On Logics of Strategic Ability based on Propositional Control

Francesco Belardinelli[1] and **Andreas Herzig**[2]
[1] IBISC, Univ. Evry, France
[2] IRIT, Univ. Toulouse, France

Abstract

Recently logics for strategic ability have gained pre-eminence in the modelisation and analysis of game-theoretic scenarios. In this paper we provide a contribution to the comparison of two popular frameworks: Concurrent Game Structures (CGS) and Coalition Logic of Propositional Control (CL-PC). Specifically, we ground the abstract abilities of agents in CGS on Propositional Control, thus obtaining a class of CGS that has the same expressive power as CL-PC. We study the computational properties of this setting. Further, we relax some of the assumptions of CL-PC so as to introduce a wider class of computationally-grounded CGS.

1 Introduction

Formal languages for reasoning about strategic behaviours of human and artificial agents have attracted much interest in recent years [Bulling *et al.*, 2010; Goranko and Jamroga, 2004]. Typically, modal languages for temporal reasoning have been extended with operators to represent strategic abilities of coalitions [Alur *et al.*, 2002; Chatterjee *et al.*, 2010; Mogavero *et al.*, 2014]. The resulting formalisms describe a rather abstract notion of agents' actions and strategies, which is appropriate for the various scenarios and use cases that have been successfully analysed within these frameworks [van der Hoek *et al.*, 2006; Čermák *et al.*, 2014].

On the other hand, substantial effort has been put towards making strategic abilities more precise, by grounding formal semantics in computational theories of agency [Wooldridge, 2000]. In this direction, Coalition Logic of Propositional Control (CL-PC) [van der Hoek and Wooldridge, 2005; Gerbrandy, 2006; van der Hoek *et al.*, 2010; 2011; Herzig *et al.*, 2011] attempts to offer an explanation of the effectivity functions of Pauly's Coalition Logic [Pauly, 2002] in terms of the agents' control over propositional atoms. Basically, the models of CL-PC consist in a partition $AP_1, \ldots, AP_n$ of the set of propositional atoms, where each AP_i is the set of atoms whose truth value is controlled by agent i. Here the three key assumptions of Propositional Control (PC) are apparent: it is *exhaustive* (every atom is controlled by *at least* one agent), *exclusive* (every atom is controlled by *at most* one agent), and actions are *unrestricted*: any assignment of i's atoms is available to i in any state. It has been argued that Propositional Control is suitable for the specification and verification of rich multi-agent systems [van der Hoek *et al.*, 2006; Troquard *et al.*, 2011; Ciná and Endriss, 2015; Herzig *et al.*, 2016]. In particular, [van der Hoek and Wooldridge, 2005] shows that if we translate Pauly's coalition formulas $\langle\langle A \rangle\rangle X \varphi$ – here written in the syntax of ATL, using the embedding of [Goranko, 2001] – into CL-PC formulas $\diamond_A \Box_{Ag \setminus A} \varphi$ then all principles of Coalition Logic are valid in CL-PC. However, CL-PC is strictly stronger than Coalition Logic: there are principles of the former that are not valid for the latter. This is mainly a consequence of the exclusiveness, exhaustiveness and unrestrictedness assumptions in CL-PC.

Our motivation for this paper is to provide a more fine-grained theoretical analysis of Propositional Control in logics for strategies, particularly w.r.t. Alternating-time Temporal Logic (ATL). We start by defining a semantics for ATL based on the same notion of Propositional Control as in CL-PC, namely, control is exclusive and exhaustive, and actions are unrestricted. These are strong hypotheses, which we will show to validate several counterintuitive principles. Thus, next we relax the exhaustiveness assumption in an attempt to move towards standard ATL. Ideally, our aim is to single out a PC-based class of Concurrent Game Structures (CGS) that has the same class of validities as standard ATL. This would mean that PC suffices as models for ATL: one might for example prove that an ATL formula is satisfiable by providing a PC-based CGS. Such a strong result is however unlikely as agents can interact in complex ways in CGS. Nonetheless, in this paper we take a first step towards this aim, by considering non-exhaustive PC with restricted actions. We analyse both semantics, including the model checking and satisfiability problems, and discuss differences w.r.t. standard ATL. Our results point out that unrestrictedness brings us immediately to the same complexity classes as full ATL.

Our paper is organized as follows. In Section 2 we present some preliminaries on CL-PC, ATL, and CGS. In Section 3 we introduce CGS-PC with unrestricted actions, and in Section 4 weak CGS-PC. For both classes we analyse the semantics and provide results for the model checking and satisfiability problems. We conclude by discussing some developments for future research. Proofs are often omitted for reasons of space, we only provide those we deem most significant.

2 Preliminaries

In this section we introduce the technical notions that will be used throughout the paper: CL-PC and CGS-based ATL. In the rest of the paper $Ag = \{1, \ldots, n\}$ is the set of agents, and AP is the set of atomic propositions. Also, we denote the complement of a set U (w.r.t. some given set $V \supseteq U$) by $\overline{U}$. Given a formula φ, we define AP_φ as the set of atoms occurring in φ. Furthermore, letting $A \subseteq Ag$ be the set of agents occurring in φ, we define Ag_φ as Ag if $A = Ag$, and as $Ag \cup \{e\}$ otherwise, for some $e \notin A$. (The fresh agent e will mimic the set of agents not occurring in φ.)

2.1 Coalition Logic of Propositional Control

The language of CL-PC is defined by the following BNF, for $p \in AP$ and $A \subseteq Ag$:

$$\varphi \quad ::= \quad p \mid \neg\varphi \mid \varphi \to \varphi \mid \diamond_A \varphi$$

where $\diamond_A \psi$ reads "coalition A has the contingent ability to achieve ψ". The other propositional connectives and the box operator $\Box_A$, dual of $\diamond_A$, are defined as standard.

Two semantics for CL-PC are provided in [van der Hoek and Wooldridge, 2005]: direct models and Kripke models. Since they are equivalent we here only present the former.

Definition 1 (Model)**.** *Given sets Ag of agents and AP of atoms, a* model $\mathcal{M}$ *is a partition $AP_1, \ldots, AP_{|Ag|}$ of AP.*

Every subset θ of AP can be identified with the valuation $\theta : AP \to \{\mathit{ff}, \mathit{tt}\}$ that assigns true tt to all atoms in θ and false ff to all atoms in $\overline{\theta}$. We will use the two presentations interchangeably, without explicit mention.

