Relaxing Exclusive Control in Boolean Games

Francesco Belardinelli
IBISC, Université d'Evry and IRIT, CNRS, Toulouse
belardinelli@ibisc.fr

Umberto Grandi
IRIT, University of Toulouse
umberto.grandi@irit.fr

Andreas Herzig
IRIT, CNRS, Toulouse
andreas.herzig@irit.fr

Dominique Longin
IRIT, CNRS, Toulouse
dominique.longin@irit.fr

Emiliano Lorini
IRIT, CNRS, Toulouse
emiliano.lorini@irit.fr

Arianna Novaro
IRIT, University of Toulouse
arianna.novaro@irit.fr

Laurent Perrussel
IRIT, University of Toulouse
laurent.perrussel@irit.fr

In the typical framework for boolean games (BG) each player can change the truth value of some propositional atoms, while attempting to make her goal true. In standard BG goals are propositional formulas, whereas in iterated BG goals are formulas of Linear Temporal Logic. Both notions of BG are characterised by the fact that agents have exclusive control over their set of atoms, meaning that no two agents can control the same atom. In the present contribution we drop the exclusivity assumption and explore structures where an atom can be controlled by multiple agents. We introduce Concurrent Game Structures with Shared Propositional Control (CGS-SPC) and show that they account for several classes of repeated games, including iterated boolean games, influence games, and aggregation games. Our main result shows that, as far as verification is concerned, CGS-SPC can be reduced to concurrent game structures with exclusive control. This result provides a polynomial reduction for the model checking problem of specifications in Alternating-time Temporal Logic on CGS-SPC.

1 Introduction

Coalition Logic of Propositional Control CL-PC was introduced by van der Hoek and Wooldridge [14] as a formal language for reasoning about capabilities of agents and coalitions in multiagent environments, later extended by the concept of transfer of control [13]. In CL-PC, capability is modeled by means of the concept of *propositional control*: it is assumed that each agent i is associated with a specific finite subset Φ_i of the finite set of all atomic variables Φ, which are the variables *controlled* by i, in the sense that i has the ability to assign a (truth) value to each variable in Φ_i but cannot change the truth values of the variables in $\Phi \setminus \Phi_i$. Control over variables is assumed to be *exclusive*: two agents cannot control the same variable, i.e., if $i \neq j$ then $\Phi_i \cap \Phi_j = \emptyset$.[1] The connection between CL-PC and Dynamic Logic of Propositional Assignments was explored by Grossi *et al.* [8].

A boolean game BG [11, 3] is a game in which each player wants to achieve a certain goal represented by a propositional formula. Boolean games correspond to the specific subclass of normal form games in which agents have binary preferences. They share with CL-PC the idea that an agent's action consists in affecting the truth values of the variables she controls. Just as in there, control over atomic propositions

[1] In CL-PC, it is also assumed that control is *complete*, that is, every variable is controlled by at least one agent (i.e., for every $p \in \Phi$ there exists an agent i such that $p \in \Phi_i$).

J. Lang (Ed.): TARK 2017
EPTCS ??, 2017, pp. 44–57, doi:10.4204/EPTCS.??.9

is exclusive in BGs. More recently, BGs were generalized to iterated boolean games IBGs [10, 9]. In IBGs, the agents' goals are formulas of Linear Temporal Logic LTL, and an agent's strategy determines an assignment of the variables controlled by the agent in every round of the game.

Gerbrandy was the first to study CL-PC without exclusive control [5]. In his games of propositional control, the value of a variable at the next state is determined by an outcome function that combines the agents' choices of values for propositional variables. Gerbrandy's language contains a coalition operator and—just as coalition logic—only allows to reason about what agents and coalitions of agents are able to achieve in a single step. Importing results from many-dimensional modal logics, Gerbrandy proved that the satisfiability problem is decidable when there are at most 2 agents, and undecidable otherwise [5, Prop.5].

The aim of the present paper is to further study models without exclusive propositional control as a basis for BGs and other game-theoretic approaches. Specifically, we introduce Concurrent Game Structures with Shared Propositional Control CGS-SPC and show their relationship with different classes of games studied in literature, including IBGs. The main result of the paper is that CGS-SPC can be reduced to CGS with Exclusive Propositional Control CGS-EPC [2] by introducing a dummy agent who controls the value of the shared variables and simulates the transition function. The reduction is polynomial, showing that the problem of verification of specifications in Alternating-time Temporal Logic on CGS-SPC can be reduced to verification in CGS-EPC. We also explore the consequences of such results in the problem of finding a winning strategy in games with shared control.

The paper is organized as follows. Section 2 provides the basic definitions of concurrent game structures with exclusive and shared control, as well as introducing the language and the semantic of Alternating-time Temporal Logic. Section 3 shows that a number of game structures introduced in the literature can be reconducted to our definition of CGS-SPC. We then prove our main result in Section 4, where we reduce the problem of ATL^* model checking for CGS-SPC to model checking of a translated ATL^* formula in a CGS-EPC suitably defined. Section 5 discusses the consequences in computational complexity of our main result, and Section 6 concludes.

2 Formal Framework

In this section we consider two classes of concurrent game structures with propositional control, suitable for the interpretation of a logic for individual and collective strategies which is introduced next. The two classes differ in the type of propositional control: exclusive in the former and shared in the latter.

2.1 CGS with Exclusive and Shared Control

We first present concurrent game structures with exclusive propositional control CGS-EPC as they have been introduced by Belardinelli and Herzig [2].[2] We then generalise them by relaxing the assumption of exclusive control.

Definition 1 (CGS-EPC)**.** *A* concurrent game structure with exclusive propositional control *is a tuple* $\mathscr{G} = \langle N, \Phi_1, \dots, \Phi_n, S, d, \tau \rangle$*, where:*

- $N = \{1, \dots, n\}$ *is a set of* agents*;*
- $\Phi = \Phi_1 \cup \dots \cup \Phi_n$ *is a set of* propositional variables *partitioned in n disjoint subsets, one for each agent;*

[2]More precisely, the CGS-EPC we consider here as our basic framework correspond to the "weak" version defined by Belardinelli and Herzig [2], as opposed to a strong version where $d(i,s) = \mathscr{A}_i$ for every $i \in N$ and $s \in S$.

- $S = 2^\Phi$ *is the set of* states*, corresponding to all valuations over* Φ*;*
- $d : N \times S \to (2^{\mathscr{A}} \setminus \emptyset)$*, for* $\mathscr{A} = 2^\Phi$*, is the* protocol function*, such that* $d(i,s) \subseteq \mathscr{A}_i$ *for* $\mathscr{A}_i = 2^{\Phi_i}$*;*
- $\tau : S \times \mathscr{A}^n \to S$ *is the* transition function *such that* $\tau(s, \alpha_1, \dots, \alpha_n) = \bigcup_{i \in N} \alpha_i$.

Intuitively, a CGS-EPC describes the interactions of a group N of agents, each one of them controlling (exclusively) a set $\Phi_i \subseteq \Phi$ of propositional atoms. The state of the CGS is an evaluation of the atoms in Φ. In each such state the protocol function returns which actions an agent can execute.

