 Axioms and rules for coalition logic

In $(Prop)$, ψ is a propositional tautology, in axiom $(\bot)$, we require $C' \subseteq C$, and for (S), $C_1 \cap C_2 = \emptyset$

$Prop$	$\vdash_{CL} \psi$
$Ag\bot$	$\vdash_{CL} \neg\langle\langle Ag\rangle\rangle\bot$
$\top$	$\vdash_{CL} \neg\langle\langle\emptyset\rangle\rangle\bot \rightarrow \langle\langle C\rangle\rangle\top$
$\bot$	$\vdash_{CL} \langle\langle C\rangle\rangle\bot \rightarrow \langle\langle C'\rangle\rangle\bot$
Ag	$\vdash_{CL} \neg\langle\langle\emptyset\rangle\rangle\neg\varphi \rightarrow \langle\langle Ag\rangle\rangle\varphi$
S	$\vdash_{CL} (\langle\langle C_1\rangle\rangle\varphi_1 \wedge \langle\langle C_2\rangle\rangle\varphi_2) \rightarrow \langle\langle C_1 \cup C_2\rangle\rangle(\varphi_1 \wedge \varphi_2)$
MP	$\vdash_{CL} \varphi \rightarrow \psi,\ \vdash_{CL} \varphi \Rightarrow \vdash_{CL} \psi$
$Distr$	$\vdash_{CL} \varphi \rightarrow \psi \Rightarrow \vdash_{CL} \langle\langle C\rangle\rangle\varphi \rightarrow \langle\langle C\rangle\rangle\psi$

The idea behind Quantified Coalition Logic (QCL) is to avoid this blow-up in the length of formulae. Informally, QCL is a propositional modal logic, containing an indexed collection of unary modal operators $\langle P\rangle\varphi$ and $[P]\varphi$. The intended interpretation of $\langle P\rangle\varphi$ is that *there exists a set of agents C, satisfying predicate P, such that C can achieve* φ. We refer to expressions P as *coalition predicates*, and we now define a language for coalition predicates; QCL will then be parameterised with respect to such a language. Of course, many coalition predicate languages are possible, with different properties, and later we will investigate another such language. Throughout the remainder of this paper, we will assume a fixed, finite set Ag of agents.

3.1 Coalition predicates

Syntactically, we introduce two atomic predicates $subseteq$ and $supseteq$, and derive other predicate forms from these. Formally, the syntax of coalition predicates is given by the following grammar:

$$P ::= subseteq(C) \mid supseteq(C) \mid \neg P \mid P \vee P$$

where $C \subseteq Ag$ is a set of agents. One can think of the atomic predicates $subseteq(C)$ and $supseteq(C)$ as a stock of $2^{|Ag|+1}$ propositions, one for each coalition, which are then to be evaluated in a given coalition Co. The circumstances under which a concrete coalition Co satisfies a coalition predicate P, are specified by a satisfaction relation "$\models_{cp}$", defined by the following four rules:

$Co \models_{cp} subseteq(C)$ iff $Co \subseteq C$
$Co \models_{cp} supseteq(C)$ iff $Co \supseteq C$
$Co \models_{cp} \neg P$ iff not $Co \models_{cp} P$
$Co \models_{cp} P_1 \vee P_2$ iff $Co \models_{cp} P_1$ or $Co \models_{cp} P_2$

Now we can be precise about what it means that "a coalition Co satisfies P": it just means $Co \models_{cp} P$. We will assume the conventional definitions of implication ($\rightarrow$), biconditional ($\leftrightarrow$), and conjunction ($\wedge$) in terms of $\neg$ and $\vee$.

Coalitional predicates $subseteq(\cdot)$ and $supseteq(\cdot)$ are in fact not independent. They are mutually definable—due to the fact that the set of all agents Ag is assumed to be finite. We then have that (Ågotnes and Walicki 2006)

$$subseteq(C) \equiv \bigwedge_{i \in Ag \setminus C} \neg supseteq(\{i\})$$

and

$$supseteq(C) \equiv \bigwedge_{C' \subseteq Ag, C \not\subseteq C'} \neg subseteq(C').$$

The reason that we include both types of predicates as primitives is a main motivating factor of this paper: we are interested in *succinctly* expressing quantification in coalition logic.

We find it convenient to make use of the following derived predicates:

$$\begin{aligned}
eq(C) &\equiv subseteq(C) \wedge supseteq(C) \\
subset(C) &\equiv subseteq(C) \wedge \neg eq(C) \\
supset(C) &\equiv supseteq(C) \wedge \neg eq(C) \\
incl(i) &\equiv supseteq(\{i\}) \\
excl(i) &\equiv \neg incl(i) \\
any &\equiv supseteq(\emptyset) \\
nei(C) &\equiv \bigvee_{i \in C} incl(i) \\
ei(C) &\equiv \neg nei(C)
\end{aligned}$$

The reader may note an obvious omission here: we have not introduced any explicit way of talking about the *cardinality* of coalitions; such predicates will be discussed in Sect. 6.

We say that a coalition predicate P is Ag*-consistent* if for some $Co \subseteq Ag$, we have $Co \models_{cp} P$, and P is Ag*-valid* if $Co \models_{cp} P$ for all $Co \subseteq Ag$.

The *model checking problem for coalition predicates* is the problem of checking whether, for given Co and P, we have $Co \models_{cp} P$ (Clarke et al. 2000). It is easy to see that this problem is decidable in polynomial time. The *satisfiability problem for coalition predicates* is the problem of deciding whether P is consistent. We get the following.

Theorem 1 *The satisfiability problem for coalition predicates is NP-complete.*

Proof There is an easy reduction from SAT (Papadimitriou 1994, p. 171) that gives NP-hardness: given an instance φ of SAT, systematically replace every Boolean variable p that occurs in φ by $incl(p)$. The resulting coalition predicate will clearly be satisfiable iff the SAT instance φ is satisfiable. To show membership in NP, we use a standard "guess and check" approach. However, we must first show that a coalition predicate P is satisfiable iff there is a coalition that is witness to this that is a "short certificate", i.e., of size at most polynomial in the size of the given coalition predicate P. The right-to-left direction is immediate, so consider the left-to-right direction. Let $Ag(P)$ denote the set of all agents named in P. Suppose $C \models_{cp} P$. It is straightforward to see that, by the semantics of coalition predicates, there must exist a sequence of sets[1]

$$C_{1_1}, \ldots, C_{1_{k_1}}, C_{2_1}, \ldots, C_{2_{k_2}}, C_{3_1}, \ldots, C_{3_{k_3}}, C_{4_1}, \ldots, C_{4_{k_4}}$$

such that

$$\begin{array}{l} C \subseteq C_{1_1}, \cdots, C \subseteq C_{1_{k_1}}, \\ C \supseteq C_{2_1}, \cdots, C \supseteq C_{2_{k_2}}, \\ C \not\subseteq C_{3_1}, \cdots, C \not\subseteq C_{3_{k_3}}, \\ C \not\supseteq C_{4_1}, \cdots, C \not\supseteq C_{4_{k_4}} \end{array}$$

where each C_{i_j} appears in a $supseteq(\ldots)$ or $subseteq(\ldots)$ predicate in P, and so

$$Ag(P) \supseteq \bigcup_{1 \leq i \leq 4} \bigcup_{1 \leq j \leq k_i} C_{i_j}$$

We refer to four sets of constraints as C_1 constraints, C_2 constraints, and so on. Note that each C_i could be empty. We reason by cases:

– Case 1: there are C_1 constraints. In this case, $C \subseteq (C_{1_1} \cap \cdots \cap C_{1_{k_1}})$, and so $|C| \leq |Ag(P)|$.
– Case 2: there are no C_1 constraints. In this case, we know that:

$$\begin{array}{l} C \supseteq (C_{2_1} \cup \cdots \cup C_{2_{k_2}}) \\ \exists x_{3_1} \in C : x_{3_1} \notin C_{3_1} \\ \ldots \\ \exists x_{3_{k_3}} \in C : x_{3_{k_3}} \notin C_{3_{k_3}} \\ \exists x_{4_1} \in C_{4_1} : x_{4_1} \notin C \\ \ldots \\ \exists x_{4_{k_4}} \in C_{4_{k_4}} : x_{4_{k_4}} \notin C \end{array}$$

So, let a^* be a new element, not occurring in $Ag(P)$ (in the case that $Ag(P) = Ag$ we are done), and define

$$C^* = C_{2_1} \cup \cdots \cup C_{2_{k_2}} \cup \{a^*\}$$

Notice that, so defined, $|C^*| \leq |Ag(P)| + 1$. By construction, C^* satisfies properties C_2, C_3, and C_4.

The remaining cases are similarly straightforward. So, to verify that a coalition predicate P is satisfiable, we can (i) guess a coalition C such that $|C| \leq |Ag(P)| + 1$; and (ii) verify that $C \models_{cp} P$, which can be done in time polynomial in the size of C and P. □

[1] Assume P is in disjunctive normal form. C must satisfy a conjunction of literals each of one of the four following forms: $subseteq(C_{1_j})$, $supseteq(C_{2_j})$, $\neg subseteq(C_{3_j})$, $\neg supseteq(C_{4_j})$.

3.2 Syntax and semantics of quantified coalition logic QCL

We now introduce Quantified Coalition Logic (QCL), an extension of Pauly's Coalition Logic (Pauly 2001). Informally, QCL is a propositional modal logic, containing an indexed collection of unary modal operators $\langle P \rangle \varphi$ and $[P]\varphi$, where P is a coalition predicate.

