I Would If I Could: Reasoning about Dynamics of Actions in Multi-Agent Systems

Rustam Galimullin[1] , **Hermine Grosinger**[2] , **Munyque Mittelmann**[3,4]
[1]University of Bergen, Bergen, Norway
[2]Örebro University, Örebro, Sweden
[3]CNRS, UMR 7030, F-93430, Villetaneuse, France
[4]Université Sorbonne Paris Nord, LIPN, F-93430, Villetaneuse, France
rustam.galimullin@uib.no, hermine.grosinger@oru.se, mittelmann@lipn.univ-paris13.fr

Abstract

Autonomous agents acting in realistic Multi-Agent Systems (MAS) should be able to adapt during their execution. Standard strategic logics, such as the Alternating-time Temporal Logic (ATL), model agents' state- or history-dependent behaviour. However, the dynamic treatment of agents' available actions and their knowledge of required actions is still rarely addressed. In this paper, we introduce ATL with Dynamic Actions (ATL-D), which models the process of granting and revoking actions, and its extension ATEL-D, which captures how such updates affect agents' knowledge. Beyond the conceptual contribution, we provide several technical results: we analyse the expressivity of our logic in relation to ATL, study its relation to normative systems, and provide complexity results for relevant computational problems.

1 Introduction

In real-world, long-term scenarios with autonomous agents such as robots, it is unrealistic to assume that they possess all the necessary abilities and knowledge required to achieve certain tasks (Grosinger 2025). In general, what the agents should or should not be able to do might change dynamically. Reasoning about the skills an agent must acquire to achieve a given goal is naturally relevant for their design and reconfiguration. At the same time, formal treatment of dynamic phenomena in Multi-Agent Systems (MAS) can be combined with verification techniques to assess the correctness and safety of evolving systems (Galimullin et al. 2026). Take the following example, adapted from (Grosinger 2025):

> Bob is an elderly person. To support his everyday life at home, he has two autonomous robots. They help him, for example, to take his medicine as he is forgetful. Robot 1 is tasked with retrieving the medicine from a drawer, while Robot 2 takes care of other tasks and thus does not have the action for opening the drawer. Now, suppose that the drawer with the medicine is stuck, and hence Robot 1 cannot get the medicine on her own. We want to *grant* Robot 2 the same action, then the two robots together can open the drawer and thus deliver the medicine to Bob. Or if Bob is prescribed a new medicine that is sensitive to heat, we want to *remove* the action from the robots to bring it into a warm area. Ultimately, we would like to be able to deal with situations when the robots are uncertain about which medicine they are handling and their precise location, while reasoning about the evolution of their skills.

Research on MAS has a long tradition of investigating how to formally model problems arising from the (possibly adversarial) interactions of multiple agents operating within the same environment. Logic-based formalisms for strategic reasoning in MAS, such as *Alternating-time Temporal Logic* (ATL) (Alur, Henzinger, and Kupferman 2002), allow expressing that an agent (or a coalition of agents) can bring about a temporal goal. *Alternating-time Temporal Epistemic Logic* (van der Hoek and Wooldridge 2003; Jamroga and van der Hoek 2004), denoted ATEL, extends ATL with epistemic operators, capturing agents' knowledge about the system and their ability to cooperate in imperfect information scenarios. ATL and ATEL have been effectively implemented in tools such as MCMAS (Lomuscio, Qu, and Raimondi 2017), STV (Kurpiewski, Jamroga, and Knapik 2019), and VITAMIN (Ferrando and Malvone 2025), and both logics have been used for verification of a wide variety of applications, including e-voting protocols (Jamroga et al. 2022), autonomous submarines (Ezekiel et al. 2011), and smart contracts (Nam and Kil 2022), to name a few.

As expressive as ATL and ATEL may be, they cannot naturally model Bob's example above. While ATL agents can dynamically react based on the system's execution state or history, these logics are defined over fixed models. In this paper, we are interested in moving from a static representation of models to a more dynamic one, in which we are able to update the set of actions that an agent has at her disposal (e.g., an action to open a drawer) and ensure the agent knows about this action update. More precisely, we want to capture that agents *know* which actions they and other agents have, which actions they lack, and which actions they need to acquire or drop to achieve a goal. Going back to Bob's example, the robots could know that, although they currently cannot deliver the medicine, if Robot 2 was granted — or *acquired* — the action of opening the drawer, they could achieve the goal together. This type of reasoning is central in what (Grosinger 2025) calls *proactive learning*.

Contribution To address this limitation, we first introduce the new logic ATL-D which extends ATL (Section 2) with

update operators to grant and revoke actions. For the new logic and its fragments, we study their expressivity, and their relationship to some approaches in the field of normative reasoning in MAS (Alechina, Logan, and Dastani 2018). Indeed, revoking actions, such as moving to a warm place in Bob's example, serves as a mechanism for regulating what agents are permitted to do. Furthermore, we show that, as for ATL, model-checking ATL-D is in P and that the problem of determining the existence of a bounded update such that a given ATL-D formula is satisfied is NP-complete.

Next, we explore the epistemic dimension and introduce ATEL-D (Section 3), which allows us not only to update agents' actions, but also to capture the epistemic consequences of such updates. It enables reasoning about the agents that know which actions they have and which actions they are lacking. Model-checking of ATEL-D is Δ_2^{P}-complete, which is the same as for ATEL. Finally, we briefly discuss some related work and conclude in Section 4.

2 ATL with Dynamic Actions

We start by considering the setting with perfect information.

2.1 Syntax and Semantics

We fix a countable set of atomic *propositions* $\mathrm{AP} = \{p, q, ...\}$, and finite sets of *agents* $\mathrm{Ag} = \{a, b, ...\}$ and *actions* $\mathrm{Ac} = \{\alpha, \beta, ...\}$. The syntax of *Alternating-time Temporal Logic with Dynamic actions* (ATL-D) extends that of ATL (Alur, Henzinger, and Kupferman 2002) with operators for granting and revoking actions, as well as a construct to express that an agent currently has a particular action.

Definition 1 (ATL-D)**.** *Formulas φ of* ATL-D *are defined by the following grammar:*

$$\varphi ::= p \mid \neg\varphi \mid (\varphi \vee \varphi) \mid \langle\langle A\rangle\rangle \mathbf{X}\varphi \mid \langle\langle A\rangle\rangle \varphi \mathbf{U} \varphi \mid \langle\langle A\rangle\rangle \varphi \mathbf{R} \varphi \mid [\pi]^{+}\varphi \mid [\pi]^{-}\varphi \mid has(a, \alpha)$$

$$\pi ::= \varphi : \alpha \to A \mid \pi, \pi$$

where $p \in \mathrm{AP}$, $\alpha \in \mathrm{Ac}$, $a \in \mathrm{Ag}$, and $A \subseteq \mathrm{Ag}$. The fragments of ATL-D *without $[\pi]^{+}\varphi$ or $[\pi]^{-}\varphi$ will be denoted as* ATL-D^{-} *and* ATL-D^{+}*, respectively. Finally, the fragment of* ATL-D *without $[\pi]^{+}\varphi$, $[\pi]^{-}\varphi$, and $has(a, \alpha)$ is* ATL[1].

Operator $\langle\langle A\rangle\rangle \mathbf{X}\varphi$ means that a coalition A can force φ to be true in the *ne***X***t* state. The operator $\langle\langle A\rangle\rangle \varphi \mathbf{U} \psi$ means that A can maintain φ **U***ntil* ψ becomes true. $\langle\langle A\rangle\rangle \varphi \mathbf{R} \psi$ means that A can maintain ψ until φ (i.e., **R***eleases* the requirement for the truth of ψ). We can also define the usual derived temporal operators for *eventually* and *always* as follows: $\langle\langle A\rangle\rangle \mathbf{F}\varphi := \langle\langle A\rangle\rangle \top \mathbf{U} \varphi$ and $\langle\langle A\rangle\rangle \mathbf{G}\varphi := \langle\langle A\rangle\rangle \bot \mathbf{R} \varphi$.

Constructs $[\pi]^{+}\varphi$ mean that after applying the *granting actions update* π, φ holds. Formulas $[\pi]^{-}\varphi$ means that after applying the *removing actions update* π, φ holds. A granting actions update (resp. a removing actions update) $\varphi : \alpha \to A$ specifies that the action α is granted to (resp. removed from) the agents in A in *every state* where φ is true. Updates of different kinds can be applied sequentially (i.e., by chaining update operators), whereas updates of the same type can be applied simultaneously (e.g., $[\pi_1, \pi_2]^{+}\varphi$). Given an update $[\varphi_1 : \alpha_1 \to A_1, ..., \varphi_k : \alpha_k \to A_k]^{+}$ (resp. $[\varphi_1 : \alpha_1 \to A_1, ..., \varphi_k : \alpha_k \to A_k]^{-}$) we will abbreviate it as $[\overrightarrow{\varphi : \alpha \to A}]_k^{+}$ (resp. $[\overrightarrow{\varphi : \alpha \to A}]_k^{-}$). Operator $has(a, \alpha)$ means that, in the current state, the action α is available to agent a. Finally, all standard abbreviations of propositional logic, like $\varphi \wedge \psi$ and $\varphi \to \psi$, and conventions for removing parentheses hold. Given a formula φ, we define its *size* $|\varphi|$ as the number of symbols occurring in φ.

Now we turn our focus to the semantics of ATL-D.

