Coalition Strategy Logic

Anonymous Author(s)
Submission Id: 710

ABSTRACT

We introduce *Coalition Strategy Logic* (CSL), which combines the intuitions behind Coalition Logic (CL) and Strategy Logic (SL). Specifically, CSL allows for arbitrary quantification over actions of groups of agents. The motivation behind CSL is two-fold. First, we show that CSL is strictly more expressive than other known coalition logics, and then we discuss its model-checking procedure. Second, we provide a sound and complete axiomatisation of the logic, which is, to the best of our knowledge, *the first axiomatisation* of any strategy logic in the literature.

KEYWORDS

Strategy Logic, Coalition Logic, Expressiveness, Model Checking, Completeness

ACM Reference Format:
Anonymous Author(s). 2025. Coalition Strategy Logic. In *Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Detroit, Michigan, USA, May 19 – 23, 2025*, IFAAMAS, 9 pages.

1 INTRODUCTION

Logics for strategic reasoning constitute a numerous family of formal tools devised to model, verify, and reason about the abilities and strategies of (groups of) autonomous agents in a competitive environment [6, 13, 27, 32, 34]. Strategies here are 'recipes' telling agents what to do in order to achieve their goals. The competitive environment part arises from the fact that in the presence of several agents trying to achieve their own goals, the actions of one agent may influence the available strategies of another agent. Such logics have been shown to be invaluable for specification and verification within various domains: neuro-symbolic reasoning [4], voting protocols [25], autonomous submarines [16], manufacturing robots [15], and so on.

The prime representatives of logics for strategic reasoning are *coalition logic* (CL) [32], *alternating-time temporal logic* (ATL), [6], and *strategy logic* (SL) [29] (and numerous variations thereof). CL extends the language of propositional logic with constructs $\langle\!\langle C \rangle\!\rangle \varphi$ meaning 'coalition C has a joint action such that φ holds in the next state (no matter what agents outside of the coalition do at the same time)'. ATL extends further the abilities of agents to force temporal goals expressed with the help of such modalities as 'Until' and 'Release'. Finally, SL allows for a more fine-tuned quantification over agents' abilities: while in both CL and ATL we have a fixed quantification prefix $\exists\forall$, in SL we can have arbitrary quantification prefixes. Thus, in SL we can reason, for example, about agents sharing their strategies, and such game-theoretic notions like dominant strategies and Nash equilibria. Hence, ATL is strictly more expressive than CL, and, in turn, SL is strictly more expressive than ATL (and its more general cousin ATL*).

The incredible expressivity of SL comes at a price: the complexity of the model checking for SL is non-elementary, and it remains quite high for different fragments as well [27]. The satisfiability problem for SL is Σ_1^1-hard [28]. The latter in particular entails that the full language of SL is not finitely, or even recursively, axiomatisable.

Sound and complete axiomatisations of CL [20, 32] and ATL [24] are now classic results in the field. However, to the best of our knowledge, *no axiomatisations of any SL's were considered in the literature before.*

In this paper, we introduce a variation of SL that we call *coalition strategy logic* (CSL). As its name suggests, the logic combines both features of CL and SL. In terms of the former, we restrict ourselves to quantification over agent's actions, rather than strategies, and in terms of the latter, we allow arbitrary quantification prefixes. Moreover, we allow action labels to be present in the language. Thus, CSL is also related to ATL *with explicit strategies* [36].

We deem our contribution as two-fold. First, we show that the introduced CSL is quite a special coalition logic, being strictly more expressive than other known coalition logics in the literature. With such a remarkable expressivity comes the *PSPACE*-complete model checking problem. Second, we provide a sound and complete axiomatisation of CSL, which is, as far as we can tell, *the first axiomatisation of a stategy logic*. With this contribution, we create a foundation for discovering axiomatisations of other strategy logics.

The rest of the paper is organised as follows. In Section 2 we present the syntax and semantics of CSL. Then, in Section 3, we situate CSL in the greater landscape of coalition logics and argue that it is more expressive than any other logic mentioned in the section. In Section 4, we discuss the complexity of the model checking problem for CSL. An axiomatisation of CSL, as well as the corresponding completeness proof, is presented in Section 5. Finally, we conclude and point out further research directions in Section 6.

2 SYNTAX AND SEMANTICS

In what follows, we use the word "countable" in its standard mathematical sense, that is: a set is countable if and only if it is either finite or in bijection with $\mathbb{N}$.

Definition 2.1. A *signature* is a triple $\alpha = \langle n, C, \mathsf{Ap} \rangle$, where $n \geq 1$ is a natural number, C is a non-empty countable set of *constants*, and Ap is a non-empty countable set of *atomic proposition* (or *atoms*) such that $\mathsf{Ap} \cap C = \emptyset$.

Fix a non-empty countable set V of *variables* that is disjoint from any other set in any given signature α. The *language of coalition strategy logic* (CSL) is defined by the following grammar:

$$\varphi := p \mid \neg\varphi \mid \ (\varphi \wedge \varphi) \mid \langle\!\langle t_1 \cdots t_n \rangle\!\rangle\, \varphi \mid \forall x \varphi,$$

where $p \in \mathsf{Ap}$, $t_i \in C \cup \mathsf{V}$, $x \in \mathsf{V}$, and all the usual abbreviations of propositional logic (such as $\vee$, $\to$, $\leftrightarrow$) and conventions for deleting parentheses hold. The existential quantifier $\exists x \varphi$ is defined as

Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), A. El Fallah Seghrouchni, Y. Vorobeychik, S. Das, A. Nowe (eds.), May 19 – 23, 2025, Detroit, Michigan, USA.

$\neg\forall x \neg\varphi$. Formula $(\!(t_1 \cdots t_n)\!)\,\varphi$ is read as 'after the agents execute actions assigned to $t_1 \cdots t_n$, φ is true', and $\forall x \varphi$ is read as 'for all actions x, φ holds'. Given a formula $\varphi \in \mathsf{CSL}$, the *size of* φ, denoted by $|\varphi|$, is the number of symbols in φ.

Definition 2.2. Given a formula φ, we define its *set of free variables* $\mathsf{FV}(\varphi)$ by the following cases:

(1) If $\varphi \in \mathsf{Ap}$ then $\mathsf{FV}(\varphi) = \emptyset$;
(2) If $\varphi = \neg\varphi_1$ then $\mathsf{FV}(\varphi) = \mathsf{FV}(\varphi_1)$;
(3) If $\varphi = \varphi_1 \wedge \varphi_2$ then $\mathsf{FV}(\varphi) = \mathsf{FV}(\varphi_1) \cup \mathsf{FV}(\varphi_2)$;
(4) if $\varphi = (\!(t_1 \cdots t_n)\!)\,\varphi_1$ then $\mathsf{FV}(\varphi) = \mathsf{FV}(\varphi_1) \cup \{t_i \mid t_i \in \mathsf{V}\}$;
(5) if $\varphi = \forall x \varphi_1$ then $\mathsf{FV}(\varphi) = \mathsf{FV}(\varphi_1) \setminus \{x\}$.

A formula φ such that $\mathsf{FV}(\varphi) = \emptyset$ is called a *closed formula*, or a *sentence*.

Definition 2.3. A *Kripke frame* is a tuple $\mathcal{F} = \langle \Sigma, S, R \rangle$, where Σ is a non-empty countable alphabet, S is a non-empty countable set of states such that $\Sigma \cap S = \emptyset$, and $R \subseteq S \times \Sigma \times S$ is a ternary relation, dubbed transition relation. We say that $\mathcal{F}$ is:

serial if for every $s \in S$ and for every $a \in \Sigma$, there is a $t \in S$ such that $\langle s, a, t \rangle \in R$;
functional whenever for all $s, t, v \in S$ and for every $a \in \Sigma$, if $\langle s, a, t \rangle \in R$ and $\langle s, a, v \rangle \in R$ then $t = v$.

Definition 2.4. A *game frame* is a tuple $\mathcal{G} = \langle n, \mathsf{Ac}, \mathcal{D}, S, R \rangle$ with triple $\langle \mathcal{D}, S, R \rangle$ being a serial and functional Kripke frame, where:

- n is a positive natural number;
- Ac is a countable set of *actions*;
- S is a countable set of *states*;
- $\mathcal{D}$ is a set of tuples of elements of Ac of length n (elements of this set will be called *decisions*);
- $R \subseteq S \times \mathcal{D} \times S$ is a ternary relation.

A *Concurrent Game Structure* (CGS) is a pair $\mathfrak{G} = \langle \mathcal{G}, \mathcal{V} \rangle$ where $\mathcal{G}$ is a game frame, and $\mathcal{V} : \mathsf{Ap} \to \mathcal{P}(S)$ is a *valuation function* assigning to each atomic proposition a subset of S.