Formulae of QCL are defined by the following grammar (with respect to a set Φ_0 of Boolean variables, a fixed set Ag of agents, and the language of coalition predicates):

$$\varphi ::= \top \mid p \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle P \rangle \varphi \mid [P]\varphi$$

where $p \in \Phi_0$ is an atomic proposition and P is a coalition predicate over Ag. As usual, we use parentheses to disambiguate formulae where necessary, and define the remaining connectives of classical logic as abbreviations: $\bot \equiv \neg\top$, $\varphi \rightarrow \psi \equiv (\neg\varphi) \vee \psi$ and $\varphi \leftrightarrow \psi \equiv (\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi)$.

Models for QCL are exactly the same as the models for CL (i.e., weak playability models), and *interpretations* for QCL are also the same as interpretations for CL, i.e., they are pairs $\mathcal{M}, s$ where $\mathcal{M}$ is a model and s is a state in $\mathcal{M}$. The satisfaction relation "$\models_{QCL}$" for QCL holds between interpretations and formulae of CL. The satisfaction relation is defined by the following inductive rules:

$\mathcal{M}, s \models_{QCL} \top$
$\mathcal{M}, s \models_{QCL} p$ iff $p \in \pi(s)$ (where $p \in \Phi_0$)
$\mathcal{M}, s \models_{QCL} \neg\varphi$ iff $\mathcal{M}, s \not\models_{QCL} \varphi$
$\mathcal{M}, s \models_{QCL} \varphi \vee \psi$ iff $\mathcal{M}, s \models_{QCL} \varphi$ or $\mathcal{M}, s \models_{QCL} \psi$
$\mathcal{M}, s \models_{QCL} \langle P \rangle \varphi$ iff $\exists C \subseteq Ag$: $C \models_{cp} P$ and $\exists S \in \mathcal{E}(C, s)$ such that $\forall s' \in S$, we have $\mathcal{M}, s' \models_{QCL} \varphi$.
$\mathcal{M}, s \models_{QCL} [P]\varphi$ iff $\forall C \subseteq Ag$: $C \models_{cp} P$ implies $\exists S \in \mathcal{E}(C, s)$ such that $\forall s' \in S$, we have $\mathcal{M}, s' \models_{QCL} \varphi$.

Remark 1 Readers familiar with modal logic may wonder why we did not introduce the universal coalition modality $[P]\varphi$ as the dual $\neg\langle P \rangle \neg\varphi$. In fact such a definition would not serve the desired purpose. Consider the pattern of quantifiers in the semantics of $\langle \cdot \rangle$: $\exists\exists\forall$. Taking the dual $\neg\langle \cdot \rangle \neg$ would yield the quantifiers $\forall\forall\exists$, rather than the desired $\forall\exists\forall$ pattern. Of course, this does not mean that $[P]\varphi$ is not definable from $\langle P \rangle \varphi$ (and the propositional connectives) in some *other* way. In fact:

$$[P]\varphi \equiv \bigwedge_{\{C \mid C \models_{pc} P\}} \langle eq(C) \rangle \varphi$$

Thus, for expressiveness, $\langle P \rangle$ together with the propositionals are adequate connectives, and $[P]\varphi$ is definable. The reason we introduce the box cooperation modality as a separate construct is one of the main motivations in this paper, as discussed before for the different predicate operators: succinctness of expression.

Let us define the dual $\langle\langle P\rangle\rangle\varphi$ as $\neg\langle P\rangle\neg\varphi$. Then we get:

$$\begin{aligned}\langle\langle P\rangle\rangle\varphi &\equiv \neg\langle P\rangle\neg\varphi\\ &\equiv \text{ not } \exists C \subseteq Ag\big(C \models_{cp} P \ \&\ \exists S \in \mathcal{E}(C,s)\forall s' \in S(\mathcal{M}, s' \models_{QCL} \neg\varphi)\big)\\ &\equiv \forall C \subseteq Ag\big(C \models_{cp} P \Rightarrow \forall S \in \mathcal{E}(C,s)\exists s' \in S : M, s' \models_{QCL} \varphi\big)\end{aligned}$$

and, similarly for $[[P]]$:

$$\begin{aligned}[[P]]\varphi &\equiv \neg[P]\neg\varphi\\ &\equiv \text{ not } \forall C \subseteq Ag\big(C \models_{cp} P \Rightarrow \exists S \in \mathcal{E}(C,s)\forall s' \in S(\mathcal{M}, s' \models_{QCL} \neg\varphi)\big)\\ &\equiv \exists C \subseteq Ag\big(C \models_{cp} P \& \forall S \in \mathcal{E}(C,s)\exists s' \in S : M, s' \models_{QCL} \varphi\end{aligned}$$

In the following we summarise the interpretation of our four modalities. We say that coalition C *can enforce* φ in s if for some $S \in \mathcal{E}(C,s)$, φ is true in all $s' \in S$. That is, C can make a choice such that, irrespective of the others' choices, φ.

$\mathcal{M}, s \models_{QCL} \langle P\rangle\varphi$	Some coalition satisfying P can enforce φ in s
$\mathcal{M}, s \models_{QCL} [P]\varphi$	All coalitions satisfying P can enforce φ in s
$\mathcal{M}, s \models_{QCL} \langle\langle P\rangle\rangle\varphi$	No coalition satisfying P can enforce $\neg\varphi$ in s
$\mathcal{M}, s \models_{QCL} [[P]]\varphi$	Some coalition satisfying P is unable to enforce $\neg\varphi$ in s

3.3 Some QCL expressions

To get a flavour of the kind of properties we can express in QCL, we present some example QCL formulae. First, note that the conventional CL/ATL ability expression is defined simply as:

$$\langle\!\langle C\rangle\!\rangle\varphi \ \equiv\ \langle eq(C)\rangle\varphi.$$

We can also succinctly express properties such as the solution concepts from *Qualitative Coalitional Games* (Wooldridge and Dunne 2004; Dunne et al. 2007). For example, a *weak veto player* for φ is an agent that must be present in any coalition that has the ability to bring about φ:

$$WVETO(i,\varphi) \ \equiv\ \neg\langle excl(i)\rangle\varphi.$$

Of course, if *no* coalition has the ability to achieve φ, then this means that *every* agent is a veto player for φ. A *strong veto player* for φ is thus an agent that is both a weak veto player for φ and that is a member of *some* coalition that can achieve φ:

$$VETO(i,\varphi) \ \equiv\ WVETO(i,\varphi) \wedge \langle incl(i)\rangle\varphi.$$

Finally, a *dictator* for φ is a weak veto player such that *every* coalition that includes that player is able to achieve φ:

$$DICT(i,\varphi) \ \equiv\ WVETO(i,\varphi) \wedge [incl(i)]\varphi$$

A coalition C is *weakly minimal for* φ if no subset of C can achieve φ:

$$WMIN(C, \varphi) \equiv \neg\langle subset(C)\rangle\varphi.$$

And C is simply *minimal* if they are weakly minimal and also able to bring about φ:

$$MIN(C, \varphi) \equiv \langle eq(C)\rangle\varphi \wedge WMIN(C, \varphi).$$

Finally, $GC(C)$ says that C is the *grand* coalition:

$$GC(C) \equiv [supset(C)]\bot$$

4 Expressiveness, axiomatisation and succinctness

We now argue that QCL is equivalent in expressive power to Coalition Logic. To begin, consider the following translation τ from QCL formulae to CL formulae.

$$\begin{aligned}
\tau(\top) &= \top \\
\tau(p) &= p \\
\tau(\neg\varphi) &= \neg\tau(\varphi) \\
\tau(\varphi_1 \vee \varphi_2) &= \tau(\varphi_1) \vee \tau(\varphi_2) \\
\tau(\langle P\rangle\varphi) &= \bigvee\nolimits_{\{C|C\models_{pc}P\}} \langle\langle C\rangle\rangle\tau(\varphi) \\
\tau([P]\varphi) &= \bigwedge\nolimits_{\{C|C\models_{pc}P\}} \langle\langle C\rangle\rangle\tau(\varphi)
\end{aligned}$$

We already know from the discussion above that we have a translation in the other direction: let us call it δ, with defining clause

$$\delta(\langle\langle C\rangle\rangle\varphi) = \langle eq(C)\rangle\delta(\varphi).$$

As an example, suppose $Ag = \{a, b, c\}$ and let $P = (supset(\{a\}) \vee supset(\{b\}) \vee supset(\{c\})) \wedge \neg eq(\{a, b, c\})$. Now, consider the QCL formula $\psi = \langle P\rangle q$. Then $\tau(\psi) = \langle\langle\{a, b\}\rangle\rangle q \vee \langle\langle\{a, c\}\rangle\rangle q \vee \langle\langle\{b, c\}\rangle\rangle q$ while $\delta(\tau(\psi)) = \langle\langle eq(\{a, b\})\rangle\rangle q \vee \langle\langle eq(\{a, c\})\rangle\rangle q \vee \langle\langle eq(\{b, c\})\rangle\rangle q$.

Hence, one can think of $\delta(\tau(\varphi))$ as a *normal form* for φ, where the only coalition predicate in φ is eq. That QCL and CL have equal expressive power follows from the fact that the two translations preserve truth.

Theorem 2 *Let $\mathcal{M}$ be a model, and s a state, and let φ be a QCL formula, and ψ a CL formula. Then*:

1. $\mathcal{M}, s \models_{QCL} \varphi$ *iff* $\mathcal{M}, s \models_{CL} \tau(\varphi)$
2. $\mathcal{M}, s \models_{CL} \psi$ *iff* $\mathcal{M}, s \models_{QCL} \delta(\psi)$

Proof The proof follows immediately from the definition of τ and δ. We consider only one of the inductive steps for each translation.

