Quand le raisonnement sur les actions rencontre les logiques stratégiques (MFI 2013)
Andreas Herzig, Emiliano Lorini, Dirk Walther

⇒ To cite this version:
Andreas Herzig, Emiliano Lorini, Dirk Walther. Quand le raisonnement sur les actions rencontre les logiques stratégiques (MFI 2013). Secondes journées francophones sur les Modèles Formels de l'Interaction (MFI 2013), Jul 2013, Lille, France. hal-03464938

HAL Id: hal-03464938
https://hal.science/hal-03464938
Submitted on 7 Dec 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Reasoning about actions meets strategic logics
Quand le raisonnement sur les actions rencontre les logiques stratégiques

Andreas Herzig* Emiliano Lorini* Dirk Walther**
lorini@irit.fr herzig@irit.fr dirkww@googlemail.com
*University of Toulouse, IRIT-CNRS
Toulouse, France
**University of 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.

Keywords: Alternating-time Temporal Logic ATL, reasoning about actions, frame problem

Résumé :
Nous introduisons ATLEA : une extension de la logique temporelle ATL (Alternating-time Temporal Logic) par des actions explicites. ATLEA permet de raisonner sur les capacités des agents à exécuter des actions tout en étant respectant des engagements. Elle permet également d’exprimer des pré- et postconditions d’actions ainsi que la disponibilité et non-disponibilité d’actions. Nous montrons que l’extension multi-agent de la solution de Reiter au problème du cadre peut être encodé en ATLEA. Nous considérons aussi une extension épistémique d’ATLEA. Nous démontrons que la logique résultante est suffisamment expressive pour raisonner sur des choix uniformes d’actions. Finalement, l’article contient des résultats de complexité de satisfaisabilité et d’ATLEA et de son extension épistémique.

Mots-clés : Alternating-time Temporal Logic ATL, raisonnement sur les actions, problème du cadre

1 Introduction

Several formalisms for reasoning about actions were suggested in AI, including situation calculus [18], event calculus [19], fluent calculus [20, 21], and so-called action languages such as A and C [8, 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.

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). Proofs are contained in the long version of the paper [13].

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 ⊢ ω : 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 ω = ω′. We write ρ(a) = ω if (a, ω) ∈ ρ; otherwise we say that ρ(a) is undefined. The partial function ρ describes the commitments of the agents a in dom(ρ) to play action ρ(a) at the current state.

Action commitment functions parameterise ATL path quantifiers. The formula $\langle\langle A\rangle\rangle_\rho \psi$ is read : “while the agents in dom(ρ) perform the actions as specified in ρ, the agents in A have a strategy to ensure the temporal property $\psi$, 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 \( \rho \) may commit both members of the coalition \( A \) (the proponents) and its opponents outside \( A \). A special case is when \( \rho = \emptyset \); then \( \langle A \rangle_\rho \) is nothing but the ATL operator \( \langle A \rangle \). For example, the formula \( \langle a, c \rangle \{ a \rightarrow \omega_a, b \rightarrow \omega_b \} \psi \) holds at a state \( w \) if, and only if, there is a strategy for coalition \( \{ a, c \} \) where \( a \) performs \( \omega_a \) at \( w \), such that for all strategies for \( \{ a, c \} \) where \( b \) performs \( \omega_b \) at \( w \), all paths resulting from the chosen strategies satisfy the temporal property \( \psi \).

We fix a set \( \Pi \) of atomic propositions, a set \( \Sigma \) of agents, and a set \( \Omega \) of action names. We assume that these three sets are countably infinite and disjoint.¹ The language of ATLEA is defined over the signature \( \langle \Pi, \Sigma, \Omega \rangle \).