Definition 2 (CGS)**.** *A* concurrent game structure *(CGS) is a tuple $\mathcal{G} = (V, \ell, d, o)$, where V is a finite set of* states*; $\ell : V \to 2^{\mathrm{AP}}$ is a* labelling function*; $d : \mathrm{Ag} \times V \to 2^{\mathrm{Ac}} \setminus \emptyset$ is an* availability function*, defining a non-empty set of actions available to agents at each state; $o : V \times \mathrm{Ac}^{\mathrm{Ag}} \to V$ is a* transition function*, assigning the outcome state to each state and all possible joint actions. We will also sometimes call a CGS a* model*.*

Given a CGS $\mathcal{G}$ and a state v, let $True(v) = \{p \in \mathrm{AP} \mid p \in \ell(v)\}$ be the set of propositions true in v. We define the size *of $\mathcal{G}$, denoted $|\mathcal{G}|$, as $|V| + \sum_{v \in V} |True(v)| + |d| + |o|$. We call $\mathcal{G}$* finite*, if $|\mathcal{G}|$ is finite.*

Observe that our definition of a CGS slightly deviates from the standard one as we specify all possible transitions in o even though agents may not have available actions in d to execute all of them. We can consider such transitions as 'enabled' and 'disabled', and below we make it clear how the dynamic operators of granting and removing actions affect the set of transitions enabled in a CGS.

Definition 3 (Paths and Strategies)**.** *A* path *$\lambda = \lambda_0 \lambda_1 ... \in V^{\omega}$ is an infinite sequence of states s.t. for every $i \geqslant 0$ there exists an action profile $\boldsymbol{\alpha} \in \prod_{a \in \mathrm{Ag}} d(a, \lambda_i)$ such that $o(\lambda_i, \boldsymbol{\alpha}) = \lambda_{i+1}$.*

A memoryless strategy *for an agent a is a function $\sigma_a : V \to \mathrm{Ac}$. We let Str_a be the set of memoryless strategies for agent a, and $\mathrm{Str}_A = \prod_{a \in A} \mathrm{Str}_a$ be the set of collections of strategies for agents a in A.*

For a state v and a coalition $A \in 2^{\mathrm{Ag}} \setminus \emptyset$, a strategy profile *of A is $\boldsymbol{\sigma}_A \in \mathrm{Str}_A$. The* outcome function *$\mathrm{Out}(v, \boldsymbol{\sigma}_A)$ returns the set of all paths starting at state v that occur when agents in A execute the strategy profile $\boldsymbol{\sigma}_A$.*

Definition 4 (Semantics ATL-D)**.** *Let $\mathcal{G} = (V, \ell, d, o)$ be a CGS, and $v \in V$ be a state. The* semantics *of* ATL-D *is defined by induction as follows (we omit Boolean cases for brevity):*

$\mathcal{G}, v \models \langle\langle A\rangle\rangle \mathbf{X}\varphi$ *iff* $\exists \boldsymbol{\sigma}_A \in \mathrm{Str}_A, \forall \lambda \in \mathrm{Out}(v, \boldsymbol{\sigma}_A) : \mathcal{G}, \lambda_1 \models \varphi$

$\mathcal{G}, v \models \langle\langle A\rangle\rangle \varphi \mathbf{U} \psi$ *iff* $\exists \boldsymbol{\sigma}_A \in \mathrm{Str}_A, \forall \lambda \in \mathrm{Out}(v, \boldsymbol{\sigma}_A) : \exists i \geqslant 0$ *s.t.* $\mathcal{G}, \lambda_i \models \psi$ *and* $\forall 0 \leqslant j < i, \mathcal{G}, \lambda_j \models \varphi$

$\mathcal{G}, v \models \langle\langle A\rangle\rangle \varphi \mathbf{R} \psi$ *iff* $\exists \boldsymbol{\sigma}_A \in \mathrm{Str}_A, \forall \lambda \in \mathrm{Out}(v, \boldsymbol{\sigma}_A) : \forall k \geqslant 0, \mathcal{G}, \lambda_k \models \psi$ *or* $\exists j \in [0, k)$ *s.t.* $\mathcal{G}, \lambda_j \models \varphi$

[1] We define ATL using $\langle\langle A\rangle\rangle \varphi \mathbf{R} \varphi$ as it is strictly more expressive than ATL defined using $\langle\langle A\rangle\rangle \mathbf{G}\varphi$ (Laroussinie, Markey, and Oreiby 2008).

$\mathcal{G}, v \models [\overline{\varphi : \alpha \to A}]^+_k \psi$ *iff* $(V, \ell, d', o), v \models \psi$, *where*
$\forall v' \in V, a \in \mathrm{Ag}, d'(a, v') =$
$= d(a, v') \cup upd^\pi(a, v')$

$\mathcal{G}, v \models [\overline{\varphi : \alpha \to A}]^-_k \psi$ *iff* $(V, \ell, d', o), v \models \psi$, *where*
$\forall v' \in V, a \in \mathrm{Ag}, d'(a, v') =$
$= d(a, v') \setminus upd^\pi(a, v')$
if $d(a, v') \not\subseteq upd^\pi(a, v')$, *and*
$d'(a, v') = d(a, v')$ *otherwise*

$\mathcal{G}, v \models has(a, \alpha)$ *iff* $\alpha \in d(a, v)$

where $upd^\pi(a, v') = \{\alpha_i \mid 1 \leqslant i \leqslant k, \mathcal{G}, v' \models \varphi_i$, *and* $a \in A_i\}$ *is the set of actions to be added to or removed from the set of available actions of agent* a *in state* v' *according to the given update* $\pi = \varphi_1 : \alpha_1 \to A_1, ..., \varphi_k : \alpha_k \to A_k$. *We will denote the new CGS* (V, ℓ, d', o) *as* $\mathcal{G}'$.

Given an update $[\varphi_1 : \alpha_1 \to A_1, ..., \varphi_k : \alpha_k \to A_k]^+$, we construct a new CGS (V, ℓ, d', o), where for each $1 \leqslant i \leqslant k$, we globally and simultaneously grant to agents in A_i the action α_i in states satisfying φ_i. For the action removing operator $[\varphi_1 : \alpha_1 \to A_1, ..., \varphi_k : \alpha_k \to A_k]^-$, we need to additionally check whether the simultaneous and global removal of actions from some agent a is possible, i.e., whether after such an update, agent a still has at least one action available. If yes, we proceed with updating the set of available actions of a, and if not, the agent retains *all* her actions[2]. The assumption that, after an update, an agent should have at least one action available is called *reasonableness* and is standard in reasoning about normative MAS (Ågotnes et al. 2007).

Example 1 (Oxygen tank)**.** *We draw inspiration from the classic example from (Bulling, Dix, and Jamroga 2010), and adapt it to our context. Imagine that two robots,* r_1 *and* r_2*, are transporting an oxygen tank in a circular environment. Their task is to deliver the tank safely to Bob who has a respiratory disorder. Robot* r_1 *can move to the left and right, but robot* r_2 *can only go left, as she cannot turn. Since the tank is heavy, they only succeed to bring the tank when moving in the same direction. The environment has three areas: Bob's room, the salon, and a balcony. The model, denoted* $\mathcal{G}^{Bob}$*, is depicted in Figure 1 (left). State* q_1 *represents the robots successfully bringing the oxygen tank to Bob's room, making the proposition* $atBob$ *true. States* q_0 *and* q_2 *correspond to the case where the robots instead bring the tank to the salon (their initial position) and the balcony, respectively. Since the temperature outside is warm, the proposition* $warm$ *is true at* q_2.

The robots can cooperate to bring the oxygen tank to Bob by moving together to the left twice, passing through the balcony. Thus, we have that $\mathcal{G}^{Bob}, q_0 \models \langle\!\langle \{r_1, r_2\} \rangle\!\rangle \mathbf{F}\, atBob$. *However, the robots cannot immediately bring it to Bob from* q_0 *(i.e., in one step). By granting the action* $right$ *to* r_2*, which she currently lacks, they can cooperate to bring the tank anywhere in one step (see Figure 1 (right)). Hence,*

$\mathcal{G}^{Bob}, q \models \neg has(r_2, right) \wedge$
$[\top : right \to \{r_2\}]^+ (has(r_2, right) \wedge \langle\!\langle \{r_1, r_2\} \rangle\!\rangle \mathbf{X}\, atBob)$

for any state q. *In other words, initially* r_2 *does not have action* $right$*, and after she acquires it universally, the robots always have a strategy to bring the tank to Bob in one step.*

It is advised not to expose the oxygen tank to warm temperatures and direct sunlight. Hence, we would then like to prevent the robots from going to the balcony before reaching Bob's room. This could be achieved by removing actions leading to the undesirable state. For instance, we have that

$\mathcal{G}^{Bob}, q_0 \models [\top : right \to \{r_2\}]^+ [\neg atBob : left \to \{r_1\}]^- \langle\!\langle \emptyset \rangle\!\rangle\, atBob\, \mathbf{R}\, \neg warm$

This means that, after the update, whatever the robots do, they are only able to go to the warm area after bringing the oxygen tank to Bob's room. The model obtained with the updates as specified in this formula is shown in Figure 2.

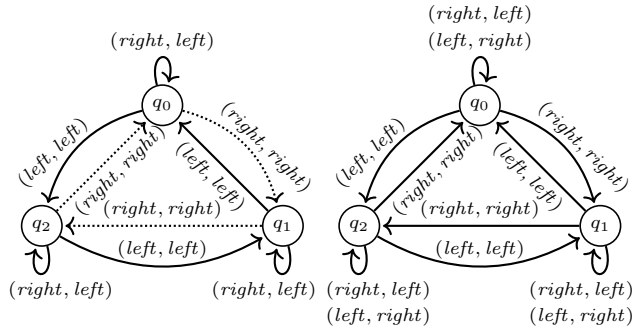

Figure 1: The CGS $\mathcal{G}^{Bob}$ (left), and its update in which the robot r_2 was granted the new action $right$ (right). Dotted arrows represent disabled transitions, which could be enabled if the robot is granted a new action.

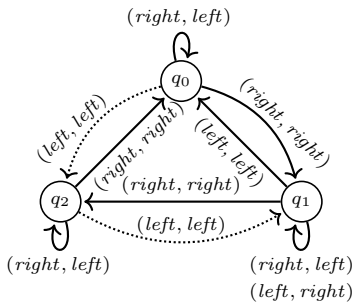

Figure 2: Update of the CGS $\mathcal{G}^{Bob}$ obtained by granting action $right$ to r_2 and removing action $left$ of r_1.