Definition 2 (Semantics for CL-PC)**.** *We define whether model $\mathcal{M}$* satisfies *formula φ according to valuation θ, or $(\mathcal{M}, \theta) \models \varphi$, as follows (we omit the clauses for propositional connectives as straightforward):*

$(\mathcal{M}, \theta) \models p$ *iff* $p \in \theta$
$(\mathcal{M}, \theta) \models \diamond_A \psi$ *iff for some* $\theta_A : \bigcup_{i \in A} AP_i \to \{\mathit{ff}, \mathit{tt}\}$, $(\mathcal{M}, \theta \oplus \theta_A) \models \psi$

where $\theta \oplus \theta_A = (\theta \cup \theta_A) \setminus (\bigcup_{i \in A} AP_i \setminus \theta_A)$ is the update *of θ according to θ_A.*

A formula φ is *true* in a model $\mathcal{M}$, or $\mathcal{M} \models \varphi$, iff $(\mathcal{M}, \theta) \models \varphi$ for all valuations $\theta \in 2^{AP}$; φ is *valid* in a class $\mathcal{K}$ of models iff $\mathcal{M} \models \varphi$ for all $\mathcal{M} \in \mathcal{K}$. Notice that given Ag and a partition $AP_1, \ldots, AP_{|Ag|}$ of AP, there exists a unique model $\mathcal{M}$.

Models for CL-PC were originally defined on *finite* sets of atoms. Here we do not make such an assumption, as it is not normally considered in temporal logics. Moreover, by the following result, the two accounts are equivalent w.r.t. satisfaction of formulas.

Lemma 1. *Given model $\mathcal{M}$ based on Ag, with partition $AP_1, \ldots, AP_{|Ag|}$ of AP, valuation θ, and formula φ, we have that $(\mathcal{M}, \theta) \models \varphi$ iff $(\mathcal{M}_\varphi, \theta_\varphi) \models \varphi$, where*

- *$\mathcal{M}_\varphi$ is the model based on the set of agents Ag_φ and atoms AP_φ, with partition $AP_1 \cap AP_\varphi, \ldots, AP_{|Ag'|} \cap AP_\varphi$ and, possibly, $AP_e = \bigcup_{i \in \overline{Ag'}} AP_i \cap AP_\varphi$ (in case the fresh agent e is present);*
- *θ_φ is the restriction of θ to AP_φ.*

We observed that, since coalition Logic corresponds to the X-fragment of ATL [Goranko, 2001], we can express the ATL formula $\langle\!\langle A \rangle\!\rangle X \psi$ in CL-PC as $\diamond_A \Box_{\overline{A}} \psi$. This remark prompts the question of interpreting ATL on CL-PC models. To do so, we introduce formally ATL and its semantics.

2.2 ATL and Concurrent Game Structures

Formulas in ATL are defined by the following BNF, for $p \in AP$ and $A \subseteq Ag$:

$$\varphi \quad ::= \quad p \mid \neg\varphi \mid \varphi \to \varphi \mid \langle\!\langle A \rangle\!\rangle X \varphi \mid \langle\!\langle A \rangle\!\rangle G \varphi \mid \langle\!\langle A \rangle\!\rangle \varphi U \varphi$$

The formula $\langle\!\langle A \rangle\!\rangle X \varphi$ reads "the agents in A have a strategy to enforce φ at the next state, *no matter what the other agents do*". Formulas $\langle\!\langle A \rangle\!\rangle G \varphi$ and $\langle\!\langle A \rangle\!\rangle (\varphi U \varphi')$ read accordingly, where X stands for 'next' and U stands for 'until'.

The semantics of ATL is in terms of a standard framework for the representation of games: Concurrent Game Structures.

Definition 3 (CGS)**.** *Given sets Ag of agents and AP of atoms, a* Concurrent Game Structure *(CGS) is a tuple $\mathcal{G} = \langle S, Act, d, \tau, \pi \rangle$ such that*

- *S is a non-empty set of states;*
- *Act is a non-empty set of individual actions;*
- *$d : Ag \times S \to (2^{Act} \setminus \{\varnothing\})$ is the protocol function that returns the actions available to agents at each state;*
- *$\tau : S \times Act^{|Ag|} \to S$ is the transition function such that, for every $s \in S$ and joint action $\alpha \in Act^{|Ag|}$, $\tau(s, \alpha)$ is defined iff $\alpha_i \in d(i, s)$ for every $i \in Ag$;*
- *$\pi : S \to 2^{AP}$ is the state-labeling function.*

We remark that in Def. 3, states, actions, and the transition function are given as completely abstract notions: nothing is specified about the internal structure of states, about the nature of actions, nor about the computational properties of the transition function. Obviously, this responds to the need for a purely mathematical concept, proper of formal semantics. Nonetheless, we are interested in filling these abstract notions with some computational content, in order to give a computationally-grounded semantics to ATL [Wooldridge, 2000], specifically by using Propositional Control. But first we provide the interpretation of ATL on CGS. Hereafter we write $s \xrightarrow{\alpha} s'$ whenever $s' = \tau(s, \alpha)$.

Definition 4 (Strategy)**.** *Given a CGS $\mathcal{G} = \langle S, Act, d, \tau, \pi \rangle$, a* strategy *for an agent $i \in Ag$ is a function $f : S \to Act$ that is compatible with d, i.e., for every $s \in S$, $f(s) \in d(i, s)$.*

Since the strategies in Def. 4 only take into account the present state of the system, they are called positional or memoryless [Bulling *et al.*, 2010]. We recall that in contexts of perfect information, as the present one, semantically there is no difference between positional strategies and strategies with perfect recall. We work with the former for simplicity.

A *collective strategy* for a coalition A is a set $f_A = \{f_i \mid i \in A\}$ of strategies. If f_A is a collective strategy, $s \in S$, and $\alpha \in Act^{|Ag|}$, then $\alpha \in \hat{f}_A(s)$ whenever *(i)* for $i \in A$, $\alpha_i \in f_i(s)$; and *(ii)* for $i \in \overline{A}$, $\alpha_i \in d(i, s)$. Further, the *outcome* $out(s, f_A)$ is the set of all infinite runs $\lambda = s, s_1, s_2, \ldots$ such that for every $j \geqslant 0$, for some $\alpha \in \hat{f}_A(s_j)$, we have $s_j \xrightarrow{\alpha} s_{j+1}$. For a run λ and $j \geqslant 0$, $\lambda[j]$ denotes the j+1-th state s_j.