**Definition 2.1 [ATLEA syntax]** The following grammar defines state formulas \( \varphi \) and path formulas \( \psi \):

\[
\begin{align*}
\varphi &::= p \mid \neg \varphi \mid \varphi \lor \varphi \mid \langle A \rangle_\rho \psi \\
\psi &::= \neg \psi \mid \bigcirc \varphi \mid \varphi \mid \varphi \cup \varphi
\end{align*}
\]

where \( p \) ranges over \( \Pi \), \( A \) ranges over finite subsets of \( \Sigma \) and \( \rho \) ranges over action commitment functions with action names from \( \Omega \). The language of ATLEA consists of state formulas. <

We sometimes omit set parentheses as in \( \langle a \rangle_{a \rightarrow \omega} \bigcirc \varphi \). For state formulas, the Boolean operators \( \land, \rightarrow, \leftrightarrow \), and the logical constants \( \top \) and \( \bot \) are defined as usual by means of \( \neg \) and \( \lor \). The commonly used temporal operators ‘sometimes’ and ‘forever’ are defined as the path formulas \( \Diamond \varphi = (\top \cup \varphi) \) and \( \Box \varphi = \neg (\top \cup \neg \varphi) \), respectively.

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

**Definition 2.2 [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) \( C \) for the signature \( \langle S, P, O \rangle \) is a tuple \( C = \langle W, V, M, Mov, E, || \cdot || \rangle \), where:

- \( W \) is a finite, non-empty set of worlds (alias states);
- \( V : W \rightarrow 2^P \) is a valuation function;
- \( M \) is a finite, non-empty set of moves;
- \( Mov : W \times S \rightarrow 2^M \setminus \emptyset \) maps a world \( w \) and an agent \( a \) to the non-empty set \( Mov(w, a) \) of moves available to \( a \) at \( w \);
- \( E : W \times M^S \rightarrow W \) is a transition function mapping a world \( w \) and a move profile \( \vec{m} = \langle m_1, \ldots, m_n \rangle \) (one move for each agent) to the world \( E(w, \vec{m}) \);
- \( || \cdot || : O \rightarrow M \) 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 \( \omega_a \) is available to agent \( a \) at \( w \) if \( \rho(a) = m_a \) 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 \( w \).² 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 \( \rho \), a strategy \( F_A \) for \( A \) is called compatible with \( \rho \) if \( F_A(a)(w) = ||\rho(a)|| \).

Clearly, when \( A \cap \rho = \emptyset \) then any strategy \( F_A \) for coalition \( A \) is compatible with \( \rho \). We denote with \( \text{strat}(A, \rho, w) \) the set of all strategies for \( A \) that are compatible with \( \rho \) 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 \( \rho \) at \( w \). This holds more generally for coalitions containing \( a \) : if \( ||\rho(a)|| \notin Mov(w, a) \) for some \( a \in A \) then \( \text{strat}(A, \rho, w) = \emptyset \).<

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:

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

The set of possible successors of \( w \) is the set of states \( E(w, \vec{m}) \) where \( \vec{m} \) ranges over \( \text{prof}(w) \). 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 \( \lambda \), and with \( \lambda[0, i] \) the initial sequence \( x_0 \cdots x_i \) of \( \lambda \).

---

¹ Infinite signatures are relevant for the analysis of the complexity of the satisfiability problem.

² The logic is defined for memoryless strategies. The extension to perfect recall strategies is straightforward.
The set \( \text{out}(w, F_A) \) of outcomes of a strategy \( F_A \) for \( A \) starting at a world \( w \) is the set of all computations \( \lambda = x_0x_1x_2 \cdots \) such that \( x_0 = w \) and, for every \( i \geq 0 \), there is a move profile \( \vec{m} = \langle m_1, \ldots, m_n \rangle \in \text{prof}(x_i) \) such that:

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

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

\[
C, w \models \langle A \rangle_{\rho} \psi \quad \text{iff} \quad \exists F_A \in \text{strat}(A, \rho, w) \quad \text{s.t.} \quad \forall F_{S \setminus A} \in \text{strat}(S \setminus A, \rho, w) \quad \text{it holds that} \quad C, \lambda \models \psi,
\]

where \( \{ \lambda \} = \text{out}(w, F_A \cup F_{S \setminus A}) \).