2.2 Expressivity

Expressivity of ATL-D and its Fragments In this section, we show that ATL-D is strictly more expressive than

[2]Note that we could also have defined the removing operator in such a way that if a removal is impossible for some agent a, we still remove *some* of her actions. In this case, however, we would need to specify an ordering over a's actions to decide which ones to remove. This variation is interesting for future work, but here, for simplicity, we do not distinguish between actions of a.

ATL, and we also make the relation between ATL-D$^+$ and ATL-D$^-$ precise.

Definition 5 (Expressivity)**.** *Let* L_1 *and* L_2 *be two languages, and let* $\varphi \in \mathsf{L}_1$ *and* $\psi \in \mathsf{L}_2$*. We say that* φ *and* ψ *are* equivalent*, when for all CGS* $\mathcal{G}$ *and state* v*, we have* $\mathcal{G}, v \models \varphi$ *if and only if* $\mathcal{G}, v \models \psi$*.*

If for every $\varphi \in \mathsf{L}_1$ *there is an equivalent* $\psi \in \mathsf{L}_2$*, we write* $\mathsf{L}_1 \preccurlyeq \mathsf{L}_2$ *and say that* L_2 *is* at least as expressive as L_1*. We write* $\mathsf{L}_1 \prec \mathsf{L}_2$ *iff* $\mathsf{L}_1 \preccurlyeq \mathsf{L}_2$ *and* $\mathsf{L}_2 \not\preccurlyeq \mathsf{L}_1$*, and we say that* L_2 *is* strictly more expressive than L_1*. Finally, if* $\mathsf{L}_1 \not\preccurlyeq \mathsf{L}_2$ *and* $\mathsf{L}_2 \not\preccurlyeq \mathsf{L}_1$*, we say that* L_1 *and* L_2 *are* incomparable*.*

Theorem 1. ATL $\prec$ ATL-D$^+$ *and* ATL-D$^+$ $\not\preccurlyeq$ ATL-D$^-$*.*

Proof. Consider formula $[\top : \beta \to \{a\}]^+\langle\langle\{a\}\rangle\rangle\mathbf{X}\neg p$ of ATL-D$^+$ and assume, for the sake of a contradiction, that there are equivalent formulas ψ of ATL and χ of ATL-D$^-$. Next, consider CGSs $\mathcal{G}_1$ and $\mathcal{G}_2$ with a single agent a and two actions $\{\alpha, \beta\}$ shown in Figure 3.

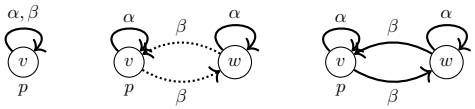

Figure 3: CGS $\mathcal{G}_1$ (left), $\mathcal{G}_2$ (middle), and $\mathcal{G}_3$ (right). Dotted arrows denote disabled transitions.

CGS $\mathcal{G}_1$ consists of a single state v satisfying p and with two self-loops for both actions α and β. In CGS $\mathcal{G}_2$, we have two states, v and w, with p being true only in v. Note that agent a does not have action β available to her in either state, i.e. β-transitions are disabled for her.

It is easy to see that $\mathcal{G}_1, v \not\models [\top : \beta \to \{a\}]^+\langle\langle\{a\}\rangle\rangle\mathbf{X}\neg p$, since a already has action β available to her, and no matter what she does, she cannot reach a $\neg p$-state as there is simply no such state in the CGS. At the same time, $\mathcal{G}_2, v \models [\top : \beta \to \{a\}]^+\langle\langle\{a\}\rangle\rangle\mathbf{X}\neg p$, as enabling action β for agent a in both states would results in $\mathcal{G}_3, v$, where $\langle\langle\{a\}\rangle\rangle\mathbf{X}\neg p$ holds.

Finally, $\mathcal{G}_1, v \models \psi$ iff $\mathcal{G}_2, v \models \psi$ due to the fact that no ATL formula can create a transition between the p-state v and $\neg p$-state w in $\mathcal{G}_2$. Hence, ATL-D$^+$ $\not\preccurlyeq$ ATL, and, together with the fact that ATL $\subseteq$ ATL-D$^+$, we get ATL $\prec$ ATL-D$^+$.

Similarly, $\mathcal{G}_1, v \models \chi$ iff $\mathcal{G}_2, v \models \chi$ as the only non-trivial removal is possible in $\mathcal{G}_1, v$: either action α or action β can be removed. In both outcomes, further non-trivial removals are not possible, and hence in both $\mathcal{G}_1, v$ and $\mathcal{G}_2, v$ the agent a can only force p. Thus, ATL-D$^+$ $\not\preccurlyeq$ ATL-D$^-$. □

Theorem 2. ATL $\prec$ ATL-D$^-$ *and* ATL-D$^-$ $\not\preccurlyeq$ ATL-D$^+$*.*

The theorem can be proved by taking $\mathcal{G}_3$ depicted in Figure 3, and $\mathcal{G}_4$, which is exactly like $\mathcal{G}_3$ with actions α and β swapped. Then we can target action β with an action removing update and show that the corresponding formula holds in one model and not in the other one. A proof sketch can be found in (Galimullin, Grosinger, and Mittelmann 2026).

Corollary 1. ATL-D$^+$ *and* ATL-D$^-$ *are incomparable,* ATL-D$^+$ $\prec$ ATL-D*, and* ATL-D$^-$ $\prec$ ATL-D*.*

The expressivity results are summarized in Figure 4.

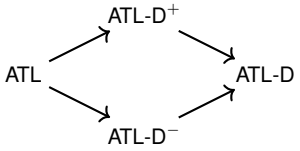

Figure 4: Overview of the expressivity results. An arrow from L_1 to L_2 means $\mathsf{L}_1 \prec \mathsf{L}_2$. The relation is transitive, and transitive arrows are omitted from the figure.

Relation to Logics for Normative Systems In this paper, we focus on a general approach to granting and removing agents' actions. Action removals have been explored before in the context of *normative* MAS, in which a *norm* or a *social law* prohibits certain transitions or actions (see (Alechina, Logan, and Dastani 2018) for an overview). Here we show how ATL-D$^-$ captures some intuitions from the literature on the topic.

Let us first consider the approach from (van der Hoek, Roberts, and Wooldridge 2007), in which, in the ATL setting, the authors consider, for a given CGS $\mathcal{G}$, *behavioural constraints* $\eta : \text{Ac} \times \text{Ag} \to 2^V$. If $v \in \eta(\alpha, a)$, then it means that agent a is not allowed to perform action α in state v. The *implementation of* η *in* $\mathcal{G}$ is defined as the new CGS $\mathcal{G} \dagger \eta$, in which all transitions that include prohibited actions are removed. Note that, as in our case, behavioural constraints must satisfy the reasonableness requirement. The pair (φ, η) is called a *social law*, where η is a behavioural constraint and φ is a goal formula.

Since behavioural constraints explicitly specify states where an action is prohibited, we cannot directly model them in ATL-D$^-$. The reason for this is that our updates are not tailored for a specific model, and instead, we use conditions expressed as formulas to pick out the necessary states. Indeed, the approach from (van der Hoek, Roberts, and Wooldridge 2007) allows, for example, removing action α from agent a in state v, while preserving the action in state v', even if v and v' are not distinguishable by any formula. We do not delve into the discussion whether such a situation is desirable, and instead note that, given a finite $\mathcal{G}$ and a goal formula φ, we can label each state of $\mathcal{G}$ with its own unique proposition that is not true in any other state of $\mathcal{G}$ and does not appear in φ. We denote such a CGS $\mathcal{G}^+$, and it is clear that $\mathcal{G}, v \models \varphi$ iff $\mathcal{G}^+, v \models \varphi$ for all v.

This modification allows us to target any particular state of a CGS, and hence model the effects of implementing a behavioural constraint η. In particular, we create an update $[\pi_\eta]^-$ by taking $\pi_\eta = p^0_v : \alpha_0 \to \{a_0\}, ..., p^n_v : \alpha_n \to \{a_n\}$ such that $v^i \in \eta(\alpha_i, a_i)$ for each $0 \leqslant i \leqslant n$, and where p^i_v is a new proposition that is true only in state v^i and not appearing in φ. Hence, we have the following result.

Theorem 3. *Let* (φ, η) *be a social law with* $\varphi \in$ ATL*, and* $\mathcal{G}$ *be a finite CGS. The social law is effective, i.e.* $\mathcal{G} \dagger \eta, v \models \langle\langle\emptyset\rangle\rangle\mathbf{G}\varphi$*, iff* $\mathcal{G}^+, v \models [\pi_\eta]^-\langle\langle\emptyset\rangle\rangle\mathbf{G}\varphi$*.*

Now, let us consider the approach from (Galimullin and Kuijer 2024), in which the authors, again in the ATL setting, consider partial functions $\zeta : \text{Ac} \times \text{Ag} \hookrightarrow \mathsf{L}$, called *atomic*

social laws, specifying conditions under which an action is allowed to be performed by an agent. The main difference from behavioural constraints is that atomic social laws specify the states in which an action is allowed using formulas of a language. For example, $\zeta(\alpha, a) = \varphi$ means that agent a is allowed to perform action α only in states satisfying φ. If for some a and α, $\zeta(\alpha, a)$ is undefined, the action is unconditionally allowed for the agent. Having a goal formula φ and an atomic social law ζ, we denote the fact that φ holds after updating the CGS $\mathcal{G}$ with ζ as $\mathcal{G}, v \models [\zeta]\varphi$.

Atomic social laws are closer to the updates of ATL-D$^-$ than behavioural constraints. For an atomic social law ζ, we construct $[\pi_\zeta]^-$ as follows: $\pi_\zeta = \neg\varphi_0 : \alpha_0 \to \{a_0\}, ..., \neg\varphi_n : \alpha_n \to \{a_n\}$ such that $\zeta(\alpha_i, a_i) = \varphi_i$ for each $0 \leqslant i \leqslant n$, i.e., if α_i is only allowed for agent a_i in φ_i-states, we remove this action from agent a_i in all $\neg\varphi_i$-states.