The intuitive meaning of action $\alpha_i \in d(i,s)$ is "assign true to all atoms in α_i, and false to all atoms in $\Phi_i \setminus \alpha_i$". The $idle_s$ action can be introduced as $\{p \in \Phi_i \mid s(p) = 1\}$, for every $i \in N$, $s \in S$. With an abuse of notation we write $d(i,s) = \alpha$ whenever $d(i,s)$ is a singleton $\{\alpha\}$.

We equally see each state $s \in S$ as a function $s : \Phi \to \{0,1\}$ returning the truth value of a propositional variable in s, so that $s(p) = 1$ iff $p \in s$. Given $\alpha = (\alpha_1, \dots, \alpha_n) \in \mathscr{A}^n$, we equally see each $\alpha_i \subseteq \Phi_i$ as a function $\alpha_i : \Phi_i \to \{0,1\}$ returning the choice of agent i for p under action α.

We now introduce a generalisation of concurrent game structures for propositional control. Namely, we relax the exclusivity requirement on the control of propositional variables, thus introducing concurrent game structures with shared propositional control CGS-SPC.

Definition 2 (CGS-SPC)**.** *A* concurrent game structure with shared propositional control *is a tuple* $\mathscr{G} = \langle N, \Phi_0, \dots, \Phi_n, S, d, \tau \rangle$ *such that:*

- *N, S, and d are defined as in Def. 1 with* $\mathscr{A} = 2^{\Phi \setminus \Phi_0}$*;*
- $\Phi = \Phi_0 \cup \Phi_1 \cup \dots \cup \Phi_n$ *is a set of* propositional variables*, where* $\Phi_0 \cup \Phi_1 \cup \dots \cup \Phi_n$ *is not necessarily a partition and* $\Phi_0 = \Phi \setminus (\Phi_1 \cup \dots \cup \Phi_n)$*;*
- $\tau : S \times \mathscr{A}^n \to S$ *is the* transition function*.*

Observe that in CGS-SPC the same atom can be controlled by multiple agents, and propositional control is not exhaustive. Additionally, the actions in $\mathscr{A}$ do not take into account propositional variables in Φ_0 because they are not controlled by anyone (though their truth value might change according to the transition function). The transition function takes care of combining the various actions and producing a consistent successor state according to some rule. Simple examples of such rules include introducing a threshold $m_p \in \mathbb{N}$ for every variable p, thus setting $p \in \tau(s, \alpha)$ iff the number of agents i with $p \in \alpha_i$ is greater than m_p. This generalises Gerbrandy's consensus games [5].[3]

Clearly, CGS-EPC can be seen as a special case of CGS-SPC in which every atom is controlled exactly by a single agent, and therefore $\{\Phi_0, \dots, \Phi_n\}$ is a partition of Φ. Moreover, τ is given in a specific form as per Definition 1.

2.2 Logics for Time and Strategies

To express relevant properties of CGS, we present the Linear-time Temporal Logic LTL [20] and the Alternating-time Temporal Logic ATL^* [1]. Firstly, state formulas φ and path formulas ψ in ATL^* are defined by the following BNF:

$$
\begin{aligned}
\varphi &::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle\langle C \rangle\rangle \psi \\
\psi &::= \varphi \mid \neg\psi \mid \psi \vee \psi \mid \bigcirc\psi \mid \psi \mathscr{U} \psi
\end{aligned}
$$

[3] The definition of τ as an arbitrary function might seem too general. Nonetheless, such a definition is needed to represent complex aggregation procedures such as those used in the games described in Sections 3.2 and 3.3.

where $p \in \Phi$ and $C \in 2^N$. The intuitive reading of $\langle\langle C \rangle\rangle \psi$ is "coalition C has a strategy to enforce ψ", that of $\bigcirc \psi$ is "ψ holds at the next state" and that of $\psi \mathscr{U} \varphi$ is "ψ will hold until φ holds".

The BNF for the language of ATL consists of all state formulas where ψ is either $\bigcirc \varphi$ or $\varphi \mathscr{U} \varphi$. On the other hand, the language of LTL consists of all path formulas in ATL*, whose state formulas are propositional atoms only. That is, formulas in LTL are defined by the following BNF:

$$\psi \quad ::= \quad p \mid \neg\psi \mid \psi \vee \psi \mid \bigcirc \psi \mid \psi \mathscr{U} \psi$$

Truth conditions of LTL and ATL* formulas are defined with respect to concurrent game structures, such as the CGS-EPC and CGS-SPC introduced above. In order to do so, we first have to provide some additional notation.

The set of *enabled joint actions* at some state s is defined as $Act(s) = \{\alpha \in \mathscr{A}^n \mid \alpha_i \in d(i,s)$ for every $i \in N\}$. Then, the set of *successors* of s is given as $Succ(s) = \{\tau(s,\alpha) \mid \alpha \in Act(s)\}$. Every $Succ(s)$ is non-empty because $d(i,s) \neq \emptyset$. An infinite sequence of states $\lambda = s_0 s_1 \ldots$ is a *computation* or a *path* if $s_{k+1} \in Succ(s_k)$ for all $k \geq 0$. For every computation λ and $k \geq 0$, $\lambda[k,\infty] = s_k, s_{k+1}, \ldots$ denotes the suffix of λ starting from s_k. Notice that $\lambda[k,\infty]$ is also a computation. When λ is clear from the context, we denote with $\alpha[k]$ the action such that $\lambda[k+1] = \tau(\lambda[k], \alpha[k])$.

A *memoryless strategy* for agent $i \in N$ is a function $\sigma_i : S \to \mathscr{A}_i$ such that $\sigma_i(s) \in d(i,s)$, returning an action for each state. For simplicity, we will assume in the rest of the paper that agents have memoryless strategies.

We let σ_C be a *joint strategy* for coalition $C \subseteq N$, i.e., a function returning for each agent $i \in C$, the individual strategy σ_i. For notational convenience we write σ for σ_N. The set $out(s,\sigma_C)$ includes all computations $\lambda = s_0 s_1 \ldots$ such that (a) $s_0 = s$; and (b) for all $k \geq 0$, there is $\alpha \in Act(s)$ such that $\sigma_C(i)(s_k) = \alpha_i$ for all $i \in C$, and $\tau(s_k,\alpha) = s_{k+1}$. Observe that $out(s,\sigma)$ is a singleton.

We are now ready to define the truth conditions for LTL and ATL* formulas with respect to a CGS-SPC $\mathscr{G}$. Formulas in ATL* are interpreted on states, while formulas in LTL are interpreted on computations.