Definition 5 (Semantics for ATL). *We define whether CGS $\mathcal{G}$ satisfies formula φ at state s, or $(\mathcal{G}, s) \models \varphi$, as follows (again we omit clauses for propositional connectives):*

$(\mathcal{G}, s) \models p$ *iff* $p \in \pi(s)$
$(\mathcal{G}, s) \models \langle\!\langle A\rangle\!\rangle X\psi$ *iff for some* f_A, *for every* $\lambda \in out(s, f_A)$, $(\mathcal{G}, \lambda[1]) \models \psi$
$(\mathcal{G}, s) \models \langle\!\langle A\rangle\!\rangle G\psi$ *iff for some* f_A, *for every* $\lambda \in out(s, f_A)$, *for every* $j \geqslant 0$, $(\mathcal{G}, \lambda[j]) \models \psi$
$(\mathcal{G}, s) \models \langle\!\langle A\rangle\!\rangle \psi U \psi'$ *iff for some* f_A, *for every* $\lambda \in out(s, f_A)$, *for some* $j \geqslant 0$, $(\mathcal{G}, \lambda[j]) \models \psi'$, *and* $j > n \geqslant 0$ *implies* $(\mathcal{G}, \lambda[n]) \models \psi$

A formula φ is *true* in a CGS $\mathcal{G}$, or $\mathcal{G} \models \varphi$, iff $(\mathcal{G}, s) \models \varphi$ for all states $s \in S$; φ is *valid* in a class $\mathcal{K}$ of CGS iff $\mathcal{G} \models \varphi$ for all CGS $\mathcal{G} \in K$. We denote by $Val(\mathcal{K})$ the set of ATL-validities in class $\mathcal{K}$.

3 CGS with Propositional Control

We observed that the notion of CGS provided in Def. 3 is fairly abstract. For instance, the transition function τ is unconstrained: given a state and an action, τ can in principle return any state in S as a successor. In this section we introduce a notion of CGS that is based on Propositional Control. As customary in CL-PC, we assume that the set AP of atoms is partitioned into sets $AP_i \subseteq AP$, where $i \in Ag$. Further, for every $P \subseteq AP$, we consider atomic actions $+P$ and $-P$, which respectively represent the action of setting the atoms in P to true and false.

We now introduce CGS with Propositional Control by providing concrete instances to the abstract elements in Def. 3.

Definition 6 (CGS-PC). *A* Concurrent Game Structure for Propositional Control *is a CGS* $\mathcal{G} = \langle S, Act, d, \tau, \pi\rangle$ *such that*

- $S = 2^{AP_1} \times \ldots \times 2^{AP_{|Ag|}}$, *where each* 2^{AP_i} *is the set of valuations* $\theta_i \subseteq AP_i$ *(which we identify with functions* $\theta_i : AP_i \to \{ff, tt\}$*);*[1]
- *Act is the set of actions* $(+P, -P')$, *for disjoint* $P, P' \subseteq AP$*;*
- *protocol d satisfies:* $(+P, -P') \in d(i, s)$ *iff* $P, P' \subseteq AP_i$*;*
- *transition function* τ *is such that, for every* $s \in S$ *and* $\alpha \in Act^{|Ag|}$, *if* $s' = \tau(s, \alpha)$, *then each* s'_i *is obtained by updating* s_i *according to* $\alpha_i = (+P, -P')$, *that is,* $s'_i = (s_i \cup P) \setminus P'$*;*
- *function* π *is such that* $\pi((s_1, \ldots, s_{|Ag|})) = \bigcup_{i\in Ag} s_i$.

By Def. 6, the actions of each agent i are *unrestricted*: all and only actions involving atoms in AP_i are enabled for i. We require sets P and P' in action $(+P, -P')$ to be disjoint for ease of presentation (the restriction can be avoided by some mechanism resolving conflicting assignments.) We refer to $+P$ (resp. $-P'$) as the *positive* (resp. *negative*) *effects* of the action. In what follows we write actions $+\{p\}$ and $-\{p\}$, for $p \in AP$, simply as $+p$ and $-p$. As it is the case for CL-PC, given sets Ag of agents and AP of atoms with partition $AP_1, \ldots, AP_{|Ag|}$, there is a unique CGS-PC built on Ag and AP. Notice also that the transition relation $\to$ such that $s \to s'$ iff $s \xrightarrow{\alpha} s'$ for some joint α, is *universal*. Indeed, given states s and s', for every $i \in Ag$, consider action $\beta_i = (+(s'_i \setminus s_i), -(s_i \setminus s'_i))$. Then, we have that $(s'_i \setminus s_i)$ and $(s_i \setminus s'_i)$ are disjoint, and $s \xrightarrow{\beta} s'$.

[1]We could as well set S to 2^{AP}, but prefer the current presentation in view of relaxing exhaustivity of control.

Clearly, CGS-PC are a particular instance of CGS, that is, the class $CGS{-}PC$ of all CGS-PC is a subclass of the class CGS of all CGS. In particular, the set $Val(CGS)$ of validities in CGS is a subset of $Val(CGS{-}PC)$. In the next section we will see that this inclusion is strict.

We now consider some formulas that are valid in $CGS{-}PC$ but not in CGS, as well as the computational properties of the former. These provide interesting insights on the impact of the assumptions underlying Propositional Control in CGS, and the distance of the latter w.r.t. standard ATL.

First, all ATL operators can be reduced to $\langle\!\langle A\rangle\!\rangle X$.

Lemma 2. *The following formulas are valid in* $CGS{-}PC$*:*

$$\langle\!\langle A\rangle\!\rangle G\varphi \leftrightarrow \varphi \land \langle\!\langle A\rangle\!\rangle X\varphi \quad (1)$$

$$\langle\!\langle A\rangle\!\rangle(\varphi U \varphi') \leftrightarrow \varphi' \lor (\varphi \land \langle\!\langle A\rangle\!\rangle X\varphi') \quad (2)$$

$$\langle\!\langle A\rangle\!\rangle F\varphi \leftrightarrow \varphi \lor \langle\!\langle A\rangle\!\rangle X\varphi \quad (3)$$

Proof. We prove (1). The '$\to$' direction holds in standard CGS. As to the '$\leftarrow$' direction, suppose $(\mathcal{G}, s) \models \varphi \land \langle\!\langle A\rangle\!\rangle X\varphi$, that is, $(\mathcal{G}, s) \models \varphi$ and for some action tuple α_A, for all action tuples $\alpha_{\overline{A}}$, $s \xrightarrow{\alpha_A \cdot \alpha_{\overline{A}}} s'$ implies $(\mathcal{G}, s') \models \varphi$. Now consider action tuple α'_A s.t. $\alpha'_i = (+\varnothing, -\varnothing)$ for every $i \in A$, and any action tuple $\alpha'_{\overline{A}}$. If $s' \xrightarrow{\alpha'_A \cdot \alpha'_{\overline{A}}} s''$ then also $s \xrightarrow{\alpha_A \cdot \alpha''_{\overline{A}}} s''$ by considering action $\alpha''_i = (+(s''_i \setminus s_i), -(s_i \setminus s''_i))$ for every $i \in \overline{A}$. But then $(\mathcal{G}, s'') \models \varphi$ by hypothesis. By reasoning along this way, we find a strategy f_A ("first ensure φ and then do nothing") such that for every $\lambda \in out(s, f_A)$ and $j \geqslant 0$, $(\mathcal{G}, \lambda[j]) \models \varphi$. The proofs for (2) and (3) are similar. ☐

As a consequence of Lemma 2, we can think of the LHS of (1)-(3) as shorthands for the RHS. However, the expansion is exponential in the length of the original formula.

