Alternating-Time Temporal Logic with Dependent Strategies

Jessica L. Newman
University of Southampton
Southampton, UK
jln1g19@soton.ac.uk

Enrico Gerding
University of Southampton
Southampton, UK
eg@soton.ac.uk

Enrico Marchioni
University of Southampton
Southampton, UK
e.marchioni@soton.ac.uk

Baharak Rastegari
University of Southampton
Southampton, UK
b.rastegari@soton.ac.uk

ABSTRACT

Alternating-Time Temporal Logic (ATL) can express statements about the strategic abilities of agents in games where agents move concurrently. However, many game-theoretic scenarios (such as Stackelberg competitions) require agents to make moves sequentially, with the actions of a given agent depending on the actions of the agents who move prior to them. To capture this, we introduce ATL with Dependent Strategies (ATLDS), which extends ATL with the ability to specify an order in which agents select actions. We characterise the sets of outcomes that are possible for a coalition to enforce when playing a normal-form game sequentially, and provide a representation theorem that allows us to convert between games and sets of enforceable outcomes generated from those games. We use this to give a sound and complete axiomatisation of ATLDS. We also show expressive equivalence with the $SL^{-}[SG]$ fragment of Strategy Logic, and provide complexity bounds for variants of the model-checking problem.

KEYWORDS

Logics for Multi-Agent Systems; ATL; Effectivity

ACM Reference Format:
Jessica L. Newman, Enrico Gerding, Enrico Marchioni, and Baharak Rastegari. 2026. Alternating-Time Temporal Logic with Dependent Strategies. In *Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), Paphos, Cyprus, May 25 – 29, 2026*, IFAAMAS, 9 pages. https://doi.org/10.65109/MYRV8798

1 INTRODUCTION

ATL [2] is a logical formalism that is commonly used to abstractly reason about the abilities of agents in multi-agent systems. ATL is defined over a transition system with an associated set of agents, called a Concurrent Game Model (CGM). Each state of the system can be viewed as a game where agents concurrently select moves, inducing a transition to another state. We can verify that certain coalitions of agents in a system have the ability to guarantee sequences of states are reached with certain temporal properties.

However, ATL can only express the ability of a coalition in a *concurrent* game, when the coalition has no knowledge of the moves chosen by other agents. We may want to model a scenario in which an agent has some sort of temporal dependency or privileged information on another agent, and therefore is able to select their move *in response* to the choice of that agent. Certain game-theoretic concepts require such a notion of dependence to express; for example, Stackelberg games, in which a leader moves first and a group of followers select moves depending on the leader's choice of move. This motivates the development of an extension of ATL that maintains most of its nice properties (e.g. only positional strategies required, simple model checking), but admits formulae that can specify dependencies between agent strategies.

Therefore, we introduce Alternating-Time Temporal Logic with Dependent Strategies (ATLDS), which extends the ATL modality $\langle\!\langle C \rangle\!\rangle$ to $\langle\!\langle C \rangle\!\rangle_P$, providing the ability to associate a permutation P of agents to a run. In a run under a permutation, at each state agents choose their next actions sequentially in the order of P, with the knowledge of the actions chosen by agents who appear prior to them in P. As an example, the ATLDS modality $\langle\!\langle \{\alpha,\gamma\} \rangle\!\rangle_{(\alpha,\beta,\gamma)} \mathrm{G}p$ expresses that if agents make moves sequentially at each state in the order α, β, γ then no matter the actions taken by β, agents α and γ are able to guarantee a proposition p always holds. We could interpret this as interacting processes being interleaved at discrete points in time in a consistent way, as opposed to interacting processes running concurrently at discrete points in time as in ATL.

We can also take an epistemic point of view, where an order (α, β, γ) means that agent β knows the strategy of agent α, and agent γ knows the strategy of both agents α and β. A formula $\neg\langle\!\langle \{\alpha,\gamma\} \rangle\!\rangle_{(\alpha,\gamma,\beta)}\varphi \wedge \langle\!\langle \{\alpha,\gamma\} \rangle\!\rangle_{(\alpha,\beta,\gamma)}\varphi$ would express, for example, that the coalition $\{\alpha,\gamma\}$ cannot achieve φ without any knowledge of opponent moves, but would be able to achieve φ if agent γ was able to learn β's strategy. For the kinds of properties expressible in ATLDS, the view of sequentially playing games interleaved in a certain order and the epistemic view are interchangeable.

ATL can be considered as a fragment of Strategy Logic (SL), introduced in [7] and further developed in [21–23]. SL allows for first-order quantification over strategies. Since for the full SL the satisfiability problem is undecidable, and model checking is intractable [25], there is interest in fragments of SL more expressive than ATL but more tractable than full SL [11]. Both ATL and ATLDS embed into SL, so ATLDS can be seen as a fragment of SL more expressive than ATL but more tractable than full SL. In fact, we show ATLDS is

Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), C. Amato, L. Dennis, V. Mascardi, J. Thangarajah (eds.), May 25 – 29, 2026, Paphos, Cyprus. © 2026 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). https://doi.org/10.65109/MYRV8798

expressively equivalent to the $SL^{-}[SG]$ fragment of SL introduced in [4], so our results also apply to this fragment.

Our main contribution is a sound and complete axiomatisation of ATLDS, given in Section 6, based on analogous axiomatisations for ATL in [16] and CTL in [10]. One difficulty with the canonical model construction to show completeness is that in CGMs, transitions are not added directly, but induced from the collective actions of agents. Therefore, given a consistent set of formulae, it is not obvious how to define the transition function to obtain a model in which all the formulae are realised. To overcome this, we provide a construction that can generate a corresponding neighbourhood model from a CGM and vice versa. In these models, instead of having to assign successors to action profiles, we can directly add the sets of states we require each coalition to be able to force. This uses effectivity functions, a way of describing strategic ability of coalitions which was originally used for social choice theory in [1] but further developed for logics of strategic ability in [13, 14, 29]. We axiomatically characterise the sets of outcomes a coalition can enforce when playing a given game under different orderings, and show that this also works in the other direction; given a collection of sets that satisfies these axioms, we can find a game in which the coalitions have exactly the strategic ability defined by these sets, in a representation theorem similar to Pauly's in [28]. The results on effectivity are provided in Section 5. We provide complexity results for model checking in Section 7, in particular when a model can be encoded in an efficient way. Some proofs are omitted for brevity.

2 RELATED WORK

There has been some work on dependencies in strategic reasoning in the context of Strategy Logic in [9], finding that some fragments allow existential strategies to depend on the past actions of all opponents, even those quantified later in the formula. In [5], Strategy Logic is extended with a team semantics-style system, to allow for independence-friendly style quantifiers. In [24], it is seen that in certain fragments of Strategy Logic, only a limited form of dependence is required, in that strategies only need to respond to the actual moves from other agents during a run, rather than other potential moves encoded in their strategy.

In [11] strategic logics that focus on conditional behaviour of agents are surveyed, each extending ATL expressiveness and embedding into Strategy Logic. One such logic, ConStR [15], introduces operators such as $[A]_\beta(\gamma_A; \langle B\rangle\gamma_B)$, stating that 'For any move of A that guarantees γ_A, there is some move of B that guarantees γ_B'. A similar modality, although without the conditioning on a goal γ_A, appears in [26]. This makes the assumption that all members of a coalition have the same information, whereas in ATLDS agents in a coalition are working towards the same goal but might have asymmetric information about opponent strategies. Related approaches at capturing conditional and coordinated behaviour include the Socially Friendly Coalition Logic [12] and the logic of Local Coalitional Goal Assignments [8]. Most of these logics look at one-step goals, apart from LCGA which has been extended to LTL goals.

ATL with Strategy Contexts [18] contains a modality $\langle\cdot C\cdot\rangle\varphi$ which fixes a strategy for C in the context of φ, without removing any strategies in the current context. This allows us to chain these modalities together to create similar order-dependent properties as in ATLDS. However, ATL with Strategy Contexts is significantly more expressive (and intractable) overall.

3 PRELIMINARIES

A **Concurrent Game Structure** (CGS) is a tuple $\mathcal{S} = (Ag, S, A, act, \delta)$ where $Ag = \{1, \ldots, n\}$ is a finite set of agents; S is a set of states; A is a nonempty set of actions; $act : Ag \times S \to \mathcal{P}(A)$ is a function that assigns to each state and agent a nonempty set of actions that agent can make at that state; and δ is a transition function that takes a state s and an action profile $\alpha = (\alpha_1, \ldots, \alpha_n)$ where each $\alpha_i \in act(i, s)$ and returns a successor state $\delta(s, \alpha)$. A **Concurrent Game Model** (CGM) is a CGS with an additional map $V : S \to \mathcal{P}(Prop)$ that assigns propositional variables to states. Given a coalition $C \subseteq Ag$ we will use the notation $\overline{C}$ to refer to the set $Ag \setminus C$.

Each state in a CGS can be thought of as a normal-form game, with outcomes as successor states. A **normal-form game** is a tuple $G = (N, (A_i)_{i\in N}, O, \pi)$ where $N = \{1, \ldots, n\}$ is a nonempty set of agents; Each A_i is a nonempty set of actions; O is a nonempty set of outcomes; and $\delta : A_1 \times \ldots \times A_n \to O$ assigns an outcome to each action profile $\alpha = (\alpha_1, \ldots, \alpha_n)$ where each $\alpha_i \in A_i$. Given a CGS $\mathcal{S} = (Ag, S, A, act, \delta)$ with state $s \in S$, to obtain the corresponding normal-form game $G_s = (N, (A_i)_{i\in N}, O, \delta)$ we set $N := Ag$, each $A_i := act(i, s)$, $O := S$, and $\delta(\alpha) := \delta(s, \alpha)$.