$(\mathscr{G},s) \models p$	iff	$s(p) = 1$
$(\mathscr{G},s) \models \neg\varphi$	iff	$(\mathscr{G},s) \not\models \varphi$
$(\mathscr{G},s) \models \varphi_1 \vee \varphi_2$	iff	$(\mathscr{G},s) \models \varphi_1$ or $(\mathscr{G},s) \models \varphi_2$
$(\mathscr{G},s) \models \langle\langle C \rangle\rangle \psi$	iff	for some σ_C, for all $\lambda \in out(s,\sigma_C)$, $(\mathscr{G},\lambda) \models \psi$
$(\mathscr{G},\lambda) \models \varphi$	iff	$(\mathscr{G},\lambda[0]) \models \varphi$
$(\mathscr{G},\lambda) \models \neg\psi$	iff	$(\mathscr{G},\lambda) \not\models \psi$
$(\mathscr{G},\lambda) \models \psi_1 \vee \psi_2$	iff	$(\mathscr{G},\lambda) \models \psi_1$ or $\mathscr{G},\lambda \models \psi_2$
$(\mathscr{G},\lambda) \models \bigcirc \psi$	iff	$(\mathscr{G},\lambda[1,\infty]) \models \psi$
$(\mathscr{G},\lambda) \models \psi_1 \mathscr{U} \psi_2$	iff	for some $i \geq 0$, $(\mathscr{G},\lambda[i,\infty]) \models \psi_2$ and $(\mathscr{G},\lambda[j,\infty]) \models \psi_1$ for all $0 \leq j < i$

We define below the model checking problem for this context.

Definition 3 (Model Checking Problem)**.** *Given a CGS-SPC $\mathscr{G}$, a state $s \in S$, and an* ATL**-formula φ, determine whether $(\mathscr{G},s) \models \varphi$.*

It is well-known that model checking for ATL* on general concurrent game structures is 2EXPTIME-complete [1]. Belardinelli and Herzig proved that model checking ATL on CGS-EPC is Δ_3^P-complete [2]. Hereafter we consider the general case of CGS-SPC and ATL*.

3 Examples of Shared Control

In this section we take three examples of iterated games from the literature, namely *iterated boolean games* [10], *influence games* [7], and *aggregation games* [6], and we show that they are all instances of our definition of a CGS-SPC.

3.1 Iterated Boolean Games

We make use of CGS-EPC to introduce iterated boolean games with LTL goals as studied by Gutierrez *et al.* [9, 10]. An *iterated boolean game* is a tuple $\langle \mathcal{G}, \gamma_1, \ldots, \gamma_n \rangle$ such that (*i*) $\mathcal{G}$ is a CGS-EPC with a trivial protocol (i.e., for every $i \in N$, $s \in S$, $d(i,s) = \mathscr{A}_i$); and (*ii*) for every $i \in N$, the *goal* γ_i is an LTL-formula.

We can generalise the above to *iterated boolean games with shared control* as follows:

Definition 4. *An* iterated boolean game with shared control *is a tuple* $\langle \mathcal{G}, \gamma_1, \ldots, \gamma_n \rangle$ *such that*

(i) $\mathcal{G}$ *is a CGS-SPC;*

(ii) for every $i \in N$, *the* goal γ_i *is an* LTL*-formula.*

Observe that function τ is thus no longer trivial. Just like CGS-SPC generalise CGS-EPC, iterated boolean games with shared control generalise standard iterated boolean games. In particular, the existence of a winning strategy can be checked via the satisfaction of an ATL^*-formula:

Proposition 1. *An agent* i *in an iterated boolean game has a winning strategy for goal* γ_i *and state* s *if and only if formula* $\langle\langle \{i\} \rangle\rangle \gamma_i$ *is satisfied in* $(\mathcal{G}, s)$*.*

Example 1. *Consider an iterated boolean game with shared control for agents* $\{1,2\}$ *and issues* $\{p,q\}$*, such that* $\Phi_1 = \{p\}$ *and* $\Phi_2 = \{p,q\}$*. Suppose that for all states* s *the transition function is such that* $\tau(s,\alpha)(q) = \alpha_2(q)$*, being agent* 2 *the only agent controlling* q*, while* $\tau(s,\alpha)(p) = 1$ *iff* $\alpha_1(p) = \alpha_2(p) = 1$*. We thus have that* $(\mathcal{G}, s) \models \langle\langle \{1,2\} \rangle\rangle \bigcirc p$ *and* $(\mathcal{G}, s) \models \neg \langle\langle \{1\} \rangle\rangle \bigcirc q$ *for all* s*.*

3.2 Influence Games

Influence games model strategic aspects of opinion diffusion on a social network. They are based on a set of variables $\mathsf{op}(i,p)$ for "agent *i* has the opinion *p*" and $\mathsf{vis}(i,p)$ for "agent *i* uses her influence power over *p*". Agents have binary opinions over all issues; hence $\neg\mathsf{op}(i,p)$ reads "agent *i* has the opinion $\neg p$".

Goals are expressed in LTL with propositional variables $\{\mathsf{op}(i,p), \mathsf{vis}(i,p) \mid i \in N, p \in \Phi\}$. We define an influence game in a compact way below, pointing to the work of Grandi *et al.* [7] for more details.

Definition 5. *An* influence game *is a tuple* $IG = \langle N, \Phi, E, S_0, \{F_{i,Inf(i)}\}_{i \in N}, \{\gamma_i\}_{i \in N} \rangle$ *where:*

- $N = \{1, \ldots, n\}$ *is a set of* agents*;*
- $\Phi = \{1, \ldots, m\}$ *is a set of* issues*;*
- $E \subseteq N \times N$ *is a directed irreflexive graph representing the* influence network*;*
- $S_0 \in \mathcal{S}$ *is the* initial state*, where states in* $\mathcal{S}$ *are tuples* (B,V)*, where* $B = (B_1, \ldots, B_n)$ *is a profile of* private opinions $B_i : \Phi \to \{0,1\}$ *indicating the opinion of agent* i *on variable* p*, and* $V = (V_1, \ldots, V_n)$ *is a profile of* visibilities $V_i : \Phi \to \{0,1\}$ *indicating whether agent* i *is using her influence power over* p*;*
- $F_{i,Inf(i)}$ *is the unanimous* aggregation function *associating a new private opinion for agent* i *based on agent* i*'s current opinion and the visible opinions of* i*'s influencers in* $Inf(i)$*;*

- γ_i *is agent i's* individual goal*, i.e., an* LTL *formula.*

Influence games are repeated games in which individuals decide whether to disclose their opinions (i.e., use their influence power over issues) or not. Once the disclosure has taken place, opinions are updated by aggregating the visible opinions of the influencers of each agent (i.e., the nodes having an outgoing edge terminating in the agent's node).

We associate to $IG = \langle N, \Phi, E, S_0, \{F_{i,Inf(i)}\}_{i\in N}, \{\gamma_i\}_{i\in N}\rangle$ a CGS-SPC $\mathscr{G}' = \langle N', \Phi'_0, \ldots, \Phi'_n, S', d', \tau'\rangle$ by letting $N' = N$; $\Phi'_0 = \{\mathsf{op}(i,p) \mid i \in \mathscr{N}, p \in \Phi\}$; $\Phi'_i = \{\mathsf{vis}(i,p) \mid p \in \Phi\}$ for $i \in N'$; $S' = 2^{\Phi'}$; $d'(i,s') = 2^{\Phi'_i}$ for $s' \in S'$; and finally for state $s' \in S'$ and action α' we let:

$$\tau'(s',\alpha')(\varphi) \ = \ \begin{cases} \alpha'_i(\mathsf{vis}(i,p)) & \text{if } \varphi = \mathsf{vis}(i,p) \\ F_{i,Inf(i)}(\vec{a},\vec{b})_{|p} & \text{if } \varphi = \mathsf{op}(i,p) \end{cases}$$

where vectors $\vec{a} = (a_1, \ldots, a_{|\Phi|})$ and $\vec{b} = (b_1, \ldots, b_{|\Phi|})$ are defined as follows, for $k \in$ *Inf(i)*:

$$a_p = \begin{cases} 1 & \text{if } \mathsf{op}(i,p) \in s' \\ 0 & \text{otherwise} \end{cases}$$

$$b_p = \begin{cases} 1 & \text{if } \alpha_k(\mathsf{vis}(k,p)) = 1 \text{ and } \mathsf{op}(k,p) \in s' \\ 0 & \text{if } \alpha_k(\mathsf{vis}(k,p)) = 1 \text{ and } \mathsf{op}(k,p) \notin s' \\ ? & \text{if } \alpha_k(\mathsf{vis}(k,p)) = 0 \end{cases}$$

Vector $\vec{a}$ represents the opinion of agent i over the issues at state s', while vector $\vec{b}$ represents the opinions of i's influencers over the issues, in case they are using their influencing. In particular, '?' indicates that the influencers of i in *Inf(i)* are not using their influence power.