Let $Prop(s) = \{p \in \mathsf{Ap} \mid s \in \mathcal{V}(p)\}$ be the set of all atomic propositions true in state s. We define the *size of CGS* $\mathfrak{G}$ as $|\mathfrak{G}| = n + |\mathsf{Ac}| + |\mathcal{D}| + |S| + |R| + \sum_{s \in S} |Prop(s)|$, where $|\mathcal{D}| = |\mathsf{Ac}|^n$. We call CGS $\mathfrak{G}$ *finite*, if $|\mathfrak{G}|$ is finite.

Definition 2.5. Given a signature $\alpha = \langle m, C, \mathsf{Ap} \rangle$, and a CGS $\mathfrak{G} = \langle n, \mathsf{Ac}, \mathcal{D}, S, R, \mathcal{V} \rangle$ we say that $\mathfrak{G}$ is constructed over α iff $m = n$ and $C = \mathsf{Ac}$.

Definition 2.6. Let φ be a sentence and $\mathfrak{G}$ be a CGS that are both constructed over the same signature α. The *satisfaction relation* $\mathfrak{G}, s \models \varphi$ is inductively defined as follows:

$\mathfrak{G}, s \models p$	iff $s \in \mathcal{V}(p)$
$\mathfrak{G}, s \models \neg\psi$	iff $\mathfrak{G}, s \not\models \psi$
$\mathfrak{G}, s \models \psi \wedge \chi$	iff $\mathfrak{G}, s \models \psi$ and $\mathfrak{G}, s \models \chi$
$\mathfrak{G}, s \models (\!(a_1 \cdots a_n)\!)\,\psi$	iff $\exists t \in S$ s.t. $\langle s, a_1, ..., a_n, t \rangle \in R$ and $\mathfrak{G}, t \models \psi$
$\mathfrak{G}, s \models \forall x \psi$	iff $\forall a \in \mathsf{Ac} : \mathfrak{G}, s \models \psi[a/x]$

where $a_1 \cdots a_n$ are constants, and $\psi[a/x]$ denotes the result of substituting every occurrence of the variable x with the constant a in ψ. We will also sometimes write $\vec{a}$ for $a_1 \cdots a_n$.

Definition 2.7. Given a formula φ whose set of free variables is $\{x_1, \ldots, x_n\}$, we denote by $C(\varphi)$ the *closure* of φ, which is the formula $\forall x_1 \cdots \forall x_n \varphi$.

Definition 2.8. Let $\mathfrak{G}$ be a CGS constructed over a signature α and φ a formula constructed α. Given a state s of $\mathfrak{G}$ we write $\mathfrak{G}, s \models \varphi$ iff $\mathfrak{G}, s \models C(\varphi)$. We say that φ is *valid in a CGS* $\mathfrak{G}$ (written $\mathfrak{G} \models \varphi$) iff $\mathfrak{G}, s \models \varphi$ for every state s. Finally, we say that φ is *valid* (written $\models \varphi$) iff it is valid in every CGS constructed over α. Given a set of formulae X, we write $\mathfrak{G}, s \models X$ if for every formula $\varphi \in X$, $\mathfrak{G}, s \models \varphi$. Finally, we write $X \models \psi$ and we say that ψ is a logical consequence of X iff $\mathfrak{G} \models X$ implies $\mathfrak{G} \models \psi$ for every CGS $\mathfrak{G}$ constructed over α.

The next proposition shows that we can give an alternative and equivalent characterization for the truth a strategic formula in a state of a CGS.

Proposition 2.9. *Let $\mathfrak{G} = \langle n, \mathsf{Ac}, \mathcal{D}, S, R, \mathcal{V} \rangle$ be a CGS, $s \in S$, and $\varphi = (\!(a_1 \cdots a_n)\!)\,\psi$ and suppose that both $\mathfrak{G}$ and φ are constructed over the same signature α. Then $\mathfrak{G}, s \models \varphi$ if and only if $\forall t \in S$: $\langle s, a_1, \ldots, a_n, t \rangle \in R$ implies $\mathfrak{G}, t \models \psi$.*

Proof. The left-to-right direction is granted because of seriality, while the converse direction holds due to totality. □

Remark 1. *Due to Proposition 2.9, we we have that $\mathfrak{G}, s \models (\!(\vec{a})\!)\,\psi$ iff $\mathfrak{G}, t \models \psi$ for the unique t such that $\langle s, \vec{a}, t \rangle \in R$.*

Example 2.10. As observed in [8], strategy logics are expressive enough to capture the intuitions behind *Stackelberg equilibrium.* Such an equilibrium is applicable to scenarios where a leader commits to a strategy, and the follower, observing the strategy of the leader, provides her best response. Stackelberg equilibria are prominent in security games [33], where the attacker observes the defender committing to a defensive strategy and then decides on the best way to attack (if at all). We can express such a scenario for the case of one-step strategies by the following CSL formula: $\forall x_d \exists x_a \forall x_e (\!(x_d, x_a, x_e)\!)\, win_a$. The formula intuitively means that for all actions of the defender, the attacker has a counter-action guaranteeing the win for all actions of the environment.

Similarly to [29], with CSL we can express the existence of deterministic *Nash equilibrium* for Boolean goals. If $\psi_1, ..., \psi_n$ are goal formulae of agents, we can assert the existence of strategy profile $x_1, ..., x_n$ such that if any agent i achieves her goal ψ_i by deviating from $x_1, ..., x_n$, then she can also achieve her goal by sticking to the action profile. The existence of such a profile can be expressed by the following CSL formula:

$$\exists x_1, ..., x_n \big(\bigwedge_{i=1}^{n} \exists y_i (\!(x_1, ..., y_i, ..., x_n)\!)\, \psi_i \to (\!(x_1, ..., x_i, ..., x_n)\!)\, \psi_i \big).$$

Example 2.11. Two horror film enthusiasts, Anna and Brita, are deliberating between which film to watch: a folk horror film or a sci-fi horror film. Anna, being indifferent to both genres, just wants to watch one of the films (goal $f_a \vee s_a$). Brita, on the other hand, has a strong preference for the folk horror film, and, additionally, she would love to watch it with Anna (goal $f_a \wedge f_b$). In this scenario, it is clear that Anna can achieve her goal while also letting Brita to achieve hers (Anna's strategy here is to watch the folk horror film). This property can be expressed by the CSL formula

$$\varphi := \exists x_a (\forall x_b (\!(x_a, x_b)\!)\, (f_a \vee s_a) \wedge \exists x_b (\!(x_a, x_b)\!)\, (f_a \wedge f_b)).$$

This is exactly the setting of socially friendly coalition logic [19], which we will look closer at in Section 3.

Consider the corresponding CGS $\mathfrak{G}$ depicted in Figure 1. The

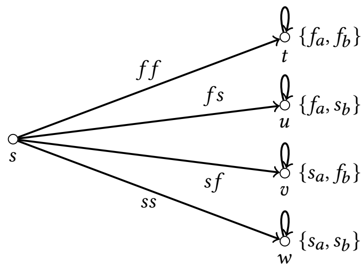

Figure 1: CGS $\mathfrak{G}$ for two agents and two actions. True atomic propositions represented in sets near corresponding states. For readability, self-loops labelled by all combinations of actions, are depicted as unlabelled loops.

CGS has five states, s, t, u, v, and w, and is defined over the set of two actions, f for 'folk horror', and s for 'sci-fi horror'. Propositional atoms correspond to the outcomes of Anna and Brita's actions with f_a and s_b being true in state u meaning that Anna watches the folk horror film, and Brita watches the sci-fi horror film.

Now, it is easy to verify that $\mathfrak{G}, s \models \varphi$ since Anna can always choose action f and thus let Brita play f as well to reach her own goal.

3 RELATION TO OTHER FORMALISMS

In order to appreciate the richness of CSL, we compare the logic to other related logics of strategic ability. In our comparison we will use two salient features of CSL. First, the logic allows for *arbitrary quantification prefixes for agents' actions*. This includes using the same strategy variable for different agents to capture *strategy sharing*. The second special feature of CSL is the presence of *explicit action labels* in its syntax.

Definition 3.1. Let L_1 and L_2 be two languages, and let $\varphi \in \mathsf{L}_1$ and $\psi \in \mathsf{L}_2$. We call φ and ψ *equivalent*, if for all CGSs $\mathfrak{G}, s$ it holds that $\mathfrak{G}, s \models \varphi$ if and only if $\mathfrak{G}, s \models \psi$.

If for all $\varphi \in \mathsf{L}_1$ there exists and equivalent $\psi \in \mathsf{L}_2$, we say that L_2 is *at least as expressive as* L_1 and write $\mathsf{L}_1 \leqslant \mathsf{L}_2$. We say that L_2 is *strictly more expressive than* L_2 (denoted by $\mathsf{L}_1 < \mathsf{L}_2$) if $\mathsf{L}_1 \leqslant \mathsf{L}_2$ and $\mathsf{L}_2 \nleqslant \mathsf{L}_1$.

