Reasoning about actions meets strategic logics

Andreas Herzig[1], Emiliano Lorini[1], and Dirk Walther[2]

[1] University of Toulouse, IRIT-CNRS
Toulouse, France
[2] TU Dresden, Theoretical Computer Science
Center for Advancing Electronics Dresden
Dresden, Germany

Abstract. We introduce ATLEA, a novel extension of Alternating-time Temporal Logic with explicit actions in the object language. ATLEA allows to reason about abilities of agents under commitments to play certain actions. Pre- and postconditions as well as availability and unavailability of actions can be expressed. We show that the multiagent extension of Reiter's solution to the frame problem can be encoded into ATLEA. We also consider an epistemic extension of ATLEA. We demonstrate that the resulting logic is sufficiently expressive to reason about uniform choices of actions. Complexity results for the satisfiability problem of ATLEA and its epistemic extension are given in the paper.

1 Introduction

Several formalisms for reasoning about actions were suggested in AI, including situation calculus [19], event calculus [20], fluent calculus [21, 22], and so-called action languages such as *A* and *C* [9, 14]. These formalisms provide languages to describe actions in terms of pre- and postconditions. We are interested in reasoning about actions within the framework of Alternating-time Temporal Logic (ATL) [2], a logic for reasoning about strategic abilities. In ATL there are no names for actions and there is no obvious way to describe the behaviour of actions. We therefore extend ATL to ATLEA: ATL with Explicit Actions in the object language. We demonstrate that the resulting logic allows to reason about multiagent actions. In particular, we show that ATLEA allows us to specify the pre- and post-conditions of actions and to check whether in a given situation an agent or coalition of agents has the capability to ensure a given outcome.

The paper is organised as follows. Section 2 introduces ATLEA, and Section 3 illustrates how pre- and postconditions of actions can be specified. We then consider an epistemic extension of ATLEA and demonstrate that it is sufficiently expressive to reason about the conditions under which an agent has a uniform choice to ensure a given state of affairs (Section 4).

2 ATL with explicit actions

An *action commitment* is a pair (a, ω) consisting of an agent a and an action name ω, also written $a \mapsto \omega$: a is committed to perform ω at the current state. An *action* *commitment function* is a finite set ρ of action commitments such that ρ is a partial function in its first argument: for every two (a, ω) and (a, ω') in ρ we have $\omega = \omega'$. We write $\rho(a) = \omega$ if $(a, \omega) \in \rho$; otherwise we say that $\rho(a)$ is undefined. The partial function ρ describes the commitments of the agents a in $\mathsf{dom}(\rho)$ to play action $\rho(a)$ at the current state.

Action commitment functions parameterise ATL path quantifiers. A formula of the form $\langle\langle A\rangle\rangle_\rho\psi$ is read: "while the agents in $\mathsf{dom}(\rho)$ perform the actions as specified in ρ, the agents in A have a strategy to ensure the temporal property ψ, no matter what the agents in $\Sigma \setminus A$ do." Just as in ATL, there is an existential quantification over the strategies of the agents in coalition A and a universal quantification over the strategies of the agents outside of A. The selection of strategies occurs simultaneously, without interdependencies between the agents. The novel part in ATLEA is that we only quantify over strategies respecting ρ. Note that in the path quantifier $\langle\langle A\rangle\rangle_\rho$, the function ρ may commit both members of the coalition A (the proponents) and its opponents outside A. A special case is when $\rho = \emptyset$: then $\langle\langle A\rangle\rangle_\rho$ is nothing but the ATL operator $\langle\langle A\rangle\rangle$. For example, the formula $\langle\langle a, c\rangle\rangle_{\{a\mapsto\omega_a, b\mapsto\omega_b\}}\psi$ holds at a state w if, and only if, there is a strategy for coalition $\{a, c\}$ where a performs ω_a at w, such that for all strategies for $\Sigma \setminus \{a, c\}$ where b performs ω_b at w, all paths resulting from the chosen strategies satisfy the temporal property ψ.

We fix a set Π of *atomic propositions*, a set Σ of *agents*, and a set Ω of *action names*. We assume that these three sets are countably infinite and disjoint.[3] The language of ATLEA is defined over the signature $\langle \Pi, \Sigma, \Omega\rangle$.

Definition 1 (ATLEA syntax). *The following grammar defines state formulas φ and path formulas ψ:*

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle\langle A\rangle\rangle_\rho\psi$$
$$\psi ::= \neg\psi \mid \bigcirc\varphi \mid \varphi\,\mathcal{U}\,\varphi$$

where p ranges over Π, A ranges over finite subsets of Σ and ρ ranges over action commitment functions with action names from Ω. The language of ATLEA *consists of state formulas.*

We sometimes omit set parentheses as in $\langle\langle a\rangle\rangle_{a\mapsto\omega}\bigcirc\varphi$. For state formulas, the Boolean operators $\wedge$, $\rightarrow$, $\leftrightarrow$, and the logical constants $\top$ and $\bot$ are defined as usual by means of $\neg$ and $\vee$. The commonly used temporal operators 'sometime' and 'forever' are defined as the path formulas $\Diamond\varphi = (\top\,\mathcal{U}\,\varphi)$ and $\Box\varphi = \neg(\top\,\mathcal{U}\,\neg\varphi)$, respectively.

Formulas are evaluated on concurrent game structures that additionally interpret action names as moves of players.

Definition 2 (CGSN). *Let $S = \{1, \ldots, n\} \subset \Sigma$, $n \geq 1$, be a finite set of agents, $P \subset \Pi$ a finite set of atomic propositions, and $O \subset \Omega$ be a finite set of action names. A* Concurrent Game Structure with action Names *(CGSN) $\mathcal{C}$ for the signature $\langle S, P, O\rangle$ is a tuple $\mathcal{C} = \langle W, V, M, Mov, E, \|\cdot\|\rangle$, where:*

- *W is a finite, non-empty set of worlds (alias states);*
- *$V : W \longrightarrow 2^P$ is a valuation function;*
- *M is a finite, non-empty set of moves;*

[3] Infinite signatures are relevant for the analysis of the complexity of the satisfiability problem.