Proposition 2. *Agent i in influence game IG has a winning strategy for goal γ_i and state S_0 if and only if formula $\langle\langle\{i\}\rangle\rangle\gamma_i$ is satisfied in the associated CGS-SPC and state s' corresponding to S_0.*

proof sketch. Let IG be an influence game and let $\mathscr{G}'$ be the CGS-SPC associated to it. Consider now an arbitrary agent i and suppose that i has a winning strategy in IG for her goal γ_i in S_0. A memoryless strategy σ_i for agent i in an influence game maps to each state actions of type $(\mathsf{reveal}(J), \mathsf{hide}(J'))$, where $J, J' \subseteq \Phi$ and $J \cap J' = \emptyset$. For any state s in IG, consisting of a valuation of opinions and visibilities, consider the state s' in $\mathscr{G}'$ where $B_i(p) = 1$ iff $\mathsf{op}(i,p) \in s'$ and $V_i(p) = 1$ iff $\mathsf{vis}(i,p) \in s'$. We now construct the following strategy for $\mathscr{G}'$:

$$\sigma'_i(s') = \{\mathsf{vis}(i,p) \mid p \in J \text{ for } \sigma_i(s) = (\mathsf{reveal}(J), \mathsf{hide}(J'))\}$$

By the semantics of the $\langle\langle\{i\}\rangle\rangle$ operator provided in Section 2.2, and by the standard game-theoretic definition of winning strategy, the statement follows easily from our construction of $\mathscr{G}'$. □

The above translation allowed to shed light over the control structure of the variables of type $\mathsf{op}(i,p)$. In fact, we can now see that $\mathsf{op}(i,p) \in \Phi'_0$ for all $i \in N$ and $p \in \Phi$.

3.3 Aggregation Games

Individuals facing a collective decision, such as members of a hiring committee or a parliamentary body, are provided with individual goals specified on the outcome of the voting process — outcome that is jointly controlled by all individuals in the group. For instance, a vote on a single binary issue using the majority rule corresponds to a game with one single variable controlled by all individuals, the majority rule playing the role of the transition function.

Similar situations have been modelled as one-shot games called *aggregation games* [6], and we now extend this definition to the case of iterated decisions:

Definition 6. *An* iterated aggregation game *is a tuple* $AG = \langle N, \Phi, F, \gamma_1, \dots, \gamma_n \rangle$ *such that:*

- N *is a set of* agents*;*
- $\Phi = \{p_1, \dots, p_m\}$ *are variables representing* issues*;*
- $F : \{0,1\}^{N \times \Phi} \to \{0,1\}$ *is an* aggregation function*, that is, a boolean function associating a collective decision with the individual opinion of the agents on the issues;*
- γ_i *for* $i \in N$ *is an* individual goal *for each agent, that is, a formula in the* LTL *language constructed over* Φ*.*

Individuals at each stage of an aggregation game only have information about the current valuation of variables in Φ, resulting from the aggregation of their individual opinions. Analogously to Proposition 2, we can obtain the following result:

Proposition 3. *An iterated aggregation game AG is an instance of a CGS-SPC. More precisely, agent i in AG has a winning strategy for goal γ_i in s if and only if formula $\langle\langle \{i\} \rangle\rangle \gamma_i$ is satisfied in the associated CGS-SPC in the corresponding state s'.*

proof sketch. Starting from an iterated aggregation game $AG = \langle N, \Phi, F, \gamma_1, \dots, \gamma_n \rangle$, construct a CGS-SPC $\mathscr{G}' = \langle N', \Phi', S', d', \tau' \rangle$ as follows. Let $N' = N$; $\Phi'_i = \Phi$ for all $i = 1, \dots, n$; and $\Phi'_0 = \emptyset$. Hence, each agent controls all variables. Let the set of actions available to each player be $d'(i,s) = 2^{\Phi'}$ for all i and s, and the transition function τ' be such that $\tau'(s, \alpha_1, \dots, \alpha_n) = F(\alpha_1, \dots, \alpha_n)$. The statement then follows easily. □

A notable example of an iterated aggregation game is the setting of iterative voting (see, e.g., [18, 17, 19]). In this setting, individuals hold preferences about a set of candidates and iteratively manipulate the result of the election in their favour until a converging state is reached. Similar situations can easily be modelled as iterated aggregation games, which have the advantage of allowing for a more refined specification of preferences via the use of more complex goals.

4 Restoring Exclusive Control

In this section we prove the main result of the paper, namely that the shared control of a CGS-SPC can be simulated in a CGS-EPC having exclusive control. In particular, any specification in ATL* satisfied in some CGS-SPC can be translated in polynomial time into an ATL*-formula satisfied in a CGS-EPC. To do so, we introduce a dummy agent to simulate the aggregation function. Moreover, we make use of an additional 'turn-taking' atom which allows us to distinguish the states where the agents choose their actions from those in which the aggregation process takes place.

We begin by inductively defining a translation function tr within ATL*. Intuitively, tr translates every ATL*-formula χ into a formula $tr(\chi)$ having roughly the same meaning, but where the one-step 'next' operator is replaced by two 'next' steps:

$$\begin{array}{lcl} tr(p) & = & p \\ tr(\neg\chi) & = & \neg tr(\chi) \\ tr(\chi \vee \chi') & = & tr(\chi) \vee tr(\chi') \\ tr(\bigcirc\chi) & = & \bigcirc\bigcirc tr(\chi) \\ tr(\chi \mathscr{U} \chi') & = & tr(\chi) \mathscr{U} tr(\chi') \\ tr(\langle\!\langle C \rangle\!\rangle \chi) & = & \langle\!\langle C \rangle\!\rangle tr(\chi) \end{array}$$

where $p \in \Phi$, $C \subseteq N$, and χ, χ' are either state- or path-formulas as suitable. Clearly, the translation is polynomial.

We then map a given CGS-SPC to a CGS-EPC.

