On the Logic of Cooperation and Propositional Control

Wiebe van der Hoek * and Michael Wooldridge

Dept of Computer Science, University of Liverpool, Liverpool L69 7ZF, United Kingdom

Abstract

Cooperation logics have recently begun to attract attention within the multi-agent systems community. Using a cooperation logic, it is possible to represent and reason about the strategic powers of agents and coalitions of agents in game-like multi-agent systems. These powers are generally assumed to be implicitly defined within the structure of the environment, and their *origin* is rarely discussed. In this paper, we study a cooperation logic in which agents are each assumed to control a set of propositional variables – the powers of agents and coalitions then derive from the allocation of propositions to agents. The basic modal constructs in this Coalition Logic of Propositional Control (CL-PC) allow us to express the fact that a group of agents can cooperate to bring about a certain state of affairs. After motivating and introducing CL-PC, we provide a complete axiom system for the logic, and prove that the model checking and satisfiability problems for CL-PC are both PSPACE-complete. We then investigate the issue of characterising *control* in CL-PC with respect to the underlying power structures of the logic, and formally investigate the relationship between CL-PC and Pauly's Coalition Logic. We conclude by discussing our results and how CL-PC sits in relation to other logics of cooperation.

Key words: Multi-agent systems, cooperation logic.

1 Introduction

Cooperation logics are logics that are intended to enable reasoning about coalitions in multi-agent systems, and in particular, the *powers* that such coalitions have. Probably the two best known examples of cooperation logics are Pauly's Coalition Logic [36,38,37], and the Alternating-time Temporal Logic (ATL) of Alur, Henzinger, and Kupferman [5]. Both of these systems are based upon the notion of a

* Corresponding author.
Email address: wiebe@csc.liv.ac.uk (Wiebe van der Hoek).

cooperation modality: a unary modal operator, indexed by a set of agents, which is used to represent the fact that this set of agents can cooperate so as to make true the state of affairs given as an argument to the operator. In Coalition Logic, for example, a formula $[1, 2](p \wedge q)$ is used to express the fact that the coalition $\{1, 2\}$ can cooperate in such a way as to make the formula $p \wedge q$ true. Although they differ on technical details, the semantics of both logics are essentially equivalent [19]: coalition C are able to achieve φ if there exists a collective strategy for C such that, by following this strategy, C can enforce φ (that is, φ is true in every outcome that could arise by following the strategy). In both logics, the strategies available to a coalition, and the outcomes consistent with these strategies, are implicitly enumerated within the logic's models: in the case of Coalition Logic, by means of *effectivity functions* (cf. [1]), and in the case of ATL, by means of a *system transition function*.

In this paper, we study a variant of cooperation logic that we refer to as the *Coalition Logic of Propositional Control* (CL-PC). The key idea in CL-PC is that each agent is assumed to control a set of propositional variables. The strategies, or choices available to an agent then correspond to the different possible assignments of truth or falsity to these propositions. On top of that, the ability of a coalition to bring about some state of affairs derives from the propositional variables that are under the overall control of the coalition.

There are at least two reasons why CL-PC is a system worthy of study in its own right:

- First, and perhaps most importantly, if we are interested in building software agents, then it is extremely natural to think of the ability of these agents in terms of setting and unsetting bits in some digital control system. Indeed, this is arguably the most fundamental kind of control that can be imagined.
- Second, in implemented systems for reasoning about cooperation in game-like multi-agent scenarios, the individual powers of agents are actually specified by allocating agents propositions that lie under their control. For example, this is exactly the approach taken in the MOCHA model checking system for ATL, where the keyword `controls` is used to indicate the fact that an agent (or "module", in the terminology of MOCHA) is uniquely able to determine the value of a propositional variable [6,3].

Against this background, the present paper makes four main contributions to the study of cooperation logics.

First, although we have taken inspiration and some methodology from ATL and Coalition Logic, we note that the basic cooperation modalities of these logics have a rather unusual modal flavour, which is neither wholly universal nor wholly existential. This is because these logics are intended to capture $\exists\forall$-ability, or α-ability [1, pp.11–12], the idea being that a coalition C have the α-ability to achieve some

	Coalition Logic	ATL	CL-PC
contingent ability			$\langle C \rangle \varphi$
α-ability	$[C]\varphi$	$\langle\langle C \rangle\rangle \bigcirc \varphi$	$\langle\langle C \rangle\rangle_\alpha \varphi$
β-ability			$\langle\langle C \rangle\rangle_\beta \varphi$

Table 1
Comparing the notation of Coalition Logic, ATL, and CL-PC for different types of ability.

state of affairs φ if C have a collective choice such that, no matter what the agents outside C do, φ will hold as a consequence of this choice. This type of ability is entirely natural when we wish to study the circumstances under which a coalition can "reliably" bring about some state of affairs. However, the $\exists\forall$ nature of cooperation modalities in these logics, and the fact that they are neither conventionally existential nor conventionally universal, means that (*i*) they are hard for "logic users" to understand, as they have some counterintuitive properties (particularly in their *dual* forms); and (*ii*) from a technical standpoint, the fact that they are a *combination* of existential and universal modality makes them somewhat awkward to work with, at least compared with conventional "box" and "diamond" modalities. In CL-PC, however, the basic cooperation modalities capture *contingent ability* – a weaker notion of ability than the α-ability of Coalition Logic and ATL. Contingent ability is the ability to achieve some state of affairs under the assumption that, apart from our actions, the world remains static. This is the type of ability that is implemented in classic AI planning systems such as STRIPS [17,28]. Although contingent ability is perhaps not of great interest in its own right, it turns out that the contingent ability constructs of CL-PC are sufficient to define α-ability (as well as a related type of ability known as β-ability). Thus, although we appear to start with a weaker notion of ability than those of Coalition Logic and ATL, it turns out that *this is in fact all we need to define these stronger types of ability*. Moreover, the contingent ability operators of CL-PC have the advantage of being "true modal diamonds": they thus have a much simpler semantics than α-ability operators, and are easier to work with from a technical point of view (for example, when using such conventional modal logic constructs such as canonical models to prove completeness [11, pp.59–61]). As an aside, we note that, unfortunately, there is no standard notation in the literature of cooperation logics: in Table 1, we compare our notation to that of Coalition Logic and ATL.

Second, although our basic cooperation modalities are conventional modal diamonds, and hence we can give them a more-or-less conventional Kripke semantics [9, p.42], we are also able to give an alternative semantics to CL-PC, which is directly based on the power structures that underpin the logic. We show that the two semantics are, in a precise sense, equivalent, and that we can thus move between the two semantics as we see fit. The advantage of this is that we can work with whichever semantics seems most appropriate to the task at hand.

Third, although complete axiomatizations are known for both ATL [19] and Coalition Logic [38,37,36], we are able to provide an axiomatization for CL-PC that draws upon the simple underlying power structures of the logic. As a consequence, our axiomatization of CL-PC has a rather different flavour to those of Coalition Logic and ATL.

Fourth, and finally, we present an analysis of *control* in CL-PC: when a coalition controls some state of affairs. In particular, we show how the control that a coalition is able to exert with respect to some state of affairs is related to the power structure underlying the logic.

The remainder of this paper is structured as follows. Following a formal definition of CL-PC, in Section 3 we present a complete axiomatization for the logic. In Section 4.1, we show how α- and β-ability modalities can be defined in terms of the basic constructs of CL-PC, and in Section 4.2, we investigate the characterisation of control in CL-PC, with particular reference to the underlying power structures. In Section 5, we investigate the computational complexity of the model checking and satisfiability problems for the logic, and show that both problems are PSPACE-complete. We conclude with a discussion on the implications of our results, and how the logic stands in relation to other similar systems.

2 The Coalition Logic of Propositional Control

In this section, we give a formal definition of CL-PC. We begin with an informal introduction to the main features of the logic (readers familiar with Coalition Logic or ATL may wish to skip or skim through this section). We then formally define the syntax of CL-PC, and give two alternative semantics. The first is a "direct" or "propositional" semantics, which has the advantages of being both simple and closely related to the intuitions underpinning the language, but has the disadvantage of being rather unconventional in the modal logic sense, and hence rather hard to work with from the point of view of proving properties such as completeness. In the second semantics, we employ the conventional Kripke-style relational structures of modal logic [11,9], making it possible to represent both the current situation and those that the agents can bring about in one and the same model. In Section 2.5, we show that these two semantics are equivalent. The obvious advantage of having two equivalent semantics for our logic is that we can choose to work which whichever semantics is most convenient for the problem at hand.

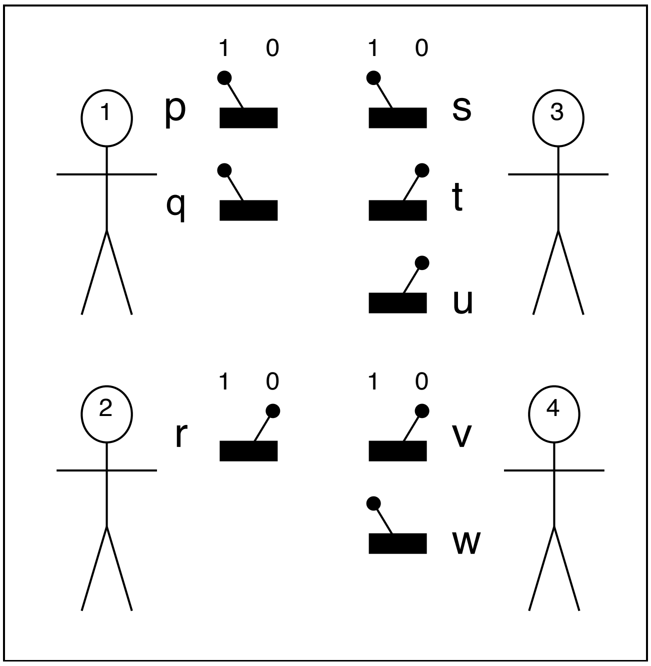

Fig. 1. In CL-PC, each agent is assumed to have unique control over the value of a set of propositional variables.

2.1 Informal Introduction

The components of the systems of interest to us are as follows. First, we assume a vocabulary At of *propositional variables*, which represent attributes of the systems that we model. Next, we assume a finite, non-empty set Ag of *agents*; a *coalition* is simply a subset of Ag. The only property that we assume of agents is that they are each able to *control* a part of their environment. We capture this by assuming that every propositional variable is controlled by exactly one agent. Thus we associate with every agent $i \in Ag$ a (possibly empty) subset At_i of At, representing those propositional variables under its control. The following example illustrates these ideas in more detail, and also informally introduces the language of CL-PC.

Example 2.1 Figure 1 illustrates the kind of scenario that we have in mind for CL-PC. Here, we have four agents, (named 1 through to 4), and seven propositional variables (p to w). Agent 1 controls propositions p and q, agent 2 controls r, agent 3 controls s, t, and u, and agent 4 controls v and w.

Agent 1 has assigned the value “1” to variables p and q – that is, both these variables have the value “true” (hereafter denoted by “`tt`”). Similarly, the variables s and w have been set to `tt` by agents 3 and 4 respectively, while all remaining variables have been set to 0 (i.e., “`ff`”). Now, given this scenario, we say that agent 1 has the contingent ability for $\neg p \wedge s$. That is, assuming that all those propositions *not* under the control of agent 1 retain their current value, then agent 1 can choose values for its propositions so as to make $\neg p \wedge s$ true. We express this in CL-PC by the following formula.

$$\langle 1 \rangle (\neg p \wedge s)$$

Thus the CL-PC expression $\langle C \rangle \varphi$ means "coalition C have the contingent ability to achieve φ".

As we noted above, contingent ability corresponds quite closely to "STRIPS-style" planning ability [17,28], in the sense that agent 1 has a plan to achieve $\neg p \wedge s$ from the scenario in Figure 1: the plan consists of one action, namely setting p to ff. However, the plan is clearly contingent, in the sense that it depends upon the value of s remaining unchanged by agent 3. If agent 3 chose to change this value, then the plan would fail.

Here is another example of a contingent ability formula that is true of Figure 1.

$$\langle 1, 3 \rangle (\neg p \wedge t \wedge \neg r)$$

A "multi-agent plan" to achieve $\neg p \wedge t \wedge \neg r$ would involve two actions: agent 1 setting p to ff, and agent 3 setting t to tt. The plan is contingent upon agent 2 leaving the value of r unchanged, at ff.

Contingent ability, as captured in the coalitional ability operator of CL-PC is clearly rather limited. It makes perfect sense however in truly *turn-based systems*, in which agents perform their actions in an alternating fashion (most games have this nature). Moreover, as we noted above, it is in fact sufficient to define α-and β-ability, both of which represent stronger and arguably more useful strains of ability. Although we will not formally define and investigate these types of ability until Section 4.1, we nevertheless provide some illustrative examples here.

In CL-PC, we write $\langle\langle C \rangle\rangle_\alpha\, \varphi$ to express the fact that coalition C are α-able to bring about φ. This means that coalition C can choose to act in such a way that *no matter what the agents outside C choose to do*, φ will be true. With respect to the scenario of Figure 1, the following formula is thus *not* true.

$$\langle\langle 1 \rangle\rangle_\alpha\, (\neg p \wedge s)$$

In contrast, the following formula *is* true in the Figure 1 scenario.

$$\langle\langle 1, 3 \rangle\rangle_\alpha\, (\neg p \wedge s)$$

We represent β-ability in CL-PC by the cooperation modality $\langle\langle \cdots \rangle\rangle_\beta$. The distinction between α-and β-ability may be understood through the following example. Suppose we have agents 1 and 3 playing a game in which agent 1 wins if, after the game is over, *the variables p and s have the opposite values* – that is, if p is tt, then s is `ff`, and vice versa. The game itself simply involves the agents choosing values for their variables.

Now, is agent 1 able to win this game? It depends upon whether we are talking about α-ability or β-ability. Agent 1 clearly does *not* have the α-ability to win this game:

$$\neg\langle\langle 1\rangle\rangle_{\alpha}\,(p \leftrightarrow \neg s)$$

This is because, when considering α-ability, agent 1 must choose first: and after choosing a value for p, agent 3 can then simply copy this value in the choice for s, thereby falsifying $p \leftrightarrow \neg s$.

However, when considering β-ability, the assumption is that agent 1 carries a *responsive role*, while agent 3 will choose first: agent 1 can then reliably win by choosing for p the opposite of whatever agent 3 chooses for s. Thus, the following CL-PC formula *is* true of Figure 1.

$$\langle\langle 1\rangle\rangle_{\beta}\,(p \leftrightarrow \neg s)$$

Notice that β-ability is strictly weaker than α-ability; we discuss this at more length in Section 4.1.

2.2 Syntax

Formulae of CL-PC are constructed from the set Ag of agents, the set At of propositional atoms, the usual operators of classical propositional logic (we use negation and disjunction as our atomic Boolean functions, together with a logical constant for truth), and the cooperation modality $\langle \cdots \rangle$. More formally, the syntax of CL-PC is given by the following BNF grammar:

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

where $p \in$ At is a propositional variable, and $C \subseteq Ag$ is a set of agents. Thus we use $\top$ as a logical constant for truth, "$\neg$" for negation, and "$\vee$" for disjunction. As usual, we define the remaining connectives of classical propositional logic as abbreviations: $\bot \hat{=} \neg\top$, $\varphi \to \psi \hat{=} \neg\varphi \vee \psi$ and $\varphi \leftrightarrow \psi \hat{=} (\varphi \to \psi) \wedge (\psi \to \varphi)$.

We refer to an expression $\langle C\rangle\varphi$ as a *cooperation modality*. Where there is no possibility of confusion, we will omit set brackets inside cooperation modalities, for example writing $\langle 1, 2\rangle$ rather than $\langle\{1, 2\}\rangle$. A CL-PC formula containing no cooperation modalities is said to be an *objective* formula.

If φ is a CL-PC formula, then let $\mathsf{At}(\varphi)$ denote the set of propositional variables occurring in φ, and let $Ag(\varphi)$ denote the set of all agents named in φ (i.e., $Ag(\varphi)$ is the union of all the coalitions occurring in cooperation modalities in φ).

2.3 Direct Semantics

We now introduce the first of the two semantics for CL-PC. We say a *model* for CL-PC is a structure:

$$\mathcal{M} = \langle Ag, \mathsf{At}, \mathsf{At}_1, \ldots, \mathsf{At}_n, \theta \rangle$$

where:

- $Ag = \{1, \ldots, n\}$ is finite, non-empty a set of *agents*;
- $\mathsf{At} = \{p, q, \ldots\}$ is finite, non-empty set of *propositional variables*;
- $\mathsf{At}_1 \ldots, \mathsf{At}_n$ is a partition of At among the members of Ag, with the intended interpretation that At_i is the subset of At representing those variables under the control of agent $i \in Ag$; and finally,
- $\theta : \mathsf{At} \to \{\mathtt{tt}, \mathtt{ff}\}$ is a propositional valuation function, which determines the initial truth value of every propositional variable.

Notice that since $\mathsf{At}_1 \ldots, \mathsf{At}_n$ is a partition of At, we have:

(1) $\mathsf{At} = \mathsf{At}_1 \cup \cdots \cup \mathsf{At}_n$
 i.e., every variable is controlled by some agent, and
(2) $\mathsf{At}_i \cap \mathsf{At}_j = \emptyset$ for $i \neq j \in Ag$
 i.e., no variable is controlled by more than one agent.