- $Mov : W \times S \longrightarrow 2^M \setminus \emptyset$ *maps a world* w *and an agent* a *to the non-empty set* $Mov(w, a)$ *of moves available to* a *at* w*;*
- $E : W \times M^S \longrightarrow W$ *is a transition function mapping a world* w *and a move profile* $\boldsymbol{m} = \langle m_1, \ldots, m_n \rangle$ *(one move for each agent) to the world* $E(w, \boldsymbol{m})$*;*
- $\| \cdot \| : O \longrightarrow M$ *is a denotation function mapping action names in* O *to moves in* M*.*

CGSNs are finite objects. We obtain infinitely many classes of CGSNs, one per signature. In a CGSN, an action name is interpreted as a move (which may interpret several action names). $Mov(w, a)$ determines which of the moves from M are available to a at state w. We say that *action* ω_a *is available to agent* a *at* w if $\|\omega_a\| = m_a$ and $m_a \in Mov(w, a)$.

A *strategy for an agent* a is a function f_a that maps every world w to a move $f_a(w) \in Mov(w, a)$ available to a at w.[4] A *strategy for a coalition* $A \subseteq S$ is a function F_A that maps every agent a from A to a strategy $F_A(a)$ for a. Given an action commitment function ρ, a strategy F_A for A is called *compatible with* ρ *at* w if for all $a \in A \cap \mathsf{dom}(\rho)$,

$$F_A(a)(w) = \|\rho(a)\|.$$

Clearly, when $A \cap \mathsf{dom}(\rho) = \emptyset$ then any strategy F_A for coalition A is compatible with ρ. We denote with $\mathsf{strat}(A, \rho, w)$ the set of all strategies for A that are compatible with ρ at w. When the interpretation of agent a's commitment is not among the moves available at w, i.e., when $\|\rho(a)\| \notin Mov(w, a)$, then no strategy for a is compatible with ρ at w. This holds more generally for coalitions containing a: if $\|\rho(a)\| \notin Mov(w, a)$ for some $a \in A$ then $\mathsf{strat}(A, \rho, w) = \emptyset$.

A move profile is used to determine a successor of a state using the transition function E. We define the set of available move profiles at state w as follows:

$$\mathsf{prof}(w) = \{\langle m_1, \ldots, m_n \rangle \mid m_i \in Mov(w, i)\}.$$

The set of *possible successors of* w is the set of states $E(w, \boldsymbol{m})$ where $\boldsymbol{m}$ ranges over $\mathsf{prof}(w)$. An infinite sequence $\lambda = x_0 x_1 x_2 \cdots$ of worlds from W is called a *computation* if x_{i+1} is a successor of x_i for all positions $i \geq 0$. $\lambda[i]$ denotes the i-th component x_i in λ, and with $\lambda[0, i]$ the initial sequence $x_0 \cdots x_i$ of λ.

The set $\mathsf{out}(w, F_A)$ of *outcomes* of a strategy F_A for A starting at a world w is the set of all computations $\lambda = x_0 x_1 x_2 \cdots$ such that $x_0 = w$ and, for every $i \geq 0$, there is a move profile $\boldsymbol{m} = \langle m_1, \ldots, m_n \rangle \in \mathsf{prof}(x_i)$ such that:

- $m_a = F_A(a)(x_i)$, for all $a \in A$; and
- $x_{i+1} = E(x_i, \boldsymbol{m})$.

A strategy F_S for all agents in the signature specifies exactly one play: $\mathsf{out}(w, F_S)$ is a singleton. A CGSN $\mathcal{C}$ for $\langle S, P, O \rangle$ allows to interpret an ATLEA formula φ if S contains all agents, P all atomic propositions, and O all action names occurring in φ. The satisfaction relation is defined as follows:[5]

[4] The logic is defined for memoryless strategies. The extension to perfect recall strategies is straightforward.

[5] We skip the cases for atomic propositions, Boolean and temporal operators; they are defined as in ATL [2].

$$\begin{aligned}\mathcal{C}, w \models \langle\!\langle A\rangle\!\rangle_\rho\psi \text{ iff there exists } F_A \in \mathsf{strat}(A,\rho,w) \text{ such that}\\ \text{for all } F_{S\setminus A} \in \mathsf{strat}(S\setminus A,\rho,w) \text{ it holds that}\\ \mathcal{C},\lambda \models \psi, \text{ where } \{\lambda\} = \mathsf{out}(w, F_A \cup F_{S\setminus A}).\end{aligned}$$

Validity and satisfiability are defined as expected: φ is valid if $\mathcal{C}, w \models \varphi$ for every state w of every CGSN $\mathcal{C}$ whose signature contains that of φ; φ is satisfiable if $\neg\varphi$ is not valid.

With ATLEA we can express the (un-)availability of actions. A formula of the form $\langle\!\langle a\rangle\!\rangle_{a\mapsto\omega_a}\bigcirc\top$ states that action ω_a is available to agent a at the current state. More generally, we have that $\mathcal{C}, w \models \langle\!\langle A\rangle\!\rangle_\rho\bigcirc\top$ iff $\|\rho(a)\| \in Mov(w,a)$ for all $a \in A \cap \mathsf{dom}(\rho)$. The other way round, to express the unavailability of ω_a to a, we have that $\mathcal{C}, w \models \neg\langle\!\langle A\rangle\!\rangle_\rho\bigcirc\top$ iff there is an $a \in \mathsf{dom}(\rho) \cap A$ such that $\|\rho(a)\| \notin Mov(w,a)$.

ATL is the fragment of ATLEA where every action commitment function is empty.[6] Without commitments ATLEA formulas can be interpreted in CGSNs with empty denotation functions, which are essentially concurrent game structures as used in ATL. A crucial difference to ATL, however, is the fact that ATLEA can detect the difference between memoryless and perfect recall strategies. Consider a CGSN for one agent with two states x and y such that $p \in V(x)$ but $p \notin V(y)$, $Mov(x,a) = \{1,2\}$ and $Mov(y,a) = \{1\}$, $x = E(x,1)$, $y = E(x,2) = E(y,1)$, and $||\omega|| = 1$. The formula $\langle\!\langle a\rangle\!\rangle_{a\mapsto\omega}\bigcirc p \wedge \langle\!\langle a\rangle\!\rangle_{a\mapsto\omega}(\top\,\mathcal{U}\,\neg p)$ is false at a state x under memoryless strategies, but it holds for strategies that allow a recall of at least one predecessor.

The proposition below illustrates that the status of some ATLEA counterparts of ATL axioms [12] depends on the interplay of the two arguments in the ATLEA operator $\langle\!\langle A\rangle\!\rangle_\rho$.