These two features of CSL, arbitrary quantification prefixes and explicit actions, on their own are not unique in the landscape of logics for strategic reasoning. Thus, arbitrary quantification prefixes are a hallmark feature of the whole family of *strategy logics* (see, e.g., [8, 29]), to which CSL belongs. Indeed, CSL can be considered as a variation of the next-time fragment of *strategy logic with simple goals* (SL[SG]) [8], which, in turn, is an extension of ATL [6] with arbitrary quantification prefixes for agents' strategies. The idea to refer to action in the language has also been explored, with, perhaps, a prime example being ATL *with explicit strategies* (ATLES) [36]. Another example of a logic with explicit actions is *action logic* (AL) [11], which we will look at closer later in this section.

Even though both of the main features of CSL have been somewhat explored in the literature, to our knowledge, CSL is the first logic for strategic reasoning that combines *both* of them. To further demonstrate the unique position of our logic in the landscape of related logics, we compare it to some known coalition logics.

Coalition logic. The original *coalition logic* (CL) [32], similarly to ATL, allows only single alternation of quantifiers in coalitional modalities. Moreover, this quantification is implicit. Thus, CL extends the language of propositional logic with constructs $\langle\langle C\rangle\rangle\varphi$ that mean 'there is a strategy for coalition C to achieve φ in the next step (whatever agents outside of the coalition do at the same time)'.

To introduce the semantics of CL, we will denote the choice of actions by coalition $C \subseteq Agt$ with $|Agt| = n$ as σ_C, and denote $Agt \setminus C$ as $\overline{C}$. Finally, $\sigma_C \cup \sigma_{\overline{C}} \in \mathcal{D}$ is a decision. The semantics of $\langle\langle C\rangle\rangle\varphi$ for a given CGS $\mathfrak{G}$ is then defined as

$$\begin{aligned}\mathfrak{G}, s \models \langle\langle C\rangle\rangle\varphi \text{ iff } &\exists\sigma_C, \forall\sigma_{\overline{C}} : \mathfrak{G}, t \models \varphi\\ &\text{with } t \in S \text{ s.t. } \langle s, \sigma_C \cup \sigma_{\overline{C}}, t\rangle \in R.\end{aligned}$$

The translation from formulas CL to formulas of CSL can be done recursively using the following schema for coalitional modalities: $tr(\langle\langle C\rangle\rangle\varphi) \rightarrow \exists\vec{x}\forall\vec{y}((\vec{x},\vec{y}))\, tr(\varphi)$, where variables $\vec{x}$ (all different) quantify over actions of C, and $\vec{y}$ (all different) quantify over actions of $Agt \setminus C$.

At the same time, one cannot refer to particular actions in CL formulas, as well as express sharing strategies between agents. We can exploit either of these features to show that CSL is strictly more expressive than CL. Indeed, consider a CSL formula $\exists x((x,x))\,\neg p$ meaning that there is an action that *both* agents 1 and 2 should use to reach a $\neg p$-state. We can construct two CGSs that are indistinguishable by any CL formulas. At the same time, $\exists x((x,x))\,\neg p$ will hold in one structure and be false in another.

Consider two structures depicted in Figure 2.

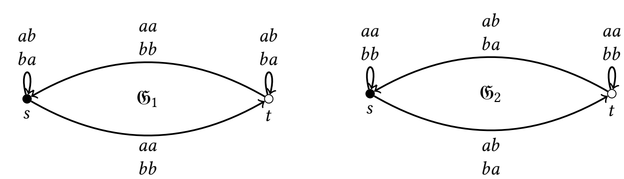

Figure 2: CGSs $\mathfrak{G}_1$ and $\mathfrak{G}_2$ for two agents and two actions. Propositional variable p is true in black states.

It is easy to see that $\mathfrak{G}_1, s$ and $\mathfrak{G}_2, s$ cannot be distinguished by any CL formula[1]. Indeed, both structures agree on the valuation of propositional variable p in corresponding states. Moreover, none of the agents, 1 and 2, can on their own force a transition from state s to state t. At the same time, the grand coalition $\{1,2\}$ can match any transition in one structure with a transition with the same effect in the other structure. Now, we can verify that $\mathfrak{G}_1, s \models \exists x((x,x))\,\neg p$ and $\mathfrak{G}_2, s \not\models \exists x((x,x))\,\neg p$. For the case of $\mathfrak{G}_1, s \models \exists x((x,x))\,\neg p$, it is enough to assign action a to x to have $\mathfrak{G}_1, s \models ((a,a))\,\neg p$. To make

[1]These structures are, in fact, bisimilar [3], and hence satisfy the same formulas of CL and ATL. The discussion of bisimulations for all the logics we mention is, however, beyond the scope of this paper, and we leave it for future work.

$\mathfrak{G}_2, s \not\models \exists x((x,x))\, \neg p$ hold, one needs to provide an action that once executed by both agents will force the transition to state t. It is easy to see that there is no such an action in $\mathfrak{G}_2, s$.

Having the translation from CL to CSL on the one hand, and the indistinguishability result on the other, we hence conclude that CSL is *strictly more expressive* than CL.

Proposition 3.2. CL < CSL.

Conditional strategic reasoning and socially friendly coalition logic. With the expressive power of CSL we can go much further than the classic CL. In particular, we can express in our logic such interesting coalition logics like *logic for conditional strategic reasoning* (ConStR) [21], *socially friendly coalition logic* (SFCL) [19], and *group protecting coalition logic* (GPCL) [19].

Presenting the semantics of the aforementioned logic is beyond the scope of this paper. However, we would like to point out that all of the logics can be captured by *basic strategy logic* (BSL) [18], which is a variant of SL where each agent has her own associated strategy variable. Differently from CSL, BSL allows for all standard temporal modalities like 'neXt', 'Until' and 'Globally (or Forever)'. At the same time, BSL does not allow for variable sharing and does not explicitly refer to actions or strategies. Moreover, it is conjectured [18] that BSL does not have a recursive axiomatisation, while CSL has a finitary complete axiomatisation (see Section 5).

Translations of all coalition logics introduced in this paragraph into formulas of BSL are presented in [18], where it is also claimed that BSL is strictly more expressive than all the aforementioned logics. The translation does not employ any temporal features of BSL apart from 'neXt', and thus the same translation also works for CSL. Moreover, we can use either strategy sharing or explicit actions to argue that CSL is strictly more expressive than the considered coalition logics.

As an example, we provide an argument for SFCL that extends the language of propositional logic with constructs $\langle\langle C\rangle\rangle(\varphi; \psi_1, ..., \psi_k)$ meaning that 'coalition C can achieve φ while also enabling $\overline{C}$ to achieve any of $\psi_1, ..., \psi_k$ (via a suitable joint action)'.

Formally, the semantics is defined as

$$\begin{aligned}\mathfrak{G}, s \models \langle\langle C\rangle\rangle(\varphi; \psi_1, ..., \psi_k) \text{ iff } & \exists\sigma_C(\forall\sigma_{\overline{C}} : \mathfrak{G}, t \models \varphi \text{ and } \forall\psi_i, \exists\sigma_{\overline{C}} : \\ & \mathfrak{G}_u \models \psi) \text{ with } t, u \in S \text{ s.t.} \\ & \langle s, \sigma_C \cup \sigma_{\overline{C}}, w\rangle \in R \text{ and } w \in \{t, u\}.\end{aligned}$$

Now, let us have another look at CGSs in Figure 2. Recall that these structures are distinguished by the CSL formula $\exists x((x,x))\, \neg p$. We claim that no formula of SFCL can distinguish $\mathfrak{G}_1, s$ from $\mathfrak{G}_2, s$. An informal sketch of the induction-based argument is as follows. For purely propositional formulas it is clear that $\mathfrak{G}_1, w$ and $\mathfrak{G}_2, w$ with $w \in \{s, t\}$ satisfy the same formulas. Now, let us consider socially-friendly coalitional modalities $\langle\langle C\rangle\rangle(\varphi; \psi_1, ..., \psi_k)$ For the case of grand coalition $C = \{1, 2\}$, it is easy to verify that any move in $\mathfrak{G}_1, w$ can be matched by a corresponding move $\mathfrak{G}_2, w$ to satisfy φ. Clearly, these transitions will require different actions by agents, but since we do not have access to action labels in SFCL, we are not able to spot the difference.

For the case of single agents, observe that yet again, every choice of, let's say, agent 1 in one structure can be matched by a choice in the other structure to the same effect. Indeed, whatever agent 1 chooses in $\mathfrak{G}_1, s$, a or b, she can only satisfy some φ that holds in both states s and t (due to the fact that the outcome is determined by what agent 2 chooses as well). Similarly in $\mathfrak{G}_2, s$. Now, goals ψ_i of agent 2 can either be satisfied in state s, state t, or both states. Hence, by the construction of CGSs, in both $\mathfrak{G}_1$ and $\mathfrak{G}_2$ for each choice of agent 1, agent 2 has an action to either stay in the current state or force the transition to another state. That the outcome of the corresponding transitions satisfy ψ_i follows from the induction hypothesis.

Proposition 3.3. ConStR < CSL, SFCL < CSL, GPCL < CSL.