Theorem 4. *Given are a CGS $\mathcal{G}$, a state v, a goal formula $\varphi \in$ ATL, and an atomic social law ζ. It holds that $\mathcal{G}, v \models [\zeta]\varphi$ iff $\mathcal{G}, v \models [\pi_\zeta]^-\varphi$.*

Theorems 3 and 4 illustrate how ATL-D$^-$ can capture some key intuitions from normative reasoning research. The full exploration of the relationship between ATL-D and logics for normative reasoning in MAS is left for future work.

2.3 Computational Problems

Model Checking Now we turn our attention to the computational problems, and we start with model checking.

Definition 6 (Model checking)**.** *Given a finite CGS $\mathcal{G} = (V, \ell, d, o)$ and a formula $\varphi \in$ ATL-D, the* global model checking problem *consists in computing $\{v \in V \mid \mathcal{G}, v \models \varphi\}$, i.e. finding all the states of $\mathcal{G}$ that satisfy φ.*

To show that model checking ATL-D is in P, we present Algorithm 1 below, omitting the standard ATL cases.

The algorithm is an extension of the ATL global model-checking algorithm, which is computable in polynomial time. The original ATL algorithm employs the function $Pre(\mathcal{G}, A, X)$ to compute the set of states from which the coalition A can force an outcome in one of the X states. Function Pre is computable in polynomial time w.r.t. $|\mathcal{G}|$. In our case, the difference is that we may also compute Pre not only for the original model but for at most $|\varphi|$ updated models as well. Updated models are computed for cases $[\pi]^+\chi$ and $[\pi]^-\chi$, where we, following the definition of semantics, globally update actions of agents in a given CGS $\mathcal{G}$ by iteratively checking the lists $[\overline{\psi : \alpha \to A}]_k^+$ and $[\overline{\psi : \alpha \to A}]_k^-$.

To update models, we recursively call the procedure $\mathrm{MC}(\mathcal{G}, \varphi)$ at most $|V|^2 \cdot |\mathrm{Ag}| \cdot k \cdot |\varphi|$ times. As mentioned, each call takes at most polynomial time, and hence the whole algorithm terminates in polynomial time w.r.t. $|\mathcal{G}|$ and $|\varphi|$.

The P-hardness of the problem follows immediately from the P-completeness of model checking for ATL (a fragment of ATL-D).

Theorem 5. *Model checking ATL-D is P-complete.*

Bounded Update Existence Problem Next, we consider the bounded update existence problem, which asks whether, for a given CGS and a goal formula, there is a bounded sequence of updates that modifies the CGS so that it satisfies the goal. We show that this problem is NP-complete.

Algorithm 1 Global model checking for ATL-D

```
1:  procedure MC(G, φ)
2:      case φ = has(a, α)
3:          return {v ∈ V | α ∈ d(a, v)}
4:      case φ = [π]^+ χ with [π]^+ = [ψ : α → A]_k^+
5:          for v' ∈ V and a ∈ Ag do
6:              X(a, v') = ∅
7:              for i ∈ [1, ..., k] do
8:                  if v' ∈ MC(G, ψ_i) and a ∈ A_i then
9:                      X(a, v') ← X(a, v') ∪ {α_i}
10:             d'(a, v') ← d(a, v') ∪ X(a, v')
11:         return MC(G', χ)
12:     case φ = [π]^- χ with [π]^- = [ψ : α → A]_k^-
13:         for v' ∈ V and a ∈ Ag do
14:             X(a, v') = ∅
15:             for i ∈ [1, ..., k] do
16:                 if v' ∈ MC(G, ψ_i) and a ∈ A_i then
17:                     X(a, v') ← X(a, v') ∪ {α_i}
18:             if d(a, v') ⊈ X(a, v') then
19:                 d'(a, v') ← d(a, v') \ X(a, v')
20:             else
21:                 d'(a, v') ← d(a, v')
22:         return MC(G', χ)
```

Definition 7 (Bounded Update Existence Problem)**.** *Given a finite CGS $\mathcal{G}$, a state v, formula φ, and a natural number n, the bounded update existence problem consists in determining whether there is a sequence $[\overrightarrow{\pi}]$ of updates $[\pi]^+$ and $[\pi]^-$ with $|\overrightarrow{\pi}| \leqslant n$ such that $\mathcal{G}, v \models [\overrightarrow{\pi}]\varphi$.*

Considering the bounded variant of the update existence problem is important when we want *economical modifications* of a given system. Indeed, dramatic changes to a system are not always desirable or possible due to resource constraints. In these cases, parameter n may represent the upper bound on how much we are willing to change the system.

Theorem 6. *The bounded update existence problem is* NP*-complete.*

Proof sketch. We show that the problem is NP-complete for all of ATL-D, ATL-D$^-$, and ATL-D$^+$. First, that the problem is in NP for all the logics follows from the fact that for given $\mathcal{G}$, v, and φ, we can guess the sequence $[\overrightarrow{\pi}]$ such that $|[\overrightarrow{\pi}]\varphi|$ is polynomial with respect to $n + |\varphi| + |\mathcal{G}|$, and model check $\mathcal{G}, v \models [\overrightarrow{\pi}]\varphi$ in polynomial time.

To demonstrate that the problem is NP-hard for ATL-D$^+$, and hence ATL-D, we employ a reduction from the NP-complete 3SAT problem. For a given 3SAT instance Ψ, we create a CGS $\mathcal{G}_\Psi$ for one agent, where states correspond to clauses in Ψ, literals, atoms, and, finally, the truth-values of atoms. Then, for this constructed $\mathcal{G}_\Psi$, we provide an ATL formula φ_Ψ that forces the agent to unambiguously choose truth-values for atoms to satisfy the goal Ψ. We claim that Ψ is satisfiable if and only if there is an update $[\pi]^+$ such that once it is executed in $\mathcal{G}_\psi$, then φ_Ψ holds. Such an update

would choose which truth-values to assign to atoms in Ψ. See the detailed proof in (Galimullin, Grosinger, and Mittelmann 2026).

The NP-hardness for ATL-D$^-$ can be obtained by a modification of the proof for ATL-D$^+$, where we consider the same φ_Ψ, but in $\mathcal{G}_\Psi$ all actions are enabled. Thus, Ψ is satisfiable if and only if there is an update $[\pi]^-$ that disables some actions for the agent, allowing for an unambiguous assignment of truth-values to atoms to satisfy φ_Ψ. □

The NP-completeness of the bounded update existence problem for ATL-D echoes the NP-completeness result for (both bounded and unbounded, i.e. without a given n) social law existence problem (Galimullin and Kuijer 2024).

3 Epistemic ATL with Dynamic Actions

While in ATL-D we can reason about granting and removing actions to affect the agents' abilities, in order to increase or decrease the autonomy of the agents, we would like them to know facts about the environment, each other, and which actions they need to acquire or drop to achieve their goals. For example, knowing which actions they lack, the agents could then *proactively* request to acquire them. To this end, we introduce an *epistemic* extension of ATL-D.

3.1 Syntax and Semantics

Epistemic ATL-D (ATEL-D) is a dynamic extension of ATEL (van der Hoek and Wooldridge 2003; Jamroga and van der Hoek 2004), which is defined over CGSs enriched with relations specifying agents' indistinguishable states.

Definition 8 (iCGS)**.** *A* concurrent game structure with imperfect information *(iCGS) is a tuple* $\mathcal{G}_i = (\mathcal{G}, \{\sim_a\}_{a\in \mathrm{Ag}})$ *where* $\mathcal{G} = (V, \ell, d, o)$ *is a CGS, and* $\sim_a \subseteq V \times V$ *is an equivalence relation (called the* observation relation*) of agent* a *s.t. if* $v \sim_a v'$*, then* $d(a, v) = d(a, v')$*. We write* $\mathcal{G}_i = (\mathcal{G}, \sim)$ *whenever the relation is clear from the context.*

Note that a standard CGS is a special case of an iCGS, where for each agent a the relation $\sim_a$ is the identity relation. The requirement that agents have the same actions in indistinguishable states is standard in the ATEL literature[3].

Definition 9 (Uniform Strategies)**.** *A* memoryless uniform strategy *for agent* a *is a function* $\sigma_a : V \to \mathrm{Ac}$ *in which for all states* v, v' *s.t.* $v \sim_a v'$*, we have* $\sigma_a(v) = \sigma_a(v')$*. With a slight abuse of notation, we let* Str_a *be the set of memoryless uniform strategies for agent* a*, and* $\mathrm{Str}_A = \prod_{a\in A} \mathrm{Str}_a$.

Intuitively, uniform strategies assign the same actions in indistinguishable states. The assumption that agents know actions available to them has epistemic side effects in our setting. As we will see, granting a new action with a condition φ to a set of agents A, also lets the agents distinguish φ-states from $\neg\varphi$-states. This is somewhat similar to *semi-private announcements* in *Dynamic Epistemic Logic* DEL (van Ditmarsch, van der Hoek, and Kooi 2008), where a set of agents A may be informed whether φ holds, and all other agents know that A were informed whether φ but do not know themselves what exactly was communicated.

[3]For the discussion of this property see (Ågotnes 2006).

Here, instead, we consider a more general setting in which an update can also inform a (possibly empty) subset B of Ag of the action update for agents in A in φ-states. This general setting subsumes semi-private announcements by taking $B = \emptyset$. At the same time, it allows us to express *public* granting and removal of actions by taking $B = \mathrm{Ag}$, where *all* agents in a system are informed of the condition of an update. As an example, we can think of chess: when one player gets her pawn promoted, the other player knows it. Similarly, if the player loses a piece, it is common knowledge that she can no longer use it.

Finally, with our general setting, we can express situations 'in-between' semi-private and public action updates. This allows us to capture scenarios, where we would like to inform only a subset B of Ag that group A has received or lost an action. We can think of such groups A and B as teammates in a board game.

To capture all of these scenarios, we now present *Epistemic* ATL-D (ATEL-D).