1. Suppose $\mathcal{M}, s \models_{QCL} \langle P\rangle\varphi$, and the theorem proven for φ. According to the truth-definition in QCL, there must be a coalition C for which $C \models_{cp} P$, and with an $S \in \mathcal{E}(C, s)$ such that for all $s' \in S$, one has $\mathcal{M}, s' \models_{QCL} \varphi$. Using the induction hypothesis, this means that there is a coalition C such that $C \models_{cp} P$ and for which $\mathcal{M}, s \models_{CL} \langle\langle C\rangle\rangle\tau(\varphi)$. This can be written using only a CL formula: $\mathcal{M}, s \models_{CL} \bigvee_{\{C|C\models_{cp}P\}} \langle\langle C\rangle\rangle\tau(\varphi)$.
   For the other direction, suppose $\mathcal{M}, s \models_{CL} \bigvee_{\{C|C\models_{cp}P\}} \langle\langle C\rangle\rangle\tau(\varphi)$. Interpreting the disjunction, this means that there is some C for which $C \models_{cp} P$ and for which $\mathcal{M}, s \models_{CL} \langle\langle C\rangle\rangle\tau(\varphi)$. Using the induction hypothesis and the truth definition of $\langle P\rangle\varphi$ in QCL, we conclude $\mathcal{M}, s \models_{QCL} \langle P\rangle\tau(\varphi)$.
2. Suppose $\mathcal{M}, s \models_{CL} \langle\langle C\rangle\rangle\psi$, and the induction hypothesis applicable to ψ. We then know that coalition C can guarantee ψ in CL. This means in QCL that $\mathcal{M}, s \models_{QCL} \langle eq(C)\rangle\delta(\psi)$. Conversely, if the latter holds, this means that C can bring about the CL formula ψ. Using the induction hypothesis and the truth definition in CL, we obtain $\mathcal{M}, s \models_{CL} \langle\langle C\rangle\rangle\psi$. □

4.1 Completeness

The translations introduced above provide the key to a complete axiomatisation of QCL. First, recall Pauly's axiomatisation of Coalition Logic (Table 1). Given this, and the translations defined previously, we obtain an axiom system for QCL-formulae as follows. First, QCL includes the δ translation of all the CL axioms and rules, and axioms that state that the δ-translation is correct: see the lower part of Table 2. On top of that, QCL is *parametrised* by an inference relation $\vdash_{cp}$ for coalition predicates. The axioms for this in Table 2 are taken from Ågotnes and Walicki (2006).

Theorem 3

1. *$\vdash_{cp}$ is sound and complete: for any P, $\models_{cp} P \Leftrightarrow \vdash_{cp} P$*
2. *For any CL formula φ, $\vdash_{CL} \varphi \Rightarrow \vdash_{QCL} \delta(\varphi)$*

Table 2 Axioms and rules for quantified coalition logic

$P0$	$\vdash_{cp} supseteq(\emptyset)$
$P1$	$\vdash_{cp} supseteq(C) \wedge supseteq(C') \leftrightarrow supseteq(C \cup C')$
$P2$	$\vdash_{cp} supseteq(C) \rightarrow \neg subseteq(C')$
$P3$	$\vdash_{cp} subseteq(C \cup \{a\}) \wedge \neg supseteq(a) \rightarrow subseteq(C)$
$P4$	$\vdash_{cp} subseteq(C) \rightarrow subseteq(C')$
Prop	$\vdash_{cp} \psi$
MP	$\vdash_{cp} \varphi \rightarrow \psi,\ \vdash_{cp} \varphi \Rightarrow \vdash_{cp} \psi$
δAx	$\vdash_{QCL} \delta(Ax)$
$\delta\langle\rangle$	$\vdash_{QCL} \langle P\rangle\varphi \leftrightarrow \bigvee_{\{C\|\vdash_{cp} eq(C)\rightarrow P\}} \langle eq(C)\rangle\varphi$
$\delta[]$	$\vdash_{QCL} [P]\varphi \leftrightarrow \bigwedge_{\{C\|\vdash_{cp} eq(C)\rightarrow P\}} \langle eq(C)\rangle\varphi$
MP	$\vdash_{QCL} \varphi \rightarrow \psi, \vdash_{QCL} \varphi \Rightarrow \vdash_{QCL} \psi$
$\delta Distr$	$\vdash_{QCL} \varphi \rightarrow \psi \Rightarrow \vdash_{QCL} \langle eq(C)\rangle\varphi \rightarrow \langle eq(C)\rangle\psi$

The condition of $P2$ is $C \not\subseteq C'$, for $P4$ it is $C \subseteq C'$, ψ in *Prop* is a propositional tautology; Ax in δAx is any CL-axiom

3. *Let φ be any QCL formula. Then $\vdash_{QCL} \varphi \leftrightarrow \delta(\tau(\varphi))$ and, in particular, $\vdash_{QCL} \varphi$ iff $\vdash_{QCL} \delta(\tau(\varphi))$.*

Proof

1. Ågotnes and Walicki (2006)
2. This is almost immediate from the definition of QCL, and proven with induction of the length $\ell(\varphi)$ of the shortest proof in CL for φ. If $\ell(\varphi) = 1$, then φ must be a CL-axiom, and hence we immediately have $\vdash_{QCL} \delta(\varphi)$, by δAx. Now suppose our lemma is proven for all ψ for which $\ell(\psi) \leq n$ and suppose that the shortest proof for φ is $\ell(\varphi) = n + 1$. Now there are two possibilities:
   (a) φ was obtained from $\vdash_{CL} \psi \to \varphi$ and $\vdash_{CL} \psi$ using *MP*. Both $\psi \to \varphi$ and ψ have a CL proof of at most length n. The induction hypothesis hence yields that we have $\vdash_{QCL} \delta(\psi \to \varphi)$, and $\vdash_{QCL} \delta(\psi)$. Moreover, as a rule in QCL we have $\delta(MP)$, which says that from $\vdash_{QCL} \delta(\psi \to \varphi)$ and $\vdash \delta(\psi)$ we may conclude $\vdash_{QCL} \delta(\varphi)$.
   (b) φ is of the form $\langle\langle C\rangle\rangle\psi_1 \to \langle\langle C\rangle\rangle\psi_2$, with an application of $Distr$ to $\vdash_{CL} \psi_1 \to \psi_2$. This implication then has a proof in CL of at most n, so the induction hypothesis guarantees $\vdash_{QCL} \delta(\psi_1 \to \delta(\psi_2))$. But the δ-translation of $Distr$ gives us $\vdash_{QCL} \delta(\langle\langle C\rangle\rangle\psi_1 \to \langle\langle C\rangle\rangle\psi_2)$, i.e., $\vdash_{QCL} \delta(\varphi)$.
3. For $\varphi = \top$ and $\varphi = p$, we have $\delta(\tau(\varphi)) = \varphi$ so the equivalence is obvious. Disjunction is also straightforward. Let us look at $\langle P\rangle\varphi$, when the equivalence is already proven for φ and $\delta(\tau(\varphi))$. We have $\delta(\tau(\langle P\rangle\varphi)) = \delta(\bigvee_{\{C|C\models_{cp}P\}} \langle\langle C\rangle\rangle \tau(\varphi)) = \bigvee_{\{C|C\models_{cp}P\}}\langle eq(C)\rangle\delta(\tau(\varphi))$. The index set $\{C \mid C \models_{cp} P\}$ is obviously equivalent to $\{C \mid \models_{cp} eq(C) \to P\}$. The latter is, by completeness of $\vdash_{cp}$, equivalent to $\{C \mid \vdash_{cp} eq(C) \to P\}$. So, we have

$$\delta(\tau(\langle P\rangle\varphi)) = \bigvee_{\{C|\vdash_{cp}eq(C)\to P\}} \langle eq(C)\rangle\delta(\tau(\varphi))$$

   Using the induction hypothesis, the right hand side is equivalent to $\bigvee_{\{C|\vdash_{cp}eq(C)\to P\}}\langle eq(C)\rangle\varphi$. The equivalence of the latter with $\langle P\rangle\varphi$ follows immediately from the cql axiom $\delta\langle\rangle$. The case for $[P]\varphi$ is similar. □

Theorem 4 (Completeness and Soundness) *Let φ be an arbitrary QCL-formula. Then: $\vdash_{QCL} \varphi$ iff $\models_{QCL} \varphi$.*

Proof We leave soundness to the reader. For completeness, suppose $\nvdash_{QCL} \varphi$. By Theorem 3.3, we have $\nvdash_{QCL} \delta(\tau(\varphi))$. By Theorem 3.2, we then have $\nvdash_{CL} \tau(\varphi)$. By completeness for the logic CL, we then know that there is a pair $\mathcal{M}, s$ for which $\mathcal{M}, s \models_{CL} \neg\tau(\varphi)$. By Theorem 2 item 1, we then conclude $\mathcal{M}, s \models_{QCL} \neg\varphi$, i.e., $\not\models_{QCL} \varphi$. □

Examples of derivable properties include:

$$\models_{QCL} [P_1]\varphi \to [P_2]\varphi \quad \text{when } \models_{cp} P_1 \to P_2$$
$$\models_{QCL} ([P_1]\varphi \wedge [P_2]\varphi) \to [P_1 \vee P_2]\varphi$$

These illustrate that we not only have primitive modal operators, but also some kind of operations over them, like negation and conjunction. This of course is very reminiscent of Boolean modal logic, where one studies algebraic operations like complement, meet and join on modal operators (Gargov and Passy 1987). We will not pursue the details of the connection here.