Action logic. A perhaps most relevant to CSL coalition logic in the literature is *action logic* (AL) [11], which is a fragment of *multi-agent PDL with quantificaiton* (mPDLQ) [10]. AL extends the language of propositional logic with so-called *modality markers* $[M]$, which are, essentially, prefixes of size $|Agt| = n$, each element of which can either be a quantifier $Q_i x_i$ with $Q_i \in \{\forall, \exists\}$ or an explicit action [9, 10]. An important feature here is that *there are no repeating variables in modality markers.* Finally, to the best of our knowledge, there is no axiomatisation of AL.

Interpreted on concurrent game structures, modality markers have the following semantics:

$$\begin{aligned}\mathfrak{G}, s \models [M]\varphi \text{ iff } & \text{ for all } x_1, ..., x_m \text{ with } \exists x_i \in M, \exists a_i \in \mathsf{Ac} \text{ s.t.} \\ & \langle s, A, s'\rangle \in R \text{ and } \mathfrak{G}, s' \models \varphi, \\ & \text{for all } A \text{ s.t. } a_1, ..., a_m \sqsubseteq A.\end{aligned}$$

Intuitively, $\mathfrak{G}, s \models [M]\varphi$ holds if and only if there is an assignment of actions to all existentially quantified variables in modality marker M such that no matter which actions are explicit in M and which actions are assigned to the universally quantified variables, the outcome state satisfies φ. Observe that this is in line with the semantics of CL as we basically choose actions for a coalition (existentially quantified variables) and verify ψ in all possible outcomes given this choice.

It is easy to see that formulas $[M]\varphi$ of AL can be trivially translated into formulas of CSL of the form $Q_1 x_1, .., Q_n x_m((t_1, ..., t_n))\, \varphi$, where $t_i := x_i$ if there is a quantifier in position i in the modality marker, and $t_i := a_i$ if there is action a_i in the ith posiiton in the modality marker. Also recall that AL does not allow for sharing strategies (while CSL does), i.e. all $x_1, ..., x_m$ in the modality marker are unique.

To show that CSL is more expressive than AL, we need to prove the following lemma.

Lemma 3.4. AL *is not at least as expressive as* CSL.

Proof. Consider $\exists x((x,x))\, \neg p \in$ CSL, and assume towards a contradiction that there is an equivalent $\varphi \in$ AL. Since we have a countably infinite set of constants C (and hence actions) at our disposal and due to the fact that φ is finite, we can assume that there are actions a and b that do not appear explicitly in φ.

Now, consider two concurrent game structures defined over two agents and two actions in Figure 2. As we have already seen in our argument for Proposition 3.2, $\mathfrak{G}_1, s \models \exists x((x,x))\, \neg p$ and $\mathfrak{G}_2, s \not\models \exists x((x,x))\, \neg p$. What is left to show is that φ cannot distinguish the two structures, i.e. $\mathfrak{G}_1, s \models \varphi$ if and only if $\mathfrak{G}_2, s \models \varphi$.

The proof is by induction on the complexity of φ. As the *Base Case*, by the construction of the structures we have that $\mathfrak{G}_1, w \models p$ if and only if $\mathfrak{G}_2, w \models p$ for $w \in \{s, t\}$ and all $p \in \mathsf{Ap}$.

Induction Hypothesis. $\mathfrak{G}_1, w \models \psi$ if and only if $\mathfrak{G}_2, w \models \psi$ for $w \in \{s, t\}$ and for all strict subformulas ψ of φ.

Boolean cases follow straightforwardly by the induction hypothesis. What is left is the case of modality markers.

Case $\varphi := [M]\psi$. First, recall that we assume that actions a and b do not appear explicitly in φ. It is enough to verify four forms of modality markers corresponding to all possible combinations of quantifiers over actions for two agents. Let $\varphi = [\exists x, \exists y]\psi$. It is easy to see that $\mathfrak{G}_1, w \models [\exists x, \exists y]\psi$ if and only if $\mathfrak{G}_2, w \models [\exists x, \exists y]\psi$ as in both CGSs the grand coalition of agents $\{1, 2\}$ has the full control over which transitions to force. Hence, each move in $\mathfrak{G}_1, w$ to a ψ-state $v \in \{s, t\}$ can be matched by a move in $\mathfrak{G}_2, w$ to the same ψ-state v, where we will have $\mathfrak{G}_1, v \models \psi$ if and only if $\mathfrak{G}_2, v \models \psi$ by the induction hypothesis. The remaining cases for modality markers can be shown similarly. □

Having the translation from AL to CSL on the one hand, and Lemma 3.4 on the other, we can conclude that CSL is strictly more expressive than AL.

Proposition 3.5. $\mathsf{AL} < \mathsf{CSL}$.

The expressivity landscape. In this section we have explored the relationship between CSL and other notable coalition logics from the literature. The overall expressivity landscape of the considered logics is presented in Figure 3.

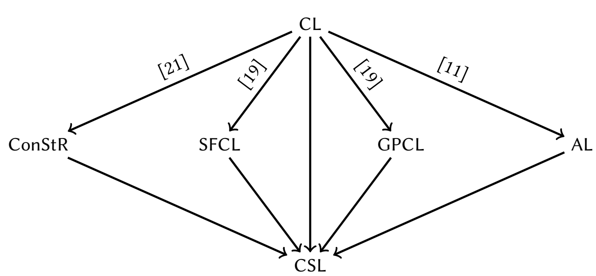

Figure 3: Overview of the expressivity results. An arrow from L_1 to L_2 means $\mathsf{L}_1 < \mathsf{L}_2$. Arrows labelled with a citation represent results from the literature. Arrows without labels are new results.

4 MODEL CHECKING

Now we turn to the complexity of the model checking problem for CSL, and show that it is *PSPACE*-complete.

Definition 4.1. Let $\mathfrak{G} = \langle n, \mathsf{Ac}, \mathcal{D}, S, R, \mathcal{V} \rangle$ be a finite CGS, $s \in S$, and closed formula $\varphi \in \mathsf{CSL}$ constructed over a signature of $\mathfrak{G}$. The *local model checking problem* for CSL consists in computing whether $\mathfrak{G}, s \models \varphi$.

Theorem 4.2. *The model checking problem for* CSL *is PSPACE-complete.*

Proof. To show that the model checking problem for CSL is in *PSPACE*, we provide an alternating recursive Algorithm 1 that takes as an input a finite CGS $\mathfrak{G}$, state of the CGS s, and a closed formula φ. The formula φ is provided in negation normal form (NNF), i.e. in equivalent rewriting, where all negations are pushed inside and appear only in front of propositional variables. To convert φ into the equivalent NNF formula, we can use propositional equivalences, interdefinability of quantifiers, and the validity $\neg (\!(\vec{t})\!)\, \varphi \leftrightarrow (\!(\vec{t})\!)\, \neg\varphi$. The size of a formula in NNF is at most linear in the size of the original formula.

Algorithm 1 An algorithm for model checking CSL

procedure MC($\mathfrak{G}, s, \varphi$)
  case $\varphi = p$
    return $s \in \mathcal{V}(p)$
  case $\varphi = \neg p$
    return not $s \in \mathcal{V}(p)$
  case $\varphi = \psi \vee \chi$
    guess $\theta \in \{\psi, \chi\}$
    return MC($\mathfrak{G}, s, \theta$)
  case $\varphi = \psi \wedge \chi$
    universally choose $\theta \in \{\psi, \chi\}$
    return MC($\mathfrak{G}, s, \theta$)
  case $\varphi = (\!(a_1, ..., a_n)\!)\, \psi$
    guess $t \in S$ such that $\langle s, a_1, ..., a_n, t \rangle \in R$
    return MC($\mathfrak{G}, t, \psi$)
  case $\varphi = \exists x \psi$
    guess $a \in \mathsf{Ac}$
    return MC($\mathfrak{G}, s, \psi[a/x]$)
  case $\varphi = \forall x \psi$
    universally choose $a \in \mathsf{Ac}$
    return MC($\mathfrak{G}, s, \psi[a/x]$)
end procedure

The correctness of the algorithm follows from the definition of the semantics. Its termination follows from the fact that every recursive call is run on a subformula of smaller size. Moreover, each call of the algorithm takes at most polynomial time, and hence it is in *APTIME*. From the fact that *APTIME = PSPACE* [12], we conclude that the model checking problem for CSL is in *PSPACE*.

The hardness is shown by the reduction from the classic satisfiability of quantified Boolean formulas (QBF). For a given QBF $\Psi := Q_1 p_1 ... Q_n p_n \psi(p_1, ..., p_n)$ with $Q_i \in \{\forall, \exists\}$, the problem consists in determining whether Ψ is true. Without loss of generality, we assume that in Ψ each variable is quantified only once.