The next equivalences follow from Lemma 2 and the universality of the transition relation $\to$.

Lemma 3. *The following formulas are valid in* $CGS{-}PC$*:*

$$\langle\!\langle A\rangle\!\rangle X\varphi \leftrightarrow (\varphi \land \langle\!\langle A\rangle\!\rangle G\varphi) \lor (\neg\varphi \land \langle\!\langle A\rangle\!\rangle F\varphi)$$

$$\langle\!\langle A\rangle\!\rangle X\varphi \leftrightarrow \langle\!\langle A\rangle\!\rangle X \langle\!\langle A\rangle\!\rangle X\varphi$$

Hereafter we consider some interesting consequences of the exclusiveness and exhaustiveness of Propositional Control, as formulated for atomic propositions, where $\dot\bigvee_{i\in A}\varphi$ abbreviates $\bigvee_{i\in A}(\varphi \land \bigwedge_{j\in A, j\neq i} \neg\varphi[j/i])$ and where $\varphi[j/i]$ results from uniformly replacing all occurrences of i in φ by j.

Lemma 4. *The following formulas are valid on* $CGS{-}PC$*:*

$$\langle\!\langle A\rangle\!\rangle X(p \lor q) \leftrightarrow \langle\!\langle A\rangle\!\rangle Xp \lor \langle\!\langle A\rangle\!\rangle Xq \quad (4)$$

$$\langle\!\langle A\rangle\!\rangle X(p \land q) \leftrightarrow \langle\!\langle A\rangle\!\rangle Xp \land \langle\!\langle A\rangle\!\rangle Xq \quad (5)$$

$$\langle\!\langle A\rangle\!\rangle Xp \leftrightarrow \dot\bigvee_{i\in A} \langle\!\langle i\rangle\!\rangle Xp \quad (6)$$

$$\langle\!\langle A\rangle\!\rangle X\neg p \leftrightarrow \dot\bigvee_{i\in A} \langle\!\langle i\rangle\!\rangle X\neg p \quad (7)$$

$$\langle\!\langle A\rangle\!\rangle Xp \leftrightarrow \langle\!\langle A\rangle\!\rangle X\neg p \quad (8)$$

Proof. As regards (4), the '←' direction is standard in CGS. As to the '→' direction, suppose that $(\mathcal{G}, s) \models \langle\!\langle A \rangle\!\rangle X(p \vee q)$. This means that for some action tuple α_A, for all action tuples $\alpha_{\overline{A}}$, $s \xrightarrow{\alpha_A \cdot \alpha_{\overline{A}}} s'$ implies $(\mathcal{G}, s') \models p \vee q$. If $(\mathcal{G}, s') \models p$, then for some $i \in A$, $p \in AP_i$, and therefore $(\mathcal{G}, s) \models \langle\!\langle i \rangle\!\rangle Xp$. By coalition monotonicity we obtain $(\mathcal{G}, s) \models \langle\!\langle A \rangle\!\rangle Xp$, and finally $(\mathcal{G}, s) \models \langle\!\langle A \rangle\!\rangle Xp \vee \langle\!\langle A \rangle\!\rangle Xq$. The case $(\mathcal{G}, s') \models q$ is analogous. The proof of (5) is similar; whereas formulas (6) and (7) are valid because control is exhaustive. □

Intuitively, (4) and (5) entail that a coalition A controls a non-tautological disjunction (resp. satisfiable conjunction) of atoms iff A controls each disjunct (resp. conjunct). Also, by (6) and (7) a coalition A controls an atom p iff p is controlled by exactly one member of A.

The following validities also illustrate the role of exclusiveness and exhaustiveness of control in CGS-PC.

Lemma 5. *The following formulas are valid on $CGS{-}PC$:*

$$\langle\!\langle Ag \rangle\!\rangle Xp \tag{9}$$

$$\langle\!\langle Ag \rangle\!\rangle X\neg p \tag{10}$$

$$\dot{\bigvee}_{i \in Ag} \langle\!\langle \varnothing \rangle\!\rangle G\big(\langle\!\langle i \rangle\!\rangle Xp \wedge \langle\!\langle i \rangle\!\rangle X\neg p\big) \tag{11}$$

Observe that the above principles (1)-(11) are sometimes too restrictive. Consider 4: it may be the case that I can throw a coin on a chessboard without being able to throw it on a white filed or a black field.

Using the above lemmas, we now prove the following result on the relationship between CL-PC and the Concurrent Game Structures for Propositional Control. Recall that in CL-PC, $\langle\!\langle A \rangle\!\rangle X\varphi$ is a shorthand for $\diamond_A \square_{\overline{A}} \varphi$. Moreover, abusing notation a bit, we consider $\langle\!\langle A \rangle\!\rangle G$-, $\langle\!\langle A \rangle\!\rangle F$-, and $\langle\!\langle A \rangle\!\rangle U$-formulas as shorthands according to (1)-(3).

Theorem 6. *For every ATL formula φ, $CL{-}PC \models \varphi$ iff $CGS{-}PC \models \varphi$.*

Proof. '⇐': Suppose $CL{-}PC \not\models \varphi$, that is, for some model $\mathcal{M}$ and valuation θ of AP, $(\mathcal{M}, \theta) \not\models \varphi$. Then, consider the (unique) CGS-PC $\mathcal{G}$ defined on the same Ag and the same partition of AP as $\mathcal{M}$. We now prove that for every ATL formula ψ, $(\mathcal{M}, \theta) \models \psi$ iff $(\mathcal{G}, \theta) \models \psi$, by induction on ψ. For $\psi = p$, $(\mathcal{M}, \theta) \models \psi$ iff $p \in \theta$, iff $(\mathcal{G}, \theta) \models \psi$. The inductive cases for propositional connectives are straightforward. As for ATL operators, if $(\mathcal{M}, \theta) \models \langle\!\langle A \rangle\!\rangle X\psi'$, then for some A-valuation θ_A, for every $\overline{A}$-valuation $\theta_{\overline{A}}$, $(\mathcal{M}, (s \oplus \theta_A) \oplus \theta_{\overline{A}}) \models \psi'$. But this means that for some A-action θ_A, for every $\overline{A}$-action $\theta_{\overline{A}}$, $\theta \xrightarrow{\theta_A \cdot \theta_{\overline{A}}} \theta'$ implies $(\mathcal{G}, \theta') \models \psi'$ by induction hypothesis, i.e., $(\mathcal{G}, \theta) \models \langle\!\langle A \rangle\!\rangle X\psi'$. As a result, $(\mathcal{G}, \theta) \not\models \varphi$, and therefore φ is not a validity in $CGS{-}PC$ either.