Definition 7. *Let $\mathscr{G} = \langle N, \Phi_0, \ldots, \Phi_n, S, d, \tau \rangle$ be a CGS-SPC. The CGS-EPC corresponding to $\mathscr{G}$ is $\mathscr{G}' = \langle N', \Phi'_1, \ldots, \Phi'_n, S', d', \tau' \rangle$ where:*

- $N' = N \cup \{*\}$;
- $\Phi' = \Phi \cup \{turn\} \cup \{c_{ip} \mid i \in N \text{ and } p \in \Phi_i\}$ *and Φ' is partitioned as follows, for agents in N':*

$$\Phi'_i = \{c_{ip} \in \Phi' \mid p \in \Phi_i\}$$
$$\Phi'_* = \{turn\} \cup \Phi$$

- $S' = 2^{\Phi'}$. *For every $s' \in S'$, let $s = (s' \cap \Phi) \in S$ be the restriction of s' on Φ;*
- *d' is defined according to the truth value of turn in s'. Specifically, given $\alpha_i \in \mathscr{A}_i$, let $\alpha'_i = \{c_{ip} \in \Phi'_i \mid p \in \alpha_i\} \in \mathscr{A}'_i$. Then, for $i \in N$ we let:*

$$d'(i, s') = \begin{cases} \{\alpha'_i \in \mathscr{A}'_i \mid \alpha_i \in d(i, s)\} & \text{if } s'(turn) = 0 \\ \emptyset & \text{if } s'(turn) = 1 \end{cases}$$

*For agent * we define:*

$$d'(*, s') = \begin{cases} +turn & \text{if } s'(turn) = 0 \\ \tau(s, \alpha), \text{ for } \alpha_i(p) = s'(c_{ip}) & \text{if } s'(turn) = 1 \end{cases}$$

where $+turn = idle_s \cup \{turn\}$.

- *τ' is defined as per Def. 1, that is, $\tau'(s', \alpha') = \bigcup_{i \in N'} \alpha'_i$.*

Intuitively, in the CGS-EPC $\mathscr{G}'$ every agent $i \in N$ manipulates local copies c_{ip} of atoms $p \in \Phi$. The aggregation function τ in $\mathscr{G}$ is mimicked by the dummy agent $*$, whose role is to observe the values of the various c_{ip}, then perform an action to aggregate them and set the value of p accordingly. Observe that agent $*$ acts only when the *turn* variable is true, in which case all the other agents set all their variables to false, i.e., they all play $\emptyset$. This is to ensure the correspondence between memory-less strategies of $\mathscr{G}$ and $\mathscr{G}'$, as shown in Lemma 5.

Note also that the size of game $\mathscr{G}'$ is polynomial in the size of $\mathscr{G}$, and that $\mathscr{G}'$ can be constructed in polynomial time from $\mathscr{G}$. To see this, observe that an upper bound on the number of variables is $\mathscr{N} \times \Phi$.

Recall that we can associate to each state $s' \in S'$ a state $s = s' \cap \Phi$ in S. For the other direction, given a state $s \in S$, there are multiple states s' that agree with s on Φ. The purpose of the next definition is to designate one such state as the canonical one.

Definition 8. *For every $s \in S$, we define the* canonical state $s'_\star = \{s' \in S' \mid s' \cap \Phi = s \text{ and } s(p) = 0 \text{ for } p \notin \Phi\}$.

$$\begin{tikzcd}
\lambda[0] \arrow[rr, "(\alpha_1 \ldots \alpha_n)[0]"] \arrow[d, dashed] & & \lambda[1] \arrow[rr, "(\beta_1 \ldots \beta_n)[1]"] \arrow[d, dashed] & & \lambda[2] \arrow[rr, "(\delta_1 \ldots \delta_n)[2]"] \arrow[d, dashed] & & \lambda[3] \arrow[r] \arrow[d, dashed] & \cdots \\
\lambda'[0] \arrow[r, "\alpha'_1 \ldots \alpha'_n", "+turn"'] & \lambda'[1] \arrow[r, "\emptyset", "\tau(\lambda'[1]_{|\Phi}, \alpha)"'] & \lambda'[2] \arrow[r, "\beta'_1 \ldots \beta'_n", "+turn"'] & \lambda'[3] \arrow[r, "\emptyset", "\tau(\lambda'[3]_{|\Phi}, \beta)"'] & \lambda'[4] \arrow[r, "\delta'_1 \ldots \delta'_n", "+turn"'] & \lambda'[5] \arrow[r, "\emptyset", "\tau(\lambda'[5]_{|\Phi}, \delta)"'] & \lambda'[6] \arrow[r] & \cdots
\end{tikzcd}$$

Figure 1: A path λ in a CGS-SPC $\mathscr{G}$ and its associated path λ' in a CGS-EPC $\mathscr{G}'$.

Observe that, in particular, in all canonical states atom *turn* is false. As an example, consider $\Phi = \{p,q\}$ and $N = \{1,2\}$. Let then $\Phi_1 = \{p\}$ and $\Phi_2 = \{p,q\}$.We thus have that $\Phi' = \{p,q,c_{1p},c_{2p},c_{2q},turn\}$. If $s = \{p\}$, we have for instance that $s' \cap \Phi = s$ for $s' = \{p,c_{1p}\}$. On the other hand, $s'_\star = \{p\}$.

We now move to define a correspondence between paths of $\mathscr{G}$ and $\mathscr{G}'$. For notational convenience, we indicate with $\lambda[k]_{|\Phi} = \lambda[k] \cap \Phi$, the restriction of state $\lambda[k]$ to variables in Φ. Given a path λ' of $\mathscr{G}'$, consider the unique infinite sequence of states λ associated to λ' defined as follows:

$$\lambda[k] = \lambda'[2k]_{|\Phi} = \lambda'[2k{+}1]_{|\Phi} \text{ for all } k \in \mathbb{N}. \quad (\dagger)$$

On the other hand, there are multiple sequences λ' that can be associated with a path λ, so that $(\dagger)$ holds true. In fact, we only know how the variables in Φ behave, while the truth values of the other variables can vary. We now make use of condition $(\dagger)$ to characterise the paths of $\mathscr{G}$ and $\mathscr{G}'$ that can be associated:

Lemma 4. *Given a CGS-SPC $\mathscr{G}$ and the corresponding CGS-EPC $\mathscr{G}'$, the following is the case:*

1. *for all paths λ' of $\mathscr{G}'$, sequence λ satisfying condition $(\dagger)$ is a path of $\mathscr{G}$;*
2. *for all paths λ of $\mathscr{G}$, for all sequences λ' satisfying $(\dagger)$, λ' is a path of $\mathscr{G}'$ iff for all k there exists a $\mathscr{G}$-action $\alpha[k]$ such that $\lambda[k] \xrightarrow{\alpha[k]} \lambda[k{+}1]$ and states $\lambda'[2k{+}1]$ and $\lambda'[2k{+}2]$ can be obtained from state $\lambda'[2k]$ by performing actions $(\alpha'_1,\ldots,\alpha'_n,\, +turn)$ and then $(\emptyset_1,\ldots,\emptyset_n,\, \tau(\lambda'[2k{+}1]_{|\Phi},\alpha))$.*

Proof. We first prove (1) by showing that λ is a path of $\mathscr{G}$, i.e., that for every k there is an action α that leads from $\lambda[k]$ to $\lambda[k{+}1]$. Suppose that $\lambda'[2k] \xrightarrow{\alpha'[2k]} \lambda'[2k{+}1] \xrightarrow{\alpha'[2k+1]} \lambda'[2(k{+}1)]$ for action $\alpha'[2k] = (\alpha'_1,\ldots,\alpha'_n,+turn)$ and action $\alpha'[2k{+}1] = (\emptyset_1,\ldots,\emptyset_n,\tau(\lambda'[2k{+}1]_{|\Phi},\alpha))$. Then, we observe that we can move from state $\lambda[k] = \lambda'[2k]_{|\Phi} = \lambda'[2k{+}1]_{|\Phi}$ to $\lambda[k{+}1] = \lambda'[2k{+}2]_{|\Phi}$ by performing action $(\alpha_1,\ldots,\alpha_n)$ such that $\alpha_i = \{p \in \Phi \mid c_{ip} \in \alpha'_i\}$ for every $i \in N$.