Proposition 1. *The following formulas are* ATLEA *valid.*

1. $\langle\!\langle A\rangle\!\rangle_\rho\bigcirc\top$ *for* $\mathsf{dom}(\rho) \cap A$ *empty*
2. $\neg\langle\!\langle A\rangle\!\rangle_\rho\bigcirc\bot$ *for* $\mathsf{dom}(\rho) \setminus A$ *empty*
3. $(\langle\!\langle A\rangle\!\rangle_\rho\bigcirc\varphi \wedge \langle\!\langle B\rangle\!\rangle_\rho\bigcirc\psi) \to \langle\!\langle A\cup B\rangle\!\rangle_\rho\bigcirc(\varphi\wedge\psi)$ *for* $A\cap B \subseteq \mathsf{dom}(\rho)$
4. $\langle\!\langle A\rangle\!\rangle_\rho\bigcirc\varphi \to \langle\!\langle A\rangle\!\rangle_{\rho'}\bigcirc\varphi$ *for* $\rho' = \rho\cup\{a\mapsto\omega\}, a\notin A$
5. $\langle\!\langle A\rangle\!\rangle_{\rho'}\bigcirc\varphi \to \langle\!\langle A\rangle\!\rangle_\rho\bigcirc\varphi$ *for* $\rho' = \rho\cup\{a\mapsto\omega\}, a\in A$
6. $\langle\!\langle A\cup\{a\}\rangle\!\rangle_\rho\bigcirc\varphi \to \langle\!\langle A\rangle\!\rangle_\rho\bigcirc\varphi$ *for* $a \in \mathsf{dom}(\rho)$
7. $\langle\!\langle A\rangle\!\rangle_\rho(\varphi\,\mathcal{U}\,\psi) \leftrightarrow ((\psi \wedge \langle\!\langle A\rangle\!\rangle_\rho\bigcirc\top) \vee (\varphi \wedge \langle\!\langle A\rangle\!\rangle_\rho\bigcirc\langle\!\langle A\rangle\!\rangle_\emptyset(\varphi\,\mathcal{U}\,\psi)))$

Item 1 generalises the ATL axiom ($\top$) (obtained when ρ is empty). Another particular case is when A is empty: then both $\langle\!\langle\emptyset\rangle\!\rangle_\rho\bigcirc\top$ is valid. Item 2 generalises the ATL axiom ($\bot$) (obtained when ρ is empty). To see that Item 1 is invalid when $\mathsf{dom}(\rho)$ and A are not disjoint, suppose $a \in \mathsf{dom}(\rho)\cap A$: then we can always find a CGSN $\mathcal{C}$ and a state w such that $\|\rho(a)\| \notin Mov(w,a)$, and then $\mathcal{C}, w \not\models \langle\!\langle A\rangle\!\rangle_\rho\bigcirc\top$. Item 3 generalises ATL's superadditivity axiom (S), relaxing the constraint of disjointness of A and B. Intuitively it says that when the actions of the agents that are in both, A and B, are

[6] We note in passing: ATL does not allow for negated path formulas, while ATLEA does (cf. Def. 1).

fixed by ρ then these agents cannot have different strategies to enforce φ and ψ. For that reason, the powers of the two coalitions can be combined. Consider the case where $A \subseteq \mathsf{dom}(\rho)$. It then follows from the monotony of $\langle\!\langle A \rangle\!\rangle_\rho$ and our superadditivity axiom (Item 3) that $\langle\!\langle A \rangle\!\rangle_\rho$ is a normal modal box operator. Items 4 and 5 are respectively about increasing commitment of opponents and releasing commitment of proponents. Item 6 is about dismissing committed proponents. Item 7 is a fixpoint axiom of ATL. It allows to rewrite formulas in a way such that ρ is empty in all modal operators of the form $\langle\!\langle A \rangle\!\rangle_\rho(\varphi \mathcal{U} \psi)$.[7] Moreover, the generalisations of the ATL inference rules of Modus Ponens and Necessitation all preserve validity. However, we leave a complete axiomatisation of ATLEA for future work.

Remark 1. As an extension of ATLEA, we may consider PDL program operators such as sequential composition, iteration and test over action descriptions. For the one-agent case, this is related to CTL with Path Relativisation [15]. It would also be interesting to study complements of actions, as well as the loop construct, which allows to formulate action commitments of the form $a \mapsto \omega^\infty$ stating that a plays the action denoted with ω at all states. We can also view $a \mapsto \omega^\infty$ as a commitment of a to play ω in all situations. In other words, ω^∞ is a strategy. This means that we can specify entire (memoryless) strategies within such an extension of ATLEA. This motivates a study of the relationships between the extension of ATLEA and other logics with representations of strategies in the object language such as ATLES [28] and Strategy Logic [6], which we leave for furture work. The equivalence in Item 7 of Proposition 1 becomes invalid if we generalise commitments from atomic actions to sequences of actions. The extension of ATLEA by the program operators of PDL is subject of ongoing work.

Theorem 1. *The satisfiability problem for* ATLEA *is ExpTime-complete.*

The ExpTime lower bound carries over from the fragment ATL [29]. The matching upper complexity bound can be shown by adapting the decision procedure for ATL [29], which is a type elimination constructions inspired from [8].

3 Reasoning about actions

We now put ATLEA to work and demonstrate its usefulness in reasoning about multiagent actions. We start by encoding in ATLEA Reiter's action descriptions in terms of complete conditions for the executability and the effects of actions. We build on the mapping of Reiter's solution to the frame problem into dynamic epistemic logics with assignments as done in [7]. We take the multiagent context into account by integrating ideas stemming from logics of propositional control. There, the set of propositional variables is partitioned among the agents, and an agent controlling a variable is the only one able to change its truth value [26].

[7] We note that ATL's S-maximality axiom $\neg\langle\!\langle S \rangle\!\rangle_\emptyset \bigcirc \neg\psi \leftrightarrow \langle\!\langle \emptyset \rangle\!\rangle_\emptyset \bigcirc \psi$ (which relates the empty coalition with the set of all agents) does not make sense in our setting: as formulas φ are evaluated in CGSNs whose signature *contains* that of φ, there is no way of 'grasping' the set of all agents S of a given model. Our ATLEA (and also the underlying version of ATL) is more general than ATL as defined in [12]. The latter is actually a *family* of logics: each member of the family is defined for a finite set of agents, yielding uncountably many axiomatisations.