In ATL, a **positional strategy** σ_i for agent i is a map assigning to each state $s \in S$ an action $\sigma_i(s) \in act(i, s)$. A joint strategy for a coalition $C \subseteq Ag$, denoted σ_C, a choice of σ_i for each $i \in C$, and the set of all such strategies is Σ_C. A trace $\lambda \in S^\omega$ is an infinite sequence of states, with $\lambda[i]$ denoting the ith state. From a state s, a joint strategy for Ag generates a trace λ where $\lambda[0] = s$ and $\lambda[i+1] = \delta(\lambda[i], \sigma_1(\lambda[i]), \ldots, \sigma_n(\lambda[i]))$. We denote this by $trace(\sigma_1, \ldots, \sigma_n)$. For a coalition C, a joint strategy σ_C induces a set $traces(\sigma_C) = \{trace(\sigma_C, \sigma_{\overline{C}}) \mid \sigma_{\overline{C}} \in \Sigma_{\overline{C}}\}$. **Memoryful strategies** extend positional ones with finite sequences of past states as input.

In this work we use an extended definition of strategy: we have a **strategy for an agent** i **under a permutation** P of Ag. By permutation, we mean an ordered tuple of all elements in a set, e.g. a permutation of $\{a, b, c\}$ could be (b, c, a). We will denote the element at position k of P (beginning at 1) as $P[k]$. Given an element i in a permutation P, we will refer to the set of elements that appear before i in P as $pre(P, i) = \{P[j] | j < k \text{ where } i = P[k]\}$. In a strategy under a permutation, agent i is able to respond to the actions chosen by agents that appear before them in the permutation. Therefore:

Definition 1. *A positional strategy for i under permutation P, denoted σ_i^P, takes a state s, and for each agent $j \in pre(P, i)$ an action $a_j \in act(j, s)$ and returns an action $\sigma_i^P(s, (a_j)_{j\in pre(P,i)}) \in act(i, s)$.*

To calculate the action chosen by a strategy σ_i^P we need the results of the strategies σ_j^P for all j that appear before i in P. We will denote the action chosen by σ_i^P at some state s as α_i^s. This value can be calculated recursively as follows: $\alpha_{P[1]}^s = \sigma_{P[1]}^P(s)$; $\alpha_{P[i]}^s = \sigma_{P[i]}^P(s, \alpha_{P[1]}^s, \ldots, \alpha_{P[i-1]}^s)$.

Each joint strategy of Ag under P, σ_{Ag}^P, induces a trace λ from a state s given by $\lambda[0] = s$; $\lambda[i+1] = \delta(\lambda[i], \alpha_{P[1]}^{\lambda[i]}, \ldots, \alpha_{P[n]}^{\lambda[i]})$. So, in the order of the permutation we compute the action chosen by each agent, feeding in the actions chosen by previous agents into the strategy of the next agent in the permutation. We denote by $\mathfrak{S}(Ag)$ the set of all permutations of Ag.

Some permutations are guaranteed to be 'better' for a coalition than others; if an agent can respond to a larger set of agents, they have more strategic power. Given a permutation P, coalition $C \subseteq Ag$, and agent $i \in C$, we denote by $bef(P, C, i) := pre(P, i) \cap \overline{C}$ the set of agents in $\overline{C}$ that appear before agent i in the permutation P. For each coalition C, we fix an ordering on permutations such that $P \leq_C P'$ iff $bef(P, C, i) \subseteq bef(P', C, i)$ for all $i \in C$.

4 ALTERNATING-TIME TEMPORAL LOGIC WITH DEPENDENT STRATEGIES

In ATLDS, each coalition modality is paired with a permutation of the agent set Ag, determining the order of moves. For example, the permutation (b, a, c) means b moves first, then a with knowledge of b's choice, and finally c with knowledge of both. We assume a given countable set of propositions $Prop$ and a finite set of agents Ag. The syntax of ATLDS is given by:

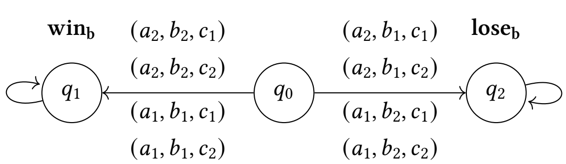
$$\varphi ::= p | \neg\varphi | (\varphi \wedge \varphi) | (\varphi \vee \varphi) | \langle\langle C \rangle\rangle_P \mathsf{X}\varphi | \langle\langle C \rangle\rangle_P \varphi \mathsf{U} \varphi$$

For $C \subseteq Ag, P \in \mathfrak{S}(Ag), p \in Prop$.

The intended interpretation of the formula $\langle\langle C \rangle\rangle_P \mathsf{X}\varphi$ is that coalition C, when acting sequentially in the order P, can force the next state to satisfy φ. Satisfaction of a formula at a state in a CGM is defined inductively as follows using strategies under a permutation (Definition 1):

- $\mathcal{M}, s \vDash p$ iff $p \in V(s)$
- $\mathcal{M}, s \vDash \varphi \wedge \psi$ iff $\mathcal{M}, s \vDash \varphi$ and $\mathcal{M}, s \vDash \psi$
- $\mathcal{M}, s \vDash \varphi \vee \psi$ iff $\mathcal{M}, s \vDash \varphi$ or $\mathcal{M}, s \vDash \psi$
- $\mathcal{M}, s \vDash \neg\varphi$ iff $\mathcal{M}, s \nvDash \varphi$
- $\mathcal{M}, s \vDash \langle\langle C \rangle\rangle_P \mathsf{X}\varphi$ iff $\exists \sigma^P_C \forall \sigma^P_{\overline{C}}.trace(s, \sigma^P_C, \sigma^P_{\overline{C}})[1] \vDash \varphi$
- $\mathcal{M}, s \vDash \langle\langle C \rangle\rangle_P \psi \mathsf{U} \varphi$ iff $\exists \sigma^P_C \forall \sigma^P_{\overline{C}}.(\mathcal{M}, trace(s, \sigma^P_C, \sigma^P_{\overline{C}})[j] \vDash \varphi$ and $\mathcal{M}, trace(s, \sigma^P_C, \sigma^P_{\overline{C}})[i] \vDash \psi)$ for some $j \geq 0$ and all $i < j$.

Example 2. Consider the following CGM:



It can be seen that under ATL semantics, $q_0 \vDash \neg\langle\langle\{b\}\rangle\rangle X win_b$. This is because if agent b chooses b_1, then a could have chosen a_2 and c chosen c_1, forcing the next state to be q_2 where win_b does not hold. Similarly if b selects b_2, a can select a_1 and c can choose c_1 so b has no strategy which guarantees that win_b holds in the next step. However, under ATLDS semantics, $q_0 \vDash \langle\langle\{b\}\rangle\rangle_{(a,b,c)} X win_b$. Under the permutation (a, b, c), b selects their move in response to the choice of a. Therefore if b chooses responses $a_1 \mapsto b_1$ and $a_2 \mapsto b_2$, no matter the choice of c they can always guarantee that the next state is q_1 where the win_b proposition holds.

The following are satisfied in all states of all CGMs for all coalitions $C \subseteq Ag$ and permutations $P \in \mathfrak{S}(Ag)$:

$$\langle\langle C \rangle\rangle_P \mathsf{X}\varphi \leftrightarrow \neg\langle\langle \overline{C} \rangle\rangle_P \mathsf{X}\neg\varphi$$
$$\langle\langle C \rangle\rangle_P \mathsf{X}\varphi \rightarrow \langle\langle C \rangle\rangle_{P'} \mathsf{X}\varphi \text{ where } P' \geq_C P$$
$$\langle\langle C \rangle\rangle_P \psi \mathsf{U}\varphi \leftrightarrow \varphi \vee (\psi \wedge \langle\langle C \rangle\rangle_P \mathsf{X} \langle\langle C \rangle\rangle_P \psi \mathsf{U} \varphi)$$

The last formula shows us that the unfolding of fixpoint operators in ATL extends to all permutations in ATLDS. The first formula shows us that we can infer facts about the strategic ability of a coalition $\overline{C}$ from facts about the strategic ability of a coalition C: if we know C can force φ at the next state under a particular ordering P, then the opposing coalition $\overline{C}$ cannot enforce the contradictory $\neg\varphi$ under the same ordering *and vice versa*. In other words, under each P the two-player zero-sum game where C wins by achieving φ and $\overline{C}$ wins by achieving $\neg\varphi$ is *determined*; exactly one of C or $\overline{C}$ achieves their goal. The second formula shows us that coalitional ability is monotonic with respect to the order agents move; if all agents in P' move at least as late as they do in P then they can achieve everything they could in P.

Syntactically, ATL usually requires the operator $\langle\langle C \rangle\rangle \psi \mathsf{R} \varphi$. The dual of the $\langle\langle C \rangle\rangle_P$ modality in ATLDS aligns with $\langle\langle \overline{C} \rangle\rangle_P$, so we only need $\langle\langle C \rangle\rangle_P \mathsf{X}\varphi$ and $\langle\langle C \rangle\rangle_P \psi \mathsf{U} \varphi$ as primitive operators in the syntax, as $\langle\langle C \rangle\rangle_P \psi \mathsf{R} \varphi$ can be defined as $\neg\langle\langle \overline{C} \rangle\rangle_P \neg\psi \mathsf{U} \neg\varphi$.