Some additional notation is convenient in what follows. We say a *coalition*, C is simply a subset of Ag, i.e., $C \subseteq Ag$. For any such $C \subseteq Ag$ we denote the *complement* of C, (i.e., $Ag \setminus C$) by $\overline{C}$. We will write At_C for $\bigcup_{i \in C} \mathsf{At}_i$. For two valuations θ and θ', and a set of propositions $\Psi \subseteq \mathsf{At}$, we write $\theta = \theta' \pmod{\Psi}$ if θ and θ' differ at most in the atoms in Ψ, and we then say that θ and θ' are the *same modulo* Ψ. We will sometimes understand the model $\mathcal{M} = \langle \mathcal{F}, \theta \rangle$ to consist of a valuation θ on top of a *frame* $\mathcal{F} = \langle Ag, \mathsf{At}, \mathsf{At}_1, \ldots, \mathsf{At}_n \rangle$. Given a model $\mathcal{M} = \langle Ag, \mathsf{At}, \mathsf{At}_1, \ldots, \mathsf{At}_n, \theta \rangle$ and a coalition C in $\mathcal{M}$, a *C-valuation* is a function:

$$\theta_C : \mathsf{At}_C \to \{\mathtt{tt}, \mathtt{ff}\}.$$

Thus a C-valuation is a function that assigns truth values to just the primitive propositions controlled by the members of the coalition C. If $\mathcal{M} = \langle Ag, \mathsf{At}, \mathsf{At}_1, \ldots, \mathsf{At}_n, \theta \rangle$ is a model, C is a coalition in $\mathcal{M}$, and θ_C is a C-valuation, then by $\mathcal{M} \oplus \theta_C$ we mean the model $\langle Ag, \mathsf{At}, \mathsf{At}_1, \ldots, \mathsf{At}_n, \theta' \rangle$, where θ' is the function defined as follows

$$\theta'(p) \hat{=} \begin{cases} \theta_C(p) & \text{if } p \in \mathsf{At}_C \\ \theta(p) & \text{otherwise} \end{cases}$$

and all other element of the model are as in $\mathcal{M}$. Thus $\mathcal{M} \oplus \theta_C$ denotes the model that is identical to $\mathcal{M}$ except that the values assigned by its valuation function to propositions controlled by members of C are as determined by θ_C: we have $\theta = \theta' \pmod{\mathsf{At}_C}$. In terms of frames: If $\mathcal{M} = \langle \mathcal{F}, \theta \rangle$ then $\mathcal{M} \oplus \theta_C = \langle \mathcal{F}, \theta_C \rangle$. Notice that the $\emptyset$-valuation $\theta_\emptyset$ is the right identity under $\oplus$, that is, $\mathcal{M} \oplus \theta_\emptyset = \mathcal{M}$, for all $\mathcal{M}$.

We define the *size* of a model $\mathcal{M} = \langle Ag, \mathsf{At}, \mathsf{At}_1, \ldots, \mathsf{At}_n, \theta \rangle$ to be $|Ag| + |\mathsf{At}|$; we denote the size of $\mathcal{M}$ by $size(\mathcal{M})$.

We interpret formulae of CL-PC with respect to models, as introduced above.

Definition 1 Given a model $\mathcal{M} = \langle Ag, \mathsf{At}, \mathsf{At}_1, \ldots, \mathsf{At}_n, \theta \rangle$ and a formula φ, we write $\mathcal{M} \models^{\mathsf{d}} \varphi$ to mean that φ is satisfied (or, equivalently, true) in $\mathcal{M}$, under the "direct" semantics. The rules defining the satisfaction relation $\models^{\mathsf{d}}$ are as follows:

$\mathcal{M} \models^{\mathsf{d}} \top$
$\mathcal{M} \models^{\mathsf{d}} p$ iff $\theta(p) = \mathtt{tt}$ (where $p \in \mathsf{At}$);
$\mathcal{M} \models^{\mathsf{d}} \neg\varphi$ iff $\mathcal{M} \not\models^{\mathsf{d}} \varphi$;
$\mathcal{M} \models^{\mathsf{d}} \varphi \vee \psi$ iff $\mathcal{M} \models^{\mathsf{d}} \varphi$ or $\mathcal{M} \models^{\mathsf{d}} \psi$;
$\mathcal{M} \models^{\mathsf{d}} \langle C \rangle \varphi$ iff there exists a C-valuation θ_C such that $\mathcal{M} \oplus \theta_C \models^{\mathsf{d}} \varphi$.

We assume the conventional definitions of satisfiability and validity: a CL-PC formula φ is ${}^{\mathsf{d}}$-*satisfiable* iff there exists a CL-PC model $\mathcal{M}$ such that $\mathcal{M} \models^{\mathsf{d}} \varphi$, and φ is ${}^{\mathsf{d}}$-*valid* iff for every CL-PC model $\mathcal{M}$ we have $\mathcal{M} \models^{\mathsf{d}} \varphi$. We write $\models^{\mathsf{d}} \varphi$ to indicate that φ is ${}^{\mathsf{d}}$-valid. Moreover, if I is a partition $\{\mathsf{At}_1, \mathsf{At}_2, \ldots, \mathsf{At}_n\}$ of the atoms At over the agents, and $\mathcal{M} \models^{\mathsf{d}} \varphi$ for every model $\mathcal{M} = \langle Ag, \mathsf{At}, \mathsf{At}_1, \ldots, \mathsf{At}_n, \theta \rangle$ based on this partition, we write $\models^{\mathsf{d}}_I \varphi$.

At this point, let us introduce the natural box dual "$[\cdots]$" of the $\langle \cdots \rangle$ cooperation modality:

$$[C]\varphi \hat{=} \neg\langle C \rangle \neg\varphi.$$

The intuitive interpretation of a formula $[C]\varphi$ is that C *cannot avoid* φ: that if, for every possible way that they choose to behave, φ "may" still be true. However it does not mean that φ will *inevitably* be true, as the following example illustrates.

Example 2.2 Suppose we have a model $\mathcal{M} = \langle \{1, 2\}, \mathsf{At}, \mathsf{At}_1, \mathsf{At}_2, \theta \rangle$ such that $\mathsf{At} = \{p, q\}$, $\mathsf{At}_1 = \emptyset$, $\mathsf{At}_2 = \{p, q\}$ and $\theta(x) = \mathtt{tt}$ for $x \in \{p, q\}$. Now, there is no 1-valuation θ_1 such that $\mathcal{M} \oplus \theta_1 \models^{\mathsf{d}} \neg p$. That is, $\mathcal{M} \models^{\mathsf{d}} [1]p$. Similarly, agent 1 cannot avoid $p \wedge q$, i.e., $\mathcal{M} \models^{\mathsf{d}} [1](p \wedge q)$. But this does not mean $p \wedge q$ is inevitable: it depends on the choice that 2 makes. Since 2 controls both p and q, we have $\mathcal{M} \models^{\mathsf{d}} \langle 2 \rangle \neg p$, $\mathcal{M} \models^{\mathsf{d}} \langle 2 \rangle \neg q$, and $\mathcal{M} \models^{\mathsf{d}} \langle 2 \rangle \neg(p \vee q)$.

Before proceeding, we pause to consider the extent to which the *number of agents in a model* affects whether or not a formula is satisfied. Let us say that a model $\langle Ag, \mathsf{At}, \mathsf{At}_1, \ldots, \mathsf{At}_n, \theta \rangle$ is a k-agent model if $|Ag| = k$. In the same way, let us say a formula φ is a k-agent formula if $|Ag(\varphi)| = k$. Now, consider the following example.

Example 2.3 (Individual *versus* Multi-agent Models) *Consider the* 1*-agent formula* $p \wedge [1]p$*. This formula is clearly satisfiable, as is witnessed by the* 2*-agent model* $\langle Ag, \mathsf{At}, \mathsf{At}_1, \mathsf{At}_2, \theta \rangle$ *such that* $Ag = \{1, 2\}$*,* $\mathsf{At} = \{p\}$*,* $\mathsf{At}_1 = \emptyset$*,* $\mathsf{At}_2 = \{p\}$*, and* $\theta(p) = \mathtt{tt}$*. However, this formula is clearly* not *satisfied in any* 1*-agent model, because the only agent in such a model must (by virtue of the fact that the formula is well-formed) be* 1*; thus agent* 1 *must control* p*, and so can choose* $\neg p$*. Hence the formula is not satisfied by the model.*

In this example, we have a 1-agent formula that is satisfiable in a 2-agent model, but not in any 1-agent model. This begs an obvious question: To what extent is the satisfiability of a formula affected by the number of agents in the models we consider for this formula? We have the following result, which says that any satisfiable k-agent formula is satisfied in a $k + 1$-agent model. In other words, if a formula is satisfiable, then it is satisfied in a model containing at most one agent in addition to those explicitly named in the formula.

Lemma 2.1 *A* CL-PC *formula* φ *is satisfiable iff it is satisfied in a model* $\mathcal{M} = \langle Ag, \mathsf{At}, \mathsf{At}_1, \ldots, \mathsf{At}_n, \theta \rangle$ *such that* $|Ag| = |Ag(\varphi)| + 1$ *and* $\mathsf{At} = \mathsf{At}(\varphi)$*.*

Proof: The direction from right to left is obvious, so suppose φ is satisfiable, let $Ag(\varphi) = \{1, \ldots, k\}$ be the agents named in φ, and let $\mathcal{M} = \langle Ag, \mathsf{At}, \mathsf{At}_1, \ldots, \mathsf{At}_k, \mathsf{At}_{k+1}, \ldots, \mathsf{At}_m, \theta \rangle = \langle \mathcal{F}, \theta \rangle$ be a model based on $\mathcal{F}$ that satisfies φ, thus $Ag = \{1, \ldots, k, k+1, \ldots, m\}$. Let S denote the agents $\{k+1, \ldots, m\}$ in Ag that are not named in φ. First, let e denote a new agent – one that is neither named in φ nor is a member of S. Now, consider the frame $\mathcal{F}' = \langle Ag', \mathsf{At}', \mathsf{At}'_1, \ldots, \mathsf{At}'_k, \mathsf{At}_e \rangle$ such that:

- $Ag' = Ag(\varphi) \cup \{e\}$;
- $\mathsf{At}' = \mathsf{At}(\varphi)$;
- $\mathsf{At}'_i = \mathsf{At}_i \cap \mathsf{At}(\varphi)$, for all $i \leq k$
- $\mathsf{At}_e = \mathsf{At}(\varphi) \setminus \bigcup_{i \leq k} \mathsf{At}'_i$.

Obviously, $|Ag'| = |Ag(\varphi)| + 1$ and $\mathsf{At}' = \mathsf{At}(\varphi)$. Finally, on the level of models, given a valuation $\theta : \mathsf{At} \to \{\mathtt{tt}, \mathtt{ff}\}$, let $\theta_{\restriction \mathsf{At}'} : \mathsf{At}' \to \{\mathtt{tt}, \mathtt{ff}\}$ be the restriction of θ to At'. Now, we claim:

$$\text{for all valuations } \theta \text{ and all subformulas } \psi \text{ of } \varphi\text{: } \langle \mathcal{F}, \theta \rangle \models^{\mathsf{d}} \psi \text{ iff } \langle \mathcal{F}', \theta_{\restriction \mathsf{At}'} \rangle \models^{\mathsf{d}} \psi$$

The proof is by induction on the structure of ψ The inductive base is where $\psi \in \mathsf{At}$, and is obvious since θ is unchanged on the relevant atoms when going from $\mathcal{F}$ to $\mathcal{F}'$. For the inductive assumption, assume that the result holds for all subformulae ψ of φ. For the inductive step, the cases for propositional connectives are immediate. So consider the case where $\chi = \langle C \rangle \psi$. Now, $\langle \mathcal{F}, \theta \rangle \models^{\mathsf{d}} \langle C \rangle \psi$ iff for some θ^i, with $\theta^i = \theta \pmod{\mathsf{At}_i}$, we have $\langle \mathcal{F}, \theta^i \rangle \models^{\mathsf{d}} \psi$. The induction hypothesis guarantees that this is equivalent to $\langle \mathcal{F}', \theta^i_{\upharpoonright \mathsf{At}'} \rangle \models^{\mathsf{d}} \psi$ and this in turn is equivalent to $\langle \mathcal{F}', \theta_{\upharpoonright \mathsf{At}'} \rangle \models^{\mathsf{d}} \langle i \rangle \psi$. □

Pauly discusses somewhat related issues in the context of Coalition Logic [36, pp.68–71]. The significance of our result will become clear later, when we use the following immediate corollary as a kind of polysize model property, of the kind that is frequently used in establishing upper bounds in the complexity theory of modal logic [9, pp.375–381].

Corollary 2.1 *If a* CL-PC *formula φ is satisfiable, then it is satisfied in a model $\mathcal{M}$ such that size$(\mathcal{M}) = |\mathsf{At}(\varphi)| + |Ag(\varphi)| + 1$.*

2.4 Kripke Semantics

The “direct” semantics presented above has the advantage of being closely related to our intuitive understanding of the logic CL-PC: the semantic rule for ability is based directly upon the way that propositional variables are partitioned among the agents in the system. However, from a technical point of view, the direct semantics has the disadvantage of being unusual and unfamiliar – and hence (arguably) hard to work with. In this section, we therefore introduce a semantics for CL-PC based on the familiar and conventional Kripke relational structures of modal logic [11,9]. Although readers may have a personal preference for one semantics over the other, we show in Section 2.5 that the two semantics are in fact equivalent, and that we can thus use whichever semantics seems most appropriate to the task in hand.

Definition 2 Let W be the set of all valuations θ over At. For every $\mathsf{At}_i \subseteq \mathsf{At}$, we assume a binary relation R_i to be given, with wR_iw' iff w and w' differ at most in the values that they assign to the atoms in At_i. Formally: $wR_iw' \Leftrightarrow w = w' \pmod{\mathsf{At}_i}$. Note that R_i is an equivalence relation. To lift the R_i relations, (corresponding to the possible choices of *individual* agents) to *arbitrary coalitions*, we need the notion of *relational composition*. Given a set S, and binary relations $R_1, R_2 \subseteq S \times S$ over S, we denote the *composition* of R_1 and R_2 by $R_1 \circ R_2$, where this is defined as follows.

$$R_1 \circ R_2 = \{(s,t) \mid \exists u : sR_1u \ \& \ uR_2t\}$$

We then define the Kripke model $\mathcal{K} = \langle W, R_1, \ldots, R_n, \Pi \rangle$, where $\Theta : W \to \mathsf{At} \to \{\mathtt{tt}, \mathtt{ff}\}$ is simply given by $\Theta(w) = w$. For any coalition $C = \{a_1, a_2, \ldots a_k\} (k \leq$

n), we define R_C as $R_{a_1} \circ R_{a_2} \circ \cdots \circ R_{a_k}$. The following result tells us that the composition is independent of the order.

Lemma 2.2 Let $\mathcal{K} = \langle W, R_1, \ldots R_n, \Theta \rangle$ be defined as in Definition 2. Then for every $i, j \in Ag$, we have: $R_i \circ R_j = R_j \circ R_i$.

Proof: Suppose $w(R_i \circ R_j)v$. Then there is a u for which wR_iu and uR_jv. This means that $w = u \pmod{\mathsf{At}_i}$ and, similarly, $u = v \pmod{\mathsf{At}_j}$. Note that hence $w = v \pmod{\mathsf{At}_i \cup \mathsf{At}_j}$. Let u' be defined as follows. It is exactly like w, except for the atoms in At_j, for which u' is as v. Then, by definition of u', we have wR_ju' (by definition, they at most differ in At_j) and $u'R_iv$ (w and u differ at most in $\mathsf{At}_i \cup \mathsf{At}_j$, hence u' and v differ at most in At_i). We conclude that $w(R_j \circ R_i)v$. $\square$

We can now see the following.

Lemma 2.3 *Let $\mathcal{K} = \langle W, R_1, \ldots R_n, \Theta \rangle$ be defined as in Definition 2. Then for all $C \subseteq Ag$, R_C is an equivalence relation.*

Proof: Reflexivity and symmetry for R_C are immediately inherited from the same properties of R_i ($i \in C$). For transitivity, suppose that uR_Cv and vR_Cw. Let us assume that $C = \{1, 2, \ldots n\}$. Then, since the order is irrelevant (Lemma 2.2), we know that there are paths $u = u_1, u_2, \ldots, u_n = v$ and $v = v_1, v_2, \ldots, v_n = w$ such that $R_iu_iu_{i+1}$ and $R_iv_iv_{i+1}$. By applying Lemma 2.2 repeatedly we find u'_i and v'_i ($i \leq n$) with a path $u = u'_1, v'_1, u'_2, v'_2, \ldots u'_n, v'_n = w$ such that $R_iu'_iv'_i$ and $R_iv'_iu'_{i+1}$. Now we can apply transitivity to the individual agent's relations, giving us a path from u to w that is in R_C. $\square$

The inductive rules defining the truth of a CL-PC formula φ under an interpretation $\mathcal{K}, w$ are defined as follows:

$\mathcal{K}, w \models^{\mathsf{k}} \top$
$\mathcal{K}, w \models^{\mathsf{k}} p$ iff $\Theta(w)(p) = w(p) = \mathtt{tt}$ (where $p \in \mathsf{At}$);
$\mathcal{K}, w \models^{\mathsf{k}} \neg\varphi$ iff $\mathcal{K} \not\models^{\mathsf{k}} \varphi$;
$\mathcal{K}, w \models^{\mathsf{k}} \varphi \vee \psi$ iff $\mathcal{K}, w \models^{\mathsf{k}} \varphi$ or $\mathcal{K}, w \models^{\mathsf{k}} \psi$;
$\mathcal{K}, w \models^{\mathsf{k}} \langle C \rangle \varphi$ iff there exists a $w' \in W$ such that wR_Cw' and $\mathcal{K}, w' \models^{\mathsf{k}} \varphi$.

Note that, given the partition I on At, there is only one model $\mathcal{K}$, which we sometimes also will denote as $\mathcal{K}_I$ [1] . We write $\mathcal{K}_I \models^{\mathsf{k}} \varphi$, but also $\models^{\mathsf{k}}_I \varphi$ as a shorthand for $\mathcal{K}_I, w \models^{\mathsf{k}} \varphi$ for every world w in $\mathcal{K}_I$. If moreover the latter holds for every $\mathcal{K}$, more precisely for every $\mathcal{K}_I$, we write $\models^{\mathsf{k}} \varphi$.

[1] As an aside, we might generalize the notion of I-validity to that in models $\mathcal{K}'_I = \langle W', R_1, \ldots R_n, \Theta \rangle$, where $W' \subseteq W$ is some subset of all the possible valuations, reflecting some background theory on the allowed propositional formulas.