3.1 Action descriptions

Let $\langle S, P, O\rangle$ be a finite signature. Let Φ be the set of propositional formulas over P. An *action description for* $\langle S, P, O\rangle$ is a tuple $\mathcal{T} = \langle \mathsf{agt}, \mathsf{poss}, \mathsf{eff}\rangle$ where

- $\mathsf{agt} : O \longrightarrow S$ associates to each action name ω an agent agt_ω;
- $\mathsf{poss} : O \longrightarrow \Phi$ associates to each action name ω a propositional formula poss_ω such that for every agent a, $\bigvee_{\omega \mid \mathsf{agt}_\omega = a} \mathsf{poss}_\omega$ is valid in propositional logic;
- $\mathsf{eff} : O \longrightarrow P \longrightarrow \Phi$ is a mapping associating to each action name ω a partial function $\mathsf{eff}_\omega : P \longrightarrow \Phi$, such that if poss_{ω_1} and poss_{ω_2} are satisfiable in propositional logic and $\mathsf{agt}_{\omega_1} \neq \mathsf{agt}_{\omega_2}$ then the domains of eff_{ω_1} and eff_{ω_2} are disjoint.

The function agt associates actions to agents who can perform them. No two agents have the same action. The propositional formula poss_ω characterises the conditions under which ω is executable by agt_ω; the constraint says that at every state, each agent has at least one action that is executable. The intuition of the function eff is that when eff_ω is defined for p then one of the things ω does is to assign to p the truth value of $\mathsf{eff}_\omega(p)$: if φ is true before ω then p is true after ω, and if φ is false before ω then p is false after ω. When eff_ω is undefined for p then ω leaves the truth value of p unchanged. The disjointness constraint guarantees that there is no state where two different agents have executable actions changing the truth value of p. This is more liberal a condition than exclusive control[8] that is common in logics of propositional control [26, 10]. We call ours *local exclusive control*.

Example 1. Consider a light that is controlled by two switches. The position of these switches is described by two propositional variables; moreover, there are variables describing whether agent a is close to switch k or not: $P = \{up_1, up_2\} \cup \{close_{a,k} \mid a, k \in \{1,2\}\}$. The light is on if the switches are either both up ($up_1 \wedge up_2$) or both down ($\neg up_1 \wedge \neg up_2$); in other words, the light is on if and only if $up_1 \leftrightarrow up_2$. There are two agents: $S = \{1,2\}$. Each agent a can toggle each switch k ($toggle_{a,k}$) or do nothing: $O = \{toggle_{a,k} \mid a, k \in \{1,2\}\} \cup \{nop_a \mid a \in \{1,2\}\}$. Therefore the action description $\mathcal{T} = \langle \mathsf{agt}, \mathsf{poss}, \mathsf{eff}\rangle$ is as follows.

- $\mathsf{agt}_{toggle_{a,k}} = \mathsf{agt}_{nop_a} = a$, for all agents a and switches k;
- $\mathsf{poss}_{toggle_{a,k}} = close_{a,k} \wedge \neg close_{a',k}$, for agents $a \neq a'$ (in order to flip a switch the agent has to be the only one close to it);
- $\mathsf{poss}_{nop_a} = \top$;
- eff_{nop_a} is undefined for all $p \in P$ (the action nop_a does not change any variable);
- $\mathsf{eff}_{toggle_{a,k}}$ is defined for up_k, and $\mathsf{eff}_{toggle_{a,k}}(up_k) = \neg up_k$.

Observe that the function eff obeys our constraints on action descriptions: for the conjunction $\mathsf{poss}_{toggle_{1,1}} \wedge \mathsf{poss}_{toggle_{2,2}}$ to be propositionally satisfiable, the domains of eff, $\mathsf{dom}(\mathsf{eff}_{toggle_{1,1}}) = \{up_1\}$ and $\mathsf{dom}(\mathsf{eff}_{toggle_{2,2}}) = \{up_2\}$, have to be disjoint, which is indeed the case.

[8] According to [10], control is exclusive when $\mathsf{agt}_{\omega_1} \neq \mathsf{agt}_{\omega_2}$ implies that the domains $\mathsf{dom}(\mathsf{eff}_{\omega_1})$ and $\mathsf{dom}(\mathsf{eff}_{\omega_2})$ are disjoint, whatever poss_{ω_1} and poss_{ω_2} are. (We have adapted the notation.)

Action descriptions are an economic description of a domain and 'count as a solution to the frame problem' [19]: the descriptions only talk about what changes and do not contain frame axioms. A given $\mathcal{T} = \langle \mathsf{agt}, \mathsf{poss}, \mathsf{eff} \rangle$ determines what Reiter calls a successor state axiom for each $p \in P$; in the situation calculus this takes the following form:

$$p(do(x,s)) \leftrightarrow \big(\bigvee_{\omega \mid p \in \mathsf{dom}(\mathsf{eff}_\omega)} (x = \omega \wedge \mathsf{eff}_\omega(p)) \big) \vee \big(p(s) \wedge \neg \bigvee_{\omega \mid p \in \mathsf{dom}(\mathsf{eff}_\omega)} x = \omega \big)$$

where x is an action variable and s is a situation variable, both universally quantified. It says that action x makes p true iff either x is an action whose precondition for making p true holds, or p was true before and x is not an action changing p.

3.2 CGSNs for $\mathcal{T}$

We now associate concurrent game structures with action names to a given action description.

Let $\langle S, P, O \rangle$ be a signature. Let $\mathcal{T} = \langle \mathsf{agt}, \mathsf{poss}, \mathsf{eff} \rangle$ be an action description and let $\mathcal{C} = \langle W, V, M, Mov, E, \| \cdot \| \rangle$ be CGSN. $\mathcal{C}$ is a *CGSN for* $\mathcal{T}$ iff:

- $M = O$;
- $Mov(w,a) = \{\omega \in O \mid \mathsf{agt}_\omega = a \;\&\; V(w) \models \mathsf{poss}_\omega\}$;
- $V(E(w, \boldsymbol{m})) =$

$$\{p \mid \exists i \in S, \mathsf{eff}_{m_i} \text{ defined for } p \;\&\; V(w) \models \mathsf{eff}_{m_i}(p)\} \cup$$
$$\{p \mid p \in V(w) \;\&\; \forall i \in S, \mathsf{eff}_{m_i} \text{ undefined for } p\};$$

- $\|\omega\| = \omega$.

In the clause for Mov, the condition $V(w) \models \mathsf{poss}_\omega$ has to be understood as truth of poss_ω in the propositional interpretation $V(w)$. Note that the clause for E corresponds to Reiter's successor state axiom.