Given a QBF $\Psi := Q_1 p_1 ... Q_n p_n \psi(p_1, ..., p_n)$, we construct a CGS over one agent $\mathfrak{G} = \langle 1, \mathsf{Ac}, \mathcal{D}, S, R, \mathcal{V} \rangle$, where $\mathsf{Ac} = \{a_1, ..., a_n\}$, $\mathcal{D} = \mathsf{Ac}$, $S = \{s, s_1, ..., s_n\}$, $R = \{\langle s, a_i, s_i \rangle \mid i \in \{1, ..., n\}\} \cup \{\langle s_i, a_j, s_i \rangle \mid i, j \in \{1, ..., n\}\}$, and $\mathcal{V}(p_i) = \{s_i\}$. Intuitively, CGS $\mathfrak{G}$ has a starting state s and a state s_i for each p_i. The agent can reach s_i from s by executing action a_i, and all the transitions from s_i's are self-loops.

The translation from the QBF Ψ into a formula φ of CSL is done recursively as follows:

$$\varphi_0 := \psi((\!(x_1)\!)\, p_1, ..., (\!(x_n)\!)\, p_n)$$
$$\varphi_k := \begin{cases} \forall x_k \varphi_{k-1} & \text{if } Q_k = \forall \\ \exists x_k \varphi_{k-1} & \text{if } Q_k = \exists \end{cases}$$
$$\varphi := \varphi_n$$

To see that

$$Q_1 p_1 ... Q_n p_n \psi(p_1, ..., p_n) \text{ is satisfiable iff } \mathfrak{G}, s \models \varphi$$

it is enough to notice that setting the truth-value of propositional variable p_i to 1 is modelled by reachability via action a_i of the state s_i, where p_i holds. Quantifiers are modelled directly as quantifiers over the agent's actions.

As an example, consider a QBF $\forall p_2 \exists p_1 \exists p_3 (p_1 \rightarrow p_2) \wedge p_3$, which is clearly satisfiable with $p_1 = 0$ and $p_3 = 1$. The formula is translated into the formula of CSL: $\forall x_2 \exists x_1 \exists x_3 ((\!(x_1)\!)\, p_1 \rightarrow (\!(x_2)\!)\, p_2) \wedge (\!(x_3)\!)\, p_3$. The corresponding CGS is presented in Figure 4, and it is easy to verify that $\mathfrak{G}, s \models \forall x_2 \exists x_1 \exists x_3 ((\!(x_1)\!)\, p_1 \rightarrow (\!(x_2)\!)\, p_2) \wedge (\!(x_3)\!)\, p_3$. □

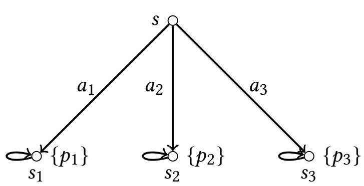

Figure 4: CGS $\mathfrak{G}$ for a single agent. Labels for self-loops are omitted for readability.

Remark 2. *The complexity of the model checking problem for a related strategy logic with simple goals (*SL[SG]*) is known to be P-complete [8]. This is due to the fact that in* SL[SG] *the quantification prefix and the operators for assigning strategies to agents always go together. Hence, for example, the* CSL *formula over two agents $\theta := \forall x \exists y \forall z ((\!(z, x)\!)\, \varphi \wedge (\!(z, y)\!)\, \psi \wedge (\!(x, z)\!)\, \chi)$ cannot be expressed in the language of* SL[SG]. *The higher complexity of* CSL *stems from the fact that quantifiers and strategy assignments are less rigid than in* SL[SG]*, and thus* CSL *is closer to the full* SL *in this regard.*

5 PROOF THEORY

5.1 Context

Perhaps the best-known results in the field are complete axiomatisations of CL [20, 32] and ATL [24] (see [23, 35] for more constructive approaches to the construction of the canonical model). Other completeness results include axiomatisations for logics based on CL and ATL, like already mentioned SFCL [19], ATLES [36], as well as *epistemic* CL [2], *resource-bounded* CL[5], *resource-bounded* ATL [31], and ATL *with finitely bounded semantics* [22], to name a few.

In the context of strategy logics, we have quite an opposite picture. Since the inception of SL [29], its completeness has been an open problem. Indeed, the satisfiability problem for the original SL is Σ^1_1-hard [28], which implies that SL is not recursively axiomatisable. This does not rule out, though, the existentence of an *infinitary* axiomatisation. A fragment of SL, called *one-goal* SL (SL[1G]), has a decidable satisfiability problem so there is a hope of having a proof system for it. However, SL[1G] subsumes ATL*, and providing a complete axiomatisation of the latter is yet another long-standing open problem.

The lack of axiomatisations of *any* (fragment of) SL can be traced back to the two main features of the logic: quantification over strategies and arbitrary quantification prefixes.

Indeed, arbitrary alternation of quantifiers in SL is quite different from the fixed quantification prefix of CL and ATL that allow only prefixes $\exists\forall$ and $\forall\exists$. Secondly, quantification over strategies[2] in SL is essentially a second-order quantification over functions. We believe that these two features combined are the root cause of the fact that no complete axiomatisations of (fragments of) SL have been proposed so far.

In CSL we focus on arbitrary quantification prefixes. To solve this sub-problem, we consider only neXt-time modalities $(\!(t_1, ..., t_n)\!)\, \varphi$ and focus on the immediate outcomes of agents' choices. Other temporal modalities, like Until and Eventually would require a more complicated construction as their truth values depend on the whole computation paths rather than just the next step (see [24] on how to deal with them in the context of ATL). Such a design choice allows us, in particular, to consider quantification over actions rather than strategies. Hence, quantification in CSL is a first-order quantification, instead of the second-order quantification of SL. The relation between various fragments of SL and the corresponding induced fragments of FOL is explored in [30].

In the completeness proof in the following section, we take as inspiration the completeness proof for *first-order modal logic* (FOML) with constant domains [17]. Our construction is quite different, though, as in FOML variables appear in n-ary predicates, and in CSL variables are placeholders for transition labels.

5.2 A Sound and Complete Axiomatisation of CSL

Definition 5.1. The *axiom system* for CSL consist of the following axiom schemata and rules, where $\vec{t} = t_1, \ldots, t_n$ for $n \geq 1$, and each t_i is either a variable or a constant, and likewise for t.

PC	Every propositional tautology
K	$((\!(\vec{t})\!)\, \varphi \wedge (\!(\vec{t})\!)\, \psi) \leftrightarrow (\!(\vec{t})\!)\, (\varphi \wedge \psi)$
N	$\neg (\!(\vec{t})\!)\, \varphi \leftrightarrow (\!(\vec{t})\!)\, \neg\varphi$
E	$\forall x \varphi \rightarrow \varphi[t/x]$
B	$\forall x (\!(\vec{t})\!)\, \varphi \rightarrow (\!(\vec{t})\!)\, \forall x \varphi$, where each t_i is different from x
MP	From $\varphi, \varphi \rightarrow \psi$, infer ψ
Nec	From φ, infer $(\!(\vec{t})\!)\, \varphi$
Gen	From $\varphi \rightarrow \psi[t/x]$, infer $\varphi \rightarrow \forall x \psi$, if t does not appear in φ

An axiomatic derivation π is a finite sequence of formulae $\varphi_1, \ldots, \varphi_m$ where for each $i \leq m$: either φ_i is an instance of one of the axiom schemata of CSL, or it is obtained by some preceding formulae in the sequence using rules MP, Nec, or Gen. We write $\vdash \varphi$ and we say that φ is *CSL derivable* (or simply derivable) iff there is a derivation π whose last element is φ. Given a set of formulae X, we write $X \vdash \varphi$ iff there is a finite subset Y of X such that $\vdash \bigwedge Y \rightarrow \varphi$.

[2] A *(memoryless) strategy* for an agent $i \in n$ is a function $\sigma_i : S \rightarrow \text{Ac}$.

We will freely use the following proposition in the rest of the paper. Its proof is relatively standard, and we omit it for brevity.

Proposition 5.2. *The following formulae are* CSL *derivable, where $\vec{t} = t_1, \dots, t_n$ and each t_i is either a constant or a variable:*

(1) $((\vec{t}))\,(\varphi \to \psi) \to (((\vec{t}))\,\varphi) \to (((\vec{t}))\,\psi)$;
(2) $\forall x(\varphi \to \psi) \to (\varphi \to \forall x\psi)$ *with* $x \notin \mathsf{FV}(\varphi)$;
(3) $\exists z(\varphi \to \forall y\psi)$ *with* $z \notin \mathsf{FV}(\forall y\psi)$.

Moreover, if $\varphi \to \psi$ is derivable, so is $((\vec{t}))\,\varphi \to ((\vec{t}))\,\varphi$.

Lemma 5.3. *Each axiom schema of* CSL *is valid and each rule of* CSL *preserves validity.*

Proof. For the sake of simplicity, we only consider closed instances of the axiom schemata. Validity of other axiom schemata and the soundness of the rules of inference can be shown similarly.