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

With ATLEA we can express the (un)satisfiability of actions. The formula \( \langle a \rangle_{\omega_a \cdot \omega_a} \top \) states that action \( \omega_a \) is available at agent \( a \) at the current state. More generally, we have that \( C, w \models \langle A \rangle_{\rho} \top \) iff \( \| \rho(a) \| \in \text{Mov}(w, a) \) for all \( a \in A \cap \text{dom}(\rho) \). The other way round, to express the unsatisfiability of \( \omega_a \) to \( a \), we have that \( C, w \models \neg \langle A \rangle_{\rho} \top \) iff there is an \( a \in \text{dom}(\rho) \cap A \) such that \( \| \rho(a) \| \not\in \text{Mov}(w, a) \).

ATL is the fragment of ATLEA where every action commitment function is empty. 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 \( \langle a \rangle_{ao \cdot ao} \top \) 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 [11] depends on the interplay of the two arguments in the ATLEA operator \( \langle A \rangle_{\rho} \).

**Proposition 2.3** The following formulas are ATLEA valid.

1. \( \langle A \rangle_{\rho} \top \) for \( \text{dom}(\rho) \cap A \) empty
2. \( \neg \langle A \rangle_{\rho} \bot \) for \( \text{dom}(\rho) \setminus A \) empty
3. \( (\langle A \rangle_{\rho} \varphi \land \langle B \rangle_{\rho} \psi) \rightarrow (\langle A \cup B \rangle_{\rho} \varphi \land \psi) \) for \( A \cap B \subseteq \text{dom}(\rho) \)
4. \( \langle A \rangle_{\rho} \varphi \rightarrow \langle A \rangle_{\rho} \varphi \)
   for \( \rho = \rho \cup \{ a \mapsto 0 \}, a \notin A \)
5. \( \langle A \rangle_{\rho} \varphi \rightarrow \langle A \rangle_{\rho} \varphi \)
   for \( \rho = \rho \cup \{ a \mapsto 0 \}, a \in A \)
6. \( (\langle A \rangle_{\rho} \varphi \land \langle A \rangle_{\rho} \top) \lor (\varphi \land \langle A \rangle_{\rho} \top) \) for \( a \in \text{dom}(\rho) \)
7. \( \langle A \rangle_{\rho} (\varphi \land \psi) \leftrightarrow (\langle \psi \land \langle A \rangle_{\rho} \top \rangle \lor (\varphi \land \langle A \rangle_{\rho} \top)) \)

Item 1 generalises the ATL axiom \( (T) \) (obtained when \( \rho \) is empty). Another particular case is when \( A \) is empty : then both \( \langle \emptyset \rangle_{\rho} \top \) is valid. Item 2 generalises the ATL axiom \( (\bot) \) (obtained when \( \rho \) is empty). To see that Item 1 is invalid when \( \text{dom}(\rho) \) and \( A \) are not disjoint, suppose \( a \in \text{dom}(\rho) \cap A \) : then we can always find a CGSN \( C \) and a state \( w \) such that \( \| \rho(a) \| \notin \text{Mov}(w, a) \), and then \( C, w \not\models \langle A \rangle_{\rho} \top \). 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 fixed by \( \rho \) then these agents cannot have different strategies to enforce \( \varphi \) and \( \psi \). For that reason, the powers of the two coalitions can be combined. 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 \( \rho \) is empty in all modal operators of the form \( \langle A \rangle_{\rho} (\varphi \land \psi) \).

---

3. We skip the cases for atomic propositions, Boolean and temporal operators. They are defined as in ATL.
4. We note in passing : ATL does not allow for negated path formulas, while ATLEA does (cf. Def. 2.1).
5. We note that ATL’s S-maximality axiom \( \neg (\langle S \rangle_{\emptyset} \top \rightarrow \psi) \leftrightarrow (\langle \emptyset \rangle_{\emptyset} \top \psi) \) (which relates the empty coalition with the set of all agents) does not make sense in our setting : as formulas \( \varphi \) are evaluated in CG-
Moreover, the generalisations of the ATL inference rules of Modus Ponens and Necessitation all preserve validity.

Consider the case where \( A \subseteq \text{dom}(\rho) \). It then follows from the monotony of \( \langle A \rangle_{\rho} \) and our superadditivity axiom (Item 3) that \( \langle A \rangle_{\rho} \) is a normal modal box operator.