Definition 10 (ATEL-D)**.** *Formulas* φ *of* ATEL-D *are defined by the following grammar:*

$$\varphi ::= p \mid \varphi \vee \varphi \mid \neg\varphi \mid \langle\langle A\rangle\rangle \mathbf{X}\varphi \mid \langle\langle A\rangle\rangle \varphi \mathbf{U} \varphi \mid \langle\langle A\rangle\rangle \varphi \mathbf{R} \varphi \mid [\pi]^+\varphi \mid [\pi]^-\varphi \mid has(a, \alpha) \mid K_a\varphi$$
$$\pi ::= \varphi : \alpha \to A, B \mid \pi, \pi$$

where $p \in \mathrm{AP}$*,* $\alpha \in \mathrm{Ac}$*,* $a \in \mathrm{Ag}$*, and* $A, B \in 2^{\mathrm{Ag}}$.

Constructs $K_a\varphi$ are read as 'agent a knows that φ'. The new updates $[\varphi : \alpha \to A, B]^+$ (resp. $[\varphi : \alpha \to A, B]^-$) mean that in all states satisfying φ, agents in A are granted action α (resp. get action α removed) and both the agents in A and in B are informed of φ, i.e., they are informed of the the action update's precondition.

Definition 11 (Semantics ATEL-D)**.** *Let* φ *be an* ATEL-D *formula,* $\mathcal{G}_i = (\mathcal{G}, \sim)$ *be an iCGS and* v *be a state. The satisfaction of* φ *on* v *in* $\mathcal{G}_i$ *is denoted* $\mathcal{G}_i, v \models \varphi$ *and is defined recursively as follows (where we omit the cases that are analogous to* ATL-D *and the case for removing actions as it is analogous to the case of adding actions):*

$\mathcal{G}_i, v \models K_a\varphi$ *iff* $\forall v' \in V : v \sim_a v'$ *implies* $\mathcal{G}, v' \models \varphi$

$\mathcal{G}_i, v \models \overline{[\varphi : \alpha \to A, B]}^+_k \psi$ *iff* $(\mathcal{G}', \sim'), v \models \psi$, *where* $\mathcal{G}'$ *as in Def. 4, and* $\forall a \in \mathrm{Ag}, \sim'_a = \sim_a \cap \sim_{\varphi_i}$ *if* $a \in A_i \cup B_i$*, and* $\sim'_a = \sim_a$ *otherwise*

where $\sim_\varphi := (\llbracket\varphi\rrbracket_{\mathcal{G}_i} \times \llbracket\varphi\rrbracket_{\mathcal{G}_i}) \cup (\llbracket\neg\varphi\rrbracket_{\mathcal{G}_i} \times \llbracket\neg\varphi\rrbracket_{\mathcal{G}_i})$ *is a* φ-partition *of the* $\mathcal{G}_i$ *with* $\llbracket\varphi\rrbracket_{\mathcal{G}_i} = \{v \in V \mid \mathcal{G}_i, v \models \varphi\}$.

Intuitively, an update $\overline{[\varphi : \alpha \to A, B]}^+_k$ (resp. $\overline{[\varphi : \alpha \to A, B]}^-_k$) globally and simultaneously grants (resp. removes) actions for agents in A_i in states that satisfy the condition φ_i. This is exactly as in Definition 4. Additionally, we inform agents in $A_i \cup B_i$ of the action update's precondition, i.e., the truth value of φ_i. To model

this, we consider the φ_i-partition[4] of the given iCGS that divides the states of the iCGS into those that satisfy φ_i and those that do not. Then, for each agent $a \in A_i \cup B_i$ that should be informed of φ_i, we 'cut' the a relations $\sim_a$ that connect φ_i- and $\neg\varphi_i$-states. In this way, agents can become less uncertain about which state they are in.

As mentioned earlier, with the action updates of ATEL-D, we can capture a plethora of action update scenarios and their epistemic consequences. For example, the ATL-D updates $[\varphi : \alpha \to A]^+$ and $[\varphi : \alpha \to A]^-$ are captured by the ATEL-D updates $[\varphi : \alpha \to A, \emptyset]^+$ and $[\varphi : \alpha \to A, \emptyset]^-$ correspondingly. Agents in A are still informed about φ as the classic assumption of iCGS is that agents know which actions are available to them, and hence have no uncertainty between the states in which they have different sets of available actions. This is exactly the setting of semi-private action granting or removing.

Similarly, fully public granting or removal of actions can be captured by $[\varphi : \alpha \to A, \mathrm{Ag}]^+$ and $[\varphi : \alpha \to A, \mathrm{Ag}]^-$, where every agent is informed about φ. Finally, we can also capture *public announcements*[5] by using, e.g., $[\varphi : \alpha \to \emptyset, \mathrm{Ag}]^+$, where agents' actions remain intact and everyone publicly and simultaneously learns the truth-value of φ.

Example 2 (Oxygen tank, continued)**.** *Let us now assume that the robots only have partial observation and may be unsure about their current location. Suppose that robot r_2 is equipped with a temperature sensor and can therefore distinguish the inside areas (Bob's room, i.e. state q_1, and the salon, i.e. state q_0) from the balcony (state q_2), which is warm. Robot r_1, instead, detects the Wi-Fi signal strength, which is only strong in Bob's room. The model representing this situation is denoted $\mathcal{G}_i^{Bob}$ and appears in Figure 5 (left).*

Similarly to Example 1, the robots cannot reach Bob's room from the salon in one step, but now they both know it (there is no uniform strategy to force the outcome to satisfy $atBob$ from all states the robots consider indistinguishable), i.e., $\mathcal{G}_i^{Bob}, q_0 \models \neg\langle\!\langle \{r_1, r_2\} \rangle\!\rangle \mathbf{X}\, atBob \wedge K_{r_1} \neg\langle\!\langle \{r_1, r_2\} \rangle\!\rangle \mathbf{X}\, atBob \wedge K_{r_2} \neg\langle\!\langle \{r_1, r_2\} \rangle\!\rangle \mathbf{X}\, atBob$. We could try to rectify it by the same granting action update as in Example 1, while informing both agents about the precondition $\top$. This update, however, which results in the model similar to the one on the left of Figure 5 but with all transitions enabled, is still not enough to guarantee that the robots can reach Bob's room in one step. This is due to the fact that their observability relations after the announcement would remain the same ($\top$ holds in all states and hence it's not informative), and they would still be uncertain whether to move to the right or the left (i.e., they have no uniform strategy). Thus, $\mathcal{G}_i^{Bob}, q_0 \not\models [\top : right \to \{r_2\}, \mathrm{Ag}]^+ \langle\!\langle \{r_1, r_2\} \rangle\!\rangle \mathbf{X}\, atBob$.

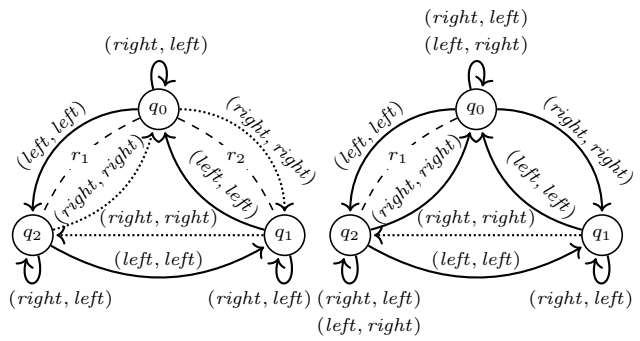

Figure 5: The CGS $\mathcal{G}_i^{Bob}$ with imperfect information is shown on the left, and its update after publicly granting the action $right$ to r_2 whenever the robots are not in Bob's room is shown in the right. Dashed lines labelled by an agent's name represent her observation relation (reflexive cases are omitted).

We can now try to grant the action $right$ to r_2 in states other than Bob's room. The resulting model is shown in Figure 5 (right). Since r_2 is informed of the precondition for her newly granted actions, she now knows that she is in the salon. However, note that it is still not enough to enable the robots to reach Bob's room in one step as robot r_1 is still uncertain whether she should go right (if in state q_0) or left (if in state q_2). Hence, there is no uniform strategy for the robots to achieve their objective, i.e., $\mathcal{G}_i^{Bob}, q_0 \not\models [\neg atBob : right \to \{r_2\}, \emptyset]^+ \langle\!\langle \{r_1, r_2\} \rangle\!\rangle \mathbf{X}\, atBob$. To allow the agents to achieve their goal, we can strengthen the precondition to $\neg atBob \wedge \neg warm$ and inform also r_1 about this. Since this formula holds only in state q_0, the resulting model (which is like in Figure 5 (right) but with the observability relation being the identity and transitions $(right, right)$ and $(left, right)$ from q_2 disabled) would inform both robots that they are in state q_0, from which they have a uniform strategy to reach Bob's room in one step. In particular, $\mathcal{G}_i^{Bob}, q \models [\neg atBob \wedge \neg warm : right \to \{r_2\}, \{r_1\}]^+ \langle\!\langle \{r_1, r_2\} \rangle\!\rangle \mathbf{X}\, atBob$ holds in all states q. This in particular implies that in the initial situation, while both robots know that they cannot reach Bob's room from the salon, they also know that after this particular update, they would be able to achieve their goal, i.e., $\mathcal{G}_i^{Bob}, q_0 \models K_r \neg\langle\!\langle \{r_1, r_2\} \rangle\!\rangle \mathbf{X}\, atBob \wedge K_r[\neg atBob \wedge \neg warm : right \to \{r_2\}, \{r_1\}]^+ \langle\!\langle \{r_1, r_2\} \rangle\!\rangle \mathbf{X}\, atBob$ for $r \in \{r_1, r_2\}$. Hence, each of the robots, based on their knowledge, could proactively request the system administrator to perform the update in order to allow the robots to reach Bob's room in one step.

3.2 Model Checking and Expressivity

ATEL-D extends ATEL with imperfect information and memoryless strategies (imperfect recall). It is known that model checking ATL with imperfect information and

[4] Such φ-partitions are inspired by the logical semantics of questions (see (Groenendijk and Stokhof 1997) for an overview, (van Benthem and Minica 2012) for the DEL approach to questions, and (Feinmann 2025) for a related discussion). The particular relation $\sim_\varphi$ has been recently used to reason about topic-based communication between agents in DEL (Velázquez-Quesada 2022; Galimullin and Velázquez-Quesada 2025).