4.2 QCL is succinct

Theorem 2 tells us that the advantage of QCL over CL is not its expressivity. Rather, the benefit of QCL is in its *succinctness* of representation. For example, for the QCL formula $\langle any\rangle q$, the translated CL formula $\tau(\langle any\rangle q)$ is exponentially longer, since it has to explicitly enumerate all coalitions in Ag. Is it however *generally* the case that $\tau(\varphi)$ is longer than φ? Since the translation does some computations under $\models_{cp}$, this is in general not the case. For instance, if $P = supseteq(\{a\}) \wedge supseteq(\{c\}) \wedge supseteq(\{b\}) \wedge (subseteq(\{a,b,c\}) \vee subseteq(\{a,b,d\}))$, then $\psi = \langle P\rangle q$ would have as a τ-translation $\langle\langle\{a,b,c\}\rangle\rangle q$, which is shorter than the original QCL-formula ψ. But then again, $\delta(\tau(\psi))$ is a QCL formula that is equivalent to ψ, but that has a size similar to $\tau(\psi)$.

To make this all precise, let us define the length $\ell(\varphi)$ of both QCL and CL formulae φ, as follows:

$$\begin{aligned}
\ell(\top) = \ell(p) &= 1\\
\ell(\varphi_1 \vee \varphi_2) &= \ell(\varphi_1) + \ell(\varphi_2) + 1\\
\ell(\neg\varphi) &= \ell(\varphi) + 1\\
\ell(\langle P\rangle\varphi) = \ell([P]\varphi) &= predsize(P) + \ell(\varphi)\\
\ell(\langle\langle C\rangle\rangle\varphi) &= coalsize(C) + \ell(\varphi)
\end{aligned}$$

with

$$\begin{aligned}
predsize(subseteq(C)) &= coalsize(C) + 1\\
predsize(supseteq(C)) &= coalsize(C) + 1\\
predsize(\neg P) &= predsize(P) + 1\\
predsize(P_1 \vee P_2) &= predsize(P_1) + predsize(P_2) + 1\\
coalsize(C) &= |C|
\end{aligned}$$

Let φ and ψ be X and Y formulae, respectively, where X and Y both range over *CL* and *QCL*. Then we say that they are equivalent with respect to some class of models if they have the same satisfying pairs $\mathcal{M}, s$, that is, for each $\mathcal{M}, s$ in the class of models it is the case that $\mathcal{M}, s \models_X \varphi$ iff $\mathcal{M}, s \models_Y \psi$. This definition naturally extends to sets of formulae.

In the following theorem we show that QCL is *exponentially more succinct* than CL, over general models. This notion of relative succinctness is taken from Lutz (2006), who demonstrates that public announcement logic is more succinct than epistemic logic.

Theorem 5 *When there are at least two agents, there is an infinite sequence of distinct QCL formulae $\varphi_0, \varphi_1, \dots$ of increasing length such that, not only is the CL formula $\tau(\varphi_i)$ equivalent to φ_i for every $i \geq 0$, but every CL formula ψ_i that is equivalent to φ_i has the property $\ell(\psi_i) \geq 2^{\ell(\varphi_i)}$.*

Before the proof, we give some definitions and intermediate results.

For $i \geq 0$, define a sequence of QCL formulae φ_i (where $\top$ denotes an arbitrary instance of a tautology in the coalition predicate language, for example $subseteq(\emptyset) \vee \neg subseteq(\emptyset)$):

- $\varphi_0 = p$
- $\varphi_{i+1} = [\top]\varphi_i$

We will show that every CL formula ψ_i equivalent to φ_i is of length at least 2^i, for all $i \geq 0$.

Note that $[\top]p$ expresses that every coalition can guarantee p.

Let $a \neq b \in Ag$ be two (distinct) agents. Define a sequence of CL formulae:

- $\psi_0 = p$
- $\psi_{i+1} = \langle\langle\emptyset\rangle\rangle\psi_i \wedge \langle\langle a\rangle\rangle\psi_i \wedge \langle\langle b\rangle\rangle\psi_i \wedge \langle\langle a, b\rangle\rangle\psi_i$

Let $Seq^*_{a,b}$ denote all sequences $\vec{s}$ over the set $\{a, b\}$ where $\vec{s} \cdot \vec{t}$ denotes appending sequence $\vec{t}$ to $\vec{s}$. We will also denote such a sequence as $\vec{st}$. Given a CL formula φ, define the set of sequences P_φ as follows, where ϵ denotes the empty sequence:

- $P_q = P_\top = \{\epsilon\}$
- $P_{\neg\varphi} = P_\varphi$
- $P_{\varphi\wedge\psi} = P_\varphi \cup P_\psi$
- $P_{\langle\langle a\rangle\rangle\varphi} = \{\epsilon\} \cup \{\vec{aw} \mid \vec{w} \in P_\varphi\}$
- $P_{\langle\langle b\rangle\rangle\varphi} = \{\epsilon\} \cup \{\vec{bw} \mid \vec{w} \in P_\varphi\}$
- $P_{\langle\langle C\rangle\rangle\varphi} = P_\varphi$ for $C \neq \{a\}$ and $C \neq \{b\}$

Note that for ψ_i, defined above, P_{ψ_i} is the set of all sequences over $\{a, b\}$ of length at most i.

Let us call $\vec{s} \in P_\varphi$ a *P_φ-maximal path* if there is no $\vec{t} \neq \epsilon$ such that $\vec{s} \cdot \vec{t} \in P_\varphi$. Note that if $\vec{s}$ is not P_{ψ_i}-maximal, then both $\vec{sa}$ and $\vec{sb}$ are in P_{ψ_i}.

Let χ be a CL formula that is equivalent to ψ_i, for some $i > 0$. We show the following:

$$P_{\psi_i} \subseteq P_\chi. \tag{1}$$

To show this, assume to the contrary that for some $\vec{x} \in Seq^*_{a,b}$, we have $\vec{x} \in P_{\psi_i}$, but $\vec{x} \notin P_\chi$. Given this $\vec{x}$, define a model $\mathcal{M} = \langle S, \mathcal{E}, \pi\rangle$ as follows:

- $S = P_{\psi_i} \cup \{A, B, Z\}$
- $\pi(s) = p$ for all $s \in S$
- The effectivity function $\mathcal{E}$ is defined as follows. For any $X \subseteq S$, $SC(X)$ is the superset closure (in S) of X: $SC(X) = \{Y \subseteq S \mid X \subseteq Y\}$. First of all, suppose that $\vec{s} \in S$ is not P_{ψ_i}-maximal. Then:

$$\mathcal{E}(C,\vec{s}) = \begin{cases} SC(\{\vec{sa}, \vec{sb}, Z\}) & \text{if } C = \emptyset \\ SC(\{\vec{sa}, Z\}) & \text{if } C = \{a\} \\ SC(\{\vec{sb}, Z\}) & \text{if } C = \{b\} \\ SC(\{\vec{sa}\}) \cup SC(\{\vec{sb}\}) \cup SC(\{Z\}) & \text{if } C = Ag \\ SC(\{Z\}) & \text{otherwise} \end{cases}$$

If $\vec{s}$ itself is already a maximal path, and does not have an extension in S, we define $\mathcal{E}(\emptyset, \vec{s}) = SC(\{A, B, Z\})$, and $\mathcal{E}(\{a\}, \vec{s}) = SC(\{A, Z\})$. Similarly, $\mathcal{E}(\{b\}, \vec{s}) = SC(\{B, Z\})$, $\mathcal{E}(Ag, \vec{s}) = SC(\{A\}) \cup SC(\{B\}) \cup SC(\{Z\})$ and $\mathcal{E}(C, \vec{s}) = SC(\{Z\})$ for any other C. Finally, for each coalition C and every state $y \in \{A, B, Z\}$, define $\mathcal{E}(C, y) = \mathcal{E}(C, \vec{s})$ for some arbitrary maximal path $\vec{s}$.

We also define a second model $\mathcal{M}' = \langle S, \mathcal{E}, \pi' \rangle$ with the same states and effectivity function, where $\pi'(\vec{x}) = \emptyset$ and $\pi'(s) = \pi(s) = \{p\}$ for all $s \neq \vec{x}$.

We argue that both $\mathcal{M}$ and $\mathcal{M}'$ are playability models:

1. $\emptyset \notin \mathcal{E}(C, s)$ for any s and C
2. $S \in \mathcal{E}(C, s)$ for any s and C, by the definition of SC
3. Ag-maximality: let $X \notin \mathcal{E}(Ag, s)$; we must show that $S \setminus X \in \mathcal{E}(\emptyset, s)$. First, consider the case that $s = \vec{s} \in P_{\psi_i}$ and that $\vec{s}$ is not a P_{ψ_i} maximal path. $X \notin \mathcal{E}(Ag, s) = SC(\{\vec{sa}\}) \cup SC(\{\vec{sb}\}) \cup SC(\{Z\})$, so $\vec{sa} \notin X$ and $\vec{sb} \notin X$ and $Z \notin X$. That means that $\vec{sa}, \vec{sb}, Z \in S \setminus X$, so $S \setminus X \in \mathcal{E}(\emptyset, s)$. Second, consider the cases that $s = \vec{s} \in P_{\psi_i}$ and that $\vec{s}$ is a maximal path, or that $s \in \{A, B, Z\}$. $X \notin SC(\{A\}) \cup SC(\{B\}) \cup SC(\{Z\})$; $A \notin X$ and $B \notin X$ and $Z \notin X$; $A, B, Z \in S \setminus X$; $S \setminus X \in \mathcal{E}(\emptyset, s) = SC(\{A, B, Z\})$.
4. Outcome-monotonicity: immediate from the definition.
5. Superadditivity: let $X_1 \in \mathcal{E}(C_1, s)$ and $X_2 \in \mathcal{E}(C_2, s)$. First consider the case that $s = \vec{s} \in P_{\psi_i}$ and $\vec{s}$ is not maximal:
   - $C_1 = C_2 = \emptyset$: $\vec{sa}, \vec{sb}, Z \in X_1$ and $\vec{sa}, \vec{sb}, Z \in X_2$, so $\vec{sa}, \vec{sb}, Z \in X_1 \cap X_2$ and thus $X_1 \cap X_2 \in \mathcal{E}(\emptyset, s)$.
   - $C_1 = \emptyset, C_2 = \{a\}$: $\vec{sa}, \vec{sb}, Z \in X_1$ and $\vec{sa}, Z \in X_2$, and thus $\vec{sa}, Z \in X_1 \cap X_2$, and $X_1 \cap X_2 \in \mathcal{E}(\{a\}, s)$.
   - $C_1 = \emptyset, C_2 = Ag$: $\vec{sa}, \vec{sb}, Z \in X_1$, and either $\vec{sa} \in X_2$ or $\vec{sb} \in X_2$ or $Z \in X_2$. Wlog. assume the former first. Then $\vec{sa} \in X_1 \cap X_2$, and thus $X_1 \cap X_2 \in \mathcal{E}(Ag, s)$.
   - $C_1 = \{a\}, C_2 = \{b\}$: $\vec{sa}, Z \in X_1$, $\vec{sb}, Z \in X_2$. $Z \in X_1 \cap X_2$, so $X_1 \cap X_2 \in SC(\{Z\})$ and thus $X_1 \cap X_2 \in \mathcal{E}(\{a, b\}, s)$.
   - $C_1 \neq \emptyset, \{a\}, \{b\}, Ag$: $Z \in X_1$. Since C_1 and C_2 are assumed to be disjunct, $C_2 \neq Ag$, and observe that then $Z \in X_2$ as well. Thus, $X \in X_1 \cap X_2$. It is either the case that (i) $C_1 \cup C_2 = Ag$ or (ii) $(C_1 \cup C_2) \neq \emptyset, \{a\}, \{b\}, Ag$. In either case $Z \in X_1 \cap X_2$ ensures that $X_1 \cap X_2 \in \mathcal{E}(C_1 \cup C_2, s)$.
   - Other cases: symmetric.