**Remark 2.4** 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 \( \omega \) at all states. We can also view \( a \mapsto \omega^\infty \) as a commitment of \( a \) to play \( \omega \) in all situations. In other words, \( \omega^\infty \) is a strategy. This means that we can specify entire (memoryless) strategies within such an extension of ATLEA. The equivalence in Item 7 of Proposition 2.3 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 2.5** The satisfiability problem for ATLEA is ExpTime-complete.

The ExpTime lower bound carries over from the fragment ATL [28]. The matching upper complexity bound can be shown by adapting the decision procedure for ATL [28], which is a type elimination constructions inspired from [7].

### 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 [6]. 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 [25].

#### 3.1 Action descriptions

Let \( \langle S, P, O \rangle \) be a finite signature. Let \( \Phi \) be the set of propositional formulas over \( P \). An action description for \( \langle S, P, O \rangle \) is a tuple \( T = (\text{agt}, \text{poss}, \text{eff}) \) where

- \( \text{agt} : O \rightarrow S \) associates to each action name \( \omega \) an agent \( \text{agt}_{\omega} \);
- \( \text{poss} : O \rightarrow \Phi \) associates to each action name \( \omega \) a propositional formula \( \text{poss}_{\omega} \), such that for every agent \( a \), \( \bigvee_{\omega | \text{agt}_{\omega} = a} \text{poss}_{\omega} \) is valid in propositional logic;
- \( \text{eff} : O \rightarrow P \rightarrow \Phi \) is a mapping associating to each action name \( \omega \) a partial function \( \text{eff}_{\omega} : P \rightarrow \Phi \), such that if \( \text{poss}_{\omega_1} \) and \( \text{poss}_{\omega_2} \) are satisfiable in propositional logic and \( \text{agt}_{\omega_1} \neq \text{agt}_{\omega_2} \) then the domains of \( \text{eff}_{\omega_1} \) and \( \text{eff}_{\omega_2} \) are disjoint.

The function \( \text{agt} \) associates actions to agents who can perform them. No two agents have the same action. The propositional formula \( \text{poss}_{\omega} \) characterises the conditions under which \( \omega \) is executable by \( \text{agt}_{\omega} \); the constraint says that at every state, each agent has at least one action that is executable. The intuition of the function \( \text{eff} \) is that when \( \text{eff}_{\omega} \) is defined for \( p \) then one of the things \( \omega \) does is to assign to \( p \) the truth value of \( \text{eff}_{\omega}(p) \); if \( \varphi \) is true before \( \omega \) then \( p \) is true after \( \omega \), and if \( \varphi \) is false before \( \omega \) then \( p \) is false after \( \omega \). When \( \text{eff}_{\omega} \) is undefined for \( p \) then \( \omega \) 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 [6] that is common in logics of propositional control [25, 9]. We call ours local exclusive control.

**Example 3.1** 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: \( P = \{up_1, up_2\} \cup \{\text{close}_{a,k} | a, k \in \{1,2\}\} \). The

6. According to [9], control is exclusive when \( \text{agt}_{\omega_1} \neq \text{agt}_{\omega_2} \) implies that the domains \( \text{dom}(\text{eff}_{\omega_1}) \) and \( \text{dom}(\text{eff}_{\omega_2}) \) are disjoint, whenever \( \text{poss}_{\omega_1} \) and \( \text{poss}_{\omega_2} \) are. (We have adapted the notation.)
light is on if the switches are either both up
\((up_1 \land up_2)\) or both down \((-up_1 \land -up_2)\); 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 \(k\) \((\text{toggle}_{a,k})\) or do nothing \((O)\) or \(\{\text{toggle}_{a,k} \mid a, k \in \{1, 2\}\} \cup \{\text{nop}_a \mid a \in \{1, 2\}\}\). Therefore the action description \(T = \langle \text{agt}, \text{poss}, \text{eff} \rangle\) is as follows.