We omit the '⇒'-direction, which is proved similarly. □

By Theorem 6, $CGS{-}PC$ is indeed the class of CGS that correspond to $CL{-}PC$ w.r.t. the language of ATL, as the two frameworks share the same set of validities. Note that Theorem 6 does not provide a complexity result, given that the expansion of abbreviations may result in exponential growth.

As anticipated in the introduction, our aim is to analyse the computational properties of ATL interpreted on CGS-PC. Specifically, we are interested in the following problems:

- **Model checking**: given a CGS-PC $\mathcal{G}$, a state s, and a formula φ, determine whether $(\mathcal{G}, s) \models \varphi$.
- **Satisfiability**: given a formula φ, determine whether φ is satisfied in some model.

As an auxiliary step to study these problems, consider the following corollary of Theorem 6 and Lemma 1.

Corollary 7. *Given CGS-PC $\mathcal{G}$ based on Ag, with partition $AP_1, \ldots, AP_{|Ag|}$ of AP, state s, and formula φ, we have that $(\mathcal{G}, s) \models \varphi$ iff $(\mathcal{G}_\varphi, s_\varphi) \models \varphi$, where*

- *$\mathcal{G}_\varphi$ is the CGS-PC based on sets AP_φ and Ag_φ as defined in Lemma 1;*
- *s_φ is the restriction of s to AP_φ.*

By Corollary 7, the size of the input for the model checking problem can be given as $|\mathcal{G}_\varphi| + |\varphi|$, where the size $|\mathcal{G}_\varphi| = |Ag_\varphi| + |AP_\varphi|$ of $\mathcal{G}_\varphi$ is polynomial in φ; whereas satisfiability can be restricted to CGS-PC built on Ag_φ and AP_φ.

We are now able to prove the following.

Theorem 8. *Both the model checking problem and the satisfiability problem for $CGS{-}PC$ are Δ_3^P-complete.*

Proof. As regards model checking, we follow [Bulling *et al.*, 2010]. For the lower bound we make use of QSAT_2, a Σ_2^P-complete problem, as an intermediate step. Specifically, given an instance $\exists p_1, \ldots, p_r \forall p_{r+1}, \ldots, p_k \varphi$ of QSAT_2, for boolean φ built on atoms $p_1, \ldots, p_k$, we consider the CGS-PC $\mathcal{G}$ defined on $Ag = \{1, 2\}$ s.t. $AP_1 = \{p_1, \ldots, p_r\}$ and $AP_2 = \{p_{r+1}, \ldots, p_k\}$. Then, QSAT_2 is reduced to model checking $\langle\!\langle 1 \rangle\!\rangle X\varphi$. Finally, to obtain Δ_3^P-hardness, $\langle\!\langle 1 \rangle\!\rangle X\varphi$ is combined with nested cooperation modalities, so as to reduce the SNSAT_3 problem along the lines of [Laroussinie *et al.*, 2008].

As to the upper bound, we outline the following procedure for checking $(M, s) \models \langle\!\langle A \rangle\!\rangle X\varphi$ with no nested modalities. First, we guess an action tuple α_A. Then, we check if the CTL formula $AX\varphi$ is true in state s of the resulting model by asking an oracle for counteractions $\alpha_{\overline{A}}$ to make $X\varphi$ false, and then we revert the oracle's reply. Nested ATL modalities can be dealt with in polynomial time.

As regards satisfiability, notice that we can check whether a formula φ is satisfiable by model checking it on the CGS-PC $\mathcal{G}_\varphi$, as defined in Corollary 7. Hence, the problem is in Δ_3^P. As to hardness, we can reduce SNSAT_3 to CGS-PC satisfiability similarly to what done above for the model checking problem. □

As a consequence of Theorem 8, model checking CGS-PC has exactly the same complexity as the implict model checking problem for ATL [Bulling *et al.*, 2010]. On the other hand, satisfiability for standard ATL is EXPTIME-complete. So the assumptions underlying PC have a noticeable computational impact on satisfiability. Observe that our complexity results differ from those for CL-PC, where both model checking and satisfiability are PSPACE complete [van der Hoek and Wooldridge, 2005].

4 Weakening CGS-PC

In Section 3 we introduced the class of CGS-PC, which we proved to have the same set of validities as CL-PC. However, we may consider some of these validities (e.g., those in Lemma 2) as being too strong and look for a more ATL-like semantics. Ideally, one would like to define a class $\mathcal{K}$ of CGS for Propositional Control such that $Val(\mathcal{K}) = Val(CGS)$. We discussed the limitation of such an endeavour when it comes to interaction between agents. Nonetheless, in this section we move a step towards standard CGS by dropping the assumptions of exhaustiveness and unrestricted actions. Specifically, we introduce the notion of *weak CGS-PC*, or CGS-PC$^-$.

Definition 7 (CGS-PC$^-$). *A* weak Concurrent Game Structure for Propositional Control *is a CGS* $\mathcal{G} = \langle S, Act, d, \tau, \pi \rangle$ *such that*

- *S, Act, τ, π are defined as for CGS-PC;*
- *the protocol d satisfies: $(+P, -P') \in d(i,s)$* only if $P, P' \subseteq AP_i$*;*

By Def. 7 the difference between CGS-PC$^-$ and CGS is that control is no longer exhaustive and protocol d no longer allows for changing any atom at any state. We will see that these small changes have a noticeable impact on validities. First of all, since an agent a might have only a subset of the set of actions $(+P, -P')$ with $P \cup P' \subseteq AP_a$ available at a state s, the transition relation $\rightarrow$ between states is neither reflexive, nor symmetric, nor transitive. (For example, reflexivity fails in a CGS-PC as soon as there is an agent without the empty action $(\varnothing, \varnothing)$ in her repertoire.) So it is not a universal relation. This contrasts with the situation for CGS-PC in Section 3.

Clearly, CGS-PC are particular instances of CGS-PC$^-$, which in turn are instances of CGS. In particular, we have the following strict inclusions:

$$Val(CGS) \subset Val(CGS\text{-}PC^-) \subset Val(CGS\text{-}PC)$$

as a result of the following lemma (and subsequent results).

Lemma 9. *Formulas (1)-(11) are not valid in $CGS\text{-}PC^-$;*

Proof. We provide counterexample CGS-PC$^-$ for (3) and (4).