Some permutations result in identical strategic ability with respect to a coalition C. The ability of an agent in C actually only depends on the subset of agents in $\overline{C}$ that she is able to respond to:

Proposition 3. *If $P \sim_C P'$ (i.e. $P \leq_C P'$ and $P' \leq_C P$) then $\mathcal{M}, s \vDash \langle\langle C \rangle\rangle_P \varphi$ iff $\mathcal{M}, s \vDash \langle\langle C \rangle\rangle_{P'} \varphi$ for $\varphi \in \{\mathsf{X}\psi, \vartheta \mathsf{U} \psi\}$*

Proof. Suppose C has some strategy σ^P_C such that $\mathcal{M}, s \vDash \langle\langle C \rangle\rangle_P \varphi$. Take the agent $i \in C$ which appears earliest in P and denote by T the set $bef(P, C, i)$. In a run, i's choice of action α_i at each state is completely determined by the joint action of their opponent α_T. Therefore alternative choices of action α'_i for α_T will never be seen in a run, so the response of agents appearing later in P to these alternative choices will not affect the set of traces generated by their joint strategy. This can be carried forward inductively to each agent $j \in C$. Since $P \sim_C P'$ means $bef(P, C, j) = bef(P', C, j)$, it is simple to obtain a strategy under P' that selects the same actions and generates the same set of traces, so satisfies the same formula. This works symmetrically in the other direction. □

We can represent a permutation P in the context of a coalition C by clustering together adjacent members of C and adjacent members of $\overline{C}$. For instance, from the perspective of the coalition $\{b, c\}$, $P = (a, b, c, d, e)$ and $P' = (a, c, b, e, d)$ are identical (in the sense that $\langle\langle\{b, c\}\rangle\rangle_P \varphi \leftrightarrow \langle\langle\{b, c\}\rangle\rangle_{P'} \varphi)$ and when we cluster together adjacent members of $\{b, c\}$ and $\{a, d, e\}$ they both result in $(\{a\}, \{b, c\}, \{d, e\})$. For a permutation P and coalition C, we will denote this $cl(P, C)$. The **alternation number** of a modality $\langle\langle C \rangle\rangle_P$ is $|cl(P, C)|$. The **alternation number of a formula**, $alt(\varphi)$, is the highest alternation number of any modality within φ. For example, $alt(\langle\langle\{b, c\}\rangle\rangle_{(a,b,c,d,e)} \mathsf{X} p) = 3$.

As in ATL, it is sufficient in ATLDS to describe satisfaction of formulae in terms of positional strategies. We have defined dependent strategies as a map that responds to actions made at a state.

However, suppose strategies were memoryful and the semantics of a formulae $\langle\langle C \rangle\rangle_P \psi$ were defined as follows, where $Q_i = \exists$ if $i \in C$ or $Q_i = \forall$ if $i \notin C$ and quantifies over strategies of $P[i]$:

$$\mathcal{M}, s \vDash \langle\langle C \rangle\rangle_P \psi \text{ iff } Q_1\sigma_1 Q_2\sigma_2 \ldots Q_n\sigma_n.\mathcal{M}, trace(s, \sigma_1, \sigma_2, \ldots, \sigma_n) \vDash \psi$$

We will denote this as $\vDash_\sigma$ and the semantics defined earlier as $\vDash_P$. In $\vDash_\sigma$, an agent further along in the permutation has access to the entire strategy of agents appearing before them. Despite this, the two semantics are the same:

Proposition 4. *$M, s \vDash_\sigma \varphi$ iff $M, s \vDash_P \varphi$*

Proof. (sketch) It can be shown that the $\vDash_\sigma$ semantics for ATLDS embed into a fragment of strategy logic. It is known that a subsuming fragment, SL[1G], is *behavioural* [24], which means that it is sufficient to have strategies that only depend on the actual history of moves from other agents. From behavioural strategies witnessing the satisfaction of a formula we can obtain memoryful winning strategies in a reachability game. Reachability games can be played optimally with positional strategies [20], so we can extract a positional strategy from this. A positional strategy in the reachability game will correspond to a positional strategy under a permutation σ_i^P in a CGM.

It is simple to show inductively that σ_i^P are sufficient for boolean and $\langle\langle C \rangle\rangle_P X\varphi$ formulae, so we will focus on the case of $\langle\langle C \rangle\rangle_P \psi U \varphi$ and construct a reachability game that corresponds to satisfaction of $\langle\langle C \rangle\rangle_P \psi U \varphi$ formulae. For every state q such that $q \vDash_\sigma \varphi$ we have a corresponding state q' in the reachability game. For every remaining state q such that $q \vDash_\sigma \psi$, we expand the state q into a tree. We create a state $q'_{P[1]}$ as the root of the tree. For each $i \in C$ and joint action $\alpha_{Pre(P,i)}$ we create a state $q_{P[i],\alpha_{Pre(P,i)}}$. We add every edge of the form $(q'_{P[i-1],\alpha_{Pre(P,i-1)}}, q'_{P[i],\alpha_{Pre(P,i)}})$. Each joint action of α_{Ag} at $q_{P[1]}$ gives a unique path through this tree, allowing us to replace the leaf node for α_{Ag} with the state corresponding to $\delta(q, \alpha_{Ag})$. We set all states labelled with $P[i] \in C$ as controlled by $P1$ and all others as controlled by $P2$. For any state q in the CGM where $q \vDash \varphi$, we put all states in the associated tree as winning states for $P1$. Then, for any state s, $s \vDash_\sigma \langle\langle C \rangle\rangle_P \psi U \varphi$ implies $P1$ has a winning strategy for the reachability game from state $s'_{P[1]}$ (or s' if $s \vDash_\sigma \varphi$). Since any state $x'_{P[i],\alpha_j}$ in the reachability game is reached via a unique joint action from $x'_{P[1]}$, a positional strategy for $P1$ over the states $x_{P[i],\alpha_{pre(P,i)}}$ where $i \in C$ gives a function from each state $x_{P[i]}$ and joint move $\alpha_{pre(P,i)}$ into an action $\alpha_i \in act(i, s)$. This defines a strategy under a permutation σ_C^P. □

This means we can freely switch between the interpretation of agents moving sequentially at each state, and the interpretation of agents having full knowledge of the strategies of agents prior to them in the given order.

The SL$^-$[SG] fragment of Strategy Logic, introduced in [4], restricts the language of Strategy Logic (which allows for full first order quantification of strategies) to blocks of strategy quantifiers followed by an operator that binds a different strategy to each agent. We can embed ATLDS into SL$^-$[SG] by replacing the ATLDS modalities for SL$^-$[SG] quantifier blocks:

$$f(\langle\langle C \rangle\rangle_P \psi) = Q_1 x_1 \ldots Q_n x_n (x_1, a_1) \ldots (x_n, a_n) f(\psi)$$

Where $a_i = P[i]$ and $Q_i = \exists$ if $P[i] \in C$, or $Q_i = \forall$ otherwise. For all other operators ϕ in the language of ATLDS, $f(\phi(\varphi_1, \ldots \varphi_k)) = \phi(f(\varphi_1), \ldots, f(\varphi_k))$. A similar mapping g can be constructed to go from SL$^-$[SG] to ATLDS formulae, by extracting P from the order agents have been bound in the quantifier block and C from which agents have been bound to existential quantifiers. The semantics of SL$^-$[SG] can be seen to correspond with the $\vDash_\sigma$ semantics of ATLDS. Therefore, as a corollary of Proposition 4 this mapping also preserves satisfaction of formulae for the $\vDash_P$ semantics of ATLDS.

Corollary 5. *For ATLDS formulae φ, $\mathcal{M}, q \vDash_P \varphi$ iff $\mathcal{M}, q \vDash_{SL^-[SG]} f(\varphi)$. For SL$^-$[SG] formulae ψ, $\mathcal{M}, q \vDash_{SL^-[SG]} \psi$ iff $\mathcal{M}, q \vDash_P g(\psi)$.*

5 EFFECTIVITY

In this section we will define various notions of 'effectivity', which abstractly represent the strategic ability of coalitions in a game. We give a construction by which we can move between CGMs and effectivity models.

5.1 Effectivity Functions

Suppose we were not interested in exactly what strategies agents have, but only what they can achieve with these strategies. We can record this with an **effectivity function**:

Definition 6. *Given some set of outcomes S and agents Ag, an effectivity function is a mapping $E : \mathcal{P}(Ag) \to \mathcal{P}(\mathcal{P}(S))$ that takes a subset of agents and returns a set of sets of outcomes.*

The common interpretation of an effectivity function is that it gives the sets of outcomes a coalition is able to 'force', in the sense that the coalition is able to guarantee an outcome falls within a particular set, but not necessarily which element of the set will be the outcome. Normal-form games have a notion of 'α-effectivity' (see [14]). A set is α-effective for a coalition if the coalition has a joint action such that, no matter the actions of agents outside the coalition, the outcome of the game will fall within the set. Given a game G, $X \in E_G^\alpha(C)$ iff $\exists \alpha_C \forall \alpha_{\overline{C}}.\delta(\alpha_C, \alpha_{\overline{C}}) \in X$.

We will parameterise effectivity functions with permutations of agents, to represent the kind of strategic ability we have in ATLDS. We will denote by $E(C, P)$ the set of sets of outcomes that coalition C can force when selecting moves in the order of permutation P.