A state formula φ of the language of ATLEA is *valid in the class of CGSNs for* $\mathcal{T}$ iff $\mathcal{C}, w \models \varphi$ for every state w of every CGSNs $\mathcal{C}$ for $\mathcal{T}$ whose signature contains that of φ. Moreover, φ is *satisfiable in a CGSN for* $\mathcal{T}$ iff $\neg\varphi$ is not satisfiable.

We can now formulate two important problems in reasoning about actions. Suppose given a signature $\langle S, P, O \rangle$, an action description $\mathcal{T}$, a formula describing the initial state φ_i and a formula describing the goal state φ_g. The *prediction problem* for a sequence of multiagent actions $\rho_1, \ldots, \rho_n$ is to decide whether it the case that

$$\varphi_i \rightarrow \langle\!\langle \mathsf{dom}(\rho_1) \rangle\!\rangle_{\rho_1} \bigcirc \cdots \langle\!\langle \mathsf{dom}(\rho_n) \rangle\!\rangle_{\rho_n} \bigcirc \varphi_g$$

is valid in the class of CGSNs for $\mathcal{T}$; the *planning problem* for a set of agents A is to decide whether it the case that

$$\varphi_i \rightarrow \langle\!\langle A \rangle\!\rangle_{\emptyset} \Diamond \varphi_g$$

is valid in the class of CGSNs for $\mathcal{T}$.

Example 2. Let us take up Example 1. Whether

$$(close_{1,1} \wedge \neg close_{2,1} \wedge close_{2,2} \wedge \neg close_{1,2} \wedge up_1 \wedge \neg up_2) \to \\ \langle\langle 1,2 \rangle\rangle_{1 \mapsto nop_1, 2 \mapsto toggle_{2,2}} \bigcirc (up_1 \leftrightarrow up_2)$$

is valid in the CGSNs for $\mathcal{T}$ is a prediction problem. Whether

$$(close_{1,1} \wedge \neg close_{2,1} \wedge close_{2,2} \wedge \neg close_{1,2}) \to \\ \langle\langle 1,2 \rangle\rangle_{\emptyset} \bigcirc (up_1 \leftrightarrow up_2)$$

is valid in the CGSNs for $\mathcal{T}$ is a planning problem. Both implications are valid in the class of CGSNs for $\mathcal{T}$.

3.3 Reduction to ATLEA satisfiability

We now show that for finite signatures, satisfiability in a CGSN for $\mathcal{T}$ can be reduced to ATLEA satisfiability.

Proposition 2. *Let $\langle S, P, O \rangle$ be a finite signature. Let $\mathcal{T}$ be an action description in $\langle S, P, O \rangle$ and let φ be a formula in $\langle S, P, O \rangle$. φ is satisfiable in a CGSN for $\mathcal{T}$ iff $\varphi \wedge \langle\langle \emptyset \rangle\rangle \Box (\bigwedge \Gamma)$ is* ATLEA *satisfiable, where Γ collects the following sets of formulas, for every $a \in S$, $p \in P$, and $\omega \in O$:*

1. $\mathsf{poss}_\omega \leftrightarrow \langle\langle \mathsf{agt}_\omega \rangle\rangle_{\mathsf{agt}_\omega \mapsto \omega} \bigcirc \top$
2. $\mathsf{eff}_\omega(p) \to \langle\langle \emptyset \rangle\rangle_{\mathsf{agt}_\omega \mapsto \omega} \bigcirc p$, *for $p \in \mathsf{dom}(\mathsf{eff}_\omega)$*
3. $\neg\mathsf{eff}_\omega(p) \to \langle\langle \emptyset \rangle\rangle_{\mathsf{agt}_\omega \mapsto \omega} \bigcirc \neg p$, *for $p \in \mathsf{dom}(\mathsf{eff}_\omega)$*
4. $\left(\bigwedge_{\omega | p \in \mathsf{dom}(\mathsf{eff}_\omega)} \neg \mathsf{poss}_\omega\right) \to (p \to \langle\langle \emptyset \rangle\rangle_{\emptyset} \bigcirc p)$
5. $\left(\bigwedge_{\omega | p \in \mathsf{dom}(\mathsf{eff}_\omega)} \neg \mathsf{poss}_\omega\right) \to (\neg p \to \langle\langle \emptyset \rangle\rangle_{\emptyset} \bigcirc \neg p)$
6. $\mathsf{poss}_\omega \to (p \to \langle\langle \emptyset \rangle\rangle_{\mathsf{agt}_\omega \mapsto \omega'} \bigcirc p)$
   for $p \in \mathsf{dom}(\mathsf{eff}_\omega)$ and $p \notin \mathsf{dom}(\mathsf{eff}_{\omega'})$;
7. $\mathsf{poss}_\omega \to (\neg p \to \langle\langle \emptyset \rangle\rangle_{\mathsf{agt}_\omega \mapsto \omega'} \bigcirc \neg p)$
   for $p \in \mathsf{dom}(\mathsf{eff}_\omega)$ and $p \notin \mathsf{dom}(\mathsf{eff}_{\omega'})$.

Formula 1 translates the information specified in $\mathcal{T}$ about the executability of ω. Formulas 2 and 3 translate the information about the effects of ω. The last four clauses are about the frame axioms and basically express that those variables p for which eff_ω is undefined are left unchanged by the execution of ω. Formulas 4 and 5 say that when none of the actions changing p is executable then the truth value of p remains unchanged. Consider formulas 6 and 7: suppose p is one of the effects of ω ($p \in \mathsf{dom}(\mathsf{eff}_\omega)$) and suppose at the present state ω is executable (poss_ω is true); then due to the local exclusive control constraint on the eff function of $\mathcal{T}$, at that state p can only be changed by agt_ω. Therefore, when agt_ω performs a different action ω' not affecting p then the truth value of p remains unchanged, whatever the other agents do.

Observe that the cardinality of Γ is polynomial in the number of symbols in the signature (more precisely: cubic). As the length of every formula in Γ is bound by the cardinality of Γ (because of items 4 and 5), the length of the formula $\bigwedge \Gamma$ is polynomial in the number of symbols in the signature, too. We can therefore polynomially embed the reasoning problems of prediction and planning into ATLEA.

4 Epistemic extension

We now sketch an epistemic extension of ATLEA along the lines of [25]. We call our logic Alternating-time Temporal Epistemic Logic with Explicit Actions, ATELEA.

4.1 ATELEA

We add knowledge modalities K_a to the language, one per agent a in Σ, and as well as common knowledge modalities C_A, one per finite subset A of Σ. We read the formula $\mathsf{K}_a\varphi$ as "a knows that φ is true" and the formula $\mathsf{C}_A\varphi$ as "the agents in A have common knowledge that φ is true".