As for (2), the right-to-left direction is clear. For the left-to-right direction, let λ' be a path associated to λ. From $(\dagger)$ we know that for any k we have that $\lambda'[2k]_{|\Phi} = \lambda[k]$ and $\lambda'[2k{+}2]_{|\Phi} = \lambda[k{+}1]$. Now by Definition 7, the only actions available to the players at $\lambda'[2k]$ are of the form $(\alpha'_1,\ldots,\alpha'_n,+turn)$, and the only action available at $\lambda'[2k{+}1]$ is $(\emptyset_1,\ldots,\emptyset_n,\tau(\lambda'[2k{+}1]_{|\Phi},\alpha))$. We can thus obtain the desired result by considering action $\alpha[k] = (\alpha_1,\ldots,\alpha_n)$, where $\alpha_i = \{p \in \Phi_i \mid c_{ip} \in \alpha'_i\}$ for each $i \in N$, and by observing that by $(\dagger)$ we have $\tau(\lambda'[2k+1]_{|\Phi},\alpha) = \tau(\lambda[k],\alpha)$. □

Figure 1 illustrates the construction of the two paths λ and λ' in the proof of Lemma 4. In particular, the second part of the lemma characterises the set of $\mathscr{G}'$-paths λ' associated to a $\mathscr{G}$-path λ: for any sequence of $\mathscr{G}$-actions that can generate path λ, we can construct a distinct $\mathscr{G}'$-path λ' that corresponds to λ, where the sequence of actions can be reconstructed by reading the values of the variables in Φ'_i in odd states $\lambda[2k+1]$.

From this set of $\mathscr{G}'$-paths λ' we can specify a subset of *canonical* paths as follows:

Definition 9. *For a path λ of $\mathcal{G}$, a* canonical *associated path $\lambda'_\star$ of $\mathcal{G}'$ is any path λ' such that $(\dagger)$ holds and $\lambda'[0] = \lambda[0]'_\star$.*

That is, a canonical path λ' associated to λ starts from the canonical state $\lambda[0]'_\star$ associated to $\lambda[0]$. The following example clarifies the concepts just introduced.

Example 2. *Consider a CGS-SPC $\mathcal{G}$ with $N = \{1,2\}$ and $\Phi = \{p,q\}$ such that $\Phi_1 = \{p\}$ and $\Phi_2 = \{p,q\}$. Let $d(i,s) = 2^{\Phi_i}$ for all $i \in N$ and $s \in S$, and let $\tau(s,\alpha)(p) = 0$ if and only if $\alpha_1(p) = \alpha_2(p) = 0$, while $\tau(s,\alpha)(q) = \alpha_2(q)$ for all $s \in S$. Namely, issue p becomes true if at least one agent makes it true, while issue q follows the decision of agent 2. Let now $\lambda = s_0 s_1 \dots$ be a path of $\mathcal{G}$ such that $s_0 = \{p\}$ and $s_1 = \{p,q\}$. Observe that there are multiple actions α such that $\tau(s_0,\alpha) = s_1$: namely, the one where both agents set p to true, or where just one of them does (and agent 2 sets q to true).*

Construct now the CGS-EPC $\mathcal{G}'$ as in Definition 7 and consider the following four sequences $\lambda' = s'_0 s'_1 s'_2 \dots$ where:

(a) $s'_0 = \{p\}$, $s'_1 = \{c_{1p}, c_{2p}, c_{2q}, p, turn\}$, $s'_2 = \{p,q\}$, $\dots$

(b) $s'_0 = \{p\}$, $s'_1 = \{c_{1p}, c_{2q}, p, turn\}$, $s'_2 = \{p,q\}$, $\dots$

(c) $s'_0 = \{p, c_{1p}\}$, $s'_1 = \{c_{1p}, c_{2q}, p, turn\}$, $s'_2 = \{p,q\}$, $\dots$

(d) $s'_0 = \{p\}$, $s'_1 = \{c_{2q}, p, turn\}$, $s'_2 = \{p,q\}$, $\dots$

Observe that (a) and (b) are both examples of canonical paths (up to the considered state), corresponding to two actions that might have led from s_0 to s_1 in $\mathcal{G}$. On the other hand, (c) is a non-example while being a path of $\mathcal{G}'$ satisfying $(\dagger)$, since s'_0 is not canonical. Finally, sequence (d) satisfies $(\dagger)$ but it is not a path of $\mathcal{G}'$, since it is not possible to obtain s'_2 from s'_1.

The next result extends the statement of Lemma 4 to paths generated by a specific strategy. Given a $\mathcal{G}'$-strategy σ'_C and a state $s' \in S'$, let $\Pi(out(s',\sigma'_C)) = \{\lambda \mid \lambda' \in out(s',\sigma'_C)\}$, i.e., all the "projections" of paths λ' in $out(s',\sigma'_C)$ to paths λ in $\mathcal{G}$, obtained through $(\dagger)$.

Lemma 5. *Given a CGS-SPC $\mathcal{G}$, the corresponding CGS-EPC $\mathcal{G}'$ is such that:*

1. *for every joint strategy σ_C in $\mathcal{G}$, there exists a strategy σ'_C in $\mathcal{G}'$ such that for every state $s \in S$ we have that $\Pi(out(s'_\star,\sigma'_C)) = out(s,\sigma_C)$;*
2. *for every joint strategy σ'_C in $\mathcal{G}'$, there exists a strategy σ_C in $\mathcal{G}$ such that for all* canonical *states $s' \in S'$ we have that $\Pi(out(s',\sigma'_C)) = out(s'_{|\Phi},\sigma_C)$.*

proof sketch. We first prove (1). Given strategy σ_C in $\mathcal{G}$, for $i \in C$ define σ'_i as follows:

$$\sigma'_i(s') \quad = \quad \begin{cases} \{c_{ip} \mid p \in \sigma_i(s) \text{ and } s = s'_{|\Phi}\} & \text{if } s'(turn) = 0 \\ \emptyset & \text{otherwise} \end{cases}$$

Observe that if $s'(turn) = 1$ agents in C are obliged to play action $\emptyset$ by Definition 7, since it is their only available action. By combining all definitions above, we get that $\Pi(out(s'_\star,\sigma'_C)) = out(s,\sigma_C)$ for an arbitrary state $s \in S$.