Definition 7. *Given a set of outcomes Q and agents Ag, an ordered effectivity function is a map $E : \mathcal{P}(Ag) \times \mathfrak{S}(Ag) \to \mathcal{P}(\mathcal{P}(Q))$.*

This definition is completely abstract and we could instantiate this with arbitrary sets. However, similar to α-effectivity, we can associate with each normal-form game a canonical ordered effectivity function:

Definition 8. *Given a normal-form game G, the π-effectivity function of G is an ordered effectivity function E_G^π such that for each coalition C and permutation of agents P,*

$$X \in E_G^\pi(C, P) \text{ iff } Q_1\alpha_1 Q_2\alpha_2 \ldots Q_n\alpha_n.\delta(\alpha_1, \alpha_2, \ldots, \alpha_n) \in X$$

Where $Q_i = \exists$ if $i \in C$ or $Q_i = \forall$ if $i \notin C$.

Each set in $E_G^\pi(C, P)$ corresponds to a strategy of C - under α-effectivity, a strategy for C was a choice of joint action, but here an agent $i \in C$ can react to the moves selected by all agents that move before her in P, as in the strategies σ_C^P defined earlier.

For a permutation $(C, \overline{C})$, the set $E^{\pi}_G(C, (C, \overline{C}))$ is equivalent to the set $E^{\alpha}_G(C)$. We will also take the effectivity for a single permutation P across all coalitions in a game G. We will denote this E^P_G, and define it as $E^P_G(C) := E^{\pi}_G(C, P)$. We will use the following properties of effectivity functions:

(1) (outcome monotonic) $X \in E(C)$ implies $Y \in E(C)$ for all $Y \supseteq X$
(2) (superadditivity) $X \in E(C)$ and $Y \in E(S)$ implies $X \cap Y \in E(C \cup S)$ for disjoint C, S
(3) (maximality) $X \notin E(C) \rightarrow \overline{X} \in E(\overline{C})$
(4) (N-maximality) $X \notin E(\emptyset) \rightarrow \overline{X} \in E(Ag)$
(5) (liveness) $\emptyset \notin E(C)$
(6) (safety) $E(C) \neq \emptyset$
(7) (order monotonic) $X \in E(C, P)$ implies $X \in E(C, P')$ for $P \leq_C P'$
(8) (regularity) $X \in E(C) \rightarrow \overline{X} \notin E(\overline{C})$
(9) (crown condition) $X \in E(Ag)$ implies there is some $x \in X$ such that $\{x\} \in E(Ag)$

Condition 4 is entailed by 3, and 8 can be derived from 2 and 5. We can apply any properties where P is omitted to ordered effectivity functions by adding a P parameter to each occurence of $E(C)$.

Definition 9. *An effectivity function is **playable** if it meets axioms 1,2 and 4-6. An effectivity function is **maximally playable** if it is playable and meets axiom 3. An ordered effectivity function is **monotonically ordered and maximally playable** if it is maximally playable and meets axiom 7 for all permutations P. An effectivity function is **truly playable** if it is playable and meets axiom 9.*

The crown condition is required to ensure that games with an infinite number of moves correspond properly to normal-form games, as detailed in [14]. In a normal-form game, each joint strategy of the grand coalition corresponds to a single outcome, so any non-singleton set must be generated by one of these through outcome monotonicity. The following is a characterisation from [14, 28] of α-effectivity in terms of the above conditions:

Theorem 10. *(Pauly Representation Theorem [14]) An effectivity function E is truly playable iff $E = E^{\alpha}_G$ for some normal form game G.*

We will build a characterisation of effectivity functions of games E^P_G and E^{π}_G in terms of these axioms.

Proposition 11. *Any E^P_G is maximally truly playable.*

Proof. Every agent has at least one action, guaranteeing safety and liveness. Outcome monotonicity comes directly from the definition of E^{π}_G. Suppose $X \notin E^P_G(C)$. This means $\neg Q_1\alpha_1 \dots Q_n\alpha_n.$ $\delta(\alpha_1, \dots, \alpha_n) \in X$ where $Q_i = \exists$ if $i \in C$ or $Q_i = \forall$ if $i \notin C$. This is equivalent to $\overline{Q}_1\alpha_1 \dots \overline{Q}_n\alpha_n.\delta(\alpha_1, \dots, \alpha_n) \in \overline{X}$ where $\overline{Q}_i = \exists$ if $Q_i = \forall$, or $\overline{Q}_i = \forall$ otherwise. This is the condition for $\overline{X} \in E^P_G(\overline{C})$, satisfying maximality. In a normal-form game, any joint action from Ag gives only one outcome, meeting the crown condition.

For any set $X \in E^P_G(C)$ there is some $Y \in E^P_G(C)$ where $Y \subseteq X$ and Y is generated by a strategy σ^P_C of C in the game G, in that $Y = \{\delta(\sigma^P_C, \sigma^P_{\overline{C}}) | \sigma^P_{\overline{C}} \in \Sigma^P_{\overline{C}}\}$. So, for disjoint coalitions C_1 and C_2, there is an Y_1 in $E^P_G(C_1)$ generated by $\sigma^P_{C_1}$ and Y_2 in $E^P_G(C_2)$ generated by $\sigma^P_{C_2}$. The strategy $(\sigma^P_{C_1}, \sigma^P_{C_2})$ generates a set $X \in E^P_G(C_1 \cup C_2)$ such that $X \subseteq Y_1 \cap Y_2$, so by outcome monotonicity $Y_1 \cap Y_2 \in E^P_G(C_1 \cup C_2)$. This is sufficient to show superadditivity, as all intersections are supersets of these intersections of sets generated by strategies. □

The following can be seen from the definition of E^{π}_G:

Proposition 12. *Given two permutations P, P', if $P \leq_C P'$ then $X \in E^P_G(C)$ implies $X \in E^{P'}_G(C)$*

In certain games, strategic power is the same no matter the order in which agents select actions:[1]

Proposition 13. *If E^{α}_G is maximal (i.e. $X \notin E^{\alpha}_G(C) \rightarrow \overline{X} \in E^{\alpha}_G(\overline{C})$), then for any ordering P of agents in G, $E^P_G = E^{\alpha}_G$.*

Proof. For right to left inclusion, suppose $X \in E^{\alpha}_G(C)$ for some $C \subseteq Ag$. This means $\exists\sigma_C\forall\sigma_{\overline{C}}.\delta(\sigma_C, \sigma_{\overline{C}}) \in X$. By Definition 8, $X \in E^{(C,\overline{C})}_G(C)$. Since $(C, \overline{C}) \leq_C P$ for all orderings P, by Prop. 12 it must be the case that $X \in E^P_G(C)$ for all P. For the left to right inclusion, suppose $X \notin E^{\alpha}_G(C)$. Since E^{α}_G is maximal, then $\overline{X} \in E^{\alpha}_G(\overline{C})$, so by the above $\overline{X} \in E^P_G(\overline{C})$. Since E^P_G is regular, then $X \notin E^{\alpha}_G(C)$. □

Maximally truly playable effectivity functions are exactly those generated by taking E^P_G:

Proposition 14. *E is maximally truly playable iff $E = E^P_G$ for some strategic game G and ordering of agents P*

Proof. ($\Longleftarrow$) From Prop. 11. ($\Longrightarrow$) If E is maximally truly playable then E is truly playable. From Theorem 10, we know when E is truly playable there is some strategic game G such that $E = E^{\alpha}_G$. Since E^{α}_G is maximal, we know by Prop. 13 that $E^{\alpha}_G = E^P_G$ for any ordering P. So, $E = E^P_G$. □

We can use this to obtain a representation theorem for π-effectivity:

Theorem 15. *E is monotonically ordered and maximally truly playable iff $E = E^{\pi}_G$ for some G.*

Proof. (sketch) ($\Longleftarrow$) Follows from 11 and 12.
($\Longrightarrow$) For each P we can use Proposition 14 to generate a game G_P such that $E(C, P) = E^P_{G_P}(C)$. We must construct a mechanism around this that ensures the game G_P is played when agents are moving in the order P. The idea will be to construct an extensive-form game G' in 2 stages: first, agents each select actions which result in some G_P being played. Then, agents play G_P.

For the first stage of G', the set of moves available to each agent will be $\mathbb{N}$. That is, each agent i is able to choose a natural number n_i. We will select G_P based on the size of the numbers chosen. So, if the natural numbers chosen are $n_{i_1} < n_{i_2} < \dots < n_{i_{|Ag|}}$, then we will select the game G_P where $P = (i_1, i_2, \dots, i_{|Ag|})$. When two agents choose the same number, we can break ties arbitrarily. In the second stage, agents play whichever G_P was selected.

We will show for this G', $X \in E^{\pi}_{G'}(C, P)$ iff $X \in E(C, P)$. Suppose $X \in E(C, P)$. We will look at the strategic ability of C moving in the order P in the constructed game G'. Let us take an agent $i \in C$, and call the set of agents that appear before them in the permutation $S = pre(P, i)$. Agent i's strategy for step 1 will be, given a joint move $a_S \in \mathbb{N}^S$ from S, to respond with $1 + max(a_S)$. Under this strategy we are guaranteed that for all $i \in C$, $a_i \geq a_j$ for all $a_j \in bef(P, C, i)$, so whichever game $G_{P'}$ is selected, it will be

[1] One example is the normal-form encoding of extensive-form games with perfect information.

such that $P \leq_C P'$. Since $X \in E(C,P) \rightarrow X \in E(C,P')$, we can force X in whichever G'_P is selected. So, $X \in E^{\pi}_{G'}(C,P)$. Now suppose $X \notin E(C,P)$. By maximality, $\overline{X} \in E(\overline{C},P)$ so by the above $\overline{X} \in E^{\pi}_{G'}(\overline{C},P)$. By regularity of π-effectivity, we know $X \notin E^{\pi}_{G'}(C,P)$.
□

5.2 Effectivity Models

From the previous subsection, we know that certain effectivity functions correspond to normal-form games. We can use this define a neighbourhood model (in the sense of [27]) with equivalent semantics to CGMs where each state is an effectivity function that points to sets of other states. The following definitions are based on state effectivity models as presented in [13].