Concurrent Epistemic Game Structures with action Names CEGSNs) are of the form

$$\mathcal{C}^+ = \langle W, V, M, Mov, E, \|\cdot\|, \{R_a\}_{a\in S}\rangle$$

where $\langle W, V, M, Mov, E, \|\cdot\|\rangle$ is a CGSN (cf. Def. 2) and where every $R_a \subseteq W \times W$ is an equivalence relation.

Given a CEGSN $\mathcal{C}^+ = \langle W, V, M, Mov, E, \|\cdot\|, \{R_a\}_{a\in S}\rangle$, the satisfaction relation $\models$ is defined as follows:

$$\mathcal{C}^+, w \models \mathsf{K}_a\varphi \text{ iff } \mathcal{C}^+, v \models \varphi \text{ for all } v \in W \text{ with } wR_a v$$
$$\mathcal{C}^+, w \models \mathsf{C}_A\varphi \text{ iff } \mathcal{C}^+, v \models \varphi \text{ for all } v \in W \text{ with } wR_A^+ v$$

where $R_A = \bigcup_{a\in A} R_a$ and where R_A^+ is the transitive closure of R_A. For the ATLEA operators the definition is as before.

We can extend the decision procedure for ATLEA to allow for the epistemic operators. This is done similarly to ATEL compared to ATL [27]. We obtain the following result.

Theorem 2. *The satisfiability problem for* ATELEA *is ExpTime-complete.*

Let us take over the concrete semantics for ATLEA given in Section 3 and consider the class of CEGSNs structures induced by an action specification. Let $\mathcal{T} = \langle \mathsf{agt}, \mathsf{poss}, \mathsf{eff}\rangle$ be an action specification and $\mathcal{C}^+ = \langle \mathcal{C}, \{R_a\}_{a\in S}\rangle$ a CEGSN for a finite signature $\langle S, P, O\rangle$. We say that $\mathcal{C}^+$ is a *CEGSN for* $\mathcal{T}$ if $\mathcal{C}$ is a CGSN for $\mathcal{T}$ as defined in Section 3.2.

As the following proposition highlights, satisfiability in a CEGSN for an action specification $\mathcal{T}$ can be reduced to ATELEA satisfiability: satisfiability with respect to the general class of CEGSNs.

Let $dg(\varphi)$ be the maximal number of nestings of ATLEA operators $\langle\langle A\rangle\rangle_\rho$ and ATELEA epistemic operators K_a or C_A within φ. Let $(\langle\langle\emptyset\rangle\rangle\Box\mathsf{C}_A)^n\psi$, for $n \geq 0$, be the formula where $\langle\langle\emptyset\rangle\rangle\Box\mathsf{C}_A$ is iterated n times. (So $(\langle\langle\emptyset\rangle\rangle\Box\mathsf{C}_A)^0\psi$ is ψ.)

Proposition 3. *Let $\mathcal{T}$ be an action specification in the finite signature $\langle S, P, O\rangle$ and let $dg(\varphi) = n$. Let φ be a formula of the language of* ATELEA *in $\langle S, P, O\rangle$. φ is satisfiable in a CEGSN for $\mathcal{T}$ iff $\varphi \wedge (\langle\langle\emptyset\rangle\rangle\Box C_S)^n(\bigwedge\Gamma) \wedge (C_S\langle\langle\emptyset\rangle\rangle\Box)^n(\bigwedge\Gamma)$ is* ATELEA *satisfiable, where Γ is the finite set of formulas defined in Proposition 2.*

The proof can be done in a way similar to that of Prop. 2.

4.2 Reasoning about uniform choices in ATELEA

An interesting aspect of our logic is that it allows us to express the concept of *uniform choice*. Specifically, we say that agent a has a *uniform choice* from the finite set of actions O to ensure that φ will be true in the next state when there exists an action in O such that a knows that by choosing this action she will ensure φ in the next state, no matter what the other agents will do. This can be expressed in ATELEA as follows:

$$\mathsf{UC}_a(O,\varphi) \overset{\mathtt{def}}{=} \bigvee_{\omega \in O} \mathsf{K}_a \langle\!\langle \{a\} \rangle\!\rangle_{\{a \mapsto \omega\}} \bigcirc \varphi$$

$\mathsf{UC}_i(O,\varphi)$ has to be read "agent a has a uniform choice from the finite set of actions O to ensure φ in the next state". This concept of uniform choice is closely related to the concept of power. In fact, a given agent a's power of achieving a certain result φ involves not only a's capability of achieving a but also a's knowledge about this capability. For example, for a thief to have the power of opening a safe, he must know the safe's combination. (See [16] for a detailed analysis of the distinction between capability and power.)

Furthermore, in ATELEA we can draw non-trivial inferences showing that, given certain initial conditions, an agent has (or has not) a uniform choice to ensure φ in the next state. Consider the following continuation of Example 1.

Example 3 (cont.). Remember that the light is on if the switches are either both up ($up_1 \wedge up_2$) or both down ($\neg up_1 \wedge \neg up_2$). Let us therefore abbreviate the equivalence $up_1 \leftrightarrow up_2$ by $lightOn$. Suppose that in the initial situation agent 1 knows that the light is off. Moreover, suppose that agent 1 knows that he is close to switch 1. Finally, let us assume that agent 1 knows that agent 2 cannot perform the action of toggling switch 1 or switch 2 because he is far away from both switches. In other words, agent 1 knows that agent 2 cannot interfere with his actions. Then we can prove that agent 1 has a uniform choice to ensure that the light is on in the next state. Indeed, it is easy to show the following formula is valid in the class of CEGSN determined by the action description $\mathcal{T}$ of Example 1:

$$\begin{aligned}(\mathsf{K}_1 \neg lightOn \wedge \mathsf{K}_1 close_{1,1} \wedge \mathsf{K}_1(\neg close_{2,1} \wedge \neg close_{2,2})) \rightarrow \\ \mathsf{UC}_1(\{toggle_{1,1}, toggle_{1,2}, nop_1\}, lightOn)\end{aligned}$$

Thanks to the common knowledge operator we can generalize the previous notion of uniform choice to coalitions of agents. It is reasonable to assume that the agents in a coalition A have the power to ensure a given outcome φ only if they can *coordinate* their actions in such a way that φ will be true in the next state. In order to achieve this level of coordination, the agents in A must have common knowledge that by performing a given joint action they will together make φ true, that is, the agents in A must have a *uniform collective choice* to ensure φ. Uniform collective choice can be formally expressed as follows. Let $A = \{1, \ldots, k\}$. Then:

$$\mathsf{UC}_A(O,\varphi) \overset{\mathtt{def}}{=} \bigvee_{\omega_1, \ldots, \omega_k \in O} \mathsf{C}_A \langle\!\langle A \rangle\!\rangle_{\{1 \mapsto \omega_1, \ldots, k \mapsto \omega_k\}} \bigcirc \varphi$$

$\mathsf{UC}_A(O,\varphi)$ has to be read "coalition A has a uniform collective choice from the set of actions O to ensure φ in the next state".