As to (3), consider the CGS-PC$^-$ $\mathcal{G} = \langle S, Act, d, \tau, \pi \rangle$, defined on set $Ag = \{a\}$ of agents, s.t. for every $s \in S$, $d(a,s) = \{(+p, -\varnothing) \mid p \in AP_a\}$, that is, agent a can only set the value of exactly one atom to true in any state. Consider now state $s_0 = \langle \theta_a \rangle$ for $\theta_a = \varnothing$. Clearly, for $q, r \in AP_a$, $(\mathcal{G}, s_0) \models \langle\!\langle a \rangle\!\rangle F(q \wedge r)$. However, $(\mathcal{G}, s_0) \not\models q \wedge r$ and $(\mathcal{G}, s_0) \not\models \langle\!\langle a \rangle\!\rangle X(q \wedge r)$.

As to (4), consider the CGS-PC$^-$ $\mathcal{G} = \langle S, Act, d, \tau, \pi \rangle$, defined on set $Ag = \{a, b\}$ of agents, s.t. for every $s \in S$, $d(a,s) = \{(+p_a, -\varnothing) \mid p_a \in AP_a\}$; while $d(b,s) = \{(+p_b, -\varnothing), (+p'_b, -\varnothing)\}$, that is, agent b can (and must) set the value of exactly one of atoms p_b, p'_b to true in any state. Consider state $s_0 = \langle \theta_a, \theta_b \rangle$ for $\theta_a = \theta_b = \varnothing$. Clearly, $(\mathcal{G}, s_0) \models \langle\!\langle a \rangle\!\rangle X(p_b \vee p'_b)$. However, $(\mathcal{G}, s_0) \not\models \langle\!\langle a \rangle\!\rangle X p_b$ and $(\mathcal{G}, s_0) \not\models \langle\!\langle a \rangle\!\rangle X p'_b$.

As to (6) and (7), suppose none of the agents has an action making p true in his repertoire: then $\langle\!\langle A \rangle\!\rangle Xp$ is true for every A, i.e., the negative condition in $\dot{\bigvee}_{i \in A} \langle\!\langle i \rangle\!\rangle Xp$ fails.

Generally speaking, (1)-(11) all fail due to restrictions on enabled actions. □

The fact that control is exclusive (for all $i, j \in Ag$, $AP_i \cap AP_j = \varnothing$), that the actions available to agent i may involve only a proper subset of 'her' actions on AP_i, and that control is constant in time makes that only a weakened version of (11) is valid.

Lemma 10. *The following formula is valid on $CGS\text{-}PC^-$:*

$$\bigvee_{i \in Ag} \langle\!\langle \varnothing \rangle\!\rangle G \big((\langle\!\langle Ag \rangle\!\rangle Xp \rightarrow \langle\!\langle i \rangle\!\rangle Xp) \wedge (\langle\!\langle Ag \rangle\!\rangle X\neg p \rightarrow \langle\!\langle i \rangle\!\rangle X \neg p) \big) \tag{12}$$

Proof. Let i be the agent s.t. $p \in AP_i$, if any; any agent in Ag, otherwise. In the latter case, if $(\mathcal{G}, s) \models \langle\!\langle Ag \rangle\!\rangle Xp$ then for some $s' \in \mathcal{G}$, $s \rightarrow s'$ and $(\mathcal{G}, s') \models p$. But the truth value of p has not changed in the transition from s to s', that is, $(\mathcal{G}, s) \models p$. Since no agent can affect the truth value of p, $(\mathcal{G}, s) \models \langle\!\langle i \rangle\!\rangle Xp$. On the other hand, if $p \in AP_i$ for some $i \in Ag$, then we have two subcases to consider. If action $+p$ is not enabled in state s, then once again the value of p does not changed from s to s', that is, $(\mathcal{G}, s) \models p$ and also $(\mathcal{G}, s) \models \langle\!\langle i \rangle\!\rangle Xp$. If $+p$ does appear in some action available to i in s, then $(\mathcal{G}, s) \models \langle\!\langle i \rangle\!\rangle Xp$ by setting the value of p to true. The proof for $(\mathcal{G}, s) \models \langle\!\langle Ag \rangle\!\rangle X \neg p$ is similar. □

On the other hand, Cor. 7 does not hold for CGS-PC$^-$. First of all, notice that, given a CGS-PC$^-$, $\mathcal{G}_\varphi$ as defined in Cor. 7 is a CGS-PC. Therefore, if $\mathcal{G}$ is the first CGS-PC$^-$ in the proof of Lemma 9, we have that $(\mathcal{G}, s) \not\models (3)$. However, $(\mathcal{G}_{(3)}, s_{(3)}) \models (3)$, as (3) is valid in the class of CGS-PC.

For CGS-PC$^-$ we have the following result, according to a different notion of restriction $\mathcal{G}_\varphi$.

Lemma 11. *Given a CGS-PC$^-$ $\mathcal{G}$ based on Ag and $AP_1, \ldots, AP_{|Ag|}$, state s, and formula φ, we have that $(\mathcal{G}, s) \models \varphi$ iff $(\mathcal{G}_\varphi, s_\varphi) \models \varphi$, where*

- *$\mathcal{G}_\varphi = \langle S_\varphi, Act_\varphi, d_\varphi, \tau_\varphi, \pi_\varphi \rangle$ is the CGS-PC$^-$ based on set Ag_φ of agents defined as in Lemma 1, and on set $AP_\varphi \cup S$ of all atoms appearing in φ together with the states in $\mathcal{G}$ as new atoms, with $AP'_i = AP_i \cap AP_\varphi$ for every $i \in Ag_\varphi$, and $AP_e = S$. Moreover, $\alpha \in d(i,s)$ iff the restriction $\alpha_{|_{AP'_i}}$ belongs to $d_\varphi(i, s')$ for $s'_e = s$ and $i \in Ag$, and $\alpha \in d(e, s')$ iff $\alpha = (-s'_e, +t)$, where t is a successor of s'_e in $\mathcal{G}$.*
- *s_φ is the tuple $(\theta_1 \cap AP_\varphi, \ldots, \theta_{Ag_\varphi} \cap AP_\varphi, s)$.*

Notice that the restriction $\mathcal{G}_\varphi$ for CGS-PC$^-$ is an infinite state systems in general, differently from the case for CL-PC and CGS-PC.

We can now consider the model checking and satisfiability problems for CGS-PC$^-$.

Theorem 12. *1. The model checking problem for $CGS\text{-}PC^-$ is Δ_3^P-complete.*

2. The satisfiability problem for $CGS\text{-}PC^-$ is EXPTIME-hard.

Proof. As to model checking, the result follows from the corresponding result for $CGS\text{-}PC$. Specifically, the lower bound follows by remarking that the CGS-PC used in the reduction of SNSAT_3 in Theorem 8 is a CGS-PC$^-$ trivially. The upper bound follows from the result available for implicit model checking of ATL w.r.t. CGS [Bulling *et al.*, 2010].