A **Standard Ordered Effectivity Model (SOEM)** is a tuple $\mathcal{M} = (Ag, Q, E_{(-)}, V)$ where: Ag is a set of agents; Q is a set of states; $E_{(-)}$ is a state effectivity function that, for each state $q \in Q$, returns a monotonically ordered and maximally playable effectivity function over Q (Definition 9); and $V : Q \rightarrow \mathcal{P}(Prop)$ assigns each state a set of propositional variables.

The effectivity function E_q at a state q determines satisfaction of $\langle\langle C\rangle\rangle_P \mathsf{X}\varphi$ formulae, by checking if $[\![\varphi]\!] \in E_q(C,P)$. To check satisfaction of a particular $\langle\langle C\rangle\rangle_P \psi \mathsf{U} \varphi$ formulae, we can define a corresponding operator U over sets of states such that $U(X) = [\![\varphi]\!] \cup ([\![\psi]\!] \cap \{z \in Q | X \in E_z(C,P)\})$. We can iterate this operator with $U^0 = [\![\varphi]\!]$, $U^i = U(U^{i-1})$ to get the states where $\langle\langle C\rangle\rangle_P \psi \mathsf{U}\varphi$ is realised in i steps. From outcome monotonicity, we get that this is a monotone operator since $X \subseteq Y$ implies that $\{z \in Q | X \in E_z(C,P)\} \subseteq \{z \in Q | Y \in E_z(C,P)\}$.

Satisfaction of an ATLDS formula in an ordered effectivity model can now be defined:

- $\mathcal{M}, s \vDash p$ iff $p \in V(s)$
- $\mathcal{M}, s \vDash \neg\varphi$ iff $s \notin [\![\varphi]\!]$
- $\mathcal{M}, s \vDash \varphi \wedge \psi$ iff $s \in [\![\varphi]\!] \cap [\![\psi]\!]$
- $\mathcal{M}, s \vDash \langle\langle C\rangle\rangle_P \mathsf{X}\varphi$ iff $[\![\varphi]\!] \in E_s(C,P)$
- $\mathcal{M}, s \vDash \varphi \vee \psi$ iff $s \in [\![\varphi]\!] \cup [\![\psi]\!]$
- $\mathcal{M}, s \vDash \langle\langle C\rangle\rangle_P \psi\mathsf{U}\varphi$ iff $s \in \bigcup_{i=0}^{\infty} U^i$

From a CGM $\mathcal{M}$, we can generate an ordered effectivity model $\mathcal{M}'$ by taking the same sets of agents and states, the same valuation function, and setting $E_q = E^{\pi}_{G_q}$, where G_q is the normal-form game at state q in the CGM. Then the following can be shown inductively:

Proposition 16. *For a CGM $\mathcal{M}$, ordered effectivity model $\mathcal{M}'$ generated from $\mathcal{M}$, and formula φ, $\mathcal{M}, s \vDash \varphi$ iff $\mathcal{M}', s \vDash \varphi$*

6 AXIOMATISATION

We can axiomatise ATLDS with an axiomatic system for propositional logic plus the following:

($\bot$). $\neg\langle\langle C\rangle\rangle_P \mathsf{X}\bot$ (Mon). $\langle\langle C\rangle\rangle_P \mathsf{X}\varphi \rightarrow \langle\langle C\rangle\rangle_{P'} \mathsf{X}\varphi$ *(for $P \leq_C P'$)*

($\top$). $\langle\langle C\rangle\rangle_P \mathsf{X}\top$ (M). $\neg\langle\langle C\rangle\rangle_P \mathsf{X}\varphi \rightarrow \langle\langle \overline{C}\rangle\rangle_P \mathsf{X}\neg\varphi$

(S). $\langle\langle C_1\rangle\rangle_P \mathsf{X}\varphi \wedge \langle\langle C_2\rangle\rangle_P \mathsf{X}\psi \rightarrow \langle\langle C_1 \cup C_2\rangle\rangle_P \mathsf{X}(\varphi \wedge \psi)$ *(for disjoint C_1, C_2)*

(FP$_U$). $\langle\langle C\rangle\rangle_P \psi\mathsf{U}\varphi \leftrightarrow \varphi \vee (\psi \wedge \langle\langle C\rangle\rangle_P \mathsf{X}\langle\langle C\rangle\rangle_P \psi\mathsf{U}\varphi)$

(LFP$_U$). $\langle\langle \emptyset\rangle\rangle_{P'} \mathsf{G}((\varphi \vee (\psi \wedge \langle\langle C\rangle\rangle_P \mathsf{X}\chi)) \rightarrow \chi) \rightarrow \langle\langle \emptyset\rangle\rangle_{P'} \mathsf{G}(\langle\langle C\rangle\rangle_P \psi\mathsf{U}\varphi \rightarrow \chi)$

Closed under Modus Ponens and:

X-Mon. $\dfrac{\varphi \rightarrow \psi}{\langle\langle C\rangle\rangle_P \mathsf{X}\varphi \rightarrow \langle\langle C\rangle\rangle_P \mathsf{X}\psi}$ G-Nec. $\dfrac{\varphi}{\langle\langle C\rangle\rangle_P \mathsf{G}\varphi}$

We use the shorthand $\langle\langle C\rangle\rangle_P \mathsf{G}\varphi := \neg\langle\langle C\rangle\rangle_P \top\mathsf{U}\neg\varphi$ to align with the usual presentation. We will omit the proof of soundness in SOEMs. Since ATLDS only differs from ATL in terms of the one-step axioms, weak completeness (i.e. every valid formula is a theorem) over SOEMs can be obtained with with a similar construction as for ATL [16] or for CTL in Chapter 9 of [10]. We will provide a brief overview of a weak completeness result for ATLDS by constructing a canonical model to satisfy any finite ATLDS-consistent formula ϕ. We omit some proofs for brevity. We will denote the set of all maximal consistent sets (MCSs) of ATLDS formulae as Γ. We denote the sets containing the formula φ by $\widehat{\varphi} = \{q \in \Gamma | \varphi \in q\}$. Take the SOEM $\mathcal{M}^{\Gamma} = (Q^{\Gamma}, E^{\Gamma}, V^{\Gamma})$ where: $Q^{\Gamma} = \Gamma$; $X \in E^{\Gamma}_q(C,P)$ iff $\exists\widehat{\varphi} \subseteq X.\langle\langle C\rangle\rangle_P \mathsf{X}\varphi \in q$; and $q \in V^{\Gamma}(p)$ iff $p \in q$.

It can be seen that this definition ensures E^{Γ} is well-defined and has the required properties of a standard ordered effectivity model. This also guarantees local consistency, in that $\widehat{\varphi} \in E^{\Gamma}_q(C,P)$ iff $\langle\langle C\rangle\rangle_P \mathsf{X}\varphi \in q$. However, there are no guarantees that an $\langle\langle C\rangle\rangle_P \psi\mathsf{U}\varphi$ formula is realised in this model, as we may not necessarily reach φ in finite steps. We can rectify this by creating a finite model for any consistent formula ψ, via a filtration of the current model.

We will assume formulae are in a normal-form where negations are directly in front of an $\langle\langle C\rangle\rangle_P \psi\mathsf{U}\varphi$ formula or atomic proposition, by repeated application of the identities $\neg(\varphi_1 \wedge \varphi_2) \equiv \neg\varphi_1 \vee \neg\varphi_2$, $\neg(\varphi_1 \vee \varphi_2) \equiv \neg\varphi_1 \wedge \neg\varphi_2$, and $\neg\langle\langle C\rangle\rangle_P \mathsf{X}\varphi \equiv \langle\langle \overline{C}\rangle\rangle_P \mathsf{X}\neg\varphi$. We use $\sim\varphi$ to refer to the normal form of $\neg\varphi$. First, we state the (slightly altered) definition of the *closure* and *extended closure* of ϕ from [16]:

Definition 17. *The closure $cl(\phi)$ of ψ is the smallest set of formulae containing ψ such that:*

- *All subformulae of ψ are in $cl(\phi)$*
- *If $\langle\langle C\rangle\rangle_P \psi\mathsf{U}\varphi \in cl(\phi)$ then $\langle\langle C\rangle\rangle_P \mathsf{X}\langle\langle C\rangle\rangle_P \psi\mathsf{U}\varphi \in cl(\phi)$*
- *If $\neg\langle\langle C\rangle\rangle_P \psi\mathsf{U}\varphi \in cl(\phi)$ then $\neg\langle\langle C\rangle\rangle_P \mathsf{X}\langle\langle C\rangle\rangle_P \psi\mathsf{U}\varphi \in cl(\phi)$*
- *If $\varphi \in cl(\phi)$ then $\sim\varphi \in cl(\phi)$*
- *$\langle\langle C\rangle\rangle_P \mathsf{X}\top \in cl(\phi)$ For all C, P*

Definition 18. *The extended closure $ecl(\phi)$ is obtained from $cl(\phi)$ by:*

- *Including all coalitions and permutations for one-step formulae: $cl^+(\phi) = cl(\phi) \cup \{\langle\langle C'\rangle\rangle_{P'} \mathsf{X}\varphi, \neg\langle\langle C'\rangle\rangle_{P'} \mathsf{X}\varphi | \langle\langle C\rangle\rangle_P \mathsf{X}\varphi \in cl(\phi), C' \subseteq Ag, P' \in \mathfrak{S}(Ag)\}$*
- *Closing $cl^+(\phi)$ under finite positive boolean combinations of formulae[2] in $cl^+(\phi)$ up to tautological equivalence.*

It can be seen that $ecl(\phi)$ is finite. We will perform a filtration of $\mathcal{M}^{\Gamma}$ through $ecl(\phi)$. First, we group states in Q^{Γ} by equivalence classes $[q] = \{s \in Q^{\Gamma} | s \cap ecl(\phi) = q \cap ecl(\phi)\}$ to get maximal consistent subsets of $ecl(\phi)$. Given a subset of states $X \subseteq Q^{\Gamma}$, we denote by $[X]$ the set $\{[x] | x \in X\}$. We mirror the definition of filtration for Kripke models [6].