Second, the cases when $s = \vec{s} \in P_{\psi_i}$ and $\vec{s}$ is maximal and when $s \in \{A, B, Z\}$ follow when $\vec{sa}$ is replaced by A and $\vec{sb}$ is replaced by B in the argument above.

Lemma 1 $\mathcal{M}, \epsilon \models \psi_i$ *and* $\mathcal{M}', \epsilon \not\models \psi_i$.

Proof In order to see that $\mathcal{M}, \epsilon \models \psi_i$ holds, observe that $(p)^{\mathcal{M}} = S$. It follows that $(\langle\!\langle C \rangle\!\rangle p)^{\mathcal{M}} = S$ for any C, and thus that $(\langle\!\langle C' \rangle\!\rangle \langle\!\langle C \rangle\!\rangle p)^{\mathcal{M}} = S$ for any C', and so on.

To show that $\mathcal{M}', \epsilon \not\models \psi_i$, note that $\vec{x}$ must be of the form

$$\vec{x} = x_1 \cdot x_2 \cdot \cdots \cdot x_k$$

with $x_i \in \{a, b\}$ and $k \leq i$. It is easy to see that $\models \psi_i \rightarrow \langle\!\langle x_1 \rangle\!\rangle \langle\!\langle x_2 \rangle\!\rangle \cdots \langle\!\langle x_k \rangle\!\rangle p$. Assume, towards a contradiction, that $\mathcal{M}', \epsilon \models \langle\!\langle x_1 \rangle\!\rangle \langle\!\langle x_2 \rangle\!\rangle \cdots \langle\!\langle x_k \rangle\!\rangle p$. In other words, $(\langle\!\langle x_2 \rangle\!\rangle \cdots \langle\!\langle x_k \rangle\!\rangle p)^{\mathcal{M}'} \in \mathcal{E}(x_1, \epsilon)$. Inspecting the definition of $\mathcal{E}$ for $C = \{a\}$ and $C = \{b\}$, we see that this means that $(\langle\!\langle x_2 \rangle\!\rangle \cdots \langle\!\langle x_k \rangle\!\rangle p)^{\mathcal{M}'} \in SC(\{x_1, Z\})$, which in particular implies that $\mathcal{M}', x_1 \models \langle\!\langle x_2 \rangle\!\rangle \cdots \langle\!\langle x_k \rangle\!\rangle p$. Repeating this argument ($k - 1$ more times), we see that this implies that $\mathcal{M}', x_1 x_2 \cdots x_k \models p$. But this is not the case, because $p \notin \pi'(x_1 \cdots x_k)$, hence the contradiction. □

Lemma 2

$$\mathcal{M}, \epsilon \models \chi \textit{ iff } \mathcal{M}', \epsilon \models \chi$$

Proof First, we show that for any $s \in \{A, B, Z\}$, we have that

$$\mathcal{M}, s \models \varphi \Leftrightarrow \mathcal{M}', s \models \varphi \tag{2}$$

for any φ by structural induction. The case $\varphi = q$, $q \in \Phi_0$, is immediate, because $\pi(s) = \pi'(s)$ for $s \in \{A, B, Z\}$. Consider $\varphi = \langle\!\langle C \rangle\!\rangle \gamma$. First, let $C = \emptyset$. For the direction to the right, let $\mathcal{M}, s \models \varphi$, i.e., $(\gamma)^{\mathcal{M}} \in \mathcal{E}(C, s)$. $\mathcal{E}(C, s) = SC(\{A, B, Z\})$, so it follows that for any $y \in \{A, B, Z\}$, $\mathcal{M}, y \models \gamma$. By the induction hypothesis, $\mathcal{M}', y \models \gamma$, and thus $\{A, B, Z\} \subseteq (\gamma)^{\mathcal{M}'}$ and we have that $(\gamma)^{\mathcal{M}'} \in \mathcal{E}(C, s)$. The direction to the left is similar. For $C = Ag$ and the direction to the right, let $(\gamma)^{\mathcal{M}} \in \mathcal{E}(Ag, s) = SC(\{A\}) \cup SC(\{B\}) \cup SC(\{Z\})$. Thus, $\mathcal{M}, y \models \gamma$ for at least one $y \in \{A, B, Z\}$. By the induction hypothesis, $\mathcal{M}', y \models \gamma$ for that $y \in \{A, B, Z\}$, and it follows that $(\gamma)^{\mathcal{M}'} \in SC(\{y\})$ and thus that $(\gamma)^{\mathcal{M}'} \in \mathcal{E}(Ag, s)$. The direction to the left is similar. The arguments for other values of C are similar to $C = \emptyset$. The Booleans are straightforward. This completes the proof of (2).

We now show that for all $\vec{s} \in P_{\psi_i}$ and $\varphi \in \mathsf{sub}(\chi)$, such that

$$\{s\vec{w} \mid \vec{w} \in P_{\varphi}\} \subseteq P_{\chi}, \tag{3}$$

we have

$$\mathcal{M}, \vec{s} \models \varphi \text{ iff } \mathcal{M}', \vec{s} \models \varphi,$$

by induction over φ. The lemma follows immediately.

Consider the case $\varphi = q, q \in \varphi_0$. If $q \neq p$, then both $\mathcal{M}, \vec{s} \not\models q$ and $\mathcal{M}', \vec{s} \not\models q$. Consider the case that $q = p$. Assume that $\mathcal{M}', \vec{s} \not\models p$. By construction of $\mathcal{M}'$, the only possibility is that $\vec{s} = \vec{x}$. Because $\vec{s} \in P_{\psi_i}$ and $\epsilon \in P_p$, we have that $\vec{s} \in P_{\chi}$ by (3). But this contradicts the assumption that $\mathcal{M}', \vec{s} \not\models p$, because by construction we have that $\vec{x} \notin P_{\chi}$. Thus, $\mathcal{M}', \vec{s} \models p$ for any $\vec{s}$, and we also have that $\mathcal{M}, \vec{s} \models p$ for any $\vec{s}$.

Consider the case $\varphi = \langle\!\langle C \rangle\!\rangle \gamma$. If $\vec{s}$ is maximal, then $\mathcal{M}, \vec{s} \models \varphi$ iff $(\gamma)^{\mathcal{M}} \in \mathcal{E}(C, \vec{s}) = \mathcal{E}(C, y)$ for arbitrary $y \in \{A, B, Z\}$ iff $\mathcal{M}, y \models \varphi$ iff, by (2), $\mathcal{M}', y \models \varphi$ iff $\mathcal{M}', \vec{s} \models \varphi$. Thus, assume that $\vec{s}$ is not maximal. For the direction to the right, let $\mathcal{M}, \vec{s} \models \varphi$, i.e., $(\gamma)^{\mathcal{M}} \in \mathcal{E}(C, \vec{s})$. Assume first that $C = \{a\}$. $(\gamma)^{\mathcal{M}} \in SC(\{\vec{sa}, Z\})$, which implies that $\mathcal{M}, \vec{sa} \models \gamma$ and $\mathcal{M}, Z \models \gamma$. We have that $\{\vec{sw} \mid w \in P_\varphi\} \subseteq P_\chi$ and because $P_\varphi = \{\epsilon\} \cup \{\vec{aw'} \mid w' \in P_\gamma\}$, it follows that $\{\vec{saw'} \mid w' \in P_\gamma\} \subseteq P_\chi$. Thus we can use the induction hypothesis to conclude that $\mathcal{M}', \vec{sa} \models \gamma$. From $\mathcal{M}, Z \models \gamma$ it follows that $\mathcal{M}', Z \models \gamma$ by (2), so $(\gamma)^{\mathcal{M}'} \in SC(\{\vec{sa}, Z\}) = \mathcal{E}(C, \vec{s})$, and $\mathcal{M}', \vec{s} \models \varphi$. For the case $C = \{b\}$, swap a and b in the above argument. The cases $C = \emptyset$ and $C \neq \emptyset, \{a\}, \{b\}, Ag$ can also be shown in the same way. For the case $C = Ag$, we have that $(\gamma)^{\mathcal{M}} \in SC(\{\vec{sa}\}) \cup SC(\{\vec{sb}\}) \cup SC(\{Z\})$. It follows that either $\mathcal{M}, \vec{sa} \models \gamma$, $\mathcal{M}, \vec{sb} \models \gamma$ or $\mathcal{M}, Z \models \gamma$. Assume wlog. the first. By the same argument as above, it follows by the induction hypothesis that $\mathcal{M}', \vec{sa} \models \gamma$, and thus that $(\gamma)^{\mathcal{M}'} \in SC(\{\vec{sa}\}) \subseteq \mathcal{E}(C, \vec{s})$, which means that $\mathcal{M}', \vec{s} \models \varphi$. The Booleans are straightforward. The direction to the left is completely symmetrical. □

Lemma 2 contradicts the fact that χ and ψ_i are equivalent. Thus, (1) holds. We can now prove the theorem.