As to satisfiability, we follow [van der Hoek *et al.*, 2006] and reduce the problem of deciding whether a given agent has a winning strategy in the two-player game PEEK-G_4 [Stockmeyer and Wong, 1979]. An instance of PEEK-G_4 consists in a quadruple (X_0, X_1, X_2, Win) where X_0, X_1 and X_2 are finite sets of propositional variables such that $X_0 \subseteq X_1 \cup X_2$, X_1 and X_2 are disjoint, and Win is a propositional formula over $X_1 \cup X_2$. The idea is that X_0 is the initial valuation and that X_1 and X_2 are variables that are respectively under the control of agent 1 and 2. At X_0, 1 starts by selecting a variable in X_1 and assigning it to either true or false (possibly leaving it unchanged); then 2 selects a variable in X_2 and assigns it; and so on. An agent wins if his move makes the winning condition φ true. The problem is to decide whether 2 has a winning strategy in a given instance (X_0, X_1, X_2, Win) of the game. Given (X_0, X_1, X_2, Win), consider the following formulas describing it in the language of ATL:

$$\varphi_0 = \Big(\bigwedge_{p \in X_0} p\Big) \wedge \Big(\bigwedge_{p \notin X_0} \neg p\Big) \wedge t_1$$

$$\varphi_1 = \langle\!\langle \varnothing \rangle\!\rangle G\Big(\big(t_1 \to \langle\!\langle \varnothing \rangle\!\rangle X \neg t_1\big) \wedge \big(\neg t_1 \to \langle\!\langle \varnothing \rangle\!\rangle X t_1\big)\Big)$$

$$\varphi_2 = \langle\!\langle \varnothing \rangle\!\rangle G\Big(t_1 \to \bigwedge_{p_1 \in X_1} \big(\langle\!\langle 1 \rangle\!\rangle X p_1 \wedge \langle\!\langle 1 \rangle\!\rangle X \neg p_1\big)\Big)$$

$$\varphi_3 = \langle\!\langle \varnothing \rangle\!\rangle G\Big(t_1 \to \bigwedge_{p_2 \in X_2} \big((p_2 \to \langle\!\langle \varnothing \rangle\!\rangle X p_2) \wedge (\neg p_2 \to \langle\!\langle \varnothing \rangle\!\rangle X \neg p_2)\big)\Big)$$

$$\varphi_4 = \langle\!\langle \varnothing \rangle\!\rangle G\Big(\neg t_1 \to \bigwedge_{p_2 \in X_2} \big(\langle\!\langle 2 \rangle\!\rangle X p_2 \wedge \langle\!\langle 2 \rangle\!\rangle X \neg p_2\big)\Big)$$

$$\varphi_5 = \langle\!\langle \varnothing \rangle\!\rangle G\Big(\neg t_1 \to \bigwedge_{p_1 \in X_1} \big((p_1 \to \langle\!\langle \varnothing \rangle\!\rangle X p_1) \wedge (\neg p_1 \to \langle\!\langle \varnothing \rangle\!\rangle X \neg p_1)\big)\Big)$$

$$\varphi_6 = \langle\!\langle \varnothing \rangle\!\rangle G \bigwedge_{p,q \in X_1 \cup X_2} \Big(\big((p \wedge q) \to \langle\!\langle \varnothing \rangle\!\rangle X (p \vee q)\big) \wedge \big((p \wedge \neg q) \to \langle\!\langle \varnothing \rangle\!\rangle X (p \vee \neg q)\big) \wedge \big((\neg p \wedge q) \to \langle\!\langle \varnothing \rangle\!\rangle X (\neg p \vee q)\big) \wedge \big((\neg p \wedge \neg q) \to \langle\!\langle \varnothing \rangle\!\rangle X (\neg p \vee \neg q)\big)\Big)$$

Formula φ_0 characterizes the initial state, where t_1 expresses that it is agent 1's turn. φ_1 says that the agents play in turns. φ_2 says that at her turn, 1 can assign each of her variables at will, while φ_3 says that at that turn, 2 cannot modify any of her variables. φ_4 and φ_5 say the same thing for 2. φ_6 says that at each turn at most one variable can change truth value. Player 2 has a strategy guaranteeing *Win* if and only if

$$(\varphi_0 \wedge \varphi_2 \wedge \varphi_3 \wedge \varphi_4 \wedge \varphi_5 \wedge \varphi_6) \to \langle\!\langle 2 \rangle\!\rangle F\, Win$$

is valid in CGS-PC$^-$. Moreover, the length of $\varphi_0 \wedge \cdots \wedge \varphi_6$ is quadratic in the cardinality of $X_1 \cup X_2$. □

We conjecture that EXPTIME membership of the satisfiability problem for $CGS\text{-}PC^-$ can be shown by proving that φ is satisfiable in $CGS\text{-}PC^-$ iff $\varphi \wedge \psi$ is satisfiable in CGS, where formula ψ is of length polynomial in the length of φ and characterizes $CGS\text{-}PC^-$. Such a formula should characterize that control is exclusive and constant: we conjecture that ψ is the formula (12) of Lemma 10.

5 Conclusion

In this paper we took the first steps to fill the gap between ATL and CL-PC by introducing two classes of CGS that are based on Propositional Control. The first class $CGS\text{-}PC$ consists in a single structure (modulo agents and partition of atoms) that can be represented in a compact way, and whose complexity is the same as CL-PC. The second $CGS\text{-}PC^-$ is a family of structures, depending on the restriction of action availability at each state, whose computational properties closely resemble those of standard ATL.

Related Literature. Recent years have witnessed a growing interest in various forms of Propositional Control, possibly combined with dynamic and epistemic aspects [van der Hoek *et al.*, 2011; Balbiani *et al.*, 2013; 2014]. It is beyond the scope of this paper to provide an exhaustive account. We focus on [van der Hoek *et al.*, 2006], which is closest to our contribution. The approach of this work is based on Simple Reactive Modules (SML), which are basically agents whose propositional control is described by rules of the form $\varphi \rightsquigarrow (+P, -P')$, where φ is a boolean condition and $(+P, -P')$ is an action. In this respect SML can be seen as a class strictly included between CGS-PC and weak CGS-PC, different from both. Also, while our motivation is mainly theoretical, the focus in [van der Hoek *et al.*, 2006] is on the verification of multi-agent systems.

Future Work. There are a many interesting extensions of the present framework for future work. A low-hanging fruit is to add control changing actions to CGS, as done in [van der Hoek *et al.*, 2010]. One of the consequences, for instance, is that formula (12) has to be weakened by dropping the temporal quantifier $\langle\!\langle \varnothing \rangle\!\rangle G$. Another interesting research direction is to lift the exclusiveness assumption and allow for multiple agents to control the same atom. The value of the atom at the next state can then be determined by a boolean function taking into account all choices of agents. We anticipate to consider various classes of boolean functions, representing interesting notions in game theory and social choice theory, in the spirit of [Grandi and Endriss, 2013]. Finally, we plan to extend our approach by an epistemic dimension, taking inspiration from recent approaches that are based on the concept of Propositional Visibility [van der Hoek *et al.*, 2011; van Benthem *et al.*, 2015; Herzig *et al.*, 2015; Charrier *et al.*, 2016], where an agent might observe or not the value of a propositional atom. Just as CGS can be built from Propositional Control, models of epistemic logic can be built from Propositional Visibility; and similarly to PC models, they can be represented in a compact way.