To prove (2), we start from a strategy σ'_C in $\mathcal{G}'$. For any state $s \in S$, define $\sigma_i(s) = \{p \in \Phi_i \mid c_{ip} \in \sigma'_i(s'_\star)\}$. Note that the assumption in Definition 7 that all variables outside of Φ are put to false at stage $2k+1$ in $\mathcal{G}'$ is crucial here. In fact, without this assumption we would only be able to prove that $\Pi(out(s',\sigma'_C)) \supseteq out(s'_{|\Phi},\sigma_C)$, as a strategy σ'_C may associate a different action to states s'_1 and s'_2 that coincide on Φ and that are realised in a path $\lambda' \in out(s',\sigma'_C)$. $\square$

By means of Lemma 5 we are able to prove the main result of this section.

Theorem 6. *Given any CGS-SPC $\mathscr{G}$, the corresponding CGS-EPC $\mathscr{G}'$ is such that for all state-formulas φ and path-formulas ψ in* ATL^* *the following holds:*

$$\text{for all } s \in S \quad (\mathscr{G}, s) \models \varphi \quad \text{iff} \quad (\mathscr{G}', s'_\star) \models tr(\varphi)$$
$$\text{for all } \lambda \text{ of } \mathscr{G} \quad (\mathscr{G}, \lambda) \models \psi \quad \text{iff} \quad (\mathscr{G}', \lambda'_\star) \models tr(\psi) \text{ for any } \lambda'_\star.$$

Proof. The proof is by induction on the structure of formulas φ and ψ. The base case for $\varphi = p$ follows from the fact that $s = s'_{|\Phi}$ for all s' associated to s, and in particular also for $s'_\star$. As to the inductive cases for boolean connectives, these follow immediately by the induction hypothesis.

Now suppose that $\varphi = \langle\langle C \rangle\rangle \psi$. As to the left-to-right direction, assume that $(\mathscr{G}, s) \models \varphi$. By the definition of the semantics, for some strategy σ_C, for all $\lambda \in out(s, \sigma_C)$, $(\mathscr{G}, \lambda) \models \psi$. By Lemma 5.1 we can find a strategy σ'_C in $\mathscr{G}'$ such that $\Pi(out(s'_\star, \sigma'_C)) = out(s, \sigma_C)$. By induction hypothesis, we know that for all $\lambda \in out(s, \sigma_C)$ we have that $(\mathscr{G}', \lambda'_\star) \models tr(\psi)$. These two facts combined imply that for all $\lambda' \in out(s'_\star, \sigma'_C)$ we have that $(\mathscr{G}', \lambda'_\star) \models tr(\psi)$, i.e., by the semantics, that $(\mathscr{G}', s'_\star) \models \langle\langle C \rangle\rangle tr(\psi)$, obtaining the desired result. The right-to-left direction can be proved similarly, by using Lemma 5.2.

Further, if φ is a state formula, $(\mathscr{G}, \lambda) \models \varphi$ iff $(\mathscr{G}, \lambda[0]) \models \varphi$, iff by induction hypothesis $(\mathscr{G}', \lambda[0]'_\star) \models tr(\varphi)$, that is, $(\mathscr{G}', \lambda'_\star) \models tr(\varphi)$.

For $\psi = \bigcirc \psi_1$, suppose that $(\mathscr{G}, \lambda[1, \infty]) \models \psi_1$. By induction hypothesis, this is the case if and only if $(\mathscr{G}', (\lambda[1, \infty])'_\star) \models tr(\psi_1)$. Recall that by (†), we have that $(\lambda[1, \infty])'_\star = \lambda'_\star[2, \infty]$. This is the case because, when moving from λ to $\lambda'_\star$, we include an additional state $\lambda'_\star[1]$ in which the aggregation takes place. Therefore, $(\mathscr{G}', \lambda'_\star[2, \infty]) \models tr(\psi_1)$, that is, $(\mathscr{G}', \lambda'_\star) \models \bigcirc \bigcirc tr(\psi_1) = tr(\psi)$. The case for $\psi = \psi_1 \mathscr{U} \psi_2$ is proved similarly. □

As a consequence of Theorem 6, if we want to model-check an ATL^*-formula φ at a state s of an CGS-SPC $\mathscr{G}$, we can check its translation $tr(\varphi)$ at the related state $s'_\star$ of the associated CGS-EPC $\mathscr{G}'$. Together with the observation that both the associated game $\mathscr{G}'$ and the translation φ are polynomial in the size of $\mathscr{G}$ and φ, we obtain the following:

Corollary 7. *The* ATL^* *model-checking problem for CGS-SPC can be reduced to the* ATL^* *model-checking problem for CGS-EPC.*

5 Computational Complexity of Shared Control Structures

The results proved in the previous sections allow us to obtain complexity results for the model checking of an ATL^* (or ATL) specification φ on a *pointed* CGS-SPC $(\mathscr{G}, s)$ defined in Definition 3.

Theorem 8. *The model-checking problem of* ATL *specifications in CGS-SPC is Δ_3^p-complete.*

Proof. As for membership, given a pointed CGS-SPC $(\mathscr{G}, s)$ and an ATL specification φ, by the translation tr introduced in Section 4 and Theorem 6 we have that $(\mathscr{G}, s) \models \varphi$ iff $(\mathscr{G}', s') \models tr(\varphi)$. Also, we observe that the CGS-EPC $\mathscr{G}'$ is of size polynomial in the size of $\mathscr{G}$, and that model checking ATL with respect to CGS-EPC is Δ_3^p-complete [2]. For hardness, it is sufficient to observe that CGS-EPC are a subclass of CGS-SPC. □

As for the verification of ATL^*, we can immediately prove the following result:

Theorem 9. *The model-checking problem of* ATL^* *specifications in CGS-SPC is PSPACE-complete.*

Proof. Membership follows by the PSPACE-algorithm for ATL^* on general CGS [4]. As for hardness, we observe that satisfiability of an LTL formula φ can be reduced to the model checking of the ATL^* formula $\langle\langle 1 \rangle\rangle \varphi$ on a CGS-SPC with a unique agent 1. ☐

In Section 3 we showed how three examples of iterated games from the literature on strategic reasoning can be modelled as CGS-SPC, and how the problem of determining the existence of a winning strategy can therefore be reduced to model checking an ATL^* specification. Let $\mathsf{E\text{-}WIN}(G,i)$ be the decision problem of deciding whether agent i has a memory-less winning strategy in game G. As an immediate consequences of Theorem 9 we obtain:

Corollary 10. *If G is an iterated boolean game with shared control,* $\mathsf{E\text{-}WIN}(G,i)$ *is in PSPACE.*

An analogous result cannot be obtained for influence and aggregation games directly. Decision problems in these structures are typically evaluated with respect to the number of agents and issues, and the size of the CGS-SPCs associated to these games are already exponential in these parameters. Therefore, in line with previous results obtained in the literature [7], we can only show the following:

Corollary 11. *If G is an influence game or an aggregation game, then* $\mathsf{E\text{-}WIN}(G,i)$ *is in PSPACE in the size of the associated CGS-SPC.*

6 Conclusion

In this contribution we have introduced a class of concurrent game structures with shared propositional control, or CGS-SPC. Then, we have interpreted popular logics for strategic reasoning ATL and ATL^* on these structures. Most importantly, we have shown that CGS-SPC are a general framework, whereby we can capture iterated boolean games and their generalisation to shared control, as well as influence and aggregation games. The main result of the paper shows that the model checking problem for CGS-SPC can be reduced to the verification of standard CGS with exclusive control, which in turn allows us to establish a number of complexity results.