Definition 19. *A filtration of an effectivity model $\mathcal{M} = (Q, E, V)$ through a set of formulae Σ is a model $\mathcal{M}^{\Sigma} = (Q^{\Sigma}, E^{\Sigma}, V^{\Sigma})$ such that: $Q^{\Sigma} = [Q]$; $X \in E_q(C,P)$ implies $[X] \in E^{\Sigma}_{[q]}$; for each $q \in Q$ and $\langle\langle C\rangle\rangle_P \mathsf{X}\varphi \in \Sigma$, $[\widehat{\varphi}] \in E^{\Sigma}_{[q]}(C,P)$ implies $\widehat{\varphi} \in E_q(C,P)$; and for each atomic proposition p, $V(p) = [V(p)]$.*

We require a filtration of $\mathcal{M}^{\Gamma}$ that ensures each E^{Γ}_q is still monotonically ordered and maximally playable. From now on, through some abuse of notation we will treat $[s]$ contextually as either an

[2] i.e. formulae joined with $\vee$ and $\wedge$

equivalence class of states or as the set $s \cap ecl(\phi)$; i.e. we will say $\varphi \in [s]$ to mean $\varphi \in s \cap ecl(\phi)$. Then, we obtain the filtration of $\mathcal{M}^{\Gamma}_{ecl(\phi)}$ defined by the model $\mathcal{M}^{ecl(\phi)} = (Q^{ecl(\phi)}, E^{ecl(\phi)}, V^{ecl(\phi)})$ where $Q^{ecl(\phi)} = \{[q] | q \in Q^{\Gamma}\}$, $[X] \in E^{ecl(\phi)}_{[q]}(C, P)$ iff $X \in E^{\Gamma}_{q'}(C, P)$ for some $q' \in [q]$, and $V^{ecl(\phi)}(p) = \{[q] | q \in V^{\Gamma}(p)\}$.

Lemma 20. *If we close $E^{ecl(\phi)}_{[q]}$ under superadditivity, outcome monotonicity, order monotonicity, and maximality, the resulting effectivity function $\boldsymbol{E}^{ecl(\phi)}_{[q]}$:*

- *is monotonically ordered and maximally playable,*
- *still makes $\mathcal{M}^{ecl(\phi)}$ a filtration (i.e. $X \in E^{\Gamma}_{q}(C, P)$ implies $[X] \in \boldsymbol{E}^{ecl(\phi)}_{[q]}(C, P)$ and for each $\langle\langle C\rangle\rangle_P \mathsf{X}\varphi \in ecl(\phi)$, $[\widehat{\varphi}] \in \boldsymbol{E}^{ecl(\phi)}_{[q]}(C, P)$ implies $\widehat{\varphi} \in E^{\Gamma}_{q}(C, P)$)*

The filtration preserves local consistency of formulae in $ecl(\phi)$ by construction:

Lemma 21. *(Local Consistency) For any $\langle\langle C\rangle\rangle_P \mathsf{X}\varphi \in ecl(\phi)$, $\langle\langle C\rangle\rangle_P \mathsf{X}\varphi \in [q]$ iff $[\widehat{\varphi}] \in \boldsymbol{E}^{ecl(\phi)}_{q}(C, P)$*

Proof. If $\langle\langle C\rangle\rangle_P \mathsf{X}\varphi \in [q]$, then by the definition of E^{Γ} and Lemma 20, $[\widehat{\varphi}] \in \mathbf{E}^{ecl(\phi)}_{q}(C, P)$. Now, suppose that $[\widehat{\varphi}] \in \mathbf{E}^{ecl(\phi)}_{[q]}(C, P)$. So, by Lemma 20, $\widehat{\varphi} \in E^{\Gamma}_{q}(C, P)$, and from the definition of $E^{\Gamma}_{q}(C, P)$ there is some $\widehat{\theta} \subseteq \widehat{\varphi}$ such that $\langle\langle C\rangle\rangle_P \mathsf{X}\theta \in q$. Since $\widehat{\theta} \subseteq \widehat{\varphi}$, $\theta \rightarrow \varphi$ is an ATLDS theorem so $\langle\langle C\rangle\rangle_P \mathsf{X}\theta \rightarrow \langle\langle C\rangle\rangle_P \mathsf{X}\varphi$ is also a theorem by X-monotonicity. Therefore, $\langle\langle C\rangle\rangle_P \mathsf{X}\varphi \in [q]$ as required. □

So we just need to show that $\langle\langle C\rangle\rangle_P \psi \mathsf{U}\varphi$ formulae are realised whenever they appear in a state $[q]$. Recall the U operator defined previously; we define an analogous operator $\widehat{U}(X) = [\widehat{\varphi}] \cup ([\widehat{\psi}] \cap \{[z] \in Q^{ecl(\phi)} | X \in \mathbf{E}^{ecl(\phi)}_{[z]}(C, P)\})$, which can be iterated such that $\widehat{U}^0 = [\widehat{\varphi}]$ and $\widehat{U}^{i+1} = \widehat{U}(\widehat{U}^i)$. We also need the following lemma, as in [16], that allows us to represent sets of states using formulae:

Lemma 22. *For any subset $Y \subseteq Q^{ecl(\phi)}$ there is a formula $\chi_Y \in ecl(\phi)$ such that for every $[y] \in Q^{ecl(\phi)}$, $\chi_Y \in [y]$ iff $[y] \in Y$.*

Lemma 23. *Take any formula of the form $\langle\langle C\rangle\rangle_P \psi \mathsf{U}\varphi$ in $ecl(\phi)$. Then, $\langle\langle C\rangle\rangle_P \psi \mathsf{U}\varphi \in [q]$ iff there is a finite i such that $[q] \in \widehat{U}^i$.*

Proof. Suppose there is a finite i such that $[q] \in \widehat{U}^i$. For all states $[s]$ in $\widehat{U}^0$, $\varphi \in [s]$, so by FP_U, $\langle\langle C\rangle\rangle_P \psi \mathsf{U}\varphi \in [s]$. Assume for some finite j that for all states $[s]$ in $\widehat{U}^j$, it is the case that $\langle\langle C\rangle\rangle_P \psi \mathsf{U}\varphi \in [s]$. For any state $[s']$ in $\widehat{U}^{j+1}$, either $\varphi \in [s']$ giving $\langle\langle C\rangle\rangle_P \psi \mathsf{U}\varphi \in [s']$, or $\psi \in [s']$ and $[\widehat{\langle\langle C\rangle\rangle_P \psi \mathsf{U}\varphi}] \in \mathbf{E}^{ecl(\phi)}_{[s']}(C, P)$. From local consistency, $\langle\langle C\rangle\rangle_P \mathsf{X} \langle\langle C\rangle\rangle_P \psi \mathsf{U}\varphi \in [s']$, so from the FP_U axiom $\langle\langle C\rangle\rangle_P \psi \mathsf{U}\varphi \in [s']$. By induction, $\langle\langle C\rangle\rangle_P \psi \mathsf{U}\varphi \in [q]$.