Proof of Theorem 5 It now only suffices to show that $\ell(\chi) \geq 2^i$ for any i. First, observe that for any i, and for any sequence $\vec{x} = x_1 \cdots x_k$ with $k \leq i$ and $x_i \in \{a, b\}$, we have that $\vec{x} \in P_{\psi_i}$. It follows immediately that $\ell(P_{\psi_i}) \geq 2^i$. From (1) it follows that $\ell(P_\chi) \geq 2^i$. For any formula φ it is the case that $\ell(\varphi) \geq |P_\varphi|$, and it thus follows that $\ell(\chi) \geq 2^i$. □

It is easy to see that the converse of Theorem 5 does *not* hold; it is not the case that CL is exponentially more succinct (in the sense of Theorem 5) than QCL. It is true that some CL formulae are strictly shorter than equivalent QCL formulae, but only by a constant factor. To see that the converse of the Theorem does not hold, take the translation $\delta(\varphi)$ of a CL formula φ. The former is effectively obtained by replacing every coalition C in φ with $subseteq(C) \wedge supseteq(C)$, increasing the length by $|C|+3$ for each C occurring in φ. Thus, there is a constant k such that $\ell(\delta(\varphi)) \leq k\ell(\varphi)$ (it suffices to take $k = 5$: an upper bound on the length of $\delta(\varphi)$ is found by assuming that every symbol in φ is a coalition symbol $C = \{a\}$).

5 Model checking and satisfiability

When one is interested in the computational properties of a logic, there are two fundamentally important questions of interest. The first relates to the complexity of *model checking*; and the second relates to the complexity of *satisfiability*.

5.1 Complexity of model checking

Model checking is currently regarded as perhaps the most important computational problem associated with any temporal/modal logic, as model checking approaches for such logics have had a substantial degree of success in industry (Clarke et al. 2000). The *explicit state model checking problem* for QCL is as follows:

> Given a model $\mathcal{M}$, state s in $\mathcal{M}$, and formula φ of QCL, is it the case that $\mathcal{M}, s \models_{QCL} \varphi$?

Notice that in this version of the problem, we assume that the components of the model $\mathcal{M}$ are *explicitly enumerated* in the input. It is known that the corresponding problem for Coalition Logic may be solved in polynomial time $O(|\mathcal{M}| \cdot |\varphi|)$ (Pauly 2001, p. 50; as may the explicit state ATL model checking problem, Alur et al. 2002). Perhaps surprisingly, the QCL model checking problem is no worse:

Theorem 6 *The explicit state model checking problem for QCL may be solved in polynomial time.*

Proof (Summary.) The obvious source of difficulty is with cooperation operators $[P]$ and $\langle P \rangle$. So consider $[P]\varphi$, as the $\langle P \rangle$ case is similar. We need to check whether $\forall C \subseteq Ag : C \models_{cp} P$ implies $\exists S \in \mathcal{E}(C, s)$ such that $\forall s' \in S$, we have $\mathcal{M}, s' \models_{QCL} \varphi$. Now consider the naive approach: exhaustively checking each $C \subseteq Ag$. We have running time $O(2^{Ag})$. But consider the representation of $\mathcal{E}$: in the input we must list the value of $\mathcal{E}$ for all $C \subseteq Ag$ and $s \in S$. The size of this representation is $O(|S| \cdot 2^{Ag})$, and so running time is polynomial in the size of the input as required. □

Of course, this result is not terribly useful, since it assumes a representation of $\mathcal{M}$ that is not feasible, since it is exponentially large in the number of agents and Boolean variables in the system. Implemented model checkers use *succinct* languages for defining models; for example, the reactive modules language (RML) of Alur and Henzinger (1999). Assuming an RML representation, Coalition Logic model checking is PSPACE-complete (van der Hoek et al. 2005b), and thus no easier than theorem proving in the same logic (Pauly 2001, p. 60). It is therefore more meaningful to ask what the model checking complexity of QCL is for such a representation. We only give a very brief summary of RML—space restrictions prevent a complete description; see Alur and Henzinger (1999), van der Hoek et al. (2005b) for details.

In reactive modules, a system is specified as a collection of *modules*, which correspond to agents. Here is a (somewhat simplified) example of an RML module:

$$
\begin{array}{ll}
 & \texttt{module}\ \mathit{toggle}\ \texttt{controls}\ x \\
\texttt{init} & [\,]\top \rightsquigarrow x' := \top \\
 & [\,]\top \rightsquigarrow x' := \bot \\
\texttt{update} & [\,]x \rightsquigarrow x' := \bot \\
 & [\,](\neg x) \rightsquigarrow x' := \top
\end{array}
$$

This agent *toggle*, controls a single Boolean variable, x. The *choices* available to the agent at any given time are defined by the `init` and `update` rules. The `init` rules define the choices available to the agent with respect to the initialisation of its variables, while the `update` rules define the agent's choices subsequently. The `init` rules define two choices for the initialisation of this variable: assign it the value $\top$ or the value $\bot$. Both of these rules can fire initially, as their conditions ($\top$) are always satisfied; in fact, only one of the available rules will ever *actually* fire, corresponding to the “choice made” by the agent on that decision round. With respect to `update` rules, the first rule says that if x has the value $\top$, then the corresponding choice is to assign it the value $\bot$, while the second rule ‘does the opposite’. In other words, the agent non-deterministically chooses a value for x initially, and then on subsequent rounds toggles this value. In summary, the actions available to an agent in any given state correspond to the rules whose l.h.s. fire against the current state of the system; the agent will actually select only one of these actions to perform, which results in it updating the variables it controls.

Theorem 7 *The model checking problem for QCL assuming an RML representation for models is PSPACE-complete.*

Proof PSPACE-hardness follows from the fact that QCL subsumes Coalition Logic, for which the corresponding problem is PSPACE-hard (van der Hoek et al. 2005b). For membership of PSPACE, the algorithm of van der Hoek et al. (2005b) can be easily adapted: when handling the $[P]$ and $\langle P\rangle$ operators, we can simply loop through each coalition in turn, then applying the relevant part of the algorithm from van der Hoek et al. (2005b). Such a loop can trivially be implemented in PSPACE. □

This result, we believe, is potentially much more interesting than that for explicit state model checking, since it tells us that QCL model checking is no more complex than Coalition Logic *even for a realistic representation of models*.

5.2 Complexity of satisfiability

While the model checking problem addresses itself to the question of the truth of a formula φ for a *specific* interpretation $\mathcal{M}, s$, the *satisfiability* problem asks whether *there exists* an interpretation $\mathcal{M}, s$ that satisfies φ. The presence of the existential quantifier here suggests that satisfiability is going to be harder than model checking, and in general, it is (at least with respect to *explicit state model checking*). For CL, we know that the complexity of satisfiability varies from NP-complete in the simplest case (Pauly 2001, p. 62) up to PSPACE-complete in the case of weak playability models (as used in the present paper; Pauly 2001, p. 66). In this section, we show that, despite its comparative succinctness, the satisfiability problem for QCL is no harder than that of CL:

Theorem 8 *The satisfiability problem for QCL (parameterised by the coalition predicate language with* $subseteq$, $supseteq$ *as primitives) is PSPACE-complete.*

Proof Since QCL subsumes Pauly’s CL, we immediately have a PSPACE lower bound: it only remains to prove the upper bound, i.e., that the problem is in PSPACE. The proof makes use of another language: another quantified variant of CL, which we will refer to as CLQ. As we will see, the relationship of CLQ to CL is roughly that of Quantified Boolean Formulae (QBF) to conventional propositional logic. Formally, for the syntax of this language, we assume a finite stock of *coalition variables*, $CV = \{c_1, \ldots, c_k\}$, and the usual quantifiers $\{\forall, \exists\}$. Let CL(CV) be the language of

Pauly's Coalition Logic in which coalition variables are allowed to be used instead of explicitly listed sets of agents. For example, $\langle\langle c\rangle\rangle p \vee q$, where $c \in CV$, will be a formula of CL (CV), with the intended meaning that the coalition denoted by c can achieve $p \vee q$. □

The syntax of CLQ is then defined by the following grammar:

$$\varphi ::= \psi \mid \neg\varphi \mid \varphi \vee \varphi \mid Qc : P\ \psi$$

where $Q \in \{\forall, \exists\}$, $c \in CV$ is a coalition variable, P is a coalition predicate, and ψ is a formula of CL(CV).

For example, the following is a formula of CLQ:

$$\exists c_1 : \mathit{supseteq}(\{1, 2\})\ \langle\langle c_1\rangle\rangle p \wedge \forall c_2 : \mathit{subseteq}(\{1, 2, 3\})\ \langle\langle c_2\rangle\rangle q \tag{4}$$

The semantics of CLQ are straightforward, and we will not present it here. (We essentially add the obvious rules to deal with quantifiers to the rules for cl.) Satisfiability for CLQ is defined in the obvious way. We assume the conventional definitions of closed formulae, and free and bound variables. Now, the following fact is immediately evident:

> Every formula φ of QCL can be translated to a formula $\theta(\varphi)$ of CLQ such that φ is satisfiable in CL iff $\theta(\varphi)$ is satisfiable in CLQ; moreover, the translation $\theta(\cdot)$ can be carried out in polynomial time (and hence the size of $\theta(\varphi)$ is polynomial in the size of φ).