Acknowledgements. Thanks are due to the AAMAS reviewers whose comments helped us to improve the paper.

References

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

[Balbiani *et al.*, 2013] Philippe Balbiani, Andreas Herzig, and Nicolas Troquard. Dynamic logic of propositional assignments: A well-behaved variant of PDL. In *Proc. LICS 2013*, pages 143–152. IEEE Computer Society, 2013.

[Balbiani *et al.*, 2014] Philippe Balbiani, Andreas Herzig, François Schwarzentruber, and Nicolas Troquard. DL-PA and DCL-PC: model checking and satisfiability problem are indeed in PSPACE. *CoRR*, abs/1411.7825, 2014.

[Bulling *et al.*, 2010] N. Bulling, J. Dix, and W. Jamroga. Model checking logics of strategic ability: Complexity. In *Specification and Verification of Multi-agent Systems*, pages 125–159. Springer, 2010.

[Čermák *et al.*, 2014] Petr Čermák, Alessio Lomuscio, Fabio Mogavero, and Aniello Murano. MCMAS-SLK: A model checker for the verification of strategy logic specifications. In *Computer Aided Verification*, volume 8559 of *LNCS*, pages 525–532. Springer, 2014.

[Charrier *et al.*, 2016] Tristan Charrier, Andreas Herzig, Emiliano Lorini, Faustine Maffre, and François Schwarzentruber. Building epistemic logic from observations and public announcements. In *Proc. KR*, 2016.

[Chatterjee *et al.*, 2010] Krishnendu Chatterjee, Thomas A. Henzinger, and Nir Piterman. Strategy logic. *Inf. Comput.*, 208(6):677–693, 2010.

[Ciná and Endriss, 2015] Giovanni Ciná and Ulle Endriss. A syntactic proof of arrow's theorem in a modal logic of social choice functions. In *Proc. AAMAS 2015*, May 2015.

[Gerbrandy, 2006] Jelle Gerbrandy. Logics of propositional control. In Nakashima et al. [2006], pages 193–200.

[Goranko and Jamroga, 2004] Valentin Goranko and Wojciech Jamroga. Comparing semantics of logics for multi-agent systems. *Synthese*, 139(2):241–280, 2004.

[Goranko, 2001] Valentin Goranko. Coalition games and alternating temporal logics. In *Proc. TARK 2001*, TARK '01, pages 259–272, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

[Grandi and Endriss, 2013] Umberto Grandi and Ulle Endriss. Lifting integrity constraints in binary aggregation. *Artif. Intell.*, 199:45–66, 2013.

[Herzig *et al.*, 2011] Andreas Herzig, Emiliano Lorini, Frédéric Moisan, and Nicolas Troquard. A dynamic logic of normative systems. *Proc. IJCAI*, 2011:228–233, 2011.

[Herzig *et al.*, 2015] Andreas Herzig, Emiliano Lorini, and Faustine Maffre. A poor man's epistemic logic based on propositional assignment and higher-order observation. In van der Hoek et al. [2015], pages 156–168.

[Herzig *et al.*, 2016] Andreas Herzig, Emiliano Lorini, Faustine Maffre, and Francois Schwarzentruber. Epistemic boolean games based on a logic of visibility and control. In Subbarao Kambhampati, editor, *Proc. IJCAI 2016*. AAAI Press, July 2016.

[Laroussinie *et al.*, 2008] François Laroussinie, Nicolas Markey, and Ghassan Oreiby. On the expressiveness and complexity of ATL. *Logical Methods in Computer Science*, 4(2), 2008.

[Mogavero *et al.*, 2014] Fabio Mogavero, Aniello Murano, Giuseppe Perelli, and Moshe Y. Vardi. Reasoning about strategies: On the model-checking problem. *ACM Trans. Comput. Log.*, 15(4):34:1–34:47, 2014.

[Nakashima *et al.*, 2006] Hideyuki Nakashima, Michael P. Wellman, Gerhard Weiss, and Peter Stone, editors. *Proc. AAMAS*. ACM, 2006.

[Pauly, 2002] Marc Pauly. A modal logic for coalitional power in games. *J. Log. Comput.*, 12(1):149–166, 2002.

[Stockmeyer and Wong, 1979] Larry J. Stockmeyer and C. K. Wong. On the number of comparisons to find the intersection of two relations. *SIAM J. Comput.*, 8(3):388–404, 1979.

[Troquard *et al.*, 2011] Nicolas Troquard, Wiebe van der Hoek, and Michael Wooldridge. Reasoning about social choice functions. *J. Philosophical Logic*, 40(4):473–498, 2011.

[van Benthem *et al.*, 2015] Johan van Benthem, Jan van Eijck, Malvin Gattinger, and Kaile Su. Symbolic model checking for dynamic epistemic logic. In van der Hoek et al. [2015], pages 366–378.

[van der Hoek and Wooldridge, 2005] Wiebe van der Hoek and Michael Wooldridge. On the logic of cooperation and propositional control. *Artif. Intell.*, 164(1-2):81–119, 2005.

[van der Hoek *et al.*, 2006] Wiebe van der Hoek, Alessio Lomuscio, and Michael Wooldridge. On the complexity of practical ATL model checking. In Nakashima et al. [2006], pages 201–208.

[van der Hoek *et al.*, 2010] Wiebe van der Hoek, Dirk Walther, and Michael Wooldridge. Reasoning about the transfer of control. *J. Artif. Intell. Res. (JAIR)*, 37:437–477, 2010.

[van der Hoek *et al.*, 2011] Wiebe van der Hoek, Nicolas Troquard, and Michael Wooldridge. Knowledge and control. In Liz Sonenberg, Peter Stone, Kagan Tumer, and Pinar Yolum, editors, *Proc. AAMAS 2011)*, pages 719–726. IFAAMAS, 2011.

[van der Hoek *et al.*, 2015] Wiebe van der Hoek, Wesley H. Holliday, and Wen-Fang Wang, editors. *Logic, Rationality, and Interaction - 5th International Workshop, LORI 2015 Taipei, Taiwan, October 28-31, 2015, Proceedings*, volume 9394 of *LNCS*. Springer, 2015.

[Wooldridge, 2000] M. Wooldridge. Computationally Grounded Theories of Agency. In *Proc. of ICMAS*, pages 13–22. IEEE Press, 2000.