Since by Lemma 2.3 every accessibility relation R_C in each model $\mathcal{K}$ is an equivalence we immediately obtain the following properties (see, e.g., [11]).

$$\begin{array}{ll} a & \models^{\mathsf{k}} \varphi \rightarrow \langle C\rangle\varphi \\ b & \models^{\mathsf{k}} \langle C\rangle\langle C\rangle\varphi \leftrightarrow \langle C\rangle\varphi \\ c & \models^{\mathsf{k}} \langle C\rangle\neg\langle C\rangle\varphi \leftrightarrow \neg\langle C\rangle\varphi \\ d & \models^{\mathsf{k}} [C]\varphi \rightarrow \varphi \\ e & \models^{\mathsf{k}} [C]\langle C\rangle\varphi \leftrightarrow \langle C\rangle\varphi \\ f & \models^{\mathsf{k}} [C][C]\varphi \leftrightarrow [C]\varphi \end{array} \tag{1}$$

Property a is the dual of d which in turn is modal axiom T and is guaranteed by reflexivity of R_C (cf. Lemma 2.3). Properties b and f are also each other duals: in f, the $\rightarrow$-direction is a special case of d, the other direction (axiom 4) is due to the transitivity of R_C. Of property c, the $\leftarrow$-direction is an instance of a, the other direction is equivalent to axiom 5, which is usually given as $\neg[C]\psi \rightarrow [C]\neg[C]\psi$. This is also what the $\leftarrow$-direction of e is expressing. Finally, the $\rightarrow$-direction of e is an instance of d.

We note in passing that $[Ag]$ is a "universal" modality, and gives us a bridge to go from "local" (i.e., truth at a world in a Kripke structure) to "global" (i.e., truth at every world in an I-Kripke structure):

$$\mathcal{K}_I \models^{\mathsf{k}} \varphi \quad \Leftrightarrow \quad \exists w : \mathcal{K}_I, w \models^{\mathsf{k}} [Ag]\varphi \tag{2}$$

Example 2.4 (Bach or Stravinsky) Let us consider two agents, each having a choice to go to either a concert by Bach or by Stravinsky. Let b_1 be the atom denoting that agent 1 chooses for Bach, and similarly for b_2 and agent 2. Obviously, I is such that each agent i has control over the atom b_i. It is understood that agent i's choice for Stravinsky is represented as $\neg b_i$. Assume that initially no agent is going to Bach (i.e., w and Θ are such that $\mathcal{K}, w \models^{\mathsf{k}} (\neg b_1 \wedge \neg b_2)$), then we have in w that no agent can force them both going to Bach $(\neg\langle 1\rangle(b_1 \wedge b_2) \wedge \neg\langle 2\rangle(b_1 \wedge b_2))$, whereas in full cooperation they can share a Bach evening $(\langle 1,2\rangle(b_1 \wedge b_2))$. On a global level, neither agent can establish an evening out together ($\mathcal{K}_I \not\models^{\mathsf{k}} \langle i\rangle(b_1 \wedge b_2)$ and $\mathcal{K}_I \not\models^{\mathsf{k}} \langle i\rangle(\neg b_1 \wedge \neg b_2)$, $i \leq 2$), but fortunately, they can cooperate to have an evening out together ($\mathcal{K}_I \models^{\mathsf{k}} \langle 1,2\rangle(b_1 \wedge b_2) \wedge \langle 1,2\rangle(\neg b_1 \wedge \neg b_2)$), but this still involves a choice ($\mathcal{K}_I \not\models^{\mathsf{k}} \langle 1,2\rangle((b_1 \wedge b_2) \wedge (\neg b_1 \wedge \neg b_2))$).

Our next task is to formally establish that the two semantics are equivalent.

Remark 2.1 From now on, given a propositional model $\mathcal{M} = \langle Ag, \mathsf{At}, \mathsf{At}_1, \dots, \mathsf{At}_n, \theta\rangle$ such as defined in Definition 1, and a Kripke model $\mathcal{K} = \langle W, R_1, \dots, R_n, \Pi\rangle$ as in Definition 2 we assume they are based on the same partition $\mathsf{At}_1, \dots \mathsf{At}_n$. Moreover, we will write w_θ for that world in W for which $\Theta(w_\theta) = \theta$.

Lemma 2.4 (Equivalence of Semantics) Let the model $\mathcal{M} = \langle Ag, \mathsf{At}, \mathsf{At}_1, \dots, \mathsf{At}_n, \theta\rangle$ be as in Definition 1, and let the Kripke model $\mathcal{K} = \langle W, R_1, \dots, R_n, \Pi\rangle$ be as in Definition 2. Then, for every CL-PC formula φ, we have:

$$\mathcal{M} \models^{\mathsf{d}} \varphi \quad \text{iff} \quad \mathcal{K}, w_\theta \models^{\mathsf{k}} \varphi$$

Proof: We prove the displayed equivalence for *any* $\mathcal{M} = \langle Ag, \mathsf{At}, \mathsf{At}_1, \dots, \mathsf{At}_n, \theta\rangle$, the proof proceeding with induction on φ. For φ being an atom p this follows from the definition of θ and w_θ. Let us look at the diamond formula $\langle C\rangle\varphi$. Suppose $\mathcal{M} \models^{\mathsf{d}} \varphi$. This means that there is some valuation θ' and a model $\mathcal{M}' = \langle Ag, \mathsf{At}, \mathsf{At}_1, \dots, \mathsf{At}_n, \theta'\rangle$ such that $\mathcal{M}' \models^{\mathsf{d}} \varphi$. The induction hypothesis gives us $\mathcal{K}, w_{\theta'} \models^{\mathsf{k}} \varphi$, and, since by definition, θ' differs from θ in at most the atoms occurring in $\bigcup_{i \in C} \mathsf{At}_i$, we have $\mathcal{K}, w_{\theta'} \models^{\mathsf{k}} \langle C\rangle\varphi$. The other direction is proven in a similar way. $\square$

Corollary 2.2

$$\models^{\mathsf{d}}_I \varphi \text{ iff } \mathcal{K}_I \models^{\mathsf{k}} \varphi \quad \text{and} \quad \models^{\mathsf{d}} \varphi \text{ iff } \models^{\mathsf{k}} \varphi$$

How do the local, the semi-general I-based, and the global semantics compare? We start by giving the obvious dependencies, and then discuss the differences between the semantics, on the fly arguing that, in general, none of the implications below can be reversed. So, let $\mathcal{K}_I$ be a Kripke model with I a partition over At, and let w be a world. Then it is easy to see that we have the following chain of implications.

$$\models^{\mathsf{k}} \varphi \quad \Rightarrow \quad \mathcal{K}_I \models^{\mathsf{k}} \varphi \quad \Rightarrow \quad \mathcal{K}_I, w \models^{\mathsf{k}} \varphi \tag{3}$$

The last implication cannot be reversed, witnessed by $\varphi = p$ and $w(p) = \mathtt{tt}$. But objective formulas are not the only counterexamples: take w as before and assume that $p \notin \mathsf{At}_i$, $q \in \mathsf{At}_i$. Then we have that $\mathcal{K}_I, w \models^{\mathsf{k}} \langle i\rangle(p \wedge q)$, whereas, even under this fixed partition I, $\mathcal{K}_I \not\models^{\mathsf{k}} \langle i\rangle(p \wedge q)$. The first implication is also not an equivalence: take I such that $q \in \mathsf{At}_i$, then $\mathcal{K}_I \models \langle i\rangle q$, but obviously $\not\models^{\mathsf{k}} \langle i\rangle q$. It seems that the notion $\mathcal{K}_I \models^{\mathsf{k}} \varphi$ is the most interesting for our current aims: the fact that i can bring about $p \wedge q$ in w is too local a property, he would not be able to bring it about when in a situation in which p would not have been true.

3 The Deductive System CL-PC

In this section, we present an axiom system for CL-PC, and show that this axiom system is sound and complete with respect to our semantics. More precisely, we employ a canonical model construction to show that the axiom system is complete with respect to the Kripke semantics for CL-PC introduced in Section 2.4. We then appeal to the fact that the Kripke and direct semantics are equivalent (Lemma 2.4) to conclude that the axiomatization is complete with respect to both.

Definition 3 By ℓ, we denote a literal p or $\neg p$. If we want to explicitly refer to the the atom on which the literal is based, we write $\ell(p)$. Recall that an objective formula is one containing no cooperation modalities. Then the axioms of our logic CL-PC are as shown in Figure 2. Given a partition I, we will write $\vdash_I^{\text{CL-PC}} \varphi$ to denote that formula φ is derivable from these axioms and the following inference rules. We will often suppress the subscript I, though, and sometimes also the superscript CL-PC.

$$(MP) \quad \vdash^{\text{CL-PC}} \varphi, \quad \vdash^{\text{CL-PC}} (\varphi \rightarrow \psi) \quad \Rightarrow \quad \vdash^{\text{CL-PC}} \psi$$

$$Nec(i) \vdash^{\text{CL-PC}} \varphi \quad \Rightarrow \quad \vdash^{\text{CL-PC}} [i]\varphi$$

First, note that all the axioms apart from *Comp*-$\cup$ are about individual agents. We see below how the axioms can be "lifted" to arbitrary coalitions.

With respect to the soundness of axioms, we note that *Prop* is obvious, and $K(i)$ states that $[i]$ is a normal modal operator. Axiom $T(i)$ in contrapositive – note that the axioms are *schemes* – reads $(\varphi \rightarrow \langle i \rangle \varphi)$: every agent has the possibility to not change his atoms, leaving the state unchanged. $B(i)$ follows from the symmetry of the R_i relation [11, p.80]: starting in an arbitrary state verifying φ, any agent i cannot avoid to go to a state in which he can return to the original state $([i]\langle i \rangle \varphi)$. For $\emptyset$, note that $R_\emptyset$ is the identity: if nobody is allowed to change some atoms, we stay where we are. Axiom *at-least*$(control)$ states that every literal can be changed by at least one of the agents, and axiom *at-most*$(control)$ guarantees that no more than one agent can modify it.

Axiom *effect*(i) states that agents can make local changes to formulas: if ψ does not involve a particular atom p that is under control of agent i, then that agent can bring about to toggle p without modifying ψ, for any objective formula ψ. Although the soundness of this axiom is obvious for propositional formulas ψ, it is

$(Prop)$	$\varphi,$	where φ is any propositional tautology
$(K(i))$	$[i](\varphi \to \psi) \to ([i]\varphi \to [i]\psi)$	
$(T(i))$	$[i]\varphi \to \varphi$	
$(B(i))$	$\varphi \to [i]\langle i\rangle\varphi$	
$(empty)$	$[\emptyset]\varphi \leftrightarrow \varphi$	
$at\text{-}least(control)$	$\ell(p) \to \bigvee_{i\in Ag} \langle i\rangle\neg\ell(p)$	
$at\text{-}most(control)$	$\ell(p) \to (\langle i\rangle\neg\ell(p) \to [j]\ell(p))$	where $i \neq j$
$(effect(i))$	$(\psi \wedge \ell(p)) \to \langle i\rangle(\psi \wedge \neg\ell(p))$	where $\begin{cases} p \in \mathsf{At}_i, \\ p \notin \mathsf{At}(\psi), \text{ and} \\ \psi \text{ is objective} \end{cases}$
$(non\text{-}effect(i))$	$\langle i\rangle\ell(p) \to [i]\ell(p)$	where $p \notin \mathsf{At}_i$
$(Comp\text{-}\cup)$	$[C_1][C_2]\varphi \leftrightarrow [C_1 \cup C_2]\varphi$	

Fig. 2. Axioms of CL-PC.

not immediately clear for general formulas φ, although this appears to be derivable, as we will see in Lemma 3.4. The *non-effect*(i)-axiom says that no agent i can change the literals based on atoms outside his control: if $p \notin \mathsf{At}(i)$ and $\langle i\rangle\ell(p)$ is true, then $\ell(p)$ must be true no matter how i toggles his atoms, and in particular, $\ell(p)$ must already be true now (by virtue of $T(i)$).

Finally, observe that Axiom *Comp*-$\cup$ is equivalent to $\langle C_1\rangle\langle C_2\rangle\varphi \leftrightarrow \langle C_1 \cup C_2\rangle\varphi$. This axiom is sound in the Kripke interpretation: if, given the interpretation θ, agent 1 can make a transition to θ_1, from which agent 2 can choose the state to be θ_2, then, equivalently, from θ, agents 1 and 2 can choose values for their atoms so that we end up in θ_2. The converse also holds: a change in valuation due to the coalition $C_1 \cup C_2$ can always be decomposed in a change by C_1 and one by C_2. (Schema *Comp*-$\cup$ is so named for its close relationship to the axiom *Comp* from dynamic logic [18, p.88] [2] .)

Formally, we state without proof:

Lemma 3.1 (Soundness) The system CL-PC is sound with respect to both the CL-PC semantics of Definition 1 and the Kripke semantics of Definition 2.

[2] The *Comp* axiom from dynamic logic axiom is $\langle p_1; p_2\rangle\varphi \leftrightarrow \langle p_1\rangle\langle p_2\rangle\varphi$.

$(K(C))$	$[C](\varphi \to \psi) \to ([C]\varphi \to [C]\psi)$	
$(T(C))$	$[C]\varphi \to \varphi$	
$(4(C))$	$[C]\varphi \to [C][C]\varphi$	
$(5(C))$	$\neg[C]\varphi \to [C]\neg[C]\varphi$	
$(\mathit{effect}(C))$	$(\varphi \wedge \ell(p)) \to \langle C\rangle(\varphi \wedge \neg\ell(p))$	where $\begin{cases} p \in \mathsf{At}_C, \text{ and} \\ p \notin \mathsf{At}(\varphi) \end{cases}$
$(\mathit{non\text{-}effect}(C))$	$(p \to [C]p) \wedge (\neg p \to [C]\neg p)$	where $p \notin \bigcup_{i \in C} \mathsf{At}_i$
(Sub)	$[\{i\} \cup C]\varphi \to [i]\varphi$	
$\mathit{Alt}(i)$	$(\varphi \wedge \ell(p)) \to \langle i\rangle(\varphi \wedge \ell(p)) \wedge \langle i\rangle(\varphi \wedge \neg\ell(p))$	where $\begin{cases} p \in \mathsf{At}_i, \text{ and} \\ p \notin \mathsf{At}(\varphi) \end{cases}$
(S')	$\langle C_1\rangle\varphi \wedge \langle C_2\rangle\psi \to \langle C_1 \cup C_2\rangle(\varphi \wedge \psi)$	where $\mathsf{At}(\varphi) \cap \mathsf{At}(\psi) = \emptyset$

Fig. 3. Some derived axiom schemes.

The following lemma shows a number of derivable formulas. It demonstrates that the individual agent properties of the logic CL-PC can be lifted to the collective level, and, moreover, property S' shows when agents can concurrently exercise their powers.

Lemma 3.2 (Derived schemes) The schemes in Figure 3 may be derived in the deductive system CL-PC (C is an arbitrary set of agents).

Proof: Follow from the semantics, once we have proven completeness. As an illustration of a derivation, however, in Appendix A we prove (S') within CL-PC. ▫

Schema (S') is the closest we have in CL-PC to Pauly's schema (S), which intuitively said that disjoint coalitions could add abilities (i.e., for disjoint C_1 and C_2, if C_1 can achieve φ and C_2 can achieve ψ, then $C_1 \cup C_2$ can achieve $\varphi \wedge \psi$). The direct CL-PC analogue of Pauly's (S), i.e., $\langle C_1\rangle\varphi \wedge \langle C_2\rangle\psi \to \langle C_1 \cup C_2\rangle(\varphi \wedge \psi)$ for disjoint C_1, C_2, is not valid, as the following example illustrates.

Example 3.1 Suppose that we have a model $\mathcal{M}$ with $Ag = \{1, 2\}$, $\mathsf{At}_1 = \{p\}$, and $\theta(p) = \mathtt{tt}$. Then clearly $\mathcal{M} \models^{\mathsf{d}} (\langle 1\rangle\neg p) \wedge (\langle 2\rangle p)$. But if $\langle C_1\rangle\varphi \wedge \langle C_2\rangle\psi \to \langle C_1 \cup C_2\rangle(\varphi \wedge \psi)$ were valid, then we would have $\mathcal{M} \models^{\mathsf{d}} \langle 1, 2\rangle(p \wedge \neg p)$, and so $\mathcal{M} \models^{\mathsf{d}} \langle 1, 2\rangle\bot$ and hence $\mathcal{M} \models^{\mathsf{d}} \bot$.

Instead, we have the weaker scheme (S'): this scheme says that, if C_1 can achieve φ, and C_2 can achieve ψ, and φ *and* ψ *have no propositional variables in common*, then $C_1 \cup C_2$ can achieve $\varphi \wedge \psi$. Note that, in contrast to Pauly's schema (S), while this schema requires that the formulae φ and ψ are disjoint with respect to their propositional variables, the coalitions C_1 and C_2 are *not* required to be disjoint.

3.1 Completeness

We are now ready to prove that the axiom system CL-PC is complete with respect to the two semantics presented earlier. We begin by fixing some notation.

Definition 4 (Notation) In the following, we use c as a variable both over individual agents $i \in Ag$ and coalitions $C \subseteq Ag$. If such a c is fixed, we want to reason about what it can achieve, and what not. For this, we partition a set of relevant atoms (which can be At, or At_c or $\mathsf{At}(\varphi)$ for a specific φ) in a subset X under control of c, say $X = \{x_1, \ldots x_k\}$ and a set $Y = \{y_1, \ldots, y_m\}$ out of control of c. Note that the x's and y's are variables over atoms: when changing focus to another coalition c' we will use the sets X and Y again, but of course then $X \subseteq \mathsf{At}(c')$. Note that in order to not obscure notation too much, we do not make c explicit in this notation, nor do we say how k and m depend on c or φ. We also will write a propositional formula ψ in a special Disjunctive Normal Form ($DNF(\psi, c)$), where $DNF(\psi, c)$ = $\pi_1 \vee \pi_2 \vee \cdots \vee \pi_d$, each π_j being a conjunction of literals over *all the atoms occurring in* ψ, and ordered along the sets $\mathsf{At}(\psi) \cap \mathsf{At}(c)$ and $\mathsf{At}(\psi) \setminus \mathsf{At}(c)$. Then we can write any π_j as

$$\pi_j \equiv \pi_j^{\hat{c}} \wedge \pi_j^{\check{c}} \equiv \bigwedge_{g \leq k} \ell_j(x_g) \wedge \bigwedge_{h \leq m} \ell_j(y_h)$$

Given a conjunction π, and a coalition or agent c, we will hence write $\pi^{\check{\imath}}$ for the conjunction of literals obtained from π by leaving out the x's, and $\pi^{\hat{\imath}}$ for the conjunction of π without the y's. Observe that hence we can interpret $\pi^{\hat{\imath}}$ as that part of π that is under control of c, and $\pi^{\check{\imath}}$ as the part out of c's control. Sometimes, we will also use $dnf(\psi)$, which is also a disjunctive normal form, but not necessarily containing all of ψ's atoms. For instance, if ψ is the formula $(p_1 \wedge q) \wedge (p_2 \vee \neg p_2)$ $(p_1, p_2 \in \mathsf{At}(C), q \notin \mathsf{At}(C))$, then $DNF(\psi, C) = (p_1 \wedge p_2 \wedge q) \vee (p_1 \wedge \neg p_2 \wedge q)$ while $dnf(\psi) = (p_1 \wedge q)$.