1. \(\text{agt}_{\text{toggle}_{a,k}} = \text{agt}_{\text{nop}_a} = a\), for all agents \(a\) and switches \(k\);
2. \(\text{poss}_{\text{toggle}_{a,k}} = \text{close}_{a,k} \land \neg \text{close}_{a',k}\), for agents \(a \neq a'\) (in order to flip a switch the agent has to be the only one close to it);
3. \(\text{poss}_{\text{nop}_a} = \top\);
4. \(\text{eff}_{\text{nop}_a}\) is undefined for all \(p \in P\) (the action \(\text{nop}_a\) does not change any variable);
5. \(\text{eff}_{\text{toggle}_{a,k}}\) is defined for \(up_k\), and \(\text{eff}_{\text{toggle}_{a,k}}(up_k) = \neg up_k\).

Observe that the function \(\text{eff}\) obeys our constraints on action descriptions: for example, \(\text{poss}_{\text{toggle}_{1,1}} \land \text{poss}_{\text{toggle}_{2,2}}\) being propositionally satisfiable, the domains \(\text{dom}(\text{eff}_{\text{toggle}_{1,1}}) = \{up_1\}\) and \(\text{dom}(\text{eff}_{\text{toggle}_{2,2}}) = \{up_2\}\) have to be disjoint, which is indeed the case.

Action descriptions are an economic description of a domain and ‘count as a solution to the frame problem’ [18]: the descriptions only talk about what changes and do not contain frame axioms. A given \(\bar{T} = \langle \text{agt}, \text{poss}, \text{eff} \rangle\) determines what Reiter calls a successor state axiom for each \(p \in P\); in the situation calculus this takes the following form:

\[
p(do(x, s)) \iff \bigvee_{\omega|p \in \text{dom}(\text{eff}_\omega)} (x = \omega \land \text{eff}_\omega(p)) \lor (p(s) \land \neg \bigvee_{\omega|p \in \text{dom}(\text{eff}_\omega)} x = \omega)
\]

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 \(T\)

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

Let \(\langle S, P, O \rangle\) a signature. Let \(T = \langle \text{agt}, \text{poss}, \text{eff} \rangle\) be an action description and let \(C = \langle W, V, M, \text{Mov}, E, \parallel \parallel \rangle\) be CGSN. \(C\) is a CGSN for \(T\) iff:

1. \(M = O\);
2. \(\text{Mov}(w, a) = \{\omega \in O \mid \text{agt}_\omega = a \land V(w) \models \text{poss}_\omega\}\);
3. \(V(E(w, \bar{m})) = \{p \mid \exists i \in S, \text{eff}_m\text{, defined for } p \land V(w) \models \text{eff}_m(p)\} \cup \{p \mid p \in V(w) \land \forall i \in S, \text{eff}_m\text{, undefined for } p\}\);
4. \(\parallel \omega \parallel = \omega\).

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

A state formula \(\varphi\) of the language of ATLEA is valid in the class of CGSNs for \(T\) iff \(C, w \models \varphi\) for every state \(w\) of every CGSNs \(C\) for \(T\) whose signature contains that of \(\varphi\). Moreover, \(\varphi\) is satisfiable in a CGSN for \(T\) 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 \(T\), a formula describing the initial state \(\varphi_i\) and a formula describing the goal state \(\varphi_g\). The prediction problem for a sequence of multiagent actions \(\rho_1, ..., \rho_n\) is to decide whether it the case that

\[
\varphi_i \rightarrow \langle\langle \text{dom}(\rho_1)\rangle\rangle_{\rho_1} \cdots \langle\langle \text{dom}(\rho_n)\rangle\rangle_{\rho_n} \parallel \varphi_g
\]

is valid in the class of CGSNs for \(T\); the planning problem for a set of agents \(A\) is to decide whether it the case that

\[
\varphi_i \rightarrow \langle\langle A\rangle\rangle_\parallel \varphi_g
\]

is valid in the class of CGSNs for \(T\).