Conversely, suppose that $\langle\langle C\rangle\rangle_P \psi \mathsf{U}\varphi \in [q]$. Let us take the set of states which meet the required condition that there is a finite i such that $[q] \in \widehat{U}^i$, call this Z. We can construct a characteristic formula χ_Z such that $\chi_Z \in [s]$ iff $[s] \in Z$. To show $[q]$ has the required condition and is therefore in Z, it is sufficient to show that $\langle\langle C\rangle\rangle_P \psi \mathsf{U}\varphi \rightarrow \chi_Z$ is an ATLDS theorem, so is in all MCSs of formulae. To show this, it is sufficient (via LFP_U, FP_U, and G-Nec) to show that $\varphi \vee (\psi \wedge \langle\langle C\rangle\rangle_P \mathsf{X}\chi_Z) \rightarrow \chi_Z$ is in every MCS. For any MCS $x \in \Gamma$, if $\varphi \vee (\psi \wedge \langle\langle C\rangle\rangle_P \mathsf{X}\chi_Z) \notin x$ then we get the required result. If $\varphi \vee (\psi \wedge \langle\langle C\rangle\rangle_P \mathsf{X}\chi_Z) \in x$, then either $\varphi \in x$ or $\psi \wedge \langle\langle C\rangle\rangle_P \mathsf{X}\chi_Z \in x$. Suppose the former; then $\langle\langle C\rangle\rangle_P \psi \mathsf{U}\varphi \in x$, so $[x] \in \widehat{U}^0$ meaning $[x] \in Z$ and $\chi_Z \in [x]$. If on the other hand, $\psi \wedge \langle\langle C\rangle\rangle_P \mathsf{X}\chi_Z \in x$ then $\psi \in x$ and $\langle\langle C\rangle\rangle_P \mathsf{X}\chi_Z \in x$ so from local consistency of M^{Γ}, $\widehat{\chi_Z} \in E^{\Gamma}_{x}(C, P)$. So from Lemma 20, $[\widehat{\chi_Z}] \in \mathbf{E}^{ecl(\phi)}_{[x]}(C, P)$. At a state [q] containing χ_Z in the filtrated model, there is a smallest finite i such that $[q] \in \widehat{U}^i$; the set of states $[\widehat{\chi_Z}]$ is finite, so we can find the maximum such i across all states, call this i'. Since $[\widehat{\chi_Z}] \in E_{[x]}(C, P)$, it must be that $[x] \in \widehat{U}^{i'+1}$, so $\chi_Z \in [x]$. So, $\varphi \vee (\psi \wedge \langle\langle C\rangle\rangle_P \mathsf{X}\chi_Z) \rightarrow \chi_Z$ is in every MCS x, so $\langle\langle C\rangle\rangle_P \psi \mathsf{U}\varphi \rightarrow \chi_Z$ is an ATLDS theorem. Since $\langle\langle C\rangle\rangle_P \psi \mathsf{U}\varphi \in [q]$, it must be that $\chi_Z \in [q]$, so $[q] \in Z$ and there is a finite i such that $[q] \in \widehat{U}^i$. □

Lemma 24. *(Truth Lemma) Given a formula $\varphi \in ecl(\phi)$ and model $\mathcal{M}^{\Gamma}_{ecl(\phi)}$, then for any state $[q]$, $\varphi \in [q]$ iff $\mathcal{M}^{\Gamma}_{ecl(\phi)}, [q] \models \varphi$.*

Proof. We show the main inductive cases. For atomic propositions, the definition of $V^{ecl(\phi)}$ guarantees this, and the boolean connectives are similarly clear. For $\langle\langle C\rangle\rangle_P \mathsf{X}\varphi$ formulae, this comes directly from Lemma 21. Note that under the inductive hypothesis for $\langle\langle C\rangle\rangle_P \psi \mathsf{U}\varphi$ formulae, the operator $\widehat{U}$ and U are equivalent, the latter providing the semantics of $\langle\langle C\rangle\rangle_P \psi \mathsf{U}\varphi$. So, Lemma 23 immediately gives us that $\langle\langle C\rangle\rangle_P \psi \mathsf{U}\varphi \in [q]$ iff $\mathcal{M}^{\Gamma}_{ecl(\phi)}, [q] \models \langle\langle C\rangle\rangle_P \psi \mathsf{U}\varphi$ as required. For both cases when $\neg\langle\langle C\rangle\rangle_P \psi \mathsf{U}\varphi \in ecl(\phi)$, note that $cl(\phi)$ is closed under negation so $\langle\langle C\rangle\rangle_P \psi \mathsf{U}\varphi \in ecl(\phi)$ also. Since each $[q]$ is a maximally consistent subset of $ecl(\phi)$, it must be that $\neg\langle\langle C\rangle\rangle_P \psi \mathsf{U}\varphi \in [q]$ iff $\langle\langle C\rangle\rangle_P \psi \mathsf{U}\varphi \notin [q]$ iff (by Lemma 23) $\mathcal{M}^{\Gamma}_{ecl(\phi)}, [q] \not\models \langle\langle C\rangle\rangle_P \psi \mathsf{U}\varphi$.

□

Therefore, every ATLDS-consistent formula has a satisfying model, which has finite states by construction, so:

Theorem 25. *The axiomatic system for ATLDS is sound and (weakly) complete wrt SOEMs.*

Corollary 26. *Every satisfiable formula of ATLDS is satisfiable in a finite-state SOEM.*

The construction in Theorem 15 should allow us to apply all of our results to CGMs, but there is a problem: the ATLDS axioms do not guarantee the 'crown condition' property which is required for the construction. However, we can show that the class of effectivity models with this condition have exactly the same validities as the class without this condition, using the fact that any playable effectivity function over a finite domain is truly playable [14]:

Proposition 27. *For ATLDS formula φ, $\models \varphi$ in the class of SOEMs iff $\models \varphi$ in the class of truly playable SOEMs*

Proof. Since truly playable models are a subclass of the standard models, any validity in SOEMs is valid in the class of truly playable models. Conversely, if φ is valid for truly playable SOEMs but fails in some SOEM, there must be a finite counter-model by Corollary 26. All finite models are truly playable, so φ must hold, giving a contradiction. Therefore φ must be valid across all SOEMs. □

Note that by Theorem 15 any truly playable effectivity function can be generated by a game and vice versa, so by Proposition 16 the valid formulae over truly playable SOEMs are exactly the validities over CGMs. Therefore:

Corollary 28. *The axiomatic system is sound and (weakly) complete wrt CGMs with ATLDS semantics.*

7 MODEL CHECKING

In ATL, the model checking problem (determining whether $\mathcal{M}, s \models \varphi$ given a CGM $\mathcal{M}$, state s, and formula φ) is PTime-Complete [2]. This procedure extends naturally to ATLDS, also yielding a polynomial-time algorithm with respect to the model size (as shown for $SL^{-}[SG]$ in [4]). However, if we are modelling a system, we might expect the transition function to have some structure that allows us to encode it compactly with respect to the set of agents and actions. In this setting ATLDS is more complex than ATL. Therefore, we will investigate the model checking problem under *efficient encoding of CGMs*. By this we mean the model checking problem restricted to the class of CGMs where the transition function δ can be encoded in space $p(|Q| \times |A| \times |Ag|)$ (for some polynomial p) and given an action profile a_{Ag} and state q we can recover the state $\delta(q, a_{Ag})$ from this encoding in polynomial time. One such approach is the implicit concurrent game models found in [19] and [17]. In this section, we will only consider CGMs with a finite number of states, actions, and agents. We will also use notions from complexity theory relating to the polynomial hierarchy [3].

In a CGM with state space Q, for a coalition C, permutation P, and subset of states $S \subseteq Q$, the set $pre(S, C, P) = \{q \in Q | S \in E^{\pi}_{G_q}(C, P)\}$ is the set of states from which C moving under permutation P can guarantee they are in S after one move. We can check if $q \in [\![\langle\!\langle C\rangle\!\rangle_P \mathsf{X}\varphi]\!]$ by checking if $q \in pre([\![\varphi]\!], C, P)$. Given two sets of states X and Y, we can define a monotone operator $\mathcal{U} : \mathcal{P}(Q) \to \mathcal{P}(Q)$ as $\mathcal{U}(S) = X \cup (Y \cap pre(S, C, P))$. When $X = [\![\varphi]\!]$ and $Y = [\![\psi]\!]$ the fixpoint of this operator is the set of states for which $[\![\langle\!\langle C\rangle\!\rangle_P \psi \mathsf{U} \varphi]\!]$; if we start at $\emptyset$, this fixpoint must be reached within $|Q|$ iterations. Similarly, $q \in [\![p]\!]$ if $p \in V(q)$, $q \in [\![\varphi \wedge \psi]\!]$ if $q \in [\![\varphi]\!] \cap [\![\psi]\!]$, and $q \in [\![\neg\varphi]\!]$ iff $q \in Q \setminus [\![\varphi]\!]$.

Proposition 29. *Model Checking problem for ATLDS under efficient encoding of CGMs is PSpace-Complete.*

Proof. We can recursively check $q \in [\![\varphi]\!]$ using the above method, keeping track of a set of states of size at most $|Q|$ for each subformula. To compute $pre([\![\psi]\!], C, P)$ for some ψ, we can iterate through all strategies at a state, and record if any of them result in a set of states within $[\![\psi]\!]$. We can encode the result of a strategy at a state as a tree, where the nodes at depth i are labelled with the agent $P[i]$. For nodes labelled with an agent $P[i] \in \overline{C}$, there is a successor for each action available to them. For nodes labelled with an agent $P[i] \in C$, there is a single successor corresponding to a choice of an action available to them. Each leaf node corresponds to an action profile, so we can calculate a successor state from this. In a depth-first exploration of this tree, we only need to store a single action profile of size $O(|Ag|)$ and the successor states so far of size $O(|Q|)$ at any one time. Therefore, the problem is in PSpace.