(N). Suppose that $\mathfrak{G}, s \models \neg((\vec{a}))\,\varphi$. By the definition of the semantics, this means that $\mathfrak{G}, s \not\models ((\vec{a}))\,\varphi$, i.e. for each $t \in S$ if $\langle s, \vec{a}, t\rangle \in R$, then we have that $\mathfrak{G}, t \not\models \varphi$. From the seriality and functionality of R, we can conclude that there is exactly one such t, and thus $\mathfrak{G}, s \models ((\vec{a}))\,\neg\varphi$. For the converse direction, suppose that $\mathfrak{G}, s \models ((\vec{a}))\,\neg\varphi$. This means that there is a t such that $\langle s, \vec{a}, t\rangle \in R$ and $\mathfrak{G}, t \not\models \varphi$. By functionality of R there is no other t related to s by means of $\vec{a}$, and thus we can conclude that $\mathfrak{G}, s \models \neg((\vec{a}))\,\varphi$.

(B). Assume that $\mathfrak{G}, s \models \forall x((\vec{t}))\,\varphi$, where x is different from every t_i. Since the formula is closed, this is just $\mathfrak{G}, s \models \forall x((\vec{a}))\,\varphi$ for some $\vec{a} \in \mathcal{D}$. By the truth definition, this is equivalent to $\mathfrak{G}, s \models ((\vec{a}))\,(\varphi[b/x])$ for every $b \in \mathsf{Ac}$, which means $\mathfrak{G}, s \models ((\vec{a}))\,\forall x\varphi$. □

Our completeness proof is based on the canonical model construction, where states are maximal consistent sets with the $\forall$-property.

Definition 5.4. Let X be a set of CSL sentences. We say that:

- X is *consistent* iff $X \nvdash \bot$
- X is maximally consistent iff it is consistent and there is no other consistent set of sentences that strictly includes X;
- X has the $\forall$-property iff for every formula φ and variable x there is a constant a such that $\varphi[a/x] \to \forall x\varphi \in X$, where $\varphi[a/x]$ is closed.

Let $\alpha = \langle n, C, \mathsf{Ap}\rangle$ be a signature, we denote by $\alpha^\star$ the signature $\langle n, C \cup C^\star, \mathsf{Ap}\rangle$ where $C^\star$ is countably infinite, and $C \cap C^\star = \emptyset$.

Next lemma shows that each consistent set of sentences over a given signature α can be extended to a consistent set of sentences over $\alpha^\star$ having the $\forall$-property. Its proof follows the standard technique in FOML [14], and we report it here for the sake of completeness.

Lemma 5.5. *If X is a consistent set of sentences over a given signature α, then there is a consistent set of sentences Y over $\alpha^\star$ such that $X \subseteq Y$, and Y has the $\forall$-property.*

Proof. Let E be an enumeration of sentences of the form $\forall x\varphi$ over $\alpha^\star$. We define a sequence of sets of sentences $Y_0, Y_1, \dots$ with $Y_0 = X$ and $Y_{n+1} = Y_n \cup \{\varphi[a/x] \to \forall x\varphi\}$ where $\forall x\varphi$ is the $n+1$-th sentence in E, and a is the first constant in the enumeration occurring neither in Y_n nor in φ. Since Y_0 is over α, Y_n is obtained by the addition of n sentences over $\alpha^\star$, and $\alpha^\star$ includes a countably infinite set of new constants, we can always find such an a.

Now we show that Y_{n+1} constructed in the described way is consistent. For this, assume towards a contradiction that Y_n is consistent and Y_{n+1} is not. This means that there is a finite set of sentences $U \subseteq Y_n$ such that $U \cup \{\varphi[a/x] \to \forall x\varphi\} \vdash \bot$. By the rules of CSL we thus obtain that (i) $U \vdash \varphi[a/x]$ and (ii) $U \vdash \neg\forall x\varphi$. Since a does not appear in Y_n, we can use the Gen rule of inference and conclude that $U \vdash \forall x\varphi$. In conjunction with (ii) this amounts to the fact that Y_n is not consistent, and hence we arrive at a contradiction.

Define Y as $\bigcup_{n\in\mathbb{N}} Y_n$. It is now easy to see that Y is consistent and has the $\forall$-property. □

The proof of the following lemma (Lindenbaum Lemma) is standard, and we omit it for brevity.

Lemma 5.6. *If X is a consistent set of sentences over a given signature, then there is a maximal consistent set of sentences Y over the same signature such that $X \subseteq Y$.*

The next two lemmas will be instrumental in the proof of the Truth Lemma, and showing that the canonical model we are to define in this proof is indeed a CGS.

Lemma 5.7. *Let X be a consistent set of sentences and suppose that $((\vec{a}))\,\varphi \in X$, then the set $Y = \{\varphi\} \cup \{\psi \mid ((\vec{a}))\,\psi \in X\}$ is also consistent.*

Proof. Suppose, towards a contradiction, that set Y is not consistent. This implies that $(\psi_1 \wedge \dots \wedge \psi_m) \to \neg\varphi$. Using Nec, we can then derive $((\vec{a}))\,\psi_1 \wedge \dots \wedge ((\vec{a}))\,\psi_m \to ((\vec{a}))\ \neg\varphi$. Since $((\vec{a}))\,\psi_i \in X$, we conclude by MP that $X \vdash ((\vec{a}))\,\neg\varphi$. Then, by N and MP we can further derive $X \vdash \neg((\vec{a}))\,\varphi$, which contradicts $((\vec{a}))\,\varphi \in X$. □

Lemma 5.8. *Let X be a maximal consistent set of sentences with the $\forall$-property over a given signature, and suppose that $((\vec{a}))\,\varphi \in X$. Then there is a consistent set of sentences Y over the same signature that has the $\forall$-property and such that $\{\varphi\} \cup \{\psi \mid ((\vec{a}))\,\psi \in X\} \subseteq Y$*

Proof. Let $Z = \{\psi \mid ((\vec{a}))\,\psi \in X\}$, E be an enumeration of sentences of the form $\forall x\xi$, and C be an enumeration of constants in the given signature. We define a sequence of sentences $\theta_0, \theta_1, \dots$ where $\theta_0 = \varphi$ and, given θ_n, $\theta_{n+1} = \theta_n \wedge \xi[a/x] \to \forall x\xi$, where $\forall x\xi$ the $n+1$-th sentence in E, and a be the first constant in C such that $Z \cup \{\theta_n \wedge (\xi[a/x] \to \forall x\xi)\}$ is consistent. We claim that we can always find an a having this property. The proof of this claim is by induction on n.

If $n = 0$, then the result follows by Lemma 5.7. Suppose now that the result holds for every $0 \le k \le n$, and it does not hold for $n+1$. This means that for some finite subset $\{\psi_1, \dots, \psi_m\}$ of Z we have that $(\psi_1 \wedge \dots \wedge \psi_m) \to (\theta_n \to \neg(\xi[a/x] \to \forall x\xi))$ is derivable. From this, by the rules of CSL, we obtain that $(((\vec{a}))\,\psi_1 \wedge \dots \wedge ((\vec{a}))\,\psi_m) \to ((\vec{a}))\,(\theta_n \to \neg(\xi[a/x] \to \forall x\xi))$ is derivable. Since we have that if $\psi_i \in Z$ then $((\vec{a}))\,\psi_i \in X$, we can conclude that (i) $((\vec{a}))\,(\theta_n \to \neg(\xi[a/x] \to \forall x\xi)) \in X$ for *every* constant a .

Now, let z be a variable that appears neither in θ_n nor in ξ. Consider the sentence $\forall z((\vec{a}))\,(\theta_n \to \neg(\xi[z/x] \to \forall x\xi))$. From the fact that X has the $\forall$-property and because of (i), it follows that $\forall z((\vec{a}))\,(\theta_n \to \neg(\xi[z/x] \to \forall x\xi)) \in X$. By the rule B, the latter implies $((\vec{a}))\,\forall z(\theta_n \to \neg(\xi[z/x] \to \forall x\xi)) \in X$, and thus, by (2) of Proposition 5.2, it holds that (ii) $((\vec{a}))\,(\theta_n \to \forall z\neg(\xi[z/x] \to \forall x\xi)) \in X$. Since $\exists z(\xi[z/x] \to \forall x\xi)$ is CSL derivable, application of the rule Nec yields $((\vec{a}))\, \exists z(\xi[z/x] \to \forall x\xi) \in X$. Finally, from $((\vec{a}))\, \exists z(\xi[z/x] \to \forall x\xi) \in X$, (ii), using Proposition 5.2 and contrapositive reasoning, we conclude that $((\vec{a}))\, \neg\theta_n \in X$. The latter implies that $\neg\theta_n \in Z$ by the construction of Z. But $Z \cup \{\theta_n\}$ was consistent by the induction hypothesis, and we have thus arrived at a contradiction.