#### Example 3.2

Let us take up Example 3.1. Whether

\[
(\text{close}_{1,1} \land \neg \text{close}_{2,1} \land \text{close}_{2,2} \land \neg \text{close}_{1,2} \land up_1 \land \neg up_2) \rightarrow \langle\langle 1, 2\rangle\rangle_{1 \rightarrow \text{nop}_1, 2 \rightarrow \text{toggle}_2} \parallel (up_1 \leftrightarrow up_2)
\]

is valid in the CGSNs for \(T\) is a prediction problem. Whether

\[
(\text{close}_{1,1} \land \neg \text{close}_{2,1} \land \text{close}_{2,2} \land \neg \text{close}_{1,2}) \rightarrow \langle\langle 1, 2\rangle\rangle_\parallel (up_1 \leftrightarrow up_2)
\]

is valid in the CGSNs for \(T\) is a planning problem. Both implications are valid in the class of CGSNs for \(T\).
3.3 Reduction to ATLEA satisfiability

We now show that for finite signatures, satisfiability in a CEGSN for \( T \) can be reduced to ATLEA satisfiability.

**Proposition 3.3** Let \( \langle S, P, O \rangle \) be a finite signature. Let \( T \) be an action description in \( \langle S, P, O \rangle \) and let \( \varphi \) be a formula in \( \langle S, P, O \rangle \). \( \varphi \) is satisfiable in a CEGSN for \( T \) iff \( \varphi \land \langle \emptyset \rangle \Box (\bigwedge \Gamma) \) is ATLEA satisfiable, where \( \Gamma \) collects the following sets of formulas, for every \( a \in S \), \( p \in P \), and \( \omega \in O : \)

1. \( \text{poss}_\omega \leftrightarrow \langle \text{agt}_\omega \rangle_{\text{agt}_\omega \to \omega} \circ T \)
2. \( \text{eff}_\omega (p) \to \langle \emptyset \rangle_{\text{agt}_\omega \to \omega} \circ p, \text{for } p \in \text{dom}(\text{eff}_\omega) \)
3. \( \neg \text{eff}_\omega (p) \to \langle \emptyset \rangle_{\text{agt}_\omega \to \omega} \circ \neg p, \text{for } p \in \text{dom}(\text{eff}_\omega) \)
4. \( \left( \bigwedge_{p \in \text{dom}(\text{eff}_\omega)} \neg \text{poss}_\omega \right) \to (p \to \langle \emptyset \rangle_{\emptyset} \circ p) \)
5. \( \left( \bigwedge_{p \in \text{dom}(\text{eff}_\omega)} \neg \text{poss}_\omega \right) \to (\neg p \to \langle \emptyset \rangle_{\emptyset} \circ \neg p) \)
6. \( \text{poss}_\omega \to (p \to \langle \emptyset \rangle_{\text{agt}_\omega \to \omega'} \circ p), \text{for } p \in \text{dom}(\text{eff}_\omega) \text{ and } p \notin \text{dom}(\text{eff}_\omega) \)
7. \( \text{poss}_\omega \to (\neg p \to \langle \emptyset \rangle_{\text{agt}_\omega \to \omega'} \circ \neg p), \text{for } p \in \text{dom}(\text{eff}_\omega) \text{ and } p \notin \text{dom}(\text{eff}_\omega) \).

Formula 1 translates the information specified in \( T \) about the executability of \( \omega \). Formulas 2 and 3 translate the information about the effects of \( \omega \). The last four clauses are about the frame axioms and basically express that those variables \( p \) for which \( \text{eff}_\omega \) is undefined are left unchanged by the execution of \( \omega \). 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 \( \omega (p \in \text{dom}(\text{eff}_\omega)) \) and suppose at the present state \( \omega \) is executable (\( \text{poss}_\omega \) is true) ; then due to the local exclusive control constraint on the eff function of \( T \), at that state \( p \) can only be changed by \( \text{agt}_\omega \). Therefore, when \( \text{agt}_\omega \) performs a different action \( \omega' \) not affecting \( p \) then the truth value of \( p \) remains unchanged, whatever the other agents do.

Observe that the cardinality of \( \Gamma \) is polynomial in the number of symbols in the signature (more precisely : cubic). As the length of every formula in \( \Gamma \) is bound by the cardinality of \( \Gamma \) (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 [24]. We call our logic Alternating-time Temporal Epistemic Logic with Explicit Actions, ATLEA.