Example 4 (cont.). Let us continue our running example and suppose that agents 1 and 2 have common knowledge that: (1) the light is off, and (2) agent 1 is close to switch 1 and far from switch 2 while agent 2 is close to switch 2 and far from switch 1. Then we can prove that the coalition $\{1, 2\}$ has a uniform collective choice to ensure that the light is on in the next state. Indeed, it is easy to show that the following formula is valid in the class of CEGSN determined by the action description $\mathcal{T}$ of Example 1:

$$\begin{aligned}&\big(\mathsf{C}_{\{1,2\}}\neg \mathit{lightOn} \wedge \mathsf{C}_{\{1,2\}}(\mathit{close}_{1,1} \wedge \mathit{close}_{2,2}) \wedge \\ &\quad \mathsf{C}_{\{1,2\}}(\neg \mathit{close}_{1,2} \wedge \neg \mathit{close}_{2,1})\big) \rightarrow \mathsf{UC}_{\{1,2\}}(O,\mathit{lightOn})\end{aligned}$$

with $O = \{\mathit{toggle}_{a,k} \mid a, k \in \{1, 2\}\} \cup \{\mathit{nop}_a \mid a \in \{1, 2\}\}$. Furthermore, we can also prove that if e.g. the agents do not have common knowledge whether the light is on then there is no uniform collective choice ensuring that the light is on. That is,

$$\begin{aligned}&(\neg\mathsf{C}_{\{1,2\}}\mathit{lightOn} \wedge \neg\mathsf{C}_{\{1,2\}}\neg \mathit{lightOn}) \rightarrow \\ &\neg\mathsf{UC}_{\{1,2\}}(O,\mathit{lightOn})\end{aligned}$$

5 Related work

Several authors have noted that while strategic logics provide an interesting abstract formalism to reason about actions and strategies, it would nevertheless be useful to have actions or strategies as first-class objects. This was tried for Coalition Logic (for example by [4, 13]) and for some very expressive logics that turned out to be undecidable (for example [17, 5, 23]). We here only overview extensions of ATL.

Alternating-time temporal logic with Actions (ATL-A) together with its epistemic extension was introduced in [1] to obtain a strategic logic for describing actions as well as their interaction with knowledge, and to solve problems with previous approaches. ATL-A corresponds to a version of ATLEA with commitment functions ρ defined over non-deterministic composition of action names and in which any such ρ can only occur in formulas of the form $\langle\langle A\rangle\rangle_\rho \bigcirc\varphi$. While we appreciate ATL-A as an interesting contribution to incorporate actions in strategic logics, we argue that the better design lies with ATLEA. The syntax of ATL-A is unwieldy as each alternative action for every agent has to be mentioned in the formula. This makes it impossible to express a's commitment $a \mapsto \omega_a$ to use action ω_a in ATL-A with a general (infinite) action signature; and even if we restrict the logic to a finite action signature the resulting ATL-A formula will be huge. Abbreviations were suggested (already in [1]) for ATL-A to be more friendly to modellers. ATL-A defines the temporal operators 'forever' and 'until' with action specifications in terms of 'next-time' and the respective fixpoint equation from ATL (cf. Item 7 in Proposition 1). While coupling one-step actions with 'next-time' formulas is conceptually clear, using fixpoint equations to define other temporal operators involves an exponential blowup in formula size, which may be an issue with reasoning complexity. Extending ATL-A to plans of actions appears to require major changes of its semantics, whereas extending ATLEA this way requires defining what it means for a coalitional strategy to be compatible with a complex action description (cf. the set strat(.) in Section 2). In [1], model checking for ATL-A was studied, while the satisfiability problem, which is relevant for synthesis and mechanism design, is not considered.

Commitment ATL, CATL, [24] is an extension of ATL with ternary operators of the form $C_i(\sigma_i,\varphi)$ with the intended reading "if it were the case that agent i committed to the strategy σ_i, then φ". The interpretation of this operator is in terms of model updates: $C_i(\sigma_i,\varphi)$ is true at world w of a given model M if and only if φ holds at w of model M' that results from eliminating from M all moves that are not consistent with agent i's strategy σ_i.[9] The complexity of the satisfiability problem for CATL has not been studied, whereas the complexity result for ATLEA and its epistemic extension is one of our main contributions here. There is also an important conceptual difference: the former considers commitments to play *strategies* while the latter considers commitments to play *actions*. From this point of view, CATL is much closer to ATL with Explicit Strategies (ATLES) by [28], where ATL-path quantifiers are parameterised with commitment functions for strategies [28], than to our ATLEA.

As for the differences between our ATLEA and Walther *et al.*'s ATLES, it is worth noting that with ATLEA we can formalise the (un-)availability of actions at states (cf. the side conditions of items 1 and 2 in Prop. 1), whereas with ATLES one can reference and reason with existing strategies but not reason about their availability. Another difference is the local nature of commitments in ATLEA, i.e., commitments to atomic actions are released after one time step (cf. Item 7 in Prop. 1).

The integration of game-theoretic concepts into the situation calculus was a subject of recent research. Belle and Lakemeyer [3] study games in extensive form (in its imperfect information version), where only one agent can act per state. Consequently no interactions have to be accounted for. They don't have path quantifiers, which allows them to define regression. De Giacomo, Lespérance and Pearce [11] have studied a multiagent version of the situation calculus in order to reason about extensive games where at most one agent can act at a given state. That agent is identified by a predicate $Control(a)$ indicating that a controls the current state. Concurrency is simulated by interleaving. They have a (first-order) language with ATL path quantifiers. For a given signature, the quantifier $\langle\!\langle A\rangle\!\rangle\bigcirc\varphi$ is basically regressed to

$$(\bigvee_{a\in A} Control(a) \wedge \bigvee_{\omega\in O}\langle\omega\rangle\varphi) \vee (\bigvee_{a\notin A} Control(a) \wedge \bigwedge_{\omega\in O}[\omega]\varphi)$$

where $\langle\omega\rangle$ and $[\omega]$ are the dynamic operators of PDL. This relies on finiteness of the set of agent and action symbols. While all these approaches do not really allow for 'true' concurrency, Reiter [18] had proposed to extend his solution to the frame problem to concurrent actions. Different from us, he allows for several actions to be performed simultaneously by the same agent and does not assume exclusive control of propositional variables. This comes with the problem of interacting preconditions: there are states where two actions ω_1 and ω_2 with inconsistent postconditions are performed concurrently. This is avoided by our condition of (local) exclusive control.