Finally, we define Y as the union of Z with every sentence in $\theta_0, \theta_1, \ldots$. It is easy to see that Y satisfies all the properties stated in the lemma. □

Definition 5.9 (Canonical Model). Given a signature $\alpha = \langle n, C, \mathsf{Ap}\rangle$, the canonical model over α is the tuple $\mathfrak{G}^C = \langle n, \mathsf{Ac}^C, \mathcal{D}^C, S^C, R^C, \mathcal{V}^C\rangle$, where:

- $\mathsf{Ac}^C = C \cup C^\star$;
- $\mathcal{D}^C = Ac^{C^n}$;
- $S^C = \{X \mid X$ is a maximally consistent set of sentences over $\alpha^\star$ with the $\forall$-property$\}$;
- for every $\vec{a} \in \mathcal{D}^c$, $\langle X, \vec{a}, Y\rangle \in R^C$ iff for every sentence φ we have that $\varphi \in Y$ implies $((\vec{a}))\, \varphi \in X$;
- $X \in \mathcal{V}^C(p)$ iff $p \in X$ for all $p \in \mathsf{Ap}$.

Proposition 5.10. *For all states X and Y of the canonical model, and for every decision $\vec{a} \in \mathcal{D}^C$, it holds that $\langle X, \vec{a}, Y\rangle \in R^C$ if and only if for every sentence φ, $((\vec{a}))\, \varphi \in X$ implies $\varphi \in Y$*

Proof. For the left-to-right direction, suppose $\langle X, \vec{a}, Y\rangle \in R^C$ and $\varphi \notin Y$. We need to show that $((\vec{a}))\, \varphi \notin X$. Since Y is maximally consistent, we have that $\neg\varphi \in Y$. From the fact that $\langle X\, \vec{a}, Y\rangle \in R^C$ it follows, by the definition of the canonical model, that $((\vec{a}))\, \neg\varphi \in X$. Since X is maximally consistent, we have that $\neg((\vec{a}))\, \neg\varphi \notin X$, which implies, by axiom N, that $((\vec{a}))\, \varphi \notin X$.

For the right-to-left direction, we again reason by contraposition. Suppose that $\langle X, \vec{a}, Y\rangle \notin R^C$, and thus, by the construction of the canonical model, there is a formula $\varphi \in Y$ such that $((\vec{a}))\, \varphi \notin X$. Since X is maximally consistent, we have that $\neg((\vec{a}))\, \varphi \in X$. By the axiom N we get $((\vec{a}))\, \neg\varphi \in X$. Thus $((\vec{a}))\, \neg\varphi \in X$ and $\neg\varphi \notin Y$ as required for the proof. □

Now we are ready to show that the canonical model is a CGS.

Proposition 5.11. *The canonical model $\mathfrak{G}^C$ is a CGS.*

Proof. We have to prove that the relation R^C of the canonical model is serial and functional.

For seriality, we have that given any state $X \in S^C$, X contains the formula $((\vec{a}))\, \top$ for any $a \in D^C$ due to $\top$ being a tautology and the application of Gen. Thus, by Lemma 5.8, for any $\vec{a} \in D^C$ there is a $Y \in S^C$ such that $\{\top\} \cup \{\psi \mid ((\vec{a}))\, \psi \in X\} \subseteq Y$, and by Proposition 5.10 we have that $\langle X, \vec{a}, Y\rangle \in R^C$

For functionality, suppose that $\langle X, \vec{a}, Y\rangle \in R^C$, $\langle X, \vec{a}, Z\rangle \in R^C$ and $Z \neq Y$. Thus there is a φ, such that $\varphi \in Y$ and $\neg\varphi \in Z$. By the definition of R^C, this implies $((\vec{a}))\, \varphi \in X$ and $((\vec{a}))\, \neg\varphi \in X$. By N, the latter is equivalent to $\neg((\vec{a}))\, \varphi \in X$, which contradicts the consistency of X. □

Lemma 5.12 (Truth Lemma). *For any state X of the canonical model $\mathfrak{G}^C$ and for any formula φ, we have that $\mathfrak{G}^C, X \models \varphi$ iff $\varphi \in X$.*

Proof. The proof is by induction on φ. The *base case*, in which $\varphi \in \mathsf{Ap}$, immediately follows from the definition of $\mathcal{V}^C$. The cases in which φ is a Boolean formula follow from the induction hypothesis and the properties of maximally consistent sets.

Case $\varphi = ((\vec{a}))\, \psi$. Suppose that $\mathfrak{G}^C, X \models \varphi$. By the definition of semantics, this means that there is a Y such that $\langle X, \vec{a}, Y\rangle \in R^C$ and $\mathfrak{G}^C, Y \models \psi$. The latter is equivalent to $\psi \in Y$ by the induction hypothesis, and by the definition of R^C we conclude that $\varphi \in X$.

Suppose that $\varphi \in X$. By Lemma 5.8, there is a maximal consistent set of sentences Y over $\alpha^\star$ that has the $\forall$-property and such that $\{\psi\} \cup \{\theta \mid ((\vec{a}))\, \theta \in X\} \subseteq Y$. By the definition of R^C this means $\langle X, \vec{a}, Y\rangle \in R^C$, which, in conjunction with the fact that $\psi \in Y$, is equivalent to $\mathfrak{G}^C, X \models \varphi$ by the induction hypothesis.

Case $\varphi = \forall x\psi$. If $\mathfrak{G}^C, X \models \varphi$, then, by the induction hypothesis, it holds that (i) $\psi[a/x] \in X$ for every $a \in \mathsf{Ac}^C$. Now, assume towards a contradiction that $\varphi \notin X$. Since X is maximally consistent, we have that $\neg\forall x\psi \in X$. Moreover, since X has the $\forall$-property, there is a constant a such that $\psi[a/x] \to \forall x\psi \in X$. Then by (i) it follows that $\forall x\psi \in X$, which contradicts $\forall x\psi \notin X$.

Suppose that $\varphi \in X$, which implies, by axiom E and MP, that $\psi[a/x] \in X$ for every $a \in \mathsf{Ac}^C$. By the induction hypothesis, we conclude that $\mathfrak{G}^C, X \models \varphi[a/x]$ for every $a \in \mathsf{Ac}^C$, which is equivalent to $\mathfrak{G}^C, X \models \forall x\varphi$ by the definition of semantics. □

We can now prove that the axiom system for CSL is sound and complete.

Theorem 5.13. *For every set of formulae X and every formula φ, we have that $X \vdash \varphi$ iff $X \models \varphi$.*

Proof. The left-to-right direction is proved by induction on the length of the derivation of $X \vdash \varphi$ using Lemma 5.3. For the other direction, assume that $X \nvdash \varphi$. This means that $X \cup \{\neg\varphi\}$ is consistent, and, by Lemmas 5.5 and 5.6, there is a maximally consistent set Z with the $\forall$-property, such that $X \cup \{\neg\varphi\} \subseteq Z$. As $\neg\varphi \in Z$, it holds that $\varphi \notin Z$, and by the truth lemma we have that $\mathfrak{G}^C, Z \models X$ and $\mathfrak{G}^C, Z \not\models \varphi$. □

6 DISCUSSION

We introduced *coalition strategy logic* (CSL), which combines features of both CL and SL, and, additionally, allows for explicit action labels in the syntax. For CSL, we showed that it is strictly more expressive than other known coalition logics, and that its model checking problem is *PSPACE*-complete. Moreover, we provided a sound and complete axiomatisation of CSL. To the best of our knowledge, it is *the first axiomatisation of any strategy logic* in the literature.

There is a plethora of open research questions that one can tackle building on our work. Perhaps the most immediate one is exploring the complexity of the satisfiability problem of CSL. In the future, we would also like to properly characterise the expressivity of CSL by providing an appropriate notion of bisimulation (akin to those in [26, Chapter 3] and [7]), results on bisimulaiton invariance, and translations into FOL.

Another avenue of exciting further research is finding an axiomatisation of an extension of CSL with LTL modalities. In such a way, we would be able to advance towards axiomatisations of such rich fragments of SL as *one-goal* SL [28] and *flat conjunctive-goal* SL [1].

REFERENCES

[1] Erman Acar, Massimo Benerecetti, and Fabio Mogavero. 2019. Satisfiability in Strategy Logic Can Be Easier than Model Checking. In *Proceedings of the 33rd AAAI*. 2638–2645. https://doi.org/10.1609/AAAI.V33I01.33012638

[2] Thomas Ågotnes and Natasha Alechina. 2019. Coalition logic with individual, distributed and common knowledge. *Journal of Logic and Computation* 29, 7 (2019), 1041–1069. https://doi.org/10.1093/LOGCOM/EXV085

[3] Thomas Ågotnes, Valentin Goranko, and Wojciech Jamroga. 2007. Alternating-time temporal logics with irrevocable strategies. In *Proceedings of the 11th TARK*, Dov Samet (Ed.). 15–24. https://doi.org/10.1145/1324249.1324256

[4] Michael E. Akintunde, Elena Botoeva, Panagiotis Kouvaros, and Alessio Lomuscio. 2020. Verifying Strategic Abilities of Neural-symbolic Multi-agent Systems. In *Proceedings of the 17th KR*, Diego Calvanese, Esra Erdem, and Michael Thielscher (Eds.). 22–32. https://doi.org/10.24963/KR.2020/3

[5] Natasha Alechina, Brian Logan, Nguyen Hoang Nga, and Abdur Rakib. 2011. Logic for coalitions with bounded resources. *Journal of Logic and Computation* 21, 6 (2011), 907–937. https://doi.org/10.1093/LOGCOM/EXQ032

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

[7] Francesco Belardinelli, Catalin Dima, and Aniello Murano. 2018. Bisimulations for Logics of Strategies: A Study in Expressiveness and Verification. In *Proceedings of the 16th KR*, Michael Thielscher, Francesca Toni, and Frank Wolter (Eds.). 425–434.