[5] Our setting differs from the classic public announcements *à la* (Plaza 2007), in which a public announcement of φ leads to an updated model with all $\neg\varphi$-states removed. Our case is closer to an alternative variation of public announcements that instead of removing states remove transitions (see more on this in (Gerbrandy and Groeneveld 1997; van Benthem and Liu 2007)).

memoryless strategies, and hence ATEL, is Δ_2^{P}-complete[6] (Schobbens 2003; Jamroga and Dix 2006). The algorithm works in a bottom-up way starting from simple subformulas $\langle\langle A\rangle\rangle\varphi$, where φ does not contain strategic modalities. For A, the algorithm guesses a uniform strategy of at most polynomial size and removes from the given model transitions that will never be used according to the guessed strategy. Finally, in the resulting model, it is enough to check whether, for all paths, φ is satisfied. This can be done by the standard labelling algorithm for *Computation Tree Logic* (CTL) (Clarke and Emerson 1981). The number of such guesses of strategies is linear.

The algorithm needs major changes for ATEL-D, as verification of a formula $\langle\langle A\rangle\rangle\varphi$ should occur within the updates whose scope contains the formula. In fact, such updates may have changed the available strategies for A by the time we need to verify $\langle\langle A\rangle\rangle\varphi$. So, we must ensure that the effects of updates are known before the algorithm handles formulas within their scope. To this end, following (Kuijer 2015), we preprocess the list of subformulas of a given formula.

Given a $\psi \in$ ATEL-D, we let $Sub(\psi)$ be the list of subformulas of ψ as well as all modalities $[\pi]^+$ and $[\pi]^-$ appearing in ψ. We label all elements of $Sub(\psi)$ by the sequence of updates, possibly empty, in the scope of which a given subformula or update modality appears. Finally, we order the list[7] by first taking update modalities that are not in the scope of other updates, and ordering their subformulas in the increasing order including the modalities themselves. Then, we proceed with update modalities that are in the scope of the already processed updates. Finally, we take the remaining subformulas and order them in the increasing order. For example, take $\psi = [p : \alpha \to \{a\}, \{b\}]^+[p : \alpha \to \{b\}, \{a\}]^- K_a\langle\langle\{b\}\rangle\rangle\mathbf{X}q$, where we abbreviate the first update as $[\pi]^+$ and the second update as $[\pi]^-$. The ordered list $Sub(\psi)$ is $\{p, [\pi]^+, p^{[\pi]^+}, [\pi]^{-[\pi]^+}, q^{[\pi]^+,[\pi]^-}, (\langle\langle\{b\}\rangle\rangle\mathbf{X}q)^{[\pi]^+,[\pi]^-}, (K_a\langle\langle\{b\}\rangle\rangle\mathbf{X}q)^{[\pi]^+,[\pi]^-}, ([\pi]^- K_a\langle\langle\{b\}\rangle\rangle\mathbf{X}q)^{[\pi]^+}, \psi\}$. Such a preprocessing guarantees that a subformula in a scope of an update is evaluated after we know the effects of the update.

Having prepared the labelled list of subformulas and update modalities, we extend the CTL labelling algorithm by labelling not only states but actions, transitions, and the observation relation as well. Thus, we label the observation relations $(v, v') \in\sim_a$ with a string of updates σ to denote that the relation between states v and v' for agent a is still present in the model after the sequence of updates σ. Similarly, we label actions of each agent at each state with $(\sigma; +)$ if the corresponding action is available to the agent in the corresponding state, and with $(\sigma; -)$ otherwise. We require this labelling to keep track of which actions are available after the sequence of updates σ.

The model checking algorithm for ATEL-D (Algorithm 2) represents the availability function $d : \mathrm{Ag} \times V \to 2^{\mathrm{Ac}} \setminus \emptyset$ as a set of all possible triples (a, v, α). We label a triple (a, v, α) with $(\epsilon; +)$ if $\alpha \in d(a, v)$, and $(\epsilon; -)$ otherwise, where ϵ is an empty string. Observe that having all triples explicitly leads to at most a polynomial increase in size.

Now we turn to the algorithm, which loops over all elements of the ordered labelled list $Sub(\varphi)$ and all states v. We write $\pm(a, v, \alpha)^\sigma$ to refer to the second element of the labelling $(\sigma; \bullet)$ of (a, v, α), i.e., to either $+$ or $-$. In the algorithm, we focus on updates, and we omit the Boolean and $has(a, \alpha)$ cases as they are straightforward.

The algorithm follows the definition of the semantics. For

Algorithm 2 Global model checking for ATEL-D

case $\psi^\sigma = (K_a\chi)^\sigma$
  $check \leftarrow true$
  for $(v, w)^\sigma \in\sim_a$ **do**
    if w is not labelled with χ^σ **then**
      $check \leftarrow false$; **break**
  if $check$ **then**
    label v with $(K_a\chi)^\sigma$
case $\psi^\sigma = ([\pi]^\bullet\chi)^\sigma$, where $\bullet \in \{+, -\}$
  if v is labelled with $\chi^{\sigma,[\pi]^\bullet}$ **then**
    label v with $([\pi]^\bullet\chi)^\sigma$
case $\psi^\sigma = ([\pi]^+)^\sigma$ with $[\pi]^+ = [\overline{\chi : \alpha \to A, B}]^+_k$
  for $a \in$ Ag **do**
    $X(a, v) = \{(v, w) \in\sim_a |\ (v, w)$ is labelled with $\sigma\}$
  for $i \in [1, ..., k]$ **do**
    if v is labelled with χ^σ_i **then**
      for $a \in A_i$ **do**
        label (a, v, α_i) with $(\sigma, [\pi]^+; +)$
      for $a \in A_i \cup B_i$ and $(v, w)^\sigma \in\sim_a$ **do**
        if not (v is labelled χ^σ_i iff w is labelled χ^σ_i) **then**
          $X(a, v) \leftarrow X(a, v) \setminus (v, w)^\sigma$
  for $a \in$ Ag and $(v, w)^\sigma \in X(a, v)$ **do**
    label (v, w) with $\sigma, [\pi]^+$
  for $a \in$ Ag and $\alpha \in$ Ac **do**
    if (a, v, α) is not labelled with $(\sigma, [\pi]^+; +)$ **then**
      label (a, v, α) with $(\sigma, [\pi]^+; \pm(a, v, \alpha)^\sigma)$
case $\psi^\sigma = ([\pi]^-)^\sigma$ with $[\pi]^- = [\overline{\chi : \alpha \to A, B}]^-_k$
  for $a \in$ Ag **do**
    $X(a, v) = \{(v, w) \in\sim_a |\ (v, w)$ is labelled with $\sigma\}$
    $Y(a, v) = \emptyset$
  for $i \in [1, ..., k]$ **do**
    if v is labelled with χ^σ_i **then**
      for $a \in A_i$ **do**
        $Y(a, v) \leftarrow Y(a, v) \cup \{\alpha_i\}$
      for $a \in A_i \cup B_i$ and $(v, w)^\sigma \in\sim_a$ **do**
        if not (v is labelled χ^σ_i iff w is labelled χ^σ_i) **then**
          $X(a, v) \leftarrow X(a, v) \setminus (v, w)^\sigma$
  for $a \in$ Ag and $(v, w)^\sigma \in X(a, v)$ **do**
    label (v, w) with $\sigma, [\pi]^-$
  $Z(a, v) = \{\alpha \mid (a, v, \alpha)$ is labelled $(\sigma; +)\}$
  if $Z(a, v) \not\subseteq Y(a, v)$ **then**
    for $(a, v, \alpha) \in Y(a, v)$ **do**
      label (a, v, α) with $(\sigma, [\pi]^-; -)$
  for $a \in$ Ag and $\alpha \in$ Ac **do**
    if (a, v, α) is not labelled with $(\sigma, [\pi]^-; -)$ **then**
      label (a, v, α) with $(\sigma, [\pi]^-; \pm(a, v, \alpha)^\sigma)$

[6] $\Delta_2^{\mathrm{P}} = \mathrm{P}^{\mathrm{NP}}$ is a class of problems solvable in polynomial time with access to an NP oracle (Papadimitriou 1994, Chapter 17).

[7] Here we omit the exact details for brevity, but our ordering is quite similar to the ones used in (Kuijer 2015; Galimullin and Velázquez-Quesada 2025).

the case $([\pi]^\bullet\chi)^\sigma$ (with $\bullet \in \{+,-\}$), we label the current state with $([\pi]^\bullet\chi)^\sigma$ if it is already labelled with $\chi^{\sigma,[\pi]^\bullet}$, i.e., if χ holds in the state after the sequence of updates $\sigma, [\pi]^\bullet$.

For the case $\psi^\sigma = ([\pi]^+)^\sigma$, we first create a set $X(a,v)$ for each agent (line 13) to store the pairs $(v,w) \in \sim_a$ that are still present in the model after the sequence of updates σ. These sets keep track of how the observability for agent a in state v may change as a result of update $[\pi]^+$. Then, for each element of $[\pi]^+$, if state v satisfies precondition χ_i^σ, we label triples (a,v,α_i) with $(\sigma,[\pi]^+;+)$ for each agent in $a \in A_i$, to denote that agent a has action α_i available to her in state v after the sequence of updates $\sigma,[\pi]^+$ (lines 14–17). Also, for a given element of $[\pi]^+$, for agents in $A_i \cup B_i$ we 'cut' those pairs $(v,w)^\sigma$ for which v and w differ on the labelling of χ_i^σ (lines 18–20). After we have gone over elements of $[\pi]^+$, we label pairs $(v,w)^\sigma \in X(a,v)$ that 'survived' the update with $\sigma,[\pi]^+$ (lines 21–22). Actions of agents that are not affected by $[\pi]^+$ keep their $+$ or $-$ signs (lines 23–25).