The results proved here open up several research directions. Firstly, in this paper we have focussed on the verification problem, but what about satisfiability and validity? The undecidability result provided by Gerbrandy [5] for $\mathsf{CL\text{-}PC}$ with shared control does not immediately transfer to CGS-SPC, as the relevant languages are different: $\mathsf{CL\text{-}PC}$ includes normal modal 'diamond-operators' $\langle C \rangle$ and 'box-operators' $[C]$, while our $\langle\langle C \rangle\rangle$ is non-normal.[4]

Further, given our reduction of CGS-SPC to CGS with exclusive control, one may wonder what the benefits of our move to shared control are. As our three examples have demonstrated, shared control allows to model in a natural way complex interactions between agents concerning the assignment of truth values to propositional variables. The strategic aspects of these games remain largely unexplored, and clean characterisations of equilibria and other game-theoretic concepts seem rather hard to prove, supporting the use of automated verification in these context.

Compact representations of CGS with exclusive control are a thriving subject of research in the formal verification community (see, e.g., [12, 16, 15]). There, so-called reactive modules define for every action whether it is available by means of a boolean formula. In future work we plan to investigate such compact representations for CGS with shared control. This requires in particular a compact representation of the transition function τ, which becomes more involved in the shared control setting.

[4] We observe that, on the other hand, following van der Hoek and Wooldridge [14] the fragment of the language of ATL without 'until' can be embedded into that of $\mathsf{CL\text{-}PC}$ by identifying $\langle\langle C \rangle\rangle \bigcirc \phi$ with $\langle C \rangle [N \setminus C] \phi$.

Finally, we conclude by remarking that a key assumption on our CGS (both with exclusive and shared control) is that agents have perfect knowledge of the environment they are interacting in and with. Indeed, in Definition 7 the dummy agent $*$ is able to mimick the aggregation function τ as she can observe the values of c_{ip} for any other agent i. In contexts of imperfect information, agents can only observe the atoms they can act upon. Hence, an interesting question is whether our reduction of CGS-SPC to CGS-EPC goes through even when imperfect information is assumed.

Acknowledgements

The authors are grateful to the three anonymous reviewers for their helpful comments. F. Belardinelli acknowledges the support of the French ANR JCJC Project SVeDaS (ANR-16-CE40-0021).

References

[1] Rajeev Alur, Thomas A. Henzinger & Orna Kupferman (2002): *Alternating-time temporal logic*. *Journal of the ACM* 49(5), pp. 672–713, doi:10.1145/585265.585270.

[2] Francesco Belardinelli & Andreas Herzig (2016): *On Logics of Strategic Ability based on Propositional Control*. In: *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI-2016)*.

[3] Elise Bonzon, Marie-Christine Lagasquie-Schiex, Jérôme Lang & Bruno Zanuttini (2006): *Boolean games revisited*. In: *Proceedings of the 17th European Conference on Artificial Intelligence (ECAI-2006)*.

[4] Nils Bulling, Jürgen Dix & Wojciech Jamroga (2010): *Model Checking Logics of Strategic Ability: Complexity**. In Mehdi Dastani, Koen V. Hindriks & John-Jules Charles Meyer, editors: *Specification and Verification of Multi-agent Systems*, Springer US, pp. 125–159, doi:10.1007/978-1-4419-6984-2_5.

[5] Jelle Gerbrandy (2006): *Logics of propositional control*. In: *Proceedings of the 5th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2006)*, doi:10.1145/1160633.1160664.

[6] Umberto Grandi, Davide Grossi & Paolo Turrini (2015): *Equilibrium Refinement through Negotiation in Binary Voting*. In: *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI-2015)*.

[7] Umberto Grandi, Emiliano Lorini, Arianna Novaro & Laurent Perrussel (2017): *Strategic Disclosure of Opinions on a Social Network*. In: *Proceedings of the 16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS-2017)*.

[8] Davide Grossi, Emiliano Lorini & François Schwarzentruber (2015): *The Ceteris Paribus Structure of Logics of Game Forms*. *Journal of Artificial Intelligence Research (JAIR)* 53, pp. 91–126, doi:10.1613/jair.4666.

[9] Julian Gutierrez, Paul Harrenstein & Michael Wooldridge (2013): *Iterated Boolean games*. In: *Proceedings of the 23th International Joint Conference on Artificial Intelligence (IJCAI-2013)*.

[10] Julian Gutierrez, Paul Harrenstein & Michael Wooldridge (2015): *Iterated Boolean games*. *Information and Computation* 242, pp. 53–79, doi:10.1016/j.ic.2015.03.011.

[11] Paul Harrenstein, Wiebe van der Hoek, John-Jules Meyer & Cees Witteveen (2001): *Boolean games*. In: *Proceedings of the 8th Conference on Theoretical Aspects of Rationality and Knowledge (TARK-2001)*.

[12] Wiebe van der Hoek, Alessio Lomuscio & Michael Wooldridge (2006): *On the complexity of practical ATL model checking*. In: *5th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2006)*, doi:10.1145/1160633.1160665.

[13] Wiebe van der Hoek, Dirk Walther & Michael Wooldridge (2010): *Reasoning About the Transfer of Control*. *Journal of Artificial Intelligence Research (JAIR)* 37, pp. 437–477, doi:10.1613/jair.2901.

[14] Wiebe van der Hoek & Michael Wooldridge (2005): *On the logic of cooperation and propositional control*. *Artificial Intelligence* 164(1-2), pp. 81–119, doi:10.1016/j.artint.2005.01.003.

[15] Xiaowei Huang, Qingliang Chen & Kaile Su (2015): *The Complexity of Model Checking Succinct Multiagent Systems*. In: *Proceedings of the 24th International Joint Conference on Artificial Intelligence, (IJCAI-2015)*.

[16] Wojciech Jamroga & Thomas Ågotnes (2007): *Modular interpreted systems*. In: *Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2007)*, doi:10.1145/1329125.1329286.

[17] Omer Lev & Jeffrey S. Rosenschein (2012): *Convergence of iterative voting*. In: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS-2012)*.

[18] Reshef Meir, Maria Polukarov, Jeffrey S. Rosenschein & Nicholas R. Jennings (2010): *Convergence to Equilibria in Plurality Voting*. In: *Proceedings of the 24th AAAI conference on Artificial Intelligence (AAAI-2010)*.

[19] Svetlana Obraztsova, Evangelos Markakis, Maria Polukarov, Zinovi Rabinovich & Nicholas R. Jennings (2015): *On the Convergence of Iterative Voting: How Restrictive Should Restricted Dynamics Be?* In: *Proceedings of the 29th AAAI Conference on Artificial Intelligence, (AAAI-2015)*.

[20] Amir Pnueli (1977): *The temporal logic of programs*. In: *Proceedings of the 18th IEEE Symposium on Foundations of Computer Science*, doi:10.1109/SFCS.1977.32.