[9] CATL models are called Action-based Alternating Transition Systems (AATSs) and are closely related to CGSNs.

6 Conclusion

We have introduced a variant of Alternating-time Temporal Logic that has explicit actions. The interesting aspect of our logic is that it combines ATL's strategic reasoning with reasoning about actions in terms of pre- and postconditions as traditionally done in AI.

In future research, we will investigate the extension by regular expressions over actions. This will allow to talk not only about uniform choices, but also about uniform strategies. Moreover, we intend to provide sound and complete axiomatizations both for ATLEA and for its epistemic extension ATELEA.

Acknowledgements

We thank the LORI reviewers for their comments. Andreas Herzig and Emiliano Lorini acknowledge the support of the LabEx project CIMI, and Dirk Walther the support of the German Research Foundation (DFG) within the Cluster of Excellence 'Center for Advancing Electronics Dresden'.

References

1. T. Ågotnes. Action and knowledge in alternating-time temporal logic. *Synthese*, 149(2):375–407, 2006.
2. R. Alur, R. Henzinger, and O. Kupferman. Alternating-time temporal logic. *Journal of the ACM*, 49(5):672–713, 2002.
3. V. Belle and G. Lakemeyer. Reasoning about imperfect information games in the epistemic situation calculus. In *Proceedings of AAAI'10*. AAAI Press, 2010.
4. S. Borgo. Coalitions in action logic. In *Proceedings of IJCAI'07*, pages 1822–1827, 2007.
5. T. Brihaye, A. Da Costa Lopes, F. Laroussinie, and N. Markey. ATL with strategy contexts and bounded memory. In *Proceedings of LFSC'09*, volume 5407 of *LNCS*, pages 92–106. Springer, 2009.
6. K. Chatterjee, T. A. Henzinger, and N. Piterman. Strategy logic. *Information and Computation*, 208(6):677–693, 2010.
7. H. van Ditmarsch, A. Herzig, and T. de Lima. From Situation Calculus to Dynamic Logic. *Journal of Logic and Computation*, 21(2):179–204, 2011.
8. E. A. Emerson. Temporal and modal logic. In *Handbook of Theoretical Computer Science (vol. B): Formal Models and Semantics*, pages 995–1072. MIT Press, 1990.
9. M. Gelfond and V. Lifschitz. Representing action and change by logic programs. *The Journal of Logic Programming*, 17:301–321, 1993.
10. J. Gerbrandy. Logics of propositional control. In *Proceedings of AAMAS'06*, pages 193–200. ACM, 2006.
11. G. De Giacomo, Y. Lespérance, and A. R. Pearce. Situation calculus based programs for representing and reasoning about game structures. In *Proceedings of KR'10*. AAAI Press, 2010.
12. V. Goranko and G. van Drimmelen. Complete axiomatization and decidability of alternating-time temporal logic. *Theoretical Computer Science*, 353(1–3):93–117, 2006.
13. A. Herzig and E. Lorini. A dynamic logic of agency I: STIT, capabilities and powers. *Journal of Logic, Language and Information*, 19(1):89–121, 2010.

14. N. Kartha and V. Lifschitz. A simple formalizations of actions using circumscription. In *Proceedings of IJCAI'95*, pages 1970–1975, 1995.
15. M. Lange and M. Latte. A CTL-based logic for program abstractions. In *Proceedings of WoLLIC'10*, volume 6188 of *LNCS*, pages 19–33. Springer-Verlag, 2010.
16. E. Lorini, N. Troquard, A. Herzig, and J. Broersen. Grounding power on actions and mental attitudes. *Logic Journal of the IGPL*, 21(3):311–331, 2013.
17. F. Mogavero, A. Murano, and M. Y. Vardi. Reasoning about strategies. In *Proceedings of FSTTCS'10*, pages 133–144, 2010.
18. R. Reiter. Natural actions, concurrency and continuous time in the situation calculus. In *Proceedings of KR'96*, pages 2–13. Morgan Kaufmann, 1996.
19. R. Reiter. *Knowledge in Action: Logical Foundations for Specifying and Implementing Dynamical Systems*. MIT Press, 2001.
20. M. Shanahan. *Solving the frame problem: a mathematical investigation of the common sense law of inertia*. MIT Press, 1997.
21. M. Thielscher. Representing the knowledge of a robot. In *Proceedings of KR'00*, pages 109–120. Morgan Kaufmann, 2000.
22. M. Thielscher. *Reasoning Robots - The Art and Science of Programming Robotic Agents*. Number 33 in Applied Logic. Springer, Dordrecht, 2005.
23. N. Troquard and D. Walther. On satisfiability in ATL with strategy contexts. In *Proceedings of JELIA'12*, volume 7519 of *LNCS*, pages 398–410. Springer, 2012.
24. W. van der Hoek, W. Jamroga, and M. Wooldridge. A logic for strategic reasoning. In *Proceedings of AAMAS'05*, pages 157–164. ACM Press, 2005.
25. W. van der Hoek and M. Wooldridge. Cooperation, knowledge, and time: Alternating-time temporal epistemic logic and its applications. *Studia Logica*, 75:125–157, 2003.
26. W. van der Hoek and M. Wooldridge. On the logic of cooperation and propositional control. *Artificial Intelligence*, 164(1–2):81–119, 2005.
27. D. Walther. ATEL with common and distributed knowledge is exptime-complete. In *Proceedings of M4M-4*, 2005.
28. D. Walther, W. van der Hoek, and M. Wooldridge. Alternating-time Temporal Logic with Explicit Strategies. In *Proceedings of TARK'07*, pages 269–278. ACM, 2007.
29. D. Walther, C. Lutz, F. Wolter, and M. Wooldridge. ATL satisfiability is indeed Exptime-complete. *Journal of Logic and Computation*, 16(6):765–787, 2006.