The case $\psi^\sigma = ([\pi]^-)^\sigma$ is similar, with additional sets $Y(a,v)$ and $Z(a,v)$ to check that we do not remove all available actions from an agent. If we do not, then we label all the actions that are removed after the update with $(\sigma,[\pi]^-;-)$ (lines 39–42), and if we do, we, following the semantics, preserve all current actions of agent a.

Finally, with such a modification of the labelling algorithm, the rest follows the bottom-up approach from (Schobbens 2003) described above with the only difference being that, given $(\langle\!\langle A\rangle\!\rangle\varphi)^\sigma$, where φ does not contain strategic modalities, we guess a uniform strategy w.r.t. to $\sim^\sigma$ such that for all agents $a \in A$, states v, and actions α, the corresponding triple (a,v,α) is labelled with $(\sigma;+)$. In this way we ensure that the evaluation of $(\langle\!\langle A\rangle\!\rangle\varphi)^\sigma$ takes into account the updates σ that may have affected strategies of agents.

The preparation of $Sub(\psi)$ can be done in polynomial time w.r.t. $|\psi|$, and Algorithm 2 spends at most polynomial amount of time w.r.t. to $|\psi|$ and $|\mathcal{G}_i|$. For the strategic cases, we call the NP oracle at most polynomial number of times.

Theorem 7. *Model checking* ATEL-D *is* Δ_2^{P}*-complete.*

As a final remark of this section, we would like to mention that all the expressivity results for ATL-D hold, *mutatis mutandis*, for ATEL-D. This follows from the fact that in proofs of theorems in Section 2.2, every CGS can be considered as an iCGS with the identity observability relation. Thus, we have for ATEL, ATEL-D, ATEL-D $^-$, and ATEL-D $^+$ analogous expressivity results as in Corollary 1.

4 Related Work and Discussion

Normative Systems In Section 2.2, we showed how ATL-D$^-$ captures some intuitions behind normative reasoning. Other related work on normative updates includes (Shoham and Tennenholtz 1995; Ågotnes, van der Hoek, and Wooldridge 2010; Alechina, Dastani, and Logan 2013; Bulling and Dastani 2016; Alechina et al. 2022; van Ditmarsch, Kuijer, and Liu 2023). In comparison, our approach is more general and allows not only removing actions, but also adding them, as well as reasoning about the epistemic outcomes of such operations.

Dynamics of Ability Reasoning about updates of CGSs has been argued for in (Galimullin et al. 2026). The updates studied in (Galimullin et al. 2025), while being quite general, neither allow updating the set of agents' actions nor consider the imperfect information setting. Whereas in our logics the update of agents' abilities (in the sense of the power to force a certain outcome) may happen as a result of action updates, in (Galimullin and Ågotnes 2021; Galimullin and Ågotnes 2022) the authors consider direct updates of coalitional powers. Thus, in (Galimullin and Ågotnes 2021), the authors consider the mechanism of granting *new* dictatorial actions to a *single* agent. In (Galimullin and Ågotnes 2022), the authors use more complex action models that allow redirecting transitions in a given CGS. However, action models can also create new states and hence the logic has a higher model checking complexity compared to ATL-D. In both works, strategic reasoning is restricted to the next-time fragment of ATL, and thus lacks the tools for reasoning about extended consequences of updates.

A series of works on *obstruction logics* (OL) (see, e.g., (Catta, Leneutre, and Malvone 2023; Catta et al. 2024; Catta et al. 2025; Catta, Galimullin, and Mittelmann 2026)) explores dynamic games between an agent that moves along a model and an agent or coalition that is able to modify the transitions of the model. Compared to various OL's, where model changes are implicit and in the scope of a quantifier, both ATL-D and ATEL-D treat model updates as first-class citizens, which allows reasoning about different sequences of updates explicitly within a single formula. Finally, models of OL's are weighted graphs, and the addition and removal of transitions are parametrised by their total cost. This is quite different from our setting, where we have no costs, and agents' actions parametrise changes in transitions.

Capacity ATL and ATEL A recent line of work (Ballot et al. 2024; Ballot et al. 2025; Ballot 2025) introduces Capacity ATL and Capacity ATEL in which models incorporate capacities as sets of actions for each agent. The logics can reason about an agent's knowledge of her capacity, but not about granting or revoking capacities. As the authors of (Ballot et al. 2025) themselves suggest for future work, capacity mappings could be expressed by logical formulas, which is similar in spirit to our approach here.

Finally, compared to the related works discussed in this Section, ATEL-D also captures the epistemic consequences of granting and revoking actions, and also can express a variety of information-changing events from DEL.

A Note on Semantics In our definition of the semantics of ATEL-D, an agent is informed of a precondition φ of an update $[\pi]^+$ even if she already has the action we are trying to grant. She is also informed about a precondition, even if an action removal is unsuccessful (see discussion after Definition 4). We could have alternatively defined the semantics such that the agent is not explicitly informed of the precondition and instead *infers* which state she is in as a result of receiving a new action. To capture this, we need to substitute $\sim_\varphi$ in Definition 11 with $\sim_\varphi^{\alpha,a} := (\llbracket\varphi\rrbracket_{\mathcal{G}_i} \times \llbracket\varphi\rrbracket_{\mathcal{G}_i}) \cup (\llbracket\neg\varphi\rrbracket_{\mathcal{G}_i} \times \llbracket\neg\varphi\rrbracket_{\mathcal{G}_i}) \cup (\llbracket has(a,\alpha)\rrbracket_{\mathcal{G}_i} \times \llbracket has(a,\alpha)\rrbracket_{\mathcal{G}_i})$ for a given action α. Then $\forall a \in \mathrm{Ag}, \sim'_a = \sim_a \cap \sim_{\varphi_i}^{\alpha_i,a}$ if

$a \in A_i \cup B_i$. With such a change, all of the technical results would also hold with minor modifications. An advantage of explicitly informing the agents of a precondition (as in Definition 11) is that it allows us to express a greater range of epistemic updates, like public announcements that cannot be generally captured by the alternative 'inferring' semantics.

Discussion We have introduced logics ATL-D and ATEL-D to reason about the dynamics of granting and revoking actions to and from agents in perfect and imperfect information settings. We have studied the logics' expressivity relative to their fragments and ATL and ATEL. We have also demonstrated that the model checking problems for ATL-D and ATEL-D are P-complete and Δ_2^{P}-complete, respectively, i.e., no harder than for ATL and ATEL. For ATL-D, we have also considered the bounded update existence problem and proved that it is NP-complete.

There is a plethora of interesting directions for further research. The most immediate open technical questions are the satisfiability of the logics and the complexity of the update existence problem for ATEL-D. We find it also tempting to study the validities of the logics and provide their axiomatisations. Already now we can point out that since the observability relation for ATEL-D is an equivalence, we get all the validities of the classic epistemic logic (Fagin et al. 2003). We also have, e.g., the validity $K_a has(a, \alpha) \leftrightarrow has(a, \alpha)$, which follows directly from the definition of an iCGS. We can additionally consider group knowledge operators like distributed and common knowledge (Fagin et al. 2003) to reason, e.g., in the case of distributed knowledge about the situations when agents can pool their knowledge together to infer which actions they collectively lack. It is also intriguing to relax the assumption that we model *knowledge* of agents in a system and consider *beliefs* instead. Then, we can explore the application of belief revision, including the classic AGM theory (Alchourrón, Gärdenfors, and Makinson 1985), and, in particular, DEL-style belief revision (Liu 2011), to our setting.

On a conceptual level, based on our work, we can move towards *proactively learning agents* (Grosinger 2025), where an agent could decide on her own which actions to learn (grant to herself) and which to unlearn (remove) when no longer needed. Another interesting direction is to explore action updates engendered by agents themselves, capturing the situations when agents with different abilities (possibly on different levels of a hierarchy) can grant each other actions or prohibit other agents' undesirable behaviours. Moreover, in our setting, we assumed that all actions possible in a given system are specified in advance. An intriguing avenue of further research is to allow granting *brand new* actions to agents in such a way that goes beyond the dictatorial setting of (Galimullin and Ågotnes 2021). Finally, we also find it interesting to chart the full expressivity landscape of dynamic strategic logics from the literature by providing a comprehensive formal comparison of ATL-D and ATEL-D with the logics mentioned above.

AI Declaration

The authors have not employed any Generative AI tools.

Acknowledgements

This research is funded by the Swedish Research Council (Vetenskapsrådet), by projects 2021-05542 and 2023-04349, on H.J. Grosinger's part.

References

Ågotnes, T.; van der Hoek, W.; Rodríguez-Aguilar, J. A.; Sierra, C.; and Wooldridge, M. J. 2007. On the logic of normative systems. In Veloso, M. M., ed., *Proceedings of the 20th IJCAI*, 1175–1180.

Ågotnes, T.; van der Hoek, W.; and Wooldridge, M. J. 2010. Robust normative systems and a logic of norm compliance. *Logic Journal of the IGPL* 18(1):4–30.

Ågotnes, T. 2006. Action and knowledge in alternating-time temporal logic. *Synthese* 149(2):375–407.

Alchourrón, C. E.; Gärdenfors, P.; and Makinson, D. 1985. On the logic of theory change: Partial meet contraction and revision functions. *Journal of Symbolic Logic* 50(2):510–530.

Alechina, N.; Giacomo, G. D.; Logan, B.; and Perelli, G. 2022. Automatic synthesis of dynamic norms for multi-agent systems. In Kern-Isberner, G.; Lakemeyer, G.; and Meyer, T., eds., *Proceedings of the 19th KR*, 12–21.

Alechina, N.; Dastani, M.; and Logan, B. 2013. Reasoning about normative update. In Rossi, F., ed., *Proceedings of the 23rd IJCAI*, 20–26. IJCAI/AAAI.

Alechina, N.; Logan, B.; and Dastani, M. 2018. Modeling norm specification and verification in multiagent systems. *FLAP* 5(2):457–490.

Alur, R.; Henzinger, T. A.; and Kupferman, O. 2002. Alternating-time temporal logic. *Journal of the ACM* 49:672–713.