Formally, the translation function θ is defined as follows:

$$\theta(\varphi) = \begin{cases} \varphi & \text{if } \varphi \text{ contains no modalities} \\ \exists c_i : P\ \langle\langle c_i\rangle\rangle\theta(\psi) & \text{if } \varphi = \langle P\rangle\psi\ (c_i \text{ is a new variable}) \\ \forall c_i : P\ \langle\langle c_i\rangle\rangle\theta(\psi) & \text{if } \varphi = [P]\psi\ (c_i \text{ is a new variable}) \\ \neg\theta(\psi) & \text{if } \varphi = \neg\psi \\ \theta(\psi_1) \vee \theta(\psi_2) & \text{if } \varphi = \psi_1 \vee \psi_2 \end{cases}$$

Thus, for example, the following QCL formula translates to the CLQ formula (4), above.

$$\langle \mathit{supseteq}(\{1, 2\})\rangle p \wedge [\mathit{subseteq}(\{1, 2, 3\})]q \tag{5}$$

The correctness of the translation θ is immediate from the semantics of QCL, CL, and CLQ.

Next, we say that a CLQ formula is in *prenex normal form* if it has the following syntactic structure:

$$\mathcal{Q}_1 \mathcal{Q}_2 \cdots \mathcal{Q}_l\ \varphi$$

where each $\mathcal{Q}_i$ is a quantifier expression (e.g., "$\exists c_2 : \mathit{subseteq}(\{1, 2, 3\})$") such that coalition variables in each $\mathcal{Q}_i$ are distinct, and φ is a formula of CL(CV) such that φ

contains no free variables (i.e., every coalition variable occurring in φ is bound to a quantifier). Now, we have the following:

> Every CLQ formula φ in the range of θ can be translated into a prenex normal form formula $\rho(\varphi)$ of CLQ such that φ is satisfiable in CLQ iff $\rho(\varphi)$ is satisfiable in CLQ; moreover, the translation $\rho(\cdot)$ can be carried out in polynomial time (and hence the size of $\rho(\varphi)$ is polynomial in the size of φ).

Of course, this statement is not *in general* true for quantified modal logics, but it holds for formulae in the range of θ, which is enough for our purposes. The translation into prenex normal form makes use of the usual translation rules for first-order formula (Gallier 1987, pp. 205–209). For example, we have the following CLQ equivalences:

$$\neg\exists c : P \ \langle\!\langle c \rangle\!\rangle\varphi \equiv \forall c : P \ \neg\langle\!\langle c \rangle\!\rangle\varphi$$
$$\neg\forall c : P \ \langle\!\langle c \rangle\!\rangle\varphi \equiv \exists c : P \ \neg\langle\!\langle c \rangle\!\rangle\varphi$$

Notice that this kind of translation into prenex normal form is not possible in QCL, since each quantifier in a QCL operator is inextricably tied to an ability operator. This is in fact why QCL does not have a prenex normal form, and this in turn explains why we introduce QCL.

The final step is to prove the following claim:

Claim 1 *The satisfiability problem for CLQ formulae in prenex normal form is in PSPACE.*

We sketch the design of a recursive Turing machine T that decides the problem in polynomial space. Formally, T takes as input a prenex normal form CLQ formula φ together with a partial *interpretation* for coalition variables, $I\colon CV \to 2^{Ag}$. The machine uses as a subroutine Pauly's PSPACE decision procedure $sat(\cdot)$ for Coalition Logic (Pauly 2001, p. 65). Initially, the machine is called with I being empty, i.e., no variables are interpreted. On input $\langle\varphi, I\rangle$, the machine behaves as follows:

1. If φ contains no quantifiers, then for every variable $c \in \mathrm{dom}\, I$, systematically substitute $I(c)$ for c in φ. After this process is complete, we will be left with a formula φ^* of CL; invoke Pauly's decision procedure $sat(\cdot)$ on φ^*, and "accept" if it is satisfiable, otherwise "reject".
2. If φ is of the form $\exists c : P \ \psi$, then for each $C \subseteq Ag$, if $C \models_{cp} P$ then invoke T with input $\langle\psi, I \cup \{c \mapsto C\}\rangle$; if any such call accepts, then "accept". If we complete the loop without such success, then "reject".
3. If φ is of the form $\forall c : P \ \psi$, then for each $C \subseteq Ag$, if $C \models_{cp} P$ then invoke T with input $\langle\psi, I \cup \{c \mapsto C\}\rangle$; if any such call rejects, then "reject". If we complete the loop without such failures, then "accept".

Correctness of the approach is immediate from construction. The algorithm clearly terminates, and operates in time $\mathrm{PSPACE}^{PSPACE} = \mathrm{PSPACE}$.

In sum, our PSPACE decision procedure for QCL is as follows: given a formula φ, invoke the Turing machine T with input $\langle\rho(\theta(\varphi)), \emptyset\rangle$. If the machine accepts on this input, then φ is satisfiable, otherwise it is unsatisfiable. $\square$

6 Coalition size

As we noted earlier, an obvious omission from our language of coalition predicates is designated predicates for expressing cardinality properties of coalitions. In this section, we explore extensions to the framework for this purpose. The obvious approach is to introduce primitive coalition predicates $geq(n)$, where $n \in \mathbb{N}$, with semantics as follows:

$$C \models geq(n) \text{ iff } |C| \geq n$$

Given this predicate, we can define several obvious derived predicates (see also Ågotnes et al. 2006 for a discussion of a similar language).

$$\begin{aligned} gt(n) &\equiv geq(n+1) \\ lt(n) &\equiv \neg geq(n) \\ leq(n) &\equiv lt(n+1) \\ maj(n) &\equiv geq(\lceil (n+1)/2 \rceil) \\ ceq(n) &\equiv (geq(n) \wedge leq(n)) \end{aligned}$$

The first natural question is whether $geq(n)$ is definable in QCL. Indeed it is:

$$geq(n) \equiv \bigvee_{C \subseteq Ag, |C| \geq n} supseteq(C) \tag{6}$$

However, we again see that such a definition leads to exponentially large formulae, which justifies extending the predicate language of QCL with an *atomic* coalition predicate $geq(n)$ for every $n \in \mathbb{N}$. Call the resulting logic QCL($\geq$), and let $\models_{cp\geq}$ and $\models_{QCL(\geq)}$ denote the satisfiability relations for QCL($\geq$) predicates and QCL($\geq$) formulae, respectively. Once again, the gain is not expressiveness but *succinctness*. As another example of the added succinctness, consider the CL formula $\langle\langle C \rangle\rangle p$. In QCL this cannot in general be written by any less complex formula than $\langle subseteq(C) \wedge supseteq(C) \rangle p$, but in QCL($\geq$) it can be simplified somewhat to $\langle supseteq(C) \wedge \neg geq(|C|+1) \rangle p$ (which in general is simpler since one of the enumerations of the agents in C is replaced by a number).

A subtle but important issue when reasoning with the logic is the way in which the natural number argument of the $geq(\ldots)$ predicate is *represented*. Suppose, (following standard practice in complexity theory), that we represent the argument in binary. Now, we ask whether a given coalition predicate P is satisfiable, where P contains a constraint $geq(n)$. Now checking the satisfiability of such constraints is not obviously in NP. The problem is that the witness C to the satisfiability of P is *exponentially larger than the constraint* $geq(n)$. Of course, if we express the natural number n in *unary*, then this is not an issue. But unary is not a realistic or practical representation for numbers. It turns out, however, that we do in fact get NP completeness for the satisfiability problem also for QCL($\geq$), although the argument requires some more work. The reason is that we can use an efficient encoding of the witness C. This was shown by Ågotnes et al. (2006) for a similar problem (cf. Sect. 7).

Let $Ag(P)$ and $subp(P)$ denote the set of agents, and the set of sub-predicates, respectively, occurring in a predicate P.

Lemma 3 *Any satisfiable QCL(≥) predicate P is satisfied by a coalition of no more than $1 + max_P$ agents, where $1 + max_P$ equals*

$$max(\{|Ag(P)|, max(\{geq(n) : geq(n) \in subp(P)\})\})$$

Proof Suppose P is a QCL(≥) predicate and $C \models_{cp\geq} P$. If $|C| \leq max_P$ we are done, so assume that $|C| > max_P$. In addition to the constraints C_1–C_4 in the proof of Theorem 1, we get two additional types of constraints corresponding to $geq(n)$ and $\neg geq(n)$ expressions, respectively: there exist a sequence of numbers

$$n_{5_1}, \ldots, n_{5_{k_5}}, n_{6_1}, \ldots, n_{6_{k_6}}$$

such that

$$|C| \geq n_{5_1}, \ \ldots, \ |C| \geq n_{5_{k_5}}$$
$$|C| \not\geq n_{6_1}, \ \ldots, \ |C| \not\geq n_{6_{k_6}}$$

In other words:

$$|C| \geq max(\{n_{5_i} : 1 \leq i \leq k_5\})$$
$$|C| \leq min(\{n_{6_i} : 1 \leq i \leq k_6\})$$

Let's call the two constraints n_5 and n_6 constraints. We proceed by the same two cases as in the proof of Theorem 1. In case 1 (there are C_1 constraints) we are done, so assume case 2 (no C_1 constraints). If $Ag(P) = Ag$ we are done ($|C| \leq Ag(P)$), so assume that there is an agent $a^* \in Ag$ such that $a^* \notin Ag(P)$. Let C' be a coalition satisfying the following: (i) $C' \subseteq C$, (ii) $C_2 \subseteq C'$ and (iii) $|C'| = max_P$. It is easy to see that such a coalition exists. Finally, let $C^* = C' \cup \{a^*\}$. It is easy to see that C^* satisfies the C_2, C_3 and C_4 constraints. By point (iii) in the construction it also satisfies the n_5 constraint. Finally, since we assumed that $|C| > max_P$ we get that $|C| \geq |C^*|$, and since we have that the n_6 constraints holds for C it thus follows that it also holds for C^*. Hence, all the constraints hold for C^*, so $C^* \models_{cp\geq} P$—and $|C^*| = 1 + max_P$. □

Theorem 9 *The satisfiability problem for QCL(≥) coalition predicates is NP-complete.*

Proof It suffices to show that a satisfiable predicate has a "short certificate"; see the proof of Theorem 1. Let P be a QCLC coalition predicate. Suppose $C \models_{cp} P$. It might be the case that C contains agents which are not mentioned in P, i.e., that $X(C, P) = C \setminus Ag(P)$ is non-empty. By inspecting the definition of the satisfiability relation, we see that the satisfaction of P by C does not depend on the actual agent names in $X(C, P)$ in the following sense: for any agent a in $X(C, P)$, we can replace a with any other agent $b \notin (C \cup Ag(P))$, and retain satisfaction of P.