The first main component of our completeness proof is Theorem 3.1. This result states that every CL-PC formula is equivalent (under either semantics) to a formula of propositional logic. (As an aside, note that this result does not imply that the modal language of CL-PC is redundant, since the translation from CL-PC involves an exponential blow-up in the size of the formula.) We prove Theorem 3.1 by way of two lemmas, which essentially show us how cooperation modalities can be eliminated from formulae, while preserving their truth.

Lemma 3.3 Using the notation as introduced in Definition 4, let c be an agent or coalition, and let $\pi \equiv \pi^{\hat{c}} \wedge \pi^{\check{c}}$ also as specified there. Then:

$$\vdash^{\text{CL-PC}} \pi^{\check{c}} \leftrightarrow \langle c\rangle(\pi^{\hat{c}} \wedge \pi^{\check{c}})$$

At first sight, it might seem strange that the equivalence in Lemma 3.3 has atoms on the right hand side which do not occur in the left hand side. The idea is simple though, and will be used in many proof-steps that are to follow. Let us, given agent i, loosely refer to $\pi^{\check{\imath}}$ and $\pi^{\hat{\imath}}$ as literals in π that are *out*, or *in* the agent's control, respectively. Then, i can achieve any combination of literals, as long as those out his control are already set in the right way. In an example: if i controls p_i and q_i but not p or q, then i can bring about $(p_1 \wedge \neg p_2 \wedge p \wedge \neg q)$ in exactly those situations in which $(p \wedge \neg q)$ is already true. For coalitions C, this works similarly.

Proof: We first prove the displayed formula for the case that c is an agent i. For the right to left direction, note that for any atom $y \notin \mathsf{At}_i$, we have

$$\vdash^{\text{CL-PC}} \langle i\rangle\ell(y) \;\rightarrow \ell(y) \qquad y \notin \mathsf{At}_i \tag{4}$$

One uses first *non-effect*(i) and then $T(i)$ to see this. Then we derive $(a) - (c)$:

$$(a) : \langle i\rangle(\pi^{\hat{\imath}} \wedge \pi^{\check{\imath}}) \rightarrow \langle i\rangle\pi^{\check{\imath}}$$

$$(b) : \langle i\rangle\pi^{\check{\imath}} \rightarrow \langle i\rangle \textstyle\bigwedge_{h\leq m} \ell(y_h)$$

$$(c) : \langle i\rangle \textstyle\bigwedge_{h\leq m} \ell(y_h) \rightarrow \bigwedge_{h\leq m} \ell(y_h)$$

Implication (a) is a property of any modal diamond, (b) uses the definition of $\pi^{\check{\imath}}$ and (c) applies (4). Since $\bigwedge_{h\leq m} \ell(y_h)$ is by definition $\pi^{\check{\imath}}$, this proves the derivability of $\langle i\rangle(\pi^{\hat{\imath}} \wedge \pi^{\check{\imath}}) \rightarrow \pi^{\check{\imath}}$.

For the other direction, first note that for any y not in At_i, we have

$$\vdash^{\text{CL-PC}} \ell(y) \rightarrow [i]\ell(y) \tag{5}$$

which follows immediately from (4) and the fact that we are dealing with literals. By using modal logical laws, we then derive $\vdash^{\text{CL-PC}} \bigwedge_{h\leq m} \ell(y_h) \;\rightarrow\; [i] \bigwedge_{h\leq m} \ell(y_h)$ and hence

$$\vdash^{\text{CL-PC}} \pi^{\check{\imath}} \rightarrow [i]\pi^{\check{\imath}} \tag{6}$$

Now let us write π as $\pi^{\check{\imath}} \wedge \pi^{\hat{\imath}} = \pi^{\check{\imath}} \wedge (\ell(x_1) \wedge \ell(x_2) \wedge \cdots \wedge \ell(x_k))$. By definition of $\pi^{\hat{\imath}}$ and $\pi^{\check{\imath}}$, we have $x_1 \notin \mathsf{At}(\pi^{\check{\imath}})$. Also, by axiom $T(i)$,

$$\vdash^{\text{CL-PC}} (\pi^{\check{\imath}} \wedge \ell(x_1)) \rightarrow \langle i\rangle(\pi^{\check{\imath}} \wedge \ell(x_1)) \tag{7}$$

and, by *effect*(i),

$$\vdash^{\text{CL-PC}} (\pi^{\check{\imath}} \wedge \neg\ell(x_1)) \rightarrow \langle i\rangle(\pi^{\check{\imath}} \wedge \ell(x_1)) \tag{8}$$

Since $\pi^{\check{i}}$ is equivalent to $(\pi^{\check{i}} \wedge \ell(x_1)) \vee (\pi^{\check{i}} \wedge \neg\ell(x_1))$, we obtain from (7) and (8):

$$\vdash^{\text{CL-PC}} \pi^{\check{i}} \to \langle i \rangle (\pi^{\check{i}} \wedge \ell(x_1)) \tag{9}$$

Repeating the argument (7) - (9) we also obtain

$$\vdash^{\text{CL-PC}} (\pi^{\check{i}} \wedge \ell(x_1)) \to \langle i \rangle (\pi^{\check{i}} \wedge \ell(x_1) \wedge \ell(x_2)) \tag{10}$$

Now note that (9) and (10) are of the form $\vdash_m \pi^{\check{i}} \to \Diamond\varphi_1$ and $\vdash_m \varphi_1 \to \Diamond\varphi_2$, respectively, in which in any modal logic m one concludes $\vdash_m \pi^{\check{i}} \to \Diamond\Diamond\varphi_2$. But in CL-PC we also have $\langle i \rangle\langle i \rangle\varphi \to \langle i \rangle\varphi$, so that we conclude, from (9) and (10),

$$\vdash^{\text{CL-PC}} \pi^{\check{i}} \to \langle i \rangle (\pi^{\check{i}} \wedge \ell(x_1) \wedge \ell(x_2)) \tag{11}$$

Repeating the argument from (7) - (11) sufficiently many times, we conclude that $\vdash^{\text{CL-PC}} \pi^{\check{i}} \to \langle i \rangle (\ell(x_1) \wedge \ell(x_2) \wedge \cdots \wedge \ell(x_k)$, or, equivalently, $\vdash^{\text{CL-PC}} \pi^{\check{i}} \to \langle i \rangle \pi^{\hat{i}}$. We combine this with (6) to finally obtain $\vdash^{\text{CL-PC}} \pi^{\check{i}} \to \langle i \rangle (\pi^{\hat{i}} \wedge \pi^{\check{i}})$.

Given that we now have proven what we want for c being an agent i, let us now assume that c is a coalition C. Choose an arbitrary order of the members of C, say $i_v, i_{v-1}, \ldots, i_2, i_1$. Let the initial conjunction be π. Axiom *Comp*-$\cup$ tells us that

$$\vdash^{\text{CL-PC}} \langle C \rangle \pi \leftrightarrow \langle i_v \rangle \langle i_{v-1} \rangle \cdots \langle i_2 \rangle \langle i_1 \rangle \pi \tag{12}$$

We now iteratively do the following. We construct a sequence of conjunctions $\pi_1, \ldots \pi_v$ such that every π_{j+1} is π_j with the literals over atoms from At_j left out. Let π_1 be π. The result for the single agent case tells us that $\vdash^{\text{CL-PC}} \pi_1^{\check{i_1}} \leftrightarrow \langle i_1 \rangle (\pi_1^{\hat{i_1}} \wedge \pi_1^{\check{i_1}})$, i.e., $\vdash^{\text{CL-PC}} \pi_1^{\check{i_1}} \leftrightarrow \langle i_1 \rangle \pi_1$. Using this equivalence in (12), we obtain

$$\vdash^{\text{CL-PC}} \langle C \rangle \pi \leftrightarrow \langle i_v \rangle \langle i_{v-1} \rangle \cdots \langle i_2 \rangle \pi_1^{\check{i_1}} \tag{13}$$

Note that in in (13), the formula $\pi_1^{\check{i_1}}$ is the conjunction π_i with all the literals over atoms from At_{i_1} left out. Let π_2 be the formula $\pi_1^{\check{i_1}}$. The single agent case again tells us that $\vdash^{\text{CL-PC}} \pi_2^{\check{i_2}} \leftrightarrow \langle i_2 \rangle \pi_2$ giving

$$\vdash^{\text{CL-PC}} \langle C \rangle \pi \leftrightarrow \langle i_v \rangle \langle i_{v-1} \rangle \cdots \langle i_3 \rangle \pi_2^{\check{i_2}} \tag{14}$$

where $\pi_2^{\check{i_2}}$ is the conjunction π_1 with the literals for agents i_1 and i_2 left out. We repeat this argument v times to eventually conclude

$$\vdash^{\text{CL-PC}} \langle C \rangle \pi \leftrightarrow \pi_v^{\check{i_v}} \tag{15}$$

where $\pi_v^{\check{i_v}}$ is the formula π_1 with all the literals involving any of C's atoms, left out. In order words, (15) says $\vdash^{\text{CL-PC}} \langle C \rangle \pi \leftrightarrow \pi^{\check{c}}$ □

Lemma 3.4 Let C be a coalition of agents, and ψ a propositional formula. Then there exist a propositional formula α such that

$$\vdash_{cl\text{-}pc} \alpha \leftrightarrow \langle C\rangle\psi \text{ and } \mathsf{At}(\alpha) \cap \mathsf{At}_C = \emptyset$$

Proof: We know that ψ is propositional, so let us use the notation introduced in Definition 4, and assume that $DNF(\psi, C) = \pi_1 \vee \cdots \vee \pi_d$ and that $X = \mathsf{At}(\psi) \cap \mathsf{At}_C$, and $Y = \mathsf{At}(\psi) \setminus X$. As in any modal logic, we have

$$\vdash_{\text{CL-PC}} \langle C\rangle \bigvee_{j \leq d} \pi_j \leftrightarrow \bigvee_{j \leq d} \langle C\rangle\pi_j \tag{$*$}$$

By Lemma 3.3, we moreover have

$$\vdash^{\text{CL-PC}} \langle C\rangle\pi_j \leftrightarrow \pi_j^{\check{C}} \tag{$**$}$$

Note that the right hand side of (**) is purely propositional. Combining (*) and (**), we have $\vdash^{\text{CL-PC}} \alpha \leftrightarrow \langle C\rangle\psi$ where α is the formula $\bigvee_{j\leq d} \pi_j^{\check{C}}$. Clearly, by definition of $\pi_j^{\check{C}}$, $\mathsf{At}(\alpha) \cap \mathsf{At}_C = \emptyset$. $\square$

Example 3.2 Let us consider the formula $\langle i\rangle\psi$ being $\langle i\rangle(p_1 \wedge (p_2 \rightarrow p_4))$. Bringing ψ in dnf gives $\langle i\rangle((p_1 \wedge \neg p_2 \wedge \neg p_4) \vee (p_1 \wedge \neg p_2 \wedge p_4) \vee (p_1 \wedge p_2 \wedge p_4))$. We consider the following cases:

(1) $\mathsf{At}_i = \{p_1, p_2, p_3, p_4\}$. There are no atoms occurring in ψ that are not under the control of agent i, so we get for α a disjunction of three occurrences of $\top$, which equals $\top$, so that our agent can always bring about ψ.
(2) $\mathsf{At}_i = \{p_1, p_2\}$. Agent i can only toggle the atoms p_1 and p_2, so the states in which he can bring about ψ are represented by $\neg p_4 \vee p_4 \vee p_4$ which is equivalent to true: agent i can always bring about ψ!
(3) $\mathsf{At}_i = \{p_2\}$. Then he can bring about ψ in those worlds satisfying $(p_1 \wedge \neg p_4) \vee (p_1 \wedge p_4) \vee (p_1 \wedge p_4)$, which is equivalent to p_1.
(4) $\mathsf{At}_i = \{p_3\}$. Then i can bring about ψ only in those situations in which ψ is already true (i.e., $\alpha = \psi$), which is as expected.

Note that Lemma 3.4 implies that every formula $[C]\psi$ is also equivalent to a propositional formula. Let X and Y again be the atoms in and out of control of C, respectively. This time, write ψ in Conjunctive Normal Form CNF(ψ, C) = $\delta_1 \wedge \cdots \wedge \delta_d$, each δ_j of the form $\delta_j^{\hat{C}} \vee \delta_j^{\check{C}} \equiv \bigvee_{g\leq k} \ell_j(x_g) \vee \bigvee_{h\leq m} \ell_j(y_h)$. We then get that $[C]\psi \equiv \bigwedge_{j\leq d} \delta_j^{\check{C}}$. As an example, suppose that CNF(ψ, C) = $(p_1 \vee \neg p_2 \vee q_1 \vee \neg q_2) \wedge (p_1 \vee p_2 \vee \neg q_1 \vee q_2)$, and that $p_1, p_2 \in \mathsf{At}(C)$, but $q_1, q_2 \notin \mathsf{At}(C)$. Then C cannot avoid that ψ is true in those cases where both $(q_1 \vee \neg q_2)$ and $(\neg q_1 \vee q_2)$ are true.

In fact, Lemma 3.4 gives us a constructive way to find the propositional formula α, given the coalition C and $DNF(\psi, C) = \pi_i \vee \cdots \vee \pi_k$ for ψ. Let us, for future reference, define this α as $F(C, \psi)$.

Theorem 3.1 (Normal Forms)
Every formula φ is equivalent in CL-PC *to a propositional formula.*

Proof: Consider the following translation $\mathcal{T}$.

$$\begin{aligned}
\mathcal{T}(\top) &= \top \\
\mathcal{T}(p) &= p \\
\mathcal{T}(\neg\varphi) &= \neg\mathcal{T}(\varphi) \\
\mathcal{T}(\varphi \wedge \psi) &= \mathcal{T}(\varphi) \wedge \mathcal{T}(\psi) \\
\mathcal{T}(\langle\!\langle C \rangle\!\rangle\varphi) &= F(C, \mathcal{T}(\varphi))
\end{aligned}$$

□

From the definition of $\mathcal{T}$ and Lemma 3.4, it is clear that φ is equivalent to $\mathcal{T}(\varphi)$. Moreover, it is easy to see that $\mathcal{T}(\varphi)$ is a propositional formula.

Example 3.3 Again, we consider the formula φ which in dnf is $((p_1 \wedge \neg p_2 \wedge \neg p_4) \vee (p_1 \wedge \neg p_2 \wedge p_4) \vee (p_1 \wedge p_2 \wedge p_4))$. We are interested in finding the propositional equivalent of $\langle 1 \rangle\langle 2 \rangle\varphi$. We consider the following cases:

(1) $\mathsf{At}_2 = \{p_1, p_2, p_4\}$, $\mathsf{At}_1 = \{p_3\}$. There are no atoms occurring in φ that are not under control of agent 2, so we get as the translation of $\langle 2 \rangle\varphi$ the formula $\top$, and the translation of $\langle 1 \rangle\top$ is $\top$.
(2) $\mathsf{At}_2 = \{p_1, p_2\}$. We know from Example 3.2 that $\mathcal{T}(\langle 2 \rangle\varphi) = \top$, giving also $\top$ for $\mathcal{T}(\langle 1 \rangle\langle 2 \rangle\varphi)$.
(3) $\mathsf{At}_2 = \{p_2\}$. Then 2 can bring about φ in those worlds satisfying p_1 (see Example 3.2). So, in case $p_1 \in \mathsf{At}_1$, we have that $\langle 1 \rangle\langle 2 \rangle\varphi \equiv \top$, saying that the coalition $\{1, 2\}$ can always bring about φ, but if $p_1 \not\in \mathsf{At}_1$, the formula $\langle 1 \rangle\langle 2 \rangle\varphi$ is equivalent to p_1, indicating all situations in which $\{1, 2\}$ can bring about φ.
(4) $\mathsf{At}_2 = \{p_3\}$. Then i can bring about φ only in those situations in which φ is already true, and we can use the analysis of Example 3.2 to reason about $\langle 1 \rangle\langle 2 \rangle\varphi$, since it is equal to $\langle 1 \rangle\varphi$.

Corollary 3.1 Let φ be an arbitrary formula, and $\mathcal{M}_I$ a propositional model based on a partition I. Then: $\mathcal{M}_I \models^{\mathsf{d}} [Ag]\varphi$ iff there is a propositional formula ψ for which $\models^{\mathsf{d}}_I \varphi \leftrightarrow \psi$ and ψ being a propositional tautology.

We can now move on to the main stage of our proof. We employ a canonical model construction, and assume some familiarity with this approach in what follows[3]: in particular, we assume some understanding of the standard definition of maximal consistent sets, Lindenbaum's lemma [18, pp.18–20], the use of these in the

[3] See [11, pp.59–61], [9, pp.196–201], or [18, pp.22–25] for introductions.

canonical model construction [18, pp.22–23].