[8] Francesco Belardinelli, Wojciech Jamroga, Damian Kurpiewski, Vadim Malvone, and Aniello Murano. 2019. Strategy Logic with Simple Goals: Tractable Reasoning about Strategies. In *Proceedings of the 28th IJCAI*, Sarit Kraus (Ed.). 88–94. https://doi.org/10.24963/IJCAI.2019/13

[9] Stefano Borgo. 2005. Modal Operators with Adaptable Semantics for Multi-agent Systems. In *Proceedings of the 5th AI*IA (LNCS, Vol. 3673)*, Stefania Bandini and Sara Manzoni (Eds.). Springer, 186–197. https://doi.org/10.1007/11558590_20

[10] Stefano Borgo. 2005. Quantificational Modal Logic with Sequential Kripke Semantics. *Journal of Applied Non-Classical Logics* 15, 2 (2005), 137–188. https://doi.org/10.3166/JANCL.15.137-188

[11] Stefano Borgo. 2007. Coalitions in Action Logic. In *Proceedings of 20th IJCAI*, Manuela M. Veloso (Ed.). 1822–1827.

[12] Ashok K. Chandra, Dexter Kozen, and Larry J. Stockmeyer. 1981. Alternation. *Journal of the ACM* 28, 1 (1981), 114–133. https://doi.org/10.1145/322234.322243

[13] Krishnendu Chatterjee, Thomas A Henzinger, and Nir Piterman. 2010. Strategy logic. *Information and Computation* 208, 6 (2010), 677–693.

[14] M. J. Cresswell and G. E. Hughes. 1996. *A New Introduction to Modal Logic*. Routledge, New York.

[15] Lavindra de Silva, Paolo Felli, Jack C. Chaplin, Brian Logan, David Sanderson, and Svetan M. Ratchev. 2017. Synthesising Industry-Standard Manufacturing Process Controllers. In *Proceedings of the 16th AAMAS*, Kate Larson, Michael Winikoff, Sanmay Das, and Edmund H. Durfee (Eds.). ACM, 1811–1813.

[16] Jonathan Ezekiel, Alessio Lomuscio, Levente Molnar, and Sandor M. Veres. 2011. Verifying Fault Tolerance and Self-Diagnosability of an Autonomous Underwater Vehicle. In *Proceedings of the 22nd IJCAI*, Toby Walsh (Ed.). 1659–1664. https://doi.org/10.5591/978-1-57735-516-8/IJCAI11-279

[17] James W. Garson. 1984. *Quantification in Modal Logic*. Springer, 249–307. https://doi.org/10.1007/978-94-009-6259-0_5

[18] Valentin Goranko. 2023. Logics for Strategic Reasoning of Socially Interacting Rational Agents: An Overview and Perspectives. *Logics* 1, 1 (2023), 4–35. https://doi.org/10.3390/logics1010003

[19] Valentin Goranko and Sebastian Enqvist. 2018. Socially Friendly and Group Protecting Coalition Logics. In *Proceedings of the 17th AAMAS*, Elisabeth André, Sven Koenig, Mehdi Dastani, and Gita Sukthankar (Eds.). IFAAMAS, 372–380.

[20] Valentin Goranko, Wojciech Jamroga, and Paolo Turrini. 2013. Strategic games and truly playable effectivity functions. *Autonomous Agents and Multi-Agent Systems* 26, 2 (2013), 288–314. https://doi.org/10.1007/S10458-012-9192-Y

[21] Valentin Goranko and Fengkui Ju. 2022. A Logic for Conditional Local Strategic Reasoning. *Journal of Logic, Language and Information* 31, 2 (2022), 167–188. https://doi.org/10.1007/S10849-022-09357-Y

[22] Valentin Goranko, Antti Kuusisto, and Raine Rönnholm. 2019. Alternating-time temporal logic ATL with finitely bounded semantics. *Theoretical Computer Science* 797 (2019), 129–155. https://doi.org/10.1016/J.TCS.2019.05.029

[23] Valentin Goranko and Dmitry Shkatov. 2009. Tableau-based decision procedures for logics of strategic ability in multiagent systems. *ACM Transactions on Computational Logic* 11, 1 (2009), 3:1–3:51. https://doi.org/10.1145/1614431.1614434

[24] Valentin Goranko and Govert van Drimmelen. 2006. Complete axiomatization and decidability of Alternating-time temporal logic. *Theoretical Computer Science* 353, 1-3 (2006), 93–117. https://doi.org/10.1016/J.TCS.2005.07.043

[25] Wojciech Jamroga, Michal Knapik, and Damian Kurpiewski. 2018. Model Checking the SELENE E-Voting Protocol in Multi-agent Logics. In *Proceedings of the 3rd E-Vote-ID (LNCS, Vol. 11143)*, Robert Krimmer, Melanie Volkamer, Véronique Cortier, Rajeev Goré, Manik Hapsara, Uwe Serdült, and David Duenas-Cid (Eds.). Springer, 100–116. https://doi.org/10.1007/978-3-030-00419-4_7

[26] Fabio Mogavero. 2010. *Logics in Computer Science*. Ph.D. Dissertation. University of Naples Federico II, Italy.

[27] Fabio Mogavero, Aniello Murano, Giuseppe Perelli, and Moshe Y. Vardi. 2014. Reasoning About Strategies: On the Model-Checking Problem. *ACM Transactions on Computational Logic* 15, 4 (2014), 34:1–34:47. https://doi.org/10.1145/2631917

[28] Fabio Mogavero, Aniello Murano, Giuseppe Perelli, and Moshe Y. Vardi. 2017. Reasoning about Strategies: on the Satisfiability Problem. *Logical Methods in Computer Science* 13, 1 (2017). https://doi.org/10.23638/LMCS-13(1:9)2017

[29] Fabio Mogavero, Aniello Murano, and Moshe Y. Vardi. 2010. Reasoning About Strategies. In *Proc. of the Conf. on Foundations of Software Technology and Theoretical Computer Science, FSTTCS*, Kamal Lodaya and Meena Mahajan (Eds.). https://doi.org/10.4230/LIPICS.FSTTCS.2010.133

[30] Fabio Mogavero and Giuseppe Perelli. 2015. Binding Forms in First-Order Logic. In *Proceedings of the 24th CSL (LIPIcs, Vol. 41)*, Stephan Kreutzer (Ed.). Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 648–665. https://doi.org/10.4230/LIPICS.CSL.2015.648

[31] Hoang Nga Nguyen, Natasha Alechina, Brian Logan, and Abdur Rakib. 2018. Alternating-time temporal logic with resource bounds. *Journal of Logic and Computation* 28, 4 (2018), 631–663. https://doi.org/10.1093/LOGCOM/EXV034

[32] Marc Pauly. 2002. A Modal Logic for Coalitional Power in Games. *Journal of Logic and Computation* 12, 1 (2002), 149–166. https://doi.org/10.1093/LOGCOM/12.1.149

[33] Arunesh Sinha, Fei Fang, Bo An, Christopher Kiekintveld, and Milind Tambe. 2018. Stackelberg Security Games: Looking Beyond a Decade of Success. In *Proceedings of the 27th IJCAI*, Jérôme Lang (Ed.). 5494–5501. https://doi.org/10.24963/IJCAI.2018/775

[34] Wiebe van der Hoek, Wojciech Jamroga, and Michael J. Wooldridge. 2005. A logic for strategic reasoning. In *Proceedings of the 4th AAMAS*, Frank Dignum, Virginia Dignum, Sven Koenig, Sarit Kraus, Munindar P. Singh, and Michael J. Wooldridge (Eds.). 157–164. https://doi.org/10.1145/1082473.1082497

[35] Dirk Walther, Carsten Lutz, Frank Wolter, and Michael J. Wooldridge. 2006. ATL Satisfiability is Indeed EXPTIME-complete. *Journal of Logic and Computation* 16, 6 (2006), 765–787. https://doi.org/10.1093/LOGCOM/EXL009

[36] Dirk Walther, Wiebe van der Hoek, and Michael J. Wooldridge. 2007. Alternating-time temporal logic with explicit strategies. In *Proceedings of the 11th TARK*, Dov Samet (Ed.). 269–278. https://doi.org/10.1145/1324249.1324285