Ballot, G.; Malvone, V.; Leneutre, J.; and Laarouchi, Y. 2024. Strategic reasoning under capacity-constrained agents. In Dastani, M.; Sichman, J. S.; Alechina, N.; and Dignum, V., eds., *Proceedings of the 23rd AAMAS*, 123–131. IFAAMAS / ACM.

Ballot, G.; Malvone, V.; Leneutre, J.; and Ma, J. 2025. Strategic reasoning with capacity-constrained agents and imperfect information. In Lynce, I.; Murano, N.; Vallati, M.; Villata, S.; Chesani, F.; Milano, M.; Omicini, A.; and Dastani, M., eds., *Proceedings of the 28th ECAI*, volume 413 of *Frontiers in Artificial Intelligence and Applications*, 1599–1606. IOS Press.

Ballot, G. 2025. *Optimal Active Cyber Defence Strategies based on Multi-Agent System Verification*. Ph.D. Dissertation, Polytechnic Institute of Paris, Palaiseau, France.

Bulling, N., and Dastani, M. 2016. Norm-based mechanism design. *Artificial Intelligence* 239:97–142.

Bulling, N.; Dix, J.; and Jamroga, W. 2010. Model checking logics of strategic ability: Complexity. In Dastani, M.; Hindriks, K. V.; and Meyer, J.-J. C., eds., *Specification and Verification of Multi-Agent Systems*. Springer. 125–159.

Catta, D.; Leneutre, J.; Malvone, V.; and Murano, A. 2024. Obstruction alternating-time temporal logic: A strategic

logic to reason about dynamic models. In Dastani, M.; Sichman, J. S.; Alechina, N.; and Dignum, V., eds., *Proceedings of the 23rd AAMAS*, 271–280. IFAAMAS / ACM.

Catta, D.; Leneutre, J.; Malvone, V.; and Ortiz, J. 2025. Coalition obstruction temporal logic: A new obstruction logic to reason about demon coalitions. In Kwok, J., ed., *Proceedings of the 34th IJCAI*, 21–28. ijcai.org.

Catta, D.; Galimullin, R.; and Mittelmann, M. 2026. On angels and demons: Strategic (de)construction of dynamic models. In Amato, C.; Dennis, L.; Mascardi, V.; and Thangarajah, J., eds., *Proceedings of the 25th AAMAS*. To appear.

Catta, D.; Leneutre, J.; and Malvone, V. 2023. Obstruction logic: A strategic temporal logic to reason about dynamic game models. In Gal, K.; Nowé, A.; Nalepa, G. J.; Fairstein, R.; and Radulescu, R., eds., *Proceedings of the 26th ECAI*, volume 372 of *Frontiers in Artificial Intelligence and Applications*, 365–372. IOS Press.

Clarke, E. M., and Emerson, E. A. 1981. Design and synthesis of synchronization skeletons using branching-time temporal logic. In Kozen, D., ed., *Logics of Programs*, volume 131 of *LNCS*, 52–71. Springer.

Ezekiel, J.; Lomuscio, A.; Molnar, L.; and Veres, S. M. 2011. Verifying fault tolerance and self-diagnosability of an autonomous underwater vehicle. In Walsh, T., ed., *Proceedings of the 22nd IJCAI*, 1659–1664. IJCAI/AAAI.

Fagin, R.; Halpern, J. Y.; Moses, Y.; and Vardi, M. 2003. *Reasoning About Knowledge*. MIT Press.

Feinmann, D. 2025. A note on relevance. *Journal of Semantics*. To appear.

Ferrando, A., and Malvone, V. 2025. VITAMIN: A compositional framework for model checking of multi-agent systems. In Rocha, A. P.; Steels, L.; and van den Herik, H. J., eds., *Proceedings of the 17th ICAART*, 648–655. SCITEPRESS.

Galimullin, R., and Ågotnes, T. 2021. Dynamic coalition logic: Granting and revoking dictatorial powers. In Ghosh, S., and Icard, T., eds., *Proceedings of the 8th LORI*, volume 13039 of *LNCS*, 88–101. Springer.

Galimullin, R., and Ågotnes, T. 2022. Action models for coalition logic. In Areces, C., and Costa, D., eds., *Proceedings of the 4th DaLí*, volume 13780 of *LNCS*, 73–89. Springer.

Galimullin, R., and Kuijer, L. B. 2024. Synthesizing social laws with ATL conditions. In Dastani, M.; Sichman, J. S.; Alechina, N.; and Dignum, V., eds., *Proceedings of the 23rd AAMAS*, 2270–2272. IFAAMAS / ACM.

Galimullin, R., and Velázquez-Quesada, F. R. 2025. Topic-based communication between agents. *Studia Logica* 113(5):1195–1243.

Galimullin, R.; Gladyshev, M.; Mittelmann, M.; and Motamed, N. 2025. Changing the rules of the game: Reasoning about dynamic phenomena in multi-agent systems. In Das, S.; Nowé, A.; and Vorobeychik, Y., eds., *Proceedings of the 24th AAMAS*, 829–838. IFAAMAS / ACM.

Galimullin, R.; Gladyshev, M.; Mittelmann, M.; and Motamed, N. 2026. The dynamic turn in strategy logics. In Amato, C.; Dennis, L.; Mascardi, V.; and Thangarajah, J., eds., *Proceedings of the 25th AAMAS*. To appear.

Galimullin, R.; Grosinger, H.; and Mittelmann, M. 2026. I Would If I Could: Reasoning about Dynamics of Actions in Multi-Agent Systems. *CoRR* abs/2604.26053.

Gerbrandy, J., and Groeneveld, W. 1997. Reasoning about information change. *Journal of Logic, Language and Information* 6(2):147–169.

Groenendijk, J., and Stokhof, M. 1997. Questions. In van Benthem, J., and ter Meulen, A., eds., *Handbook of Logic and Language*. North Holland / Elsevier. 1055–1124.

Grosinger, H. J. 2025. The next level of long-term agent autonomy - proactively acquiring knowledge and abilities. In Das, S.; Nowé, A.; and Vorobeychik, Y., eds., *Proceedings of the 24th AAMAS*, 2859–2864. IFAAMAS / ACM.

Jamroga, W., and Dix, J. 2006. Model checking abilities under incomplete information is indeed Δ_2^{P}-complete. In Dunin-Keplicz, B.; Omicini, A.; and Padget, J. A., eds., *Proceedings of the 4th EUMAS*, volume 223 of *CEUR Workshop Proceedings*. CEUR-WS.org.

Jamroga, W., and van der Hoek, W. 2004. Agents that know how to play. *Fundamenta Informaticae* 63(2-3):185–219.

Jamroga, W.; Masko, L.; Mikulski, L.; Pazderski, W.; Penczek, W.; Sidoruk, T.; and Kurpiewski, D. 2022. Verification of multi-agent properties in electronic voting: A case study. In Fernández-Duque, D.; Palmigiano, A.; and Pinchinat, S., eds., *Proceedings of the 14th AiML*, 531–556. College Publications.

Kuijer, L. B. 2015. An arrow-based dynamic logic of norms. In Gutierrez, J.; Mogavero, F.; Murano, A.; and Wooldridge, M., eds., *Proceedings of the 3rd SR*, 1–11.

Kurpiewski, D.; Jamroga, W.; and Knapik, M. 2019. STV: model checking for strategies under imperfect information. In Elkind, E.; Veloso, M.; Agmon, N.; and Taylor, M. E., eds., *Proceedings of the 18th AAMAS*, 2372–2374. IFAAMAS.

Laroussinie, F.; Markey, N.; and Oreiby, G. 2008. On the Expressiveness and Complexity of ATL. *Logical Methods in Computer Science* Volume 4, Issue 2.

Liu, F. 2011. *Reasoning about Preference Dynamics*, volume 354 of *Synthese Library*. Springer.

Lomuscio, A.; Qu, H.; and Raimondi, F. 2017. MCMAS: an open-source model checker for the verification of multi-agent systems. *International Journal on Software Tools for Technology Transfer* 19(1):9–30.

Nam, W., and Kil, H. 2022. Formal verification of blockchain smart contracts via ATL model checking. *IEEE Access* 10:8151–8162.

Papadimitriou, C. H. 1994. *Computational complexity*. Addison-Wesley.

Plaza, J. 2007. Logics of public communications. *Synthese* 158(2):165–179.

Schobbens, P. 2003. Alternating-time logic with imperfect recall. In van der Hoek, W.; Lomuscio, A.; de Vink, E. P.; and Wooldridge, M. J., eds., *Proceedings of the 1st LCMAS*, volume 85 of *ENTCS*, 82–93. Elsevier.

Shoham, Y., and Tennenholtz, M. 1995. On social laws for artificial agent societies: Off-line design. *Artificial Intelligence* 73(1-2):231–252.

van Benthem, J., and Liu, F. 2007. Dynamic logic of preference upgrade. *Journal of Applied Non-Classical Logics* 17(2):157–182.

van Benthem, J., and Minica, S. 2012. Toward a dynamic logic of questions. *Journal of Philosophical Logic* 41(4):633–669.

van der Hoek, W., and Wooldridge, M. J. 2003. Cooperation, knowledge, and time: Alternating-time temporal epistemic logic and its applications. *Studia Logica* 75(1):125–157.

van der Hoek, W.; Roberts, M.; and Wooldridge, M. J. 2007. Social laws in alternating time: effectiveness, feasibility, and synthesis. *Synthese* 156(1):1–19.

van Ditmarsch, H.; Kuijer, L. B.; and Liu, M. 2023. An arrow-based dynamic logic of normative systems and its decidability. In Alechina, N.; Herzig, A.; and Liang, F., eds., *Proceedings of the 9th LORI*, volume 14329 of *LNCS*, 63–76. Springer.

van Ditmarsch, H.; van der Hoek, W.; and Kooi, B. 2008. *Dynamic Epistemic Logic*, volume 337 of *Synthese Library*. Springer.

Velázquez-Quesada, F. R. 2022. Communication between agents in dynamic epistemic logic. *CoRR* abs/2210.04656.