Definition 5 (Canonical model) We define the canonical model $\mathcal{K}^{can}$ for CL-PC as $\mathcal{K}^{can} = \langle W^{can}, R_1^{can}, \ldots, R_n^{can}, \Theta^{can} \rangle$, where:

- W^{can} is the set of CL-PC maximal consistent sets;
- $R_i^{can}\Gamma\Delta$ iff for all φ: $\varphi \in \Delta \Rightarrow \langle i \rangle \varphi \in \Gamma$.
- $\Theta^{can}(\Gamma)(p) = \mathtt{tt}$ iff $p \in \Gamma$.

Lemma 3.5 The canonical model $\mathcal{K}^{can}$ defined in Definition 5 is isomorphic to the Kripke model $\mathcal{K}$ of Definition 2. More precisely, there exists a bijective function $f : W^{can} \to W$ such that:

(1) for all $\Gamma \in W^{can}$ and $p \in \mathsf{At}$: $\Theta^{can}(\Gamma)(p) = \mathtt{tt}$ iff $\Theta(f(\Gamma))(p) = \mathtt{tt}$
(2) $R_i^{can}\Gamma\Delta$ iff $R_i f(\Gamma) f(\Delta)$

Proof: Theorem 3.1 tells us that any CL-PC formula is equivalent to a propositional formula. Since we assumed a finite number of propositional atoms, there are finitely many propositional formulas. It follows that for world $\Gamma \in W^{can}$ there is a corresponding world w_Γ in the Kripke model $\mathcal{K}$ which exactly satisfies the literals in Γ. It is also the case that for every valuation w_Γ in $\mathcal{K}$, there is a corresponding world Γ in W^{can}. Let us call this isomorphism f: $W^{can} \to W$, where W^{can} is the domain of the canonical model, and W is the domain of the Kripke model $\mathcal{K}$, as defined in Definition 2. Define it formally as

$$f(\Gamma) = w_\Gamma, \text{ for the unique } w_\Gamma \text{ such that for all } p\ (p \in \Gamma \Leftrightarrow w_\Gamma(p) = \mathtt{tt})$$

This is a bijection: first of all, every world w is the image of some $\Gamma \in W^{can}$. Moreover, suppose that $\Gamma \neq \Delta$. To show that $f(\Gamma) \neq f(\Delta)$, suppose the latter is not the case: $f(\Gamma)$ and $f(\Delta)$ coincide. This means that Γ and Δ contain exactly the same literals, and, since they are maximal consistent sets, also the same objective formulas. But, by Theorem 3.1 then, they contain the same LC-PC formulas, i.e., $\Gamma = \Delta$, contradicting our initial assumption, hence $f(\Gamma) \neq f(\Delta)$.

It is obvious, by construction of f, that Θ^{can} and Θ agree on all atoms. This proves item 1.

Now we have to show that the definition of R_i^{can} in the canonical model corresponds to that of R_i in the Kripke model $\mathcal{K}$, or, more precisely, that

$$R_i{}^{can}\Gamma\Delta \quad \text{iff} \quad [f(\Gamma) = f(\Delta) \pmod{\mathsf{At}_i}]$$

To prove the displayed equivalence, suppose first that that $R_i^{can}\Gamma\Delta$, and, to derive a contradiction, that $f(\Gamma) \neq f(\Delta) \pmod{\mathsf{At}_i}$. The latter means that there is an atom $p \in (\mathsf{At} \setminus \mathsf{At}_i)$, such that for some literal $\ell(p)$, we have $\ell(p)$ being false under the valuation $f(\Gamma)$, but $\ell(p)$ being true under $f(\Delta)$. By the first item of this Lemma, we know that then $\ell(p) \notin \Gamma, \ell(p) \in \Delta$. The definition of R_i^{can} then yields that $\langle i \rangle \ell(p) \in \Gamma$. But this is impossible, since $\langle i \rangle \ell(p) \in \Gamma$ implies (use *non-effect*(i)) $[i]\ell(p) \in \Gamma$, which, by $T(i)$, implies $\ell(p) \in \Gamma$, in contradiction with what was derived earlier.

For the other direction, assume that $[f(\Gamma) = f(\Delta) \pmod{\mathsf{At}_i}]$. Let φ be an arbitrary formula in Δ: we have to show that $\langle i \rangle \varphi$ in Γ. Let π be the conjunction of all the literals in Δ. This is the strongest propositional formula in Δ, and, by Theorem 3.1, the strongest formula in it. Hence we have $\vdash_{cl\text{-}pc} \pi \rightarrow \varphi$. We can write π as a conjunction $(\pi^{\hat{\imath}} \wedge \pi^{\check{\imath}})$, in which $\pi^{\hat{\imath}}$ is the conjunction over all $\ell(p)$ with $p \in \mathsf{At}_i$, and $\pi^{\check{\imath}}$ a conjunction of $\ell(p_i)$, with $p_i \notin \mathsf{At}_i$. Since $f(\Gamma) = f(\Delta) \pmod{\mathsf{At}_i}$, we have that $\pi^{\check{\imath}} \in \Gamma$. By Lemma 3.3, we know that $\vdash_{cl\text{-}pc} \pi^{\check{\imath}} \rightarrow \langle i \rangle(\pi^{\hat{\imath}} \wedge \pi^{\check{\imath}})$, and hence $\langle i \rangle(\pi^{\hat{\imath}} \wedge \pi^{\check{\imath}}) \in \Gamma$. But then, by definition of π, we have $\langle i \rangle \pi \in \Gamma$ and hence $\langle i \rangle \varphi \in \Gamma$.

This completes the proof that the models $\mathcal{K}$ and $\mathcal{K}^{can}$ are isomorphic, and hence verify the same formulas. ▫

We now come to our main completeness result. (For readability, we suppress the set of agents Ag, the set of atoms At and the partition I of At over Ag, which should be the same in all cases.)

Theorem 3.2 (Completeness) The following are equivalent, for any formula φ.

(*i*) φ is CL-PC-consistent;
(*ii*) φ is true in one of the worlds of the canonical model $\mathcal{K}^{can}$ (see Definition 5);
(*iii*) φ is true in one of the worlds of $\mathcal{K}$ (see Definition 2);
(*iv*) φ is true in some propositional model $\mathcal{M}$ (see Definition 1).

Proof: Let φ be consistent. Then it is contained in a maximal consistent set Γ. The construction of the canonical model is standard, as is the proof of its coincidence property: $\mathcal{K}^{can}, \Delta \models \psi$ iff $\varphi \in \Delta$ (see, e.g., [9, pp.196–201]). From this, it follows that $\mathcal{K}^{can}, \Gamma \models \varphi$, and hence this proves (*i*) implies (*ii*). The other direction follows from Lemma 3.1. Lemma 3.5 states that items (*ii*) and (*iii*) are equivalent, and the equivalence of (*iii*) and (*iv*) is guaranteed by Lemma 2.4. ▫

4 Varieties of Ability and Control

In the introduction to this paper, we claimed that contingent ability, as captured in our "$\langle \cdots \rangle$" cooperation modalities, was sufficient to define other, richer (and arguably more realistic) types of ability and control. Our aim in this section is to justify this claim. We begin by showing how contingent ability may be used to define α- and β-ability modalities. In Section 4.2, we go on to show how the notion of *control* may be characterised in CL-PC.

4.1 Characterising Alpha and Beta Ability

As we noted above, the cooperation modality $\langle \cdots \rangle$ is not *strategic*. That is, a formula $\langle C \rangle \varphi$ simply means that the coalition C have some choice available to them, such that if they make this choice *and nothing else changes*, then φ will be true. But of course, multiagent encounters are typically *strategic* in nature, where the decision that one agent should make depends on the decisions that other agents may make. In this section, we show how it is possible to define *strategic* cooperation modalities in terms of CL-PC constructs.

In the study of coalitional ability, the most common form of strategic capability studied is known as *α-ability*, or "$\exists\forall$"-capability (see, e.g., [37, pp.311–312]). Intuitively, a coalition C is said to have α-ability for some proposition φ if there exists a choice σ_C for C such that, for all choices $\sigma_{\overline{C}}$ for $\overline{C}$, if C make choice σ_C and $\overline{C}$ make choice $\sigma_{\overline{C}}$, then φ will be true. (Note the "$\exists\forall$" pattern of quantifiers in this definition.) In other words, C having the α-ability to bring about φ means that C have a choice such that, *no matter what the other agents do*, if they make this choice, then φ will be true. This notion of cooperative ability typically forms the semantic basis of cooperation modalities [36].

Implicit within the notion of α-ability is the fact that C have *no knowledge* of the choice that the other agents make; they are not allowed to see the choice of $\overline{C}$ and then decide what to do, but rather they must make their decision *first*. This motivates the notion of β-ability (i.e., "$\forall\exists$"-ability): coalition C is said to have the β-ability for φ if for every choice $\sigma_{\overline{C}}$ available to $\overline{C}$, there exists a choice σ_C for C such that if $\overline{C}$ choose $\sigma_{\overline{C}}$ and C choose σ_C, then φ will result. Thus C being β-able to φ means that no matter what the other agents do, C have a choice such that, if they make this choice, then φ will be true. Note the "$\forall\exists$" pattern of quantifiers: C are implicitly allowed to make their choice while being aware of the choice made by $\overline{C}$.

It is easy to see that both α- and β-ability represent stronger notions of ability than that in the basic cooperation modalities of CL-PC. But using the cooperation modalities of CL-PC, we can in fact define coalition modalities "$\langle\langle \cdots \rangle\rangle_\alpha$" and "$\langle\langle \cdots \rangle\rangle_\beta$" capturing α and β-ability respectively, and of course their duals, "$[\![\cdots]\!]_\alpha$" and "$[\![\cdots]\!]_\beta$".

$$\begin{aligned}
\langle\!\langle C\rangle\!\rangle_\alpha\, \varphi &\mathrel{\hat{=}} \langle C\rangle[\overline{C}]\varphi\\
\langle\!\langle C\rangle\!\rangle_\beta\, \varphi &\mathrel{\hat{=}} [\overline{C}]\langle C\rangle\varphi\\
[\![C]\!]_\alpha\varphi &\mathrel{\hat{=}} \neg\langle\!\langle C\rangle\!\rangle_\alpha\, \neg\varphi\\
[\![C]\!]_\beta\varphi &\mathrel{\hat{=}} \neg\langle\!\langle C\rangle\!\rangle_\beta\, \neg\varphi
\end{aligned}$$

As an aside, the double bracket notation, "$\langle\!\langle\cdots\rangle\!\rangle$" it motivated by its use in ATL [5]: the derived modality $\langle\!\langle\cdots\rangle\!\rangle_\alpha$ is very close to the ATL "next" cooperation modality, written $\langle\!\langle\cdots\rangle\!\rangle\bigcirc$.

One may wonder whether definitions like those given above cannot be used in arbitrary coalition logics, such ATL. However, the fact that we can "reduce" α- and β-ability to the contingent ability is due to the fact that our system is basically *asynchronous*. That is, the effect of the agents' choices are independent of each other – the effect of a group's decision is exactly the sum of the effects of its individuals. Expressed more precisely: for any coalition $C = \{c_1, \ldots, c_k\}$, any ability formula $\langle C\rangle\varphi$ can always be rewritten as $\langle c_1\rangle\langle c_2\rangle \ldots \langle c_k\rangle\varphi$, expressing that the actions of the agents do not interfere with each other, and group actions can always be unravelled as a sequence of individual actions.

Let us consider the relationships between the various types of ability introduced in this paper. First, note that we have the following equivalence.

$$\begin{aligned}
\langle\!\langle C\rangle\!\rangle_\beta\, \varphi &\mathrel{\hat{=}} [\overline{C}]\langle C\rangle\varphi\\
&\leftrightarrow [\overline{C}][C]\langle C\rangle\varphi\\
&\leftrightarrow [Ag]\langle C\rangle\varphi
\end{aligned}$$

Secondly, (cf. $T(C)$), we have both $\models^{\mathsf{d}} \langle\!\langle C\rangle\!\rangle_\alpha\, \varphi \to \langle C\rangle\varphi$ and $\models^{\mathsf{d}} \langle\!\langle C\rangle\!\rangle_\beta\, \varphi \to \langle C\rangle\varphi$.

At the global level, β-ability and contingent ability collapse: saying that, for a given partition of atoms among agents, the coalition C have the β-ability to achieve φ is the same as saying that they have the contingent ability to achieve φ in every situation based on this partition. More formally, we have the following equivalence.

$$\models^{\mathsf{d}}_I \langle\!\langle C\rangle\!\rangle_\beta\, \varphi \quad\Leftrightarrow\quad \models^{\mathsf{d}}_I \langle C\rangle\varphi \tag{16}$$

However, note that α- and β-ability *are* distinct at the global level. Formally, we have the following.

$$\models^{\mathsf{d}}_I \langle\!\langle C\rangle\!\rangle_\alpha\, \varphi \quad\not\Leftrightarrow\quad \models^{\mathsf{d}}_I \langle\!\langle C\rangle\!\rangle_\beta\, \varphi$$

To see this, let $\{p\} = \mathsf{At}_C$. Then we have $\models^{\mathsf{d}}_I \langle C\rangle(p \leftrightarrow q)$, but obviously not

$\models^{\mathsf{d}}_{I} \langle\langle C\rangle\rangle_{\alpha} (p \leftrightarrow q)$. Instead, we get the following implications between the different types of ability.

$$\models^{\mathsf{d}}_{I} \langle\langle C\rangle\rangle_{\alpha} \varphi \rightarrow \langle\langle C\rangle\rangle_{\beta} \varphi \tag{17}$$

$$\models^{\mathsf{d}}_{I} \langle\langle C\rangle\rangle_{\beta} \varphi \rightarrow \langle C\rangle\varphi \tag{18}$$

Given that the definition of the α cooperation modalities more closely resembles those of Pauly's coalition logic, the following question arises: To what extent does Pauly's axiomatization of Coalition Logic carry over to CL-PC with α cooperation modalities? Consider the following axioms, which are just Pauly's axioms for Coalition Logic translated directly into CL-PC (see [36, pp.54–56]).

$(\alpha\text{-}N\bot)$ $\neg\langle\langle Ag\rangle\rangle_{\alpha} \bot$

$(\alpha\text{-}\top)$ $\langle\langle \emptyset\rangle\rangle_{\alpha} \bot \rightarrow \langle\langle C\rangle\rangle_{\alpha} \top$

$(\alpha\text{-}\bot)$ $\langle\langle C\rangle\rangle_{\alpha} \bot \rightarrow \langle\langle C'\rangle\rangle_{\alpha} \bot$ $\quad (C' \subseteq C)$

$(\alpha\text{-}N)$ $\neg\langle\langle \emptyset\rangle\rangle_{\alpha} \neg\varphi \rightarrow \langle\langle Ag\rangle\rangle_{\alpha} \varphi$

$(\alpha\text{-}S)$ $(\langle\langle C_1\rangle\rangle_{\alpha} \varphi \wedge \langle\langle C_2\rangle\rangle_{\alpha} \psi) \rightarrow \langle\langle C_1 \cup C_2\rangle\rangle_{\alpha} (\varphi \wedge \psi)$ $\quad (C_1 \cap C_2 = \emptyset)$

We have the following.

Theorem 4.1 Axiom schemes $(\alpha\text{-}N\bot)$, $(\alpha\text{-}\top)$, $(\alpha\text{-}\bot)$, $(\alpha\text{-}N)$, and $(\alpha\text{-}S)$ may all be derived within the axiom system CL-PC.

Proof: See Appendix B. ▫

4.2 Characterising Control

In this section, we address ourselves to the problem of characterising the circumstances under which a coalition *controls* some state of affairs. We begin with the following observation: Control is closely related to ability, but it is not the same thing. For, consider any propositional tautology $\top$. According to all of the definitions of ability we have considered thus far, *any* coalition (even $\emptyset$) can achieve $\top$. But we would not be inclined to take seriously any coalition that claimed to *control* $\top$ – this would be rather like claiming to control the heavens just because the sun happened to rise every morning. So, being able to control some state of affairs φ means not just being able to achieve φ, it also means being able to achieve $\neg\varphi$. This motivates the following.

Definition 6 (Contingent control) *Where C is a coalition and φ is a formula of* CL-PC*, we write $controls(C, \varphi)$ to mean that C can choose φ to be either true or false:*

$$controls(C, \varphi) \mathrel{\hat{=}} \langle C \rangle \varphi \wedge \langle C \rangle \neg \varphi$$

The following result suggests that this definition seems to be operating at least partly in the way intended: by using the $controls(\ldots)$ connective, we can capture the distribution of propositions among agents in a model.

Lemma 4.1 Let $\mathcal{M} = \langle Ag, \mathsf{At}, \mathsf{At}_1, \ldots, \mathsf{At}_n, \theta \rangle$ be a model for CL-PC, $i \in Ag$ be an agent, and $p \in \mathsf{At}$ be a propositional variable in $\mathcal{M}$. Then:

$$\mathcal{M} \models^{\mathsf{d}} controls(i, p) \quad \text{iff} \quad p \in \mathsf{At}_i.$$

Proof: For the left-to-right implication, assume $\mathcal{M} \models^{\mathsf{d}} controls(i, p)$. Then there are two cases to consider: where $\theta(p) = \mathtt{ff}$ and $\theta(p) = \mathtt{tt}$. Consider the first case. Since $\mathcal{M} \models^{\mathsf{d}} controls(i, p)$, there exists an i-valuation θ_i such that $\mathcal{M} \oplus \theta_i \models^{\mathsf{d}} p$. Since $\mathcal{M}$ differs from $\mathcal{M} \oplus \theta_i$ only in the values it assigns to variables in the domain of θ_i, it follows that $p \in \text{dom } \theta_i$, and hence $p \in \mathsf{At}_i$. The second case is essentially identical. For the right-to-left implication, assume $p \in \mathsf{At}_i$, and consider any two i-valuations θ_i^- and θ_i^+ such that $\theta_i^+(p) = \mathtt{tt}$ and $\theta_i^-(p) = \mathtt{ff}$, but on all atoms $q \neq p$, $\theta_i^+(q) = \theta_i(q) = \theta^-(q)$. Clearly, $\mathcal{M} \oplus \theta_i^+ \models^{\mathsf{d}} p$ and $\mathcal{M} \oplus \theta_i^- \models^{\mathsf{d}} \neg p$, so $\mathcal{M} \models^{\mathsf{d}} \langle i \rangle p \wedge \langle i \rangle \neg p$, and hence by definition $\mathcal{M} \models^{\mathsf{d}} controls(i, p)$. □

In other cooperation logics, there is not much we can really say about when a coalition controls some state of affairs: either it controls it, or it does not control it. However, with CL-PC, we have more information to go on: we know that propositional variables are controlled by agents. We can use this structure to extract some general rules with respect to when a coalition controls a formula, in terms of the logical structure of that formula, the propositional variables that make up the formula, and the variables that are controlled by the coalition in question. So, we will now consider the issue of how we might generalise Lemma 4.1 to *arbitrary* formulas φ.