4.1 ATLEA

We add knowledge modalities \( K_a \) to the language, one per agent \( a \) in \( \Sigma \), and as well as common knowledge modalities \( C_A \), one per finite subset \( A \) of \( \Sigma \). We read the formula \( K_a \varphi \) as “\( a \) knows that \( \varphi \) is true” and the formula \( C_A \varphi \) as “the agents in \( A \) have common knowledge that \( \varphi \) is true”.

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

\[
C = \langle W, V, M, Mov, E, \| \cdot \|, \{R_a\}_{a \in S} \rangle
\]

where \( \langle W, V, M, Mov, E, \| \cdot \| \rangle \) is a CEGSN (cf. Def. 2.2) and where every \( R_a \subseteq W \times W \) is an equivalence relation.

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

\[
C, w \models K_a \varphi \text{ iff } C, v \models \varphi \text{ for all } v \in W \text{ s.t. } wR_av
\]

\[
C, w \models C_A \varphi \text{ iff } C, v \models \varphi \text{ for all } v \in W \text{ s.t. } wR_A^+v
\]

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 ATL compared to ATL [26]. We obtain the following result.

**Theorem 4.1** The satisfiability problem for ATLEA 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 \( T = \langle \text{agt}, \text{poss}, \text{eff} \rangle \) be an action specification and \( C = \langle C, \{R_a\}_{a \in S} \rangle \) a CEGSN for a finite signature \( \langle S, P, O \rangle \). We say that \( C \) is a CEGSN for \( T \) if \( C \) is a CEGSN for \( T \) 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 \textit{ATELEA} satisfiability: satisfiability with respect to the general class of CEGSNs.

Let \( d_{\text{fg}}(\varphi) \) be the maximal number of nestings of \textit{ATELEA} operators \( \langle \langle A \rangle \rangle \rho \) and \textit{ATELEA} epistemic operators \( K_a \) or \( C_A \) within \( \varphi \). Let \( (\langle \langle \emptyset \rangle \rangle \boxtimes C_A)^n \psi \), for \( n \geq 0 \), be the formula where \( \langle \langle \emptyset \rangle \rangle \boxtimes C_A \) is iterated \( n \) times. (So \( (\langle \emptyset \rangle \rangle \boxtimes C_A)^0 \psi \) is \( \psi \).

**Proposition 4.2** Let \( T \) be an action specification in the finite signature \( \langle S, P, O \rangle \) and let \( d_{\text{fg}}(\varphi) = n \). Let \( \varphi \) be a formula of the language of \textit{ATELEA} in \( \langle S, P, O \rangle \). \( \varphi \) is satisfiable in a CEGSN for \( T \) iff

\[
\varphi \land (\langle \langle \emptyset \rangle \rangle \boxtimes C_S)^n(\bigwedge \Gamma) \land (C_S \langle \langle \emptyset \rangle \rangle \boxtimes)^n(\bigwedge \Gamma)
\]

is \textit{ATELEA} satisfiable, where \( \Gamma \) is the finite set of formulas defined in Proposition 3.3.

The proof can be done in a way similar to that of Prop. 3.3.

**4.2 Reasoning about uniform choices in \textit{ATELEA}**

An interesting aspect of our logic is that it allows us to express that a given agent \( a \) has the power to ensure \( \varphi \) true in the next state by choosing an action from a finite set of actions \( O \). We say that agent \( a \) has a \textit{uniform choice} from the finite set of actions \( O \) to ensure that \( \varphi \) 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 \( \varphi \) in the next state, no matter what the other agents will do. This can be expressed in \textit{ATELEA} as follows:

\[
\text{UC}_a(O,\varphi) \overset{\text{def}}{=} \bigvee_{\omega \in O} K_a(\langle \{ a \} \rangle_{\{ a \rightarrow \omega \}}) \varphi
\]

\( \text{UC}_a(O,\varphi) \) has to be read “agent \( a \) has a uniform choice from the finite set of actions \( O \) to ensure \( \varphi \) in the next state”.