Of course, satisfaction of P might depend on the *size* of $X(C, P)$. More precisely: the *only* information needed about $X(C, P)$ in order to determine that C satisfies P, is the *size* of $X(C, P)$—the actual names of the agents in $X(C, P)$ are not needed in order to interpret P. This makes it possible to encode the satisfying coalition C more efficiently than just enumerating the agents it contains: we only need to list the agents $C \cap Ag(P)$, and give the size of $X(C, P)$. The latter can be encoded in binary. By Lemma 3 a satisfiable predicate is satisfied by a coalition with $X(C, P)$ no greater than $max(\{geq(n) : geq(n) \in subp(P)\})$. Thus, to verify that a QCLC predicate P, with any occurrences of numbers n in $geq(n)$ encoded in binary, is satisfiable, we can (i) guess a coalition C such that $|C| \leq max_P + 1$ encoded in such a way that only agents in $Ag(P)$ are listed explicitly and the remainder of the coalition is only represented by a number m encoded in binary; and (ii) verify that $C \models_{cp} P$. It is easy to see that the size of the encoding of the model is polynomial in the size of the encoding of the predicate, and that (ii) can be done in time polynomial in the size of C and P. □

It is straightforward to lift the translation τ from QCL to CL to the case when also the additional predicates of QCL(≥) are allowed, and it is easy to see that Theorem 2 holds also for QCL(≥) formulae. For axiomatisation, we only need to add axioms for the $geq(n)$ predicates to the predicate calculus. That can be achieved simply by adding (6) as an axiom schema. A more “direct” axiomatisation of $geq(n)$ is shown in Table 3, taken from Ågotnes et al. (2006). Let $\vdash_{cp\geq}$ denote derivability in the QCL predicate calculus (from Table 2) extended with the axioms in Table 3. The following is easily obtained from a similar result in Ågotnes et al. (2006):

Lemma 4 *The QCL(≥) predicate calculus is sound and complete: for any QCL(≥) predicate P, $\models_{cp\geq} P \Leftrightarrow \vdash_{cp\geq} P$.*

Let $\vdash_{QCL(\geq)}$ denote derivability in the system obtained by replacing $\vdash_{cp}$ with $\vdash_{cp\geq}$ in the definition of $\vdash_{QCL}$ (Table 2).

Theorem 10 (Completeness and Soundness) *Let φ be a QCL(≥)-formula. Then: $\vdash_{QCL(\geq)} \varphi$ iff $\models_{QCL(\geq)} \varphi$.*

6.1 An example

To illustrate the use of QCL(≥) for reasoning about multi-agent systems, consider the expression of *majority voting*:

> An electorate of n voters wishes to select one of two outcomes ω_1 and ω_2. They want to use a simple majority voting protocol, so that outcome ω_i will be selected iff a majority of the n voters state a preference for it. No coalition of less than

Table 3 Extra predicate calculus axioms for QCL(≥)

$(MIN0)$	$\vdash_{cp} geq(0)$
$(MIN1)$	$\vdash_{cp} geq(n) \rightarrow geq(m) \;\; (m < n)$
$(MIN2)$	$\vdash_{cp} \;\; supseteq(\{a_1\}) \wedge \cdots \wedge supseteq(\{a_k\}) \rightarrow geq(k) \; \forall i \neq j \; a_i \neq a_j$

majority size should be able to select an outcome, and *any* majority should be able to choose the outcome (i.e., the selection procedure is not influenced by the "names" of the agents in a coalition). One outcome must be selected, but both outcomes should not be selected simultaneously.

We express these requirements as follows. First: *any majority should be able to select an outcome:*

$$([maj(n)]\omega_1) \wedge ([maj(n)]\omega_2)$$

No coalition that is not a majority can select an outcome:

$$(\neg\langle\neg maj(n)\rangle\omega_1) \wedge (\neg\langle\neg maj(n)\rangle\omega_2)$$

Either outcome ω_1 or ω_2 must result:

$$\langle any\rangle(\omega_1 \vee \omega_2)$$

Both outcomes cannot be selected simultaneously:

$$\langle any\rangle\neg(\omega_1 \wedge \omega_2)$$

Notice that majority voting cannot be succinctly specified using regular Coalition Logic.

7 Related work and conclusions

Quantified Coalition Logic (QCL) adds a limited but useful form of quantification to Coalition Logic. The computational problems of model checking and satisfiability for QCL are no worse (although of course no better) than the corresponding problems for Coalition Logic. The motivation for our work is succinctness rather than expressiveness: while QCL is exactly as expressive as Coalition Logic, it is exponentially more succinct.

While first-order temporal logics have been studied in the literature, and CL can be seen as the next-time fragment of ATL which again is a generalisation of the branching-time temporal logic Computational Tree Logic (CTL), we are not aware of any other work on quantification in CL or ATL. Lately, there has been some work on generalising the coalition modalities in another direction: to explicitly include actions and strategies (van der Hoek et al. 2005a; Ågotnes 2006).

Languages similar to the coalition predicate languages of QCL and QCL($\geq$) are discussed in Ågotnes and Walicki (2006), Ågotnes et al. (2006) in the context of epistemic logic, where the arguments to the predicates are sets of formulae rather than sets of agents. The predicate (corresponding to) $supseteq(X)$ means that the reasoner knows at least the formulae in X; the predicate $subseteq(X)$ means that he knows at most X; the predicate $geq(m)$ that he knows at least m formulae, etc. A crucial difference is that the set of all possible formulae that can be known is assumed to be *infinite*, as opposed to the assumption that the set of all possible agents is assumed to be finite in QCL and QCL($\geq$). Dropping the assumption of a finite bound on the number of agents in coalition logic might be interesting. In that case, we can no longer quantify over all possible formulae/agents by using conjunction or disjunction, and many of the definability results mentioned in this paper do not hold. For example, as discussed in the mentioned works, neither $subseteq(\cdot)$ nor $geq(\cdot)$ would be definable by $supseteq(\cdot)$, but the two operators $supseteq(\cdot)$ and $geq(\cdot)$ form an adequate set—they can express all the predicates discussed in this paper (even when there is no bound on the number of agents).

Further opportunities for future work include a more detailed understanding of the relationship between QCL and Boolean modal logic.

References

Ågotnes, T. (2006). Action and knowledge in alternating-time temporal logic. *Synthese, 149*(2), 377–409.

Ågotnes, T., van der Hoek, W., & Wooldridge, M. (2006). On the logic of coalitional games. In *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2006)*, Hakodate, Japan.

Ågotnes, T., & Walicki, M. (2006). Complete axiomatizations of finite syntactic epistemic states. In M. Baldoni, U. Endriss, A. Omicini, & P. Torroni (Eds.), *Declarative Agent Languages and Technologies III: Third International Workshop, DALT 2005, Utrecht, The Netherlands*, July 25, 2005. *Selected and revised papers of lecture notes in computer science (LNCS)* (Vol. 3904, pp. 33–50). Heidelberg: Springer.

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

Alur, R., Henzinger, T. A., & Kupferman, O. (2002). Alternating-time temporal logic. *Journal of the ACM, 49*(5), 672–713.

Clarke, E. M., Grumberg, O., & Peled, D. A. (2000). *Model checking*. Cambridge, MA: The MIT Press.

Dunne, P. E., van der Hoek, W., & Wooldridge, M. (2007). A logical characterisation of qualitative coalitional games. *Journal of Applied Non-Classical Logics, 17*(4), 477–509.

Gallier, J. (1987). *Logic for computer science: Foundations of automatic theorem proving*. Wiley.

Gargov, G., & Passy, S. (1987). A note on Boolean modal logic. In *Mathematical logic and applications* (pp. 253–263). Plenum Press.

Lutz, C. (2006). Complexity and succinctness of public announcement logic. In *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2006)*, Hakodate, Japan.

Papadimitriou, C. H. (1994). *Computational complexity*. Reading, MA: Addison-Wesley.

Pauly, M. (2001). Logic for social software. Ph.D. thesis, University of Amsterdam. ILLC Dissertation Series 2001-10.

van der Hoek, W., Jamroga, W., & Wooldridge, M. (2005a). A logic for strategic reasoning. In *Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2005)* (pp. 157–153). The Netherlands: Utrecht.

van der Hoek, W., & Lomuscio, A., & Wooldridge, M. (2005b). On the complexity of practical ATL model checking. In *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2006)*, Hakodate, Japan.

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