First of all, note that the fact that $controls(i, \varphi)$ is true, does not mean that $\mathsf{At}(\varphi) \subseteq \mathsf{At}_i$. As a counterexample, suppose that p is true, and $\mathsf{At}_i = \{q\}$. Then obviously $controls(i, p \wedge q)$, but $\mathsf{At}(p \wedge q) = \{p, q\} \not\subseteq \{q\} = \mathsf{At}_i$. This example also shows that although the set At_i is defined independently of the actual valuation θ in $\mathcal{M}$, whether i controls φ may depend on the specific θ, since in the above example $controls(i, p \wedge q)$ would not have been true in a model $\mathcal{M}'$ in which p was false. Secondly, the fact that $\mathsf{At}(\varphi) \subseteq \mathsf{At}_i$ does not guarantee that i controls φ, for which $(p \vee \neg p)$ for agent i with $\mathsf{At}_i = \{p\}$ is a witness. In order for any coalition C to control φ, the formula φ must be a *contingency*, i.e., neither a contradiction nor a tautology. Conversely, if φ *is* a contingent formula, the grand coalition *Ag* controls it:

Observation 4.1 Let I be a partition $\mathsf{At}_1, \ldots, \mathsf{At}_n$ of the atoms At. Then, for every formula φ:

$$\models^{\mathsf{d}}_I \mathit{controls}(Ag, \varphi) \text{ iff } (\not\models^{\mathsf{d}}_I \varphi \text{ and } \not\models^{\mathsf{d}}_I \neg\varphi)$$

Proof: From left to right is obvious, so assume φ is contingent and let the model $\mathcal{M}_I$ be $\langle Ag, \mathsf{At}, \mathsf{At}_1, \ldots, \mathsf{At}_n, \theta\rangle$. Then, for some $\mathcal{M}'_I = \langle Ag, \mathsf{At}, \mathsf{At}_1, \ldots, \mathsf{At}_n, \theta'\rangle$ and for some $\mathcal{M}''_I = \langle Ag, \mathsf{At}, \mathsf{At}_1, \ldots, \mathsf{At}_n, \theta''\rangle$ we have $\mathcal{M}' \models^{\mathsf{d}} \varphi$ and $\mathcal{M}'' \models^{\mathsf{d}} \neg\varphi$. Note that we have $\theta = \theta' \pmod{\mathsf{At}}$ and $\theta = \theta'' \pmod{\mathsf{At}}$, which finishes the proof. □

To appreciate the relevance of a fixed partition I in Observation 4.1, note that in the proof, the two obtained models $\mathcal{M}'_I$ and $\mathcal{M}''_I$ are based on the same partition I. Indeed, we have the following non-equivalence:

$$\models^{\mathsf{d}} \mathit{controls}(Ag, \varphi) \not\Leftrightarrow (\not\models^{\mathsf{d}} \varphi \text{ and } \not\models^{\mathsf{d}} \neg\varphi) \tag{19}$$

Although from left to right is a valid implication, here is a witness that refutes the right to left direction: take for φ the formula $\mathit{controls}(i, p)$. From Lemma 4.1, we know that this formula is true in a model $\mathcal{M}$ iff $p \in \mathsf{At}_i$. Obviously, there are partitions I for which this is the case, and others, for which it is not, from which we see that the right hand side of (19) is true. However, at the same time, $\not\models^{\mathsf{d}} \mathit{controls}(Ag, \mathit{controls}(i, p))$, because, in an arbitrary model $\mathcal{M}_I$ based on a partition I, we either have $p \in \mathsf{At}_i$ (in which case $\models^{\mathsf{d}}_I \mathit{controls}(i, p)$) or $p \notin \mathsf{At}_i$ (giving $\models^{\mathsf{d}}_I \neg\mathit{controls}(i, p)$). In both cases, by Observation 4.1, we have $\not\models^{\mathsf{d}}_I \mathit{controls}(Ag, \mathit{controls}(i, p))$ and hence, by the classical counterpart of (3) and Corollary 2.2, $\not\models^{\mathsf{d}} \mathit{controls}(Ag, \mathit{controls}(i, p))$

We now consider the more general question of when a coalition C controls a formula φ. We settle it first for the case with a fixed partition (in what follows, we use some notation and terminology that was introduced in Section 3). We cannot generalise Observation 4.1 to the following: Let I be a partition $\mathsf{At}_1, \ldots, \mathsf{At}_n$ of the atoms At. Then, for every formula φ and every coalition C, $\models^{\mathsf{d}}_I \mathit{controls}(C, \varphi)$ iff:

$$(\not\models^{\mathsf{d}}_I \varphi \text{ and } \not\models^{\mathsf{d}}_I \neg\varphi \text{ and } \exists\varphi' : \models^{\mathsf{d}}_I (\varphi' \leftrightarrow \varphi) \text{ and } \mathsf{At}(\varphi') \subseteq \mathsf{At}(C))$$

Although the only if direction holds, a counterexample for the if direction is the formula $\mathit{controls}(i, (p_i \leftrightarrow q))$, $\mathsf{At}(i) = \{p_i\}$. We *do* have the following, however.

Theorem 4.2 Let C be a coalition, φ a formula, and $\mathcal{M}_I = \langle Ag, \mathsf{At}, \mathsf{At}_1, \ldots, \mathsf{At}_n, \theta\rangle$ a propositional model. Then: $\mathcal{M}_I \models^{\mathsf{d}} \mathit{controls}(C, \varphi)$ iff there exists a propositional formula ψ with the following properties:

(a) $\models^{\mathsf{d}}_{I} \psi \leftrightarrow \varphi$

(b) Let $DNF(\psi, C)$ equal $(\pi_1^{\hat{C}} \wedge \pi_1^{\check{C}}) \vee \cdots \vee (\pi_d^{\hat{C}} \wedge \pi_d^{\check{C}})$ with $\mathsf{At}(\pi_j^{\hat{C}}) = X = \mathsf{At}(\psi) \cap \mathsf{At}(C)$. Then, for some $j \leq d$:

$(b1)$ $\mathcal{M}_I \models^{\mathsf{d}} \pi_j^{\check{C}}$

$(b2)$ There is some $\ell(X) = \ell(x_1) \wedge \cdots \wedge \ell(x_k)$ such that $\ell(X) \wedge \pi_j^{\check{C}}$ does not occur in $DNF(\psi, C)$.

Proof: Suppose that (a) and (b) are true, then we have that $\mathcal{M}_I \models^{\mathsf{d}} \pi_j^{\check{C}}$. By soundness, and Lemma 3.3, we have $\mathcal{M}_I \models^{\mathsf{d}} \langle C\rangle(\pi_j^{\check{C}} \wedge \pi_j^{\hat{C}})$, and hence $\mathcal{M}_I \models^{\mathsf{d}} \langle C\rangle\varphi$. We also have $\vdash^{\text{CL-PC}} (\ell(X) \wedge \pi_j^{\check{C}}) \rightarrow \neg\varphi$ and thus $\models^{\mathsf{d}}_{I} (\ell(X) \wedge \pi_j^{\check{C}}) \rightarrow \neg\varphi$. But since $\pi_j^{\check{C}}$ is already true in $\mathcal{M}_I$, and all the atoms in $\ell(X)$ belong to C, we have $\mathcal{M}_I \models^{\mathsf{d}} \langle C\rangle(\ell(X) \wedge \pi_j^{\check{C}})$, and hence $\mathcal{M}_I \models^{\mathsf{d}} \langle C\rangle\neg\varphi$. Conversely, suppose $\mathcal{M}_I \models^{\mathsf{d}} \mathit{controls}(C, \varphi)$, i.e., $\mathcal{M}_I \models^{\mathsf{d}} \langle C\rangle\varphi$ and $\mathcal{M}_I \models^{\mathsf{d}} \langle C\rangle\neg\varphi$. By Lemma 3.3, we find a ψ such that $\models^{\mathsf{d}}_{I} \psi \leftrightarrow \varphi$, and X and Y as usual, with $DNF(\psi, C) = \bigvee_{i \leq d} \pi_i$, each π_i of type $\pi^{\hat{C}} \wedge \pi^{\check{C}}$. The model $\mathcal{M}_I$ must decide on one of the disjunctions, i.e. for some π_j $(j \leq d)$, we have $\mathcal{M}_I \models^{\mathsf{d}} \pi_j$, i.e., $\mathcal{M}_I \models^{\mathsf{d}} \pi_j^{\hat{C}} \wedge \pi_j^{\check{C}}$. Now, suppose that every combination $\ell(X)$ occurs as $\ell(X) \wedge \pi_j^{\check{C}}$ in $DNF(\psi, C)$. By propositional reasoning, we then obtain that $\vdash^{\text{CL-PC}} \pi_j^{\check{C}} \rightarrow \varphi$, and, by necessitation and soundness, $\models^{\mathsf{d}} [C](\pi_j^{\check{C}} \rightarrow \varphi)$. Since C has no control over the atoms in $\pi_j^{\check{C}}$, and $\pi_j^{\check{C}}$ is already true in $\mathcal{M}_I$, we have $\mathcal{M}_I \models^{\mathsf{d}} [C]\pi_j^{\check{C}}$. All in all, we see that $\mathcal{M}_I \models^{\mathsf{d}} [C]\varphi$. But this is in contradiction with $\mathcal{M}_I \models^{\mathsf{d}} \langle C\rangle\neg\varphi$, and hence not every combination $\ell(X)$ occurs as $\ell(X) \wedge \pi_j^{\check{C}}$ in $DNF(\psi, C)$. ▫

This notion of control still has a local flavour: if an atom $p \in \mathsf{At}_i$ and $q \notin \mathsf{At}_i$, for instance, then $\mathit{controls}(i, p \wedge q)$ would be true in a model $\mathcal{M}$ for which $\mathcal{M} \models^{\mathsf{d}} q$, but not in models $\mathcal{M}'$ for which $\mathcal{M}' \models^{\mathsf{d}} \neg q$. In fact, the formulas that are globally under control of a coalition C have a very natural determination, for a fixed interpretation I:

Theorem 4.3 Let C be a coalition, φ a formula, and I a fixed partition of At. Then: $\models^{\mathsf{d}}_{I}$ $\mathit{controls}(C, \varphi)$ iff there exists a propositional formula ψ with the following properties:

(a) $\models^{\mathsf{d}}_{I} \psi \leftrightarrow \varphi$

(b) Let $DNF(\psi, C)$ be $\bigvee_{j \leq d}(\pi_j^{\hat{C}} \wedge \pi_j^{\check{C}})$. Then $\models^{\mathsf{d}}_{I} \bigvee_{j \leq d} \pi_j^{\check{C}}$ and for every $\pi_j^{\check{C}}$ in $DNF(\psi, C)$, there is an conjunction $\ell(X)$ over literals in X such that $\ell(X) \wedge \pi_j^{\check{C}}$ does not occur in $DNF(\psi, C)$.

Proof: Immediately from Theorem 4.2. ▫

Corollary 4.1 $\models^{\mathsf{d}}$ $\mathit{controls}(C, \varphi)$ iff for every partition I of At, the conditions (a) and (b) of Theorem 4.3 are met.

Note that two formulas $\langle C\rangle\varphi$ and $\langle C'\rangle\varphi'$ can be equivalent, while on the one hand C controls φ, but C' need not control φ': we have for instance $controls(Ag, p)$, for any p, but not $controls(C, \langle\overline{C}\rangle p)$, even though $\langle Ag\rangle p$ and $\langle C\rangle\langle\overline{C}\rangle p$ are equivalent.

It seems that rather than the local notion of control, the notion of Theorem 4.3 is the more interesting one. It has a local counterpart, though, inspired by (2). Let us say that a coalition C fully controls a property φ in $\mathcal{M}$, notation $\mathcal{M} \models^{\mathsf{d}} Controls(C, \varphi)$, if $\mathcal{M} \models^{\mathsf{d}} [Ag]controls(C, \varphi)$. We then obtain (where $\mathcal{M}$ is based on partition I):

$$\mathcal{M} \models^{\mathsf{d}} Controls(C, \varphi) \text{ iff } \models^{\mathsf{d}}_{I} controls(C, \varphi) \tag{20}$$

5 Model Checking and Satisfiability

Whenever one considers reasoning within some logic, there are two basic problems that one must address: *model checking* and *satisfiability checking*. Model checking is the problem of determining, for any given model and formula, whether the formula is satisfied by the model; the satisfiability problem, in contrast, asks whether there is any way a given formula could be satisfied. For many modal and temporal logics, the model checking problem is computationally easy when compared to the satisfiability checking problem, and for this reason there is currently much interest in reasoning via model checking, rather than satisfiability checking (see e.g., [15,20,12] for discussions).

In this section, we characterise the complexity of these two problems with respect to CL-PC. We show that, in fact, their complexity coincides: both problems are PSPACE-complete. Note that in the remainder of this section, we are assuming the use of the *direct* semantics.

5.1 Model Checking

We begin by considering model checking for CL-PC. Formally, the decision problem is as follow.

CL-PC MODEL CHECKING:
Instance: CL-PC model $\mathcal{M}$ and formula φ.
Answer: "Yes" if $\mathcal{M} \models^{\mathsf{d}} \varphi$, "no" otherwise.

1. function $eval(\varphi, \langle Ag, \mathsf{At}, \mathsf{At}_1, \ldots, \mathsf{At}_n, \theta\rangle)$ returns $\mathtt{tt}$ or $\mathtt{ff}$
2.   if $\varphi \in \mathsf{At}$ then
3.   return $\theta(\varphi)$
4.   elsif $\varphi = \neg\psi$ then
5.   return not $eval(\psi, \langle Ag, \mathsf{At}, \mathsf{At}_1, \ldots, \mathsf{At}_n, \theta\rangle)$
6.   elsif $\varphi = \psi_1 \vee \psi_2$ then
7.   return $eval(\psi_1, \langle Ag, \mathsf{At}, \mathsf{At}_1, \ldots, \mathsf{At}_n, \theta\rangle)$
8.   or $eval(\psi_2, \langle Ag, \mathsf{At}, \mathsf{At}_1, \ldots, \mathsf{At}_n, \theta\rangle)$
9.   elsif $\varphi = \langle C\rangle\psi$ then
10.   for each C-valuation θ_C
11.   if $eval(\psi, \langle Ag, \mathsf{At}, \mathsf{At}_1, \ldots, \mathsf{At}_n, \theta\rangle \oplus \theta_C)$ then return $\mathtt{tt}$
12.   end-for
13.   return $\mathtt{ff}$
14.   endif
15. end-function

Fig. 4. A polynomial space model checking algorithm for CL-PC.

Theorem 5.1 *The* CL-PC MODEL CHECKING *problem is* PSPACE*-complete.*

Proof: For membership of PSPACE, consider the algorithm presented in Figure 4 [4]. Partial correctness – that is, $eval(\varphi, \mathcal{M}) = \mathtt{tt}$ iff $\mathcal{M} \models^{\mathsf{d}} \varphi$ – is by an easy induction on the structure of formulae. Termination follows from the fact that the algorithm is recursively analytic (i.e., the only recursive calls are on strict sub-formulae) with atomic propositions being the recursive base. The loop in lines 10–12 clearly terminates as there are finitely many C-valuations. With respect to the space requirements of the algorithm, for an input formula φ, the algorithm will require at most $|\varphi|$ recursive calls. A recursive call requires no more space than the original input instance, and since there are at most $|\varphi|$ recursive steps, the recursion stack must record at most $|\varphi|$ recursive calls, each of which requires no more space than the original input instance. Note that the loop in lines 10–12 merely involves binary counting using the propositional variables At_C. Thus, the algorithm requires only space polynomial in $|\varphi|$.

To show PSPACE hardness, we reduce QSAT [35, p.456] to the CL-PC model checking problem. An instance of QSAT is given by a quantified Boolean formula in prenex normal form:

$$\exists x_1, \forall x_2, \exists x_3, \ldots, Q_m x_m \cdot \varphi(x_1, x_2, x_3, \ldots, x_m) \tag{21}$$

where $x_1, x_2, x_3, \ldots, x_m$ are propositional variables, the quantifier Q_m is "$\exists$" if m is odd and "$\forall$" otherwise, and $\varphi(x_1, x_2, x_3, \ldots, x_m)$ is a propositional logic formula (the *matrix*) over $x_1, x_2, x_3, \ldots, x_m$. The goal is to determine whether this formula is

[4] Note that we are not presenting this as a *practical* algorithm for model checking – its purpose is merely to establish membership in PSPACE.

true: that is, whether there is some value for x_1 such that for all values of x_2, there is some value for x_3 such that, ..., such that the matrix $\varphi(x_1, x_2, x_3, \ldots, x_m)$ is true under this valuation. We can create an instance of the CL-PC model checking problem from an instance (21) as follows. To create the model $\mathcal{M} = \langle Ag, \mathsf{At}, \mathsf{At}_1, \ldots, \mathsf{At}_n, \theta\rangle$, set $\mathsf{At} = \{x_1, \ldots, x_m\}$, and create an agent a_i for each quantifier Q_i. Then set $\mathsf{At}_i = \{x_i\}$. Finally, define θ by $\theta(x_i) = \mathtt{ff}$ for all $x_i \in \mathsf{At}$. (In fact, the valuation θ could be set to anything – it makes no difference.) Then define the input formula to the model checking problem to be

$$\langle a_1\rangle[a_2]\langle a_3\rangle \cdots \{a_m\} \;\; \varphi(x_1, x_2, x_3, \ldots, x_m) \tag{22}$$

where the final cooperation modality, (i.e., $\{a_m\}$), is $\langle a_m\rangle$ if m is odd and $[a_m]$ if m is even. We claim that this formula is satisfied in the model we have created iff the input formula (21) is true: the result follows immediately from the semantics of $\langle\cdots\rangle$ and $[\cdots]$ and the allocation of variables to agents. Hence the model checking problem for CL-PC is PSPACE-hard, and we conclude that it is PSPACE-complete. □

For readers familiar with the literature on model checking for cooperation logics, this result may seem surprisingly negative. After all, the two closest relatives to CL-PC, (i.e., Pauly's coalition logic [38,37,36] and ATL [5]) both have tractable (polynomial time) model checking problems. Closer examination gives an explanation. The ATL model checking problem may be solved in time $O(|M| \cdot |\varphi|)$, where $|M|$ is the size of the model M against which the formula is to be checked, and $|\varphi|$ is the size of the formula that is to be checked. However, models for ATL (and Coalition Logic) are more like our Kripke structures for CL-PC than the direct semantic models we use. That is, they are assumed to contain an explicitly enumerated set of states corresponding to the possible situations that agents may cooperate to bring about – and in general, this set of states will be exponentially large in the number of agents and the number of propositional atoms. In contrast, our (direct) models for CL-PC are exponentially more succinct – the fact that the model checking problem for ATL and Coalition Logic appear to be tractable is thus only part of the story.