Furthermore, in \textit{ATELEA} we can draw non-trivial inferences showing that, given certain initial conditions, an agent has (or has not) a uniform choice to ensure \( \varphi \) in the next state. Consider the following continuation of Example 3.1.

**Example 4.3 (cont.)** Remember that the light is on if the switches are either both up \((\text{up}_1 \land \text{up}_2)\)

or both down \((\neg \text{up}_1 \land \neg \text{up}_2)\). Let us therefore abbreviate the equivalence \( \text{up}_1 \leftrightarrow \text{up}_2 \) by \text{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 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 \( T \) of Example 3.1:

\[
(K_1 \neg \text{lightOn} \land K_1 \text{close}_{1,1} \land K_1 (\neg \text{close}_{2,1} \land \neg \text{close}_{2,2})) \rightarrow \text{UC}_1(\{\text{toggle}_{1,1}, \text{toggle}_{1,2}, \text{nop}_1\}, \text{lightOn})
\]

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 \( \varphi \) only if they can \textit{coordinate} their actions in such a way that \( \varphi \) 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 \( \varphi \) true, that is, the agents in \( A \) must have a \textit{uniform collective choice} to ensure \( \varphi \). Uniform collective choice can be formally expressed as follows. Let \( A = \{1, \ldots, k\} \). Then:

\[
\text{UC}_A(O,\varphi) \overset{\text{def}}{=} \bigvee_{\omega_1, \ldots, \omega_k \in O} C_A(\langle \langle A \rangle \rangle_{\{1 \rightarrow \omega_1, \ldots, k \rightarrow \omega_k\}}) \varphi
\]

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

**Example 4.4 (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 \( T \):
of Example 3.1:
\[(C_{\{1,2\}} \neg \text{lightOn} \land C_{\{1,2\}}(\text{close}_1 \land \text{close}_2) \land C_{\{1,2\}}(\neg \text{close}_1 \land \neg \text{close}_2)) \rightarrow \text{UC}_{\{1,2\}}(O, \text{lightOn})\]

with
\[
O = \{ \text{toggle}_{a,k} \mid a, k \in \{1, 2\} \} \cup \{ \text{nop}_a \mid a \in \{1, 2\} \}
\]

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,
\[
(\neg C_{\{1,2\}} \text{lightOn} \land \neg C_{\{1,2\}} \neg \text{lightOn}) \rightarrow \neg \text{UC}_{\{1,2\}}(O, \text{lightOn})
\]

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, 12]) and for some very expressive logics that turned out to be undecidable (for example [16, 5, 22]). 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 \(\rho\) defined over non-deterministic composition of action names and in which any such \(\rho\) can only occur in formulas of the form \(\langle A \rangle^\rho \varphi\). 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 \(a \rightarrow \omega_a\) to use action \(\omega_a\) 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 2.3). 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 \text{strat}(\cdot) 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, [23] 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 \(\sigma_i\), then \(\varphi\)”. 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 \(\varphi\) holds at \(w\) of model \(M'\) that results from eliminating from \(M\) all moves that are not consistent with agent \(i\)’s strategy \(\sigma_i\). 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 [27], where ATL-path quantifiers are parameterised with commitment functions for strategies [27], 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. 2.3), 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. 2.3).

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

7. CATL models are called Action-based Alternating Transition Systems (AATSs) and are closely related to CGSNs.
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 [10] 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 $\text{Control}(a)$ 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 $\langle\langle A \rangle\rangle \varphi$ is basically regressed to

$$
\left( \bigvee_{a \in A} \text{Control}(a) \land \bigvee_{\omega \in O} \langle \omega \rangle \varphi \right) \lor \\
\left( \bigvee_{a \notin A} \text{Control}(a) \land \bigwedge_{\omega \in O} [\omega] \varphi \right)
$$

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 [17] 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 $\omega_1$ and $\omega_2$ with inconsistent postconditions are performed concurrently. This is avoided by our condition of (local) exclusive control.

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

Un grand merci à tous les trois relecteurs de MFI 2013 pour leur travail : leurs commentaires détaillés et précis nous ont permis d’améliorer l’article.

Références