For hardness, we reduce from the PSpace-Complete problem TQBF. Suppose wlog we have a formula in the form $\exists x_1 \forall y_1 \ldots \exists x_n \forall y_n . \phi(x_1, y_1, \ldots x_n, y_n)$ where ϕ is a boolean formula with free variables $x_1, y_1, \ldots x_n, y_n$ and each quantifier ranges over $\{0, 1\}$. We can create a CGM with $\mathcal{M}$ with $Ag = \{x_1, y_1 \ldots, x_n, y_n\}$, $Q = \{q_0, t, f\}$, $A = \{0, 1, *\}$, for all agents i, $act(i, q_0) = \{0, 1\}$ and $act(i, t) = act(i, f) = \{*\}$, and transition function $\delta(t, *) = t$, $\delta(f, *) = f$, and $\delta(q_0, a_{x_1}, a_{y_1}, \ldots, a_{x_n}, a_{y_n})$ outputs t whenever $\phi(a_{x_1}, a_{y_1}, \ldots, a_{x_n}, a_{y_n})$ is true and f otherwise. If we have a single propositional variable p which holds only at state t, it can now be seen that $\mathcal{M}, q_0 \models \langle\!\langle \{x_1, x_2, \ldots, x_n\}\rangle\!\rangle_{(x_1, y_1, \ldots, x_n, y_n)} \mathsf{X} p$ iff $\exists x_1 \forall y_1 \ldots \exists x_n \forall y_n . \phi(x_1, y_1, \ldots x_n, y_n)$. □

However, we can do better if we bound the number of agents:

Proposition 30. *Model Checking for ATLDS under efficient encoding of CGMs with a fixed number of agents is in NP ∩ co-NP.*

Proof. All steps except evaluating $q \in pre(S, C, P)$ run in polynomial time. With a fixed number of agents $|Ag|$, the size of a strategy restricted to q is polynomially bounded by $O(|A|^{|Ag|})$. We can guess a joint strategy σ^P_C and verify whether it ensures reaching S against all strategies of $\overline{C}$ in polynomial time, meaning the problem is in NP. The complement problem reduces to checking $\neg\varphi$, so is also in NP, yielding membership in NP ∩ co-NP. □

With an unbounded number of agents, the complexity of model checking depends on the alternation number of the formula:

Proposition 31. *Model Checking for ATLDS under efficient encoding of CGMs of formulae φ such that $alt(\varphi) \leq k$ is in Δ^P_{k+1}.*

Proof. For coalition C, permutation P beginning with $P[1] \in C$, with alternation number k, it is the case that $q \in pre(S, C, P)$ iff

$$\exists a_{C_1} \forall a_{S_2} \exists a_{C_3} \ldots ((\exists/\forall) a_{(C_k/S_k)} (\delta(q, a_{C_1}, \ldots, a_{(C_k/S_k)}) \in S)$$

where each $C_i \subseteq C, S_i \subseteq \overline{C}$ and $a_C \in \Pi_{i \in C}(act(i, q))$. Checking $\delta(q, a_{C_1}, \ldots, a_{(C_k/S_k)}) \in S)$ can be done in polynomial time, so this is in Σ^P_k by definition. Similarly, checking $q \in pre(S, C, P)$ when the first agent in $P[1]$ is in $\overline{C}$ and $alt(\varphi) \leq k$ is in Π^P_k. Since a Σ^P_k oracle can check $q \in pre(S, C, P)$ in constant time, we can solve the model checking problem with alternation number bounded by k in polynomial time: so, it is in Δ^P_{k+1}. □

8 CONCLUSION

We have presented ATLDS, extending ATL with a modality that encodes an order in which agents move. We have provided a sound and complete axiomatic system for ATLDS, and given complexity bounds for variants of the model-checking problem, demonstrating how the quantifier-alternation behaviour of ATLDS makes certain decision problems more difficult than in ATL. We have also given a transformation between ATLDS and SL^{-}[SG], linking these results to a known fragment of Strategy Logic.

We could extend the syntax of ATLDS to obtain an ATL*-like logic where modalities are followed by arbitrary LTL formulae - presumably this would align with the ATL*-like fragment of strategy logic, SL^{-}[1G] [21]. However, since an axiomatisation has not yet been found for ATL*, it is unlikely the completeness results of this paper could be extended in an obvious way. It may be more fruitful to use the ATLDS modality defined in this paper to construct a variant of the alternating-time μ–calculus. The dependence in ATLDS mirrors that of first-order quantifiers, so another natural extension would be to capture the dependence of, for example, Henkin quantifiers. This could be achieved by replacing strategies under a permutation with strategies under a partial order.

REFERENCES

[1] J. Abdou and H. Keiding. 1991. *Effectivity Functions in Social Choice.* Springer Netherlands. https://doi.org/10.1007/978-94-011-3448-4

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

[3] Sanjeev Arora and Boaz Barak. 2009. *The polynomial hierarchy and alternations.* Cambridge University Press, 95–105.

[4] 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 Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19.* International Joint Conferences on Artificial Intelligence Organization, 88–94. https://doi.org/10.24963/ijcai.2019/13

[5] Dylan Bellier. 2024. *Strategic reasoning with dependencies : hyperteam logics, realizable strategies, dependency matrices.* Theses. Université de Rennes. https://theses.hal.science/tel-04959292

[6] Patrick Blackburn, Maarten de Rijke, and Yde Venema. 2001. *Models.* Cambridge University Press, 50–122.

[7] Krishnendu Chatterjee, Thomas A. Henzinger, and Nir Piterman. 2010. Strategy logic. *Information and Computation* 208, 6 (2010), 677–693. https://doi.org/10.1016/j.ic.2009.07.004 Special Issue: 18th International Conference on Concurrency Theory (CONCUR 2007).

[8] Sebastian Enqvist and Valentin Goranko. 2022. The Temporal Logic of Coalitional Goal Assignments in Concurrent Multiplayer Games. *ACM Trans. Comput. Logic* 23, 4, Article 21 (Oct. 2022), 58 pages. https://doi.org/10.1145/3517128

[9] Patrick Gardy, Patricia Bouyer, and Nicolas Markey. 2020. Dependences in Strategy Logic. *Theory of Computing Systems* 64, 3 (01 Apr 2020), 467–507. https://doi.org/10.1007/s00224-019-09926-y

[10] Robert Goldblatt. 1992. *Logics of time and computation.* Center for the Study of Language and Information Publications.

[11] 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

[12] Valentin Goranko and Sebastian Enqvist. 2018. Socially Friendly and Group Protecting Coalition Logics. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems* (Stockholm, Sweden) *(AAMAS '18).* International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 372–380.

[13] Valentin Goranko and Wojciech Jamroga. 2016. State and path coalition effectivity models of concurrent multi-player games. *Autonomous Agents and Multi-Agent Systems* 30, 3 (01 May 2016), 446–485. https://doi.org/10.1007/s10458-015-9294-4

[14] Valentin Goranko, Wojciech Jamroga, and Paolo Turrini. 2013. Strategic games and truly playable effectivity functions. *Autonomous Agents and Multi-Agent Systems* 26, 2 (01 Mar 2013), 288–314. https://doi.org/10.1007/s10458-012-9192-y

[15] Valentin Goranko and Fengkui Ju. 2022. A Logic for Conditional Local Strategic Reasoning. *Journal of Logic, Language and Information* 31, 2 (01 Jun 2022), 167–188. https://doi.org/10.1007/s10849-022-09357-y

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

[17] Wojciech Jamroga and Jürgen Dix. 2008. Model Checking Abilities of Agents: A Closer Look. *Theory of Computing Systems* 42, 3 (01 Apr 2008), 366–410. https://doi.org/10.1007/s00224-007-9080-z

[18] François Laroussinie and Nicolas Markey. 2015. Augmenting ATL with strategy contexts. *Information and Computation* 245 (2015), 98–123. https://doi.org/10.1016/j.ic.2014.12.020

[19] Francois Laroussinie, Nicolas Markey, and Ghassan Oreiby. 2008. On the Expressiveness and Complexity of ATL. *Logical methods in computer science* 4, Issue 2, 2:7 (2008).

[20] René Mazala. 2002. *Infinite Games.* Springer Berlin Heidelberg, Berlin, Heidelberg, 23–38. https://doi.org/10.1007/3-540-36387-4_2

[21] Fabio Mogavero, Aniello Murano, Giuseppe Perelli, and Moshe Y. Vardi. 2012. What Makes Atl* Decidable? A Decidable Fragment of Strategy Logic. In *CONCUR 2012 – Concurrency Theory*, Maciej Koutny and Irek Ulidowski (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 193–208.

[22] Fabio Mogavero, Aniello Murano, Giuseppe Perelli, and Moshe Y. Vardi. 2014. Reasoning About Strategies: On the Model-Checking Problem. *ACM Trans. Comput. Logic* 15, 4, Article 34 (Nov. 2014), 47 pages. https://doi.org/10.1145/2631917

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

[24] Fabio Mogavero, Aniello Murano, and Luigi Sauro. 2014. A Behavioral Hierarchy of Strategy Logic. In *Computational Logic in Multi-Agent Systems*, Nils Bulling, Leendert van der Torre, Serena Villata, Wojtek Jamroga, and Wamberto Vasconcelos (Eds.). Springer International Publishing, Cham, 148–165.

[25] Fabio Mogavero, Aniello Murano, and Moshe Y. Vardi. 2010. Reasoning About Strategies. In *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2010) (Leibniz International Proceedings in Informatics (LIPIcs), Vol. 8)*, Kamal Lodaya and Meena Mahajan (Eds.). Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 133–144. https://doi.org/10.4230/LIPIcs.FSTTCS.2010.133

[26] Pavel Naumov and Yuan Yuan. 2021. Intelligence in Strategic Games. *Journal of Artificial Intelligence Research* 71 (July 2021), 521–556. https://doi.org/10.1613/jair.1.12883

[27] Eric Pacuit. 2017. *Neighborhood Semantics for Modal Logic.* Springer International Publishing. https://doi.org/10.1007/978-3-319-67149-9

[28] Marc Pauly. 2001. *Logic for Social Software.* Ph.D. Dissertation. University of Amsterdam.

[29] Marc Pauly. 2002. A Modal Logic for Coalitional Power in Games. *Journal of Logic and Computation* 12, 1 (02 2002), 149–166. https://doi.org/10.1093/logcom/12.1.149 arXiv:https://academic.oup.com/logcom/article-pdf/12/1/149/3657514/120149.pdf