The distinction between succinct specification and explicit enumeration of states of models in model checking, and the effect that this has on the apparent complexity of model checking, has been noted elsewhere – see [41] for a detailed discussion. Donini and colleagues show how SMV, the well-known symbolic model checker for CTL, can be used to efficiently solve quantified boolean formulae (QBF), a well-known PSPACE-complete problem [13]. This is despite the fact that CTL model checking, like ATL and Coalition Logic model checking, can ostensibly be done in polynomial time [15,12]. The explanation for the fact that an apparently polynomial time algorithm for CTL model checking can be used to efficiently solve PSPACE-complete problems is exactly the same as that of the PSPACE-completeness of CL-PC model checking: the CTL model checking algorithm works in time polynomial in the number of states, but the model specification language used in SMV allows an extremely succinct description of this state set.

It is worth noting that our result suggests that the *actual* complexity of ATL model checking (i.e., the complexity of the problem that the MOCHA model checker solves) is much higher than the apparent complexity of the explicit state model checking problem. Our result suggests it is at least PSPACE-hard, and may be worse.

5.2 Satisfiability Checking

The next problem we consider is that of *satisfiability checking*: the problem of determining whether or not there is any way a given CL-PC formula could be true. Note that, for ATL, the satisfiability problem is EXPTIME-complete [14], whereas for Coalition Logic, the complexity of the satisfiability problem varies from NP-complete up to PSPACE-complete, depending on the variation considered [36, pp.73–75].

CL-PC SATISFIABILITY CHECKING:
Instance: CL-PC formula φ.
Answer: "Yes" if φ is satisfiable, "no" otherwise.

Theorem 5.2 *The* CL-PC SATISFIABILITY *problem is* PSPACE*-complete.*

Proof: Given a formula φ of CL-PC, the following non-deterministic algorithm decides whether it is satisfiable:

(1) Guess a model $\mathcal{M}$ such that $size(\mathcal{M}) = |Ag(\varphi)| + |\mathsf{At}(\varphi)| + 1$.
(2) Verify that $\mathcal{M} \models^{\mathsf{d}} \varphi$.

Step (1) can be done in (non-deterministic) polynomial space, and by Theorem 5.1, step (2) can be done in polynomial space. Moreover, by Corollary 2.1, if φ is satisfiable, then it is satisfiable in a model of the type guessed in step (1). Hence the problem is in NPSPACE, and since NPSPACE = PSPACE, the problem is in PSPACE.

To establish PSPACE hardness, we will reduce QSAT to CL-PC SATISFIABILITY (recall the definition of QSAT from Theorem 5.1). The idea of the reduction is basically the same as that of Theorem 5.1, but as we have no model to check against, we must add a side condition to the formula (22) that we create to ensure the correct distribution of controlled propositions amongst agents. Thus, given an instance (21) of QSAT, the instance of CL-PC SATISFIABILITY that we create is as follows.

$$(22) \wedge \bigwedge_{i=1}^{m} controls(a_i, x_i) \tag{23}$$

Now, by Lemma 4.1, in any model that satisfies (23), proposition x_i will be under the control of agent a_i. We now claim that (23) is satisfiable iff the QSAT instance (21) is true. The result again follows immediately from the semantics of $\langle \cdots \rangle$ and

[$\cdots$]. The CL-PC SATISFIABILITY problem is thus PSPACE-hard, and as we proved above that it is in PSPACE, we conclude it is PSPACE-complete. □

6 Related Work

Over the past three decades, researchers from many disciplines have attempted to develop a general purpose logic of ability. Within the artificial intelligence (AI) community, it was understood that such a logic could be used in order to gain a better understanding of STRIPS-style AI planning systems [17,28,2]. The most notable early effort in this direction was Moore's dynamic epistemic logic [29,30]. Moore was particularly interested in capturing some of the interactions between knowledge, action, and ability. For example, he was the first to formalise the idea of "knowledge pre-conditions": what an agent must know in order to achieve some goal, or carry out some action. To use an overworked example, consider agent A that must speak to agent B via telephone. In order to accomplish this goal, A needs to know B's telephone number – this is thus a knowledge pre-condition. However, as Moore noted, not having immediate access to B's telephone number does not preclude the goal from being accomplished – because A has an action available (looking up B's number in the telephone directory) that will furnish it with this knowledge. Moore also made the important distinction between an agent knowing a plan that will achieve some goal, and the knowledge that there exists a plan that will achieve a goal. Using the terminology of quantified modal logic [27, pp.183–188], in the former case, the agent has *de re* knowledge of the plan (in the sense that not only does the agent know that there is a plan that will achieve the goal, it knows what the plan is – it knows the *identity* of the plan), while in the latter case, the agent has *de dicto* knowledge of the plan (the agent may be ignorant about the identity of the plan). Moore's formalism was able to capture these distinctions. Moore's work was subsequently enhanced by many other researchers, perhaps most notably, Morgenstern [31,32]. These distinctions also informed later attempts to integrate a logic of ability into more general logics of rational action in autonomous agents [47,45] (see [46] for a survey of such logics).

In a somewhat parallel thread of research, researchers in the philosophy of action have developed a range of logics underpinned by rather similar ideas and motivations. A typical example is that of Brown, who developed a logic of individual ability in the mid-1980s [10]. Brown's main claim was that modal logic was a useful tool for the analysis of ability, and that previous – unsuccessful – attempts to characterise ability in modal logic were based on an over-simple semantics. Brown's account of the semantics of ability was as follows [10, p.5]:

> [An agent can achieve A] at a given world iff *there exists* a relevant cluster of worlds, at *every* world of which A is true.

Notice the $\exists\forall$ pattern of quantifiers in this account. Brown immediately noted that this gave the resulting logic a rather unusual flavour, neither properly existential nor properly universal [10, p.5]:

> Cast in this form, the truth condition [for ability] involves *two* metalinguistic quantifiers (one existential and one universal). In fact, [the character of the ability operator] should be a little like each.

As we noted in the introduction, contemporary logics of ability – Coalition Logic and ATL – are based on exactly the same idea. It is worth noting that many similar approaches have been developed in the logical theory of action: examples include the "seeing to it that" logic of Belnap and Perloff [8,7,26], Segerberg's logic of "bringing it about" [42], and of course program logics such as dynamic logic [21]. A useful survey of such work was published in 1992 [43].

The recent surge of interest in logics of strategic ability was sparked by two largely independent developments: Pauly's development of Coalition Logic [38,37,36,39], and the development of ATL by Alur, Henzinger, and Kupferman [5,19,14]. Although these logics are very closely related (indeed, Coalition Logic can be formally understood as the "next-time" fragment of ATL [19]), the motivation and background to the two systems is strikingly different.

Pauly's Coalition Logic was developed in an attempt to shed some light on the links between logic – and in particular, modal logic – and the mathematical theory of games [33]. Pauly showed how the semantic structures underpinning a family of logics of cooperative ability 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.

ATL, however, emerged from a rather different research community, and was developed with an entirely different set of motivations in mind. The development of ATL is closely linked with the development of branching-time temporal logics for the specification and verification of reactive systems [16,15,44]. Perhaps the best known branching-time temporal logic is Computation Tree Logic (CTL) [15]. CTL is a temporal logic that is interpreted over tree-like structures, in which nodes represent time points and arcs represent transitions between time points. In distributed systems applications, the set of all paths through a tree structure is assumed to correspond to the set of all possible computations of a system. CTL combines *path quantifiers* "A" and "E" for expressing that a certain series of events will happen on all paths and on some path respectively, with *tense modalities* for expressing that something will happen eventually on some path ($\Diamond$), always on some path ($\Box$) and so on. Thus, for example, by using CTL-like logics, one may express properties such as "on all possible computations, the system never enters a fail state", which is represented by the CTL formula $\mathsf{A}\Box\neg \mathit{fail}$. Although they have proved to be enormously useful in the specification and verification of reactive systems, logics such as CTL are of limited value for reasoning about *multi-agent* systems, in which system components (agents) cannot be assumed to be benevolent, but may have competing or conflicting goals. The kinds of properties we wish to express of such systems are the powers that the system components have. For example, we might wish to express the fact that "agents 1 and 2 can cooperate to ensure that the system never enters a fail state". It is not possible to capture such statements using CTL-like logics. The best one can do is either state that something will inevitably happen, or else that it may possibly happen: CTL-like logics have no notion of agency.

Alur, Henzinger, and Kupferman developed ATL in an attempt to remedy this deficiency. The key insight in ATL is that path quantifiers can be replaced by cooperation modalities: the ATL expression $\langle\langle C\rangle\rangle\varphi$, where C is a group of agents, expresses the fact that the group C can cooperate to ensure that φ. Thus, for example, the fact that agents 1 and 2 can ensure that the system never enters a fail state may be captured in ATL by the following formula: $\langle\langle 1, 2\rangle\rangle\Box\neg \mathit{fail}$. ATL generalises CTL because the path quantifiers A ("on all paths…") and E ("on some paths…") can be simulated in ATL by the cooperation modalities $\langle\langle\emptyset\rangle\rangle$ ("the empty set of agents can cooperate to…") and $\langle\langle\Sigma\rangle\rangle$ ("the grand coalition of all agents can cooperate to…").

One reason for the interest in ATL is that it shares with it ancestor CTL the computational tractability of its model checking problem [12] [5] . This led to the development of an ATL model checking system called MOCHA [6,3]. With MOCHA, one specifies a model against which a formula is to be checked using a model specification language called REACTIVE MODULES [4]. REACTIVE MODULES is a guarded command language, which provides a number of mechanisms for the structured specification of models, based upon the notion of a "module", which is basically the REACTIVE SYSTEMS terminology for an agent. Interestingly, however, it is ultimately necessary to define for every variable in a REACTIVE MODULES system which module (i.e., agent) controls it. The powers of agents and coalitions then derive from the ability to control these variables: and as we noted in the introduction, this observation was one of the motivations for considering CL-PC as a system in its own right.

ATL has begun to attract increasing attention as a formal system for the specification and verification of multi-agent systems. Examples of such work include formalising the notion of role using ATL [40], the development of epistemic extensions to ATL [23,25,24], and the use of ATL for specifying and verifying cooperative mechanisms [39].

[5] It also shares with CTL the intractability of its satisfiability problem [14].

It is worth noting that CL-PC is closely related to the formalism of *quantified Boolean formulae*. The logic of quantified Boolean formulae is an extension of propositional logic which permits quantification over propositional variables. Although perhaps not widely studied as an independent formalism, quantified Boolean formulae play an important role in the theory of computational complexity [35, p.455] (indeed, we used them when proving the complexity of CL-PC model checking and satisfiability); in addition, they are used in symbolic model checking algorithms [12, pp.66–67]. Readers may wonder whether quantified Boolean formulae might therefore be used directly to reason about the kinds of scenarios that we represent using the modal language of CL-PC. We believe that there is considerable value in the modal language of CL-PC. The family relationship that the calculus of quantified Boolean formulae bears to CL-PC is roughly that which the first-order relational calculus bears to conventional modal logic. Thus, CL-PC can be reduced to quantified Boolean formulae, in much the same way that conventional modal logic can be reduced to first-order relational calculus [11, pp.12–14]. But, just as the modal logic community believes that the language of modal logic is preferable to that of the first-order relational calculus for many situations [9, pp.xii–xiii], so we believe that CL-PC is preferable to that of quantified Boolean formulae for the scenarios we are interested in. Perhaps most importantly, CL-PC provides us with a language in which *agency* and *ability* are first-class components – and these notions are not present in quantified Boolean formulae. Attempting to represent our cooperation scenarios using quantified Boolean formulae, while arguably possible, would thus lose or obscure these aspects, which are explicitly present in CL-PC.

We finally mention related work of Harrenstein et al., which studies distribution of *decision variables* over agents in a game theoretic setting. In their so-called *Boolean Games* ([34]), two players partition the set of propositional atoms, which then represent moves in an extensive game form. Such games allow for operations that yield a Boolean algebra, and [34] defines winning strategies and determinacy on them. One of the main themes in Harrenstein's thesis is to describe ways that relate logical concepts to game theoretical ones [22]. Boolean Games inspire him to define a notion of generalized logical consequence, in which one does not quantify over arbitrary valuations, but rather those that are equivalent modulo a given set of variables. This, again, is then linked to game theoretical notions, like the existence of a winning strategy for one of the two players.

7 Conclusion

In this paper, we have introduced a logic of strategic cooperative ability in which the powers of individual agents within the system are defined by associating a set of propositional variables with each agent, the idea being that a set of propositional variables represents exactly the part of the environment under the corresponding agent's control. The choices available to an agent in this case correspond to the different assignments of truth or falsity that the agent can give to the variables under its control, and the powers of a coalition derive from the propositions under the control of its members. We then argued that there are several good, natural reasons why CL-PC is worth studying as a system in its own right. As well as giving two alternative, but provably equivalent semantics for CL-PC, we gave a complete axiomatization for the logic, established the complexity of its model checking and satisfiability problems, and investigated how a variety of different notions of ability and control could be captured in the logic, including α- and β-ability.

There remain a number of obvious challenges for future research. For example, in this analysis we have assumed a *static* power structure: the assignment of propositional atoms to agents is assumed to be fixed throughout the evaluation of a formula. We believe it will be interesting to investigate *dynamic* power structures, where agents are permitted to pass the control of their propositional atoms to other agents. Another possibility is to consider a dynamic logic-style extensions, in which atomic programs in the dynamic component of the logic correspond to assignments of truth or falsity that agents make to their variables. Finally, yet another interesting issue is the axioms that may arise by including ATL-style temporal operators into the logic.

Acknowledgements We would like to extend our sincerest thanks to Marc Pauly, who introduced us to Coalition Logic, and inspired us to write this paper. This research was supported by the EPSRC under grant RG/S62727/01.

References

[1] J. Abdou and H. Keiding. *Effectivity Functions in Social Choice Theory*. Kluwer Academic Publishers: Dordrecht, The Netherlands, 1991.

[2] J. F. Allen, J. Hendler, and A. Tate, editors. *Readings in Planning*. Morgan Kaufmann Publishers: San Mateo, CA, 1990.

[3] R. Alur, L. de Alfaro, T. A. Henzinger, S. C. Krishnan, F. Y. C. Mang, S. Qadeer, S. K. Rajamani, and S. Taşiran. MOCHA user manual. University of Berkeley Report, 2000.

[4] R. Alur and T. A. Henzinger. Reactive modules. *Formal Methods in System Design*, 15(11):7–48, July 1999.

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

[6] R. Alur, T. A. Henzinger, F. Y. C. Mang, S. Qadeer, S. K. Rajamani, and S. Taşiran. Mocha: Modularity in model checking. In *CAV 1998: Tenth International Conference on Computer-aided Verification, (LNCS Volume 1427)*, pages 521–525. Springer-Verlag: Berlin, Germany, 1998.

[7] N. Belnap. Backwards and forwards in the modal logic of agency. *Philosophy and Phenomenological Research*, LI(4):777–807, December 1991.

[8] N. Belnap and M. Perloff. Seeing to it that: a canonical form for agentives. *Theoria*, 54:175–199, 1988.

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

[10] M. A. Brown. On the logic of ability. *Journal of Philosophical Logic*, 17:1–26, 1988.

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

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

[13] F. M. Donini, P. Liberatore, F. Massacci, and M. Schaerf. Solving qbf with smv. In *Proceedings of the Eighth International Conference on Principles of Knowledge Representation and Reasoning (KR 2002)*, pages 578–589, Toulouse, France, 2002.

[14] G. van Drimmelen. Satisfiability in alternating-time temporal logic. In *Eighteenth Annual IEEE Symposium on Logic in Computer Science (LICS 2003)*, pages 208–217, Ottawa, Canada, 2003.

[15] E. A. Emerson. Temporal and modal logic. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science Volume B: Formal Models and Semantics*, pages 996–1072. Elsevier Science Publishers B.V.: Amsterdam, The Netherlands, 1990.

[16] E. A. Emerson and J. Srinivasan. Branching time logic. In J. W. de Bakker, W.-P. de Roever, and G. Rozenberg, editors, *REX School-Workshop on Linear Time, Branching Time and Parial Order in Logics and Models for Concurrency (LNCS Volume 354)*, pages 123–172. Springer-Verlag: Berlin, Germany, 1988.

[17] R. E. Fikes and N. Nilsson. STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 2:189–208, 1971.

[18] R. Goldblatt. *Logics of Time and Computation (CSLI Lecture Notes Number 7)*. Center for the Study of Language and Information, Ventura Hall, Stanford, CA 94305, 1987. (Distributed by Chicago University Press).

[19] V. Goranko. Coalition games and alternating temporal logics. In J. van Benthem, editor, *Proceeding of the Eighth Conference on Theoretical Aspects of Rationality and Knowledge (TARK VIII)*, pages 259–272, Siena, Italy, 2001.

[20] J. Y. Halpern and M. Y. Vardi. Model checking versus theorem proving: A manifesto. In V. Lifschitz, editor, *AI and Mathematical Theory of Computation — Papers in Honor of John McCarthy*, pages 151–176. The Academic Press: London, England, 1991.

[21] D. Harel, D. Kozen, and J. Tiuryn. *Dynamic Logic*. The MIT Press: Cambridge, MA, 2000.

[22] P. Harrenstein. *Logic in Conflict*. PhD thesis, Utrecht University, 2004.

[23] W. van der Hoek and M. Wooldridge. Tractable multiagent planning for epistemic goals. In *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2002)*, pages 1167–1174, Bologna, Italy, 2002.

[24] W. van der Hoek and M. Wooldridge. Model checking cooperation, knowledge, and time — a case study. *Research in Economics*, 57(3), September 2003.

[25] W. van der Hoek and M. Wooldridge. Time, knowledge, and cooperation: Alternating-time temporal epistemic logic and its applications. *Studia Logica*, 2003.

[26] J. F. Horty and N. Belnap. The deliberative STIT: A study of action, omission, ability, and obligation. *Journal of Philosophical Logic*, 24(6):583–644, December 1995.

[27] G. E. Hughes and M. J. Cresswell. *Introduction to Modal Logic*. Methuen and Co., Ltd., 1968.

[28] V. Lifschitz. On the semantics of STRIPS. In M. P. Georgeff and A. L. Lansky, editors, *Reasoning About Actions & Plans — Proceedings of the 1986 Workshop*, pages 1–10. Morgan Kaufmann Publishers: San Mateo, CA, 1986.

[29] R. C. Moore. Reasoning about knowledge and action. In *Proceedings of the Fifth International Joint Conference on Artificial Intelligence (IJCAI-77)*, Cambridge, MA, 1977.

[30] R. C. Moore. A formal theory of knowledge and action. In J. F. Allen, J. Hendler, and A. Tate, editors, *Readings in Planning*, pages 480–519. Morgan Kaufmann Publishers: San Mateo, CA, 1990.

[31] L. Morgenstern. A first-order theory of planning, knowledge, and action. In J. Y. Halpern, editor, *Proceedings of the 1986 Conference on Theoretical Aspects of Reasoning About Knowledge*, pages 99–114. Morgan Kaufmann Publishers: San Mateo, CA, 1986.

[32] L. Morgenstern. Knowledge preconditions for actions and plans. In *Proceedings of the Tenth International Joint Conference on Artificial Intelligence (IJCAI-87)*, pages 867–874, Milan, Italy, 1987.

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

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

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

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

[37] M. Pauly. A logical framework for coalitional effectivity in dynamic procedures. *Bulletin of Economic Research*, 53(4):305–324, 2002.

[38] M. Pauly. A modal logic for coalitional power in games. *Journal of Logic and Computation*, 12(1):149–166, 2002.

[39] M. Pauly and M. Wooldridge. Logic for mechanism design — a manifesto. In *Proceedings of the 2003 Workshop on Game Theory and Decision Theory in Agent Systems (GTDT-2003)*, Melbourne, Australia, 2003.

[40] M. Ryan and P.-Y. Schobbens. Agents and roles: Refinement in alternating-time temporal logic. In J.-J. Ch. Meyer and M. Tambe, editors, *Intelligent Agents VIII: Proceedings of the Eigth International Workshop on Agent Theories, Architectures, and Languages, ATAL-2001 (LNAI Volume 2333)*, pages 100–114, 2002.

[41] P. Schnoebelen. The complexity of temporal logic model checking. In P. Balbiani, N.-Y. Suzuki, F. Wolter, and M. Zakharyascev, editors, *Advanced in Modal Logic Volume 4*, pages 393–436. King's College Publications: London, 2003.

[42] K. Segerberg. Bringing it about. *Journal of Philosophical Logic*, 18:327–347, 1989.

[43] K. Segerberg. Getting started: Beginnings in the logic of action. *Studia Logica*, 51(3/4):347–378, 1992.

[44] M. Y. Vardi. Branching vs. linear time: Final showdown. In T. Margaria and W. Yi, editors, *Proceedings of the 2001 Conference on Tools and Algorithms for the Construction and Analysis of Systems, TACAS 2001 (LNCS Volume 2031)*, pages 1–22. Springer-Verlag: Berlin, Germany, April 2001.

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

[46] M. Wooldridge and N. R. Jennings. Intelligent agents: Theory and practice. *The Knowledge Engineering Review*, 10(2):115–152, 1995.

[47] M. Wooldridge and N. R. Jennings. The cooperative problem solving process. *Journal of Logic and Computation*, 9(4):563–592, 1999.

A A Derivation in the CL-PC Deductive System

We now give the proof of property S' of Figure 3. In order to circumvent to much clutter in notation, we can get rid of some subscripts by formulating S' as under the assumption that $(\mathsf{At}(\psi) \cap \mathsf{At}(\psi') = \emptyset)$, we have $\vdash^{\text{CL-PC}} \langle C \rangle \psi \wedge \langle D \rangle \psi' \rightarrow \langle C \cup D \rangle (\psi \wedge \psi')$.

Proof: By virtue of Theorem 3.1, we may assume that ψ and ψ' are purely propositional. Let $\pi = \pi_1 \vee \cdots \vee \psi_d$ be the disjunctive normal form for ψ, and $\pi' = \psi'_1 \vee \cdots \vee \pi'_{d'}$ that of ψ'. Since the assumption states that ψ and ψ' have no atoms in common, let $X = \{x_1, \ldots, x_k\}$ and $Y = \{y_1 \ldots y_m\}$ be all the atoms in $\mathsf{At}(\psi)$, all the x's under control of C, and none of the y's controlled by C. Similarly, define $X' = \{x'_1, \ldots, x'_{k'}\}$ and $Y' = \{y'_1 \ldots y'_{m'}\}$ for ψ' and D. We then know that

$(X \cup Y) \cap (X' \cup Y') = \emptyset$, and $(X \cup Y \cup X' \cup Y') = (\mathsf{At}(\psi) \cup \mathsf{At}(\psi'))$. Assume that every π_i ($i \leq d$) is of the form $\ell_i(x_1) \wedge \ldots$ *and* $\ell_i(x_k) \wedge \ell_i(y_1) \wedge \cdots \wedge \ell_i(y_m)$, and similarly for every π'_i with $i \leq d'$: $\pi'_i = \ell'_i(x'_1) \wedge \cdots \wedge \ell'_i(x'_{k'}) \wedge \ell'_i(y'_1) \wedge \cdots \wedge \ell'_i(y'_{m'})$. First of all, we observe, by *Prop*:

$$\vdash_{\text{CL-PC}} (\psi \wedge \psi') \leftrightarrow \bigvee_{i \leq d, j \leq d'} (\pi_i \wedge \pi'_j) \tag{A.1}$$

We obtain, by *Prop*, and the fact that $\langle C \rangle$ and $\langle D \rangle$ are diamonds:

$$\vdash^{\text{CL-PC}} (\langle C \rangle \psi \wedge \langle D \rangle \psi') \leftrightarrow (\bigvee_{i \leq d} \langle C \rangle \pi_i \wedge \bigvee_{j \leq d'} \langle D \rangle \pi'_j) \tag{A.2}$$

By Lemma 3.3, every $\langle C \rangle \pi_i$ is equivalent to $\pi_i^{\check{C}}$, and every $\langle D \rangle \pi'_j$ is equivalent to $\pi_j'^{\check{D}}$. Hence, we have

$$\vdash^{\text{CL-PC}} (\langle C \rangle \psi \wedge \langle D \rangle \psi') \leftrightarrow (\bigvee_{i \leq d} \pi_i^{\check{C}} \wedge \bigvee_{j \leq d'} \pi_j'^{\check{D}}) \tag{A.3}$$

by *Prop* we have

$$\vdash^{\text{CL-PC}} (\bigvee_{i \leq d} \pi_i^{\check{C}} \wedge \bigvee_{j \leq d'} \pi_j'^{\check{D}}) \leftrightarrow \bigvee_{i \leq d, j \leq d'} (\pi_i^{\check{C}} \wedge \pi_j'^{\check{D}}) \tag{A.4}$$

Let Π be a conjunct in the $DNF(\psi \wedge \psi', C))$, say $\Pi = \ell(x_1) \wedge \cdots \wedge \ell(x_k) \wedge \ell(x'_1) \wedge \cdots \wedge \ell(x'_{k'}) \wedge \ell(y_1) \wedge \cdots \wedge \ell(y_m) \wedge \cdots \wedge \ell(y'_1) \wedge \cdots \wedge \ell(y'_{m'})$. In an obvious way, we can write such a Π as $(\pi_i \wedge \pi'_j)$, for some $i \leq d, j \leq d'$, and, conversely, for every such $(\pi_i \wedge \pi'_j)$ there is a corresponding Π. Let us write Π_{ij} for it. From (A.1) we know that

$$\vdash^{\text{CL-PC}} (\psi \wedge \psi') \leftrightarrow \bigvee_{i \leq d, j \leq d'} \Pi_{ij} \tag{A.5}$$

Note that we also have $\vdash^{\text{CL-PC}} (\pi_i^{\check{C}} \wedge \pi_j'^{\check{D}}) \rightarrow \Pi_{ij}^{C \check{\cup} D}$: if any atom p is not under control of $C \cup D$, it is neither under control of C, nor of D, and hence such $p \in (Y \cup Y')$. Moreover, from Lemma 3.3, we know $\vdash^{\text{CL-PC}} \Pi_{ij}^{C \check{\cup} D} \rightarrow \langle C \cup D \rangle (\Pi_{ij}^{C \hat{\cup} D} \wedge \Pi_{ij}^{C \check{\cup} D})$. Combining those two observations gives

$$\vdash^{\text{CL-PC}} (\pi_i^{\check{C}} \wedge \pi_j'^{\check{D}}) \rightarrow \langle C \cup D \rangle (\Pi_{ij}^{C \hat{\cup} D} \wedge \Pi_{ij}^{C \check{\cup} D}) \tag{A.6}$$

We put the disjunction in place again, and push it through the diamond operator:

$$\vdash_{\text{CL-PC}} \bigvee_{i \leq d, j \leq d'} (\pi_i^{\check{C}} \wedge \pi_j'^{\check{D}}) \rightarrow \langle C \cup D \rangle \bigvee_{i \leq d, j \leq d'} (\Pi_{ij}^{C \hat{\cup} D} \wedge \Pi_{ij}^{C \check{\cup} D}) \tag{A.7}$$

Now we are done: observe that by (A.5), the right hand side of (A.7) is equal to $\langle C \cup D \rangle(\psi \wedge \psi')$, and use (A.3), (A.4) and (A.7). $\square$

B Deriving Pauly's Axioms in CL-PC

In this Appendix, we show that analogues of Pauly's axioms for Coalition Logic can be proved within the proof system CL-PC (see Section 4.1). Note that since we have established that the axiom system CL-PC is complete, it suffices to show that these schemes are valid within CL-PC. Also note that in the proofs that follow, we use whichever semantics seems appropriate to the task at hand.

Lemma B.1 (α-$N\bot$) $\vdash_{cl\text{-}pc} \neg\langle\langle Ag\rangle\rangle_\alpha \bot$.

Proof: Expanding out the definition of $\langle\langle\cdots\rangle\rangle_\alpha$ gives the following as our goal: $\neg\langle Ag\rangle[Ag \setminus Ag]\bot$, which in turn is equivalent to: $\neg\langle Ag\rangle\neg\langle\emptyset\rangle\top$. By (), this is equivalent to $\neg\langle Ag\rangle\neg\top$, or $[Ag]\top$, a derivable formula (use Necessitation). $\square$

Lemma B.2 (α-$\top$) $\vdash_{cl\text{-}pc} \neg\langle\langle\emptyset\rangle\rangle_\alpha \bot \rightarrow \langle\langle C\rangle\rangle_\alpha \top$.

Proof: Expanding out the definition of $\langle\langle\cdots\rangle\rangle_\alpha$ tells us that our goal is to prove the following: $\neg\langle\emptyset\rangle[\overline{\emptyset}]\bot \rightarrow \langle C\rangle[\overline{C}]\top$. Rewriting the expressions and expanding the definition of $[\cdots]$, we derive the following as our goal: $\neg\langle\emptyset\rangle\neg\langle Ag\rangle\top \rightarrow \langle C\rangle\neg\langle\overline{C}\rangle\bot$. We show that the right hand side of this is in fact derivable in CL-PC, and then we are done. This RHS is equivalent to $\langle C\rangle[\overline{C}]\top$; derivable since $[\overline{C}]\top$ is derivable (by Necessitation) and to that, we can apply the dual of $T(C)$. $\square$

Lemma B.3 (α-$\bot$) $\vdash_{cl\text{-}pc} \langle\langle C\rangle\rangle_\alpha \bot \rightarrow \langle\langle C'\rangle\rangle_\alpha \bot$ *(where $C' \subseteq C$).*

Proof: Expanding out, we get the following as our goal: $\models^{\mathsf{k}} \langle C\rangle[\overline{C}]\bot \rightarrow \langle C'\rangle[\overline{C'}]\bot$ (where $C' \subseteq C$). Further expansion yields $\langle C\rangle\neg\langle\overline{C}\rangle\top \rightarrow \langle C'\rangle\neg\langle\overline{C'}\rangle\top$. It is easy to see (by arguments similar to the preceding Lemma) that LHS is equivalent to $\bot$, which is sufficient. $\square$

Lemma B.4 (α-N) $\vdash_{cl\text{-}pc} \neg\langle\langle\emptyset\rangle\rangle_\alpha \neg\varphi \rightarrow \langle\langle Ag\rangle\rangle_\alpha \varphi$.

Proof: Expanding out gives the following as our goal: $\models^{\mathsf{k}} \neg\langle\emptyset\rangle[\overline{\emptyset}]\neg\varphi \rightarrow \langle Ag\rangle[\overline{Ag}]\varphi$. This in turn reduces to the following: $\models^{\mathsf{k}} \neg\langle\emptyset\rangle\neg\langle Ag\rangle\varphi \rightarrow \langle Ag\rangle[\emptyset]\varphi$. So, assume $\mathcal{K}, w \models^{\mathsf{k}} \neg\langle\emptyset\rangle\neg\langle Ag\rangle\varphi$ for arbitrary $\mathcal{K}, w$. By the definition of $[\cdots]$, we thus have $\mathcal{K}, w \models^{\mathsf{k}} [\emptyset]\langle Ag\rangle\varphi$. Since $\models^{\mathsf{k}} \psi \leftrightarrow [\emptyset]\psi$, we thus have $\mathcal{K}, w \models^{\mathsf{k}} \langle Ag\rangle\varphi$, and by deploying this equivalence again, we obtain $\mathcal{K}, w \models^{\mathsf{k}} \langle Ag\rangle[\emptyset]\varphi$, and we are done. $\square$

Lemma B.5 *(α-S)* $\vdash_{cl\text{-}pc} (\langle\!\langle C_1 \rangle\!\rangle_\alpha \varphi \wedge \langle\!\langle C_2 \rangle\!\rangle_\alpha \psi) \rightarrow \langle\!\langle C_1 \cup C_2 \rangle\!\rangle_\alpha (\varphi \wedge \psi)$ *(where $C_1 \cap C_2 = \emptyset$).*

Proof: Suppose the antecedent is true in w, i.e., suppose that $\mathcal{K}, w \models^{\mathsf{k}} \langle C_1 \rangle [\overline{C_1}] \varphi \wedge \langle C_2 \rangle [\overline{C_2}] \psi$, then our goal is to show

$$\mathcal{K}, w \models^{\mathsf{k}} \langle C_1 \cup C_2 \rangle [\overline{C_1 \cup C_2}] (\varphi \wedge \psi) \tag{B.1}$$

Our assumption gives us worlds w_1 and w_2 such that

$$R_{C_1} w w_1 \ \& \ \forall v (R_{\overline{C_1}} v w_1 \Rightarrow \mathcal{K}, w_1 \models^{\mathsf{k}} \varphi) \tag{B.2}$$

and

$$R_{C_2} w w_2 \ \& \ \forall v (R_{\overline{C_2}} v w_2 \Rightarrow \mathcal{K}, w_2 \models^{\mathsf{k}} \psi) \tag{B.3}$$

To prove our goal (B.1), we first construct a world w_{12} as follows: it is as the valuation w, but as for the atoms in $\mathsf{At}(C_i)$, we let $w_{1,2}(p) = w_i(p)$ $(i \leq 2)$. This is well-defined, since $C_1 \cap C_2 = \emptyset$. We have $R_{C_1 \cup C_2} w w_{12}$, so if we can prove

$$\mathcal{K}, w_{12} \models^{\mathsf{k}} [\overline{C_1 \cup C_2}] (\varphi \wedge \psi) \tag{B.4}$$

we are done. To this end, choose an arbitrary v for which $R_{\overline{C_1 \cup C_2}} w_{12} v$. We argue that we must have $R_{\overline{C_1}} w_1 v$, that is, $w_1 = v \pmod{\overline{C_1}}$. Since w_1 and w_{12} differ at most in $\mathsf{At}(C_2)$, and w_{12} and v differ at most in $\mathsf{At}(\overline{C_1 \cup C_2})$, we know that w_1 and v differ at most in $\mathsf{At}(C_2 \cup (\overline{C_1 \cup C_2})$, from which follows that $w_1 = v \pmod{\overline{C_1}}$ (since $C_2 \cup (\overline{C_1 \cup C_2}) \subseteq \overline{C_1}$; here we use again that $C_1 \cap C_2 = \emptyset$). But this yields indeed $R_{\overline{C_1}} w_1 v$, and hence $\mathcal{K}, v \models^{\mathsf{k}} \varphi$. In the same way, we obtain that $R_{\overline{C_2}} w_2 v$, from which $\mathcal{K}, v \models^{\mathsf{k}} \psi$ follows. Since we now have for an arbitrary v with $R_{\overline{C_1 \cup C_2}} w_{12} v$ that $\mathcal{K}, v \models^{\mathsf{k}} (\varphi \wedge \psi)$, we have proven (B.4), and thereby (B.1).

$\square$