Alternating-time Temporal Logic with Explicit Strategies

Dirk Walther
Computer Science Deptartment
University of Liverpool
Liverpool, L69 3BX

Wiebe van der Hoek
Computer Science Deptartment
University of Liverpool
Liverpool, L69 3BX

Michael Wooldridge
Computer Science Deptartment
University of Liverpool
Liverpool, L69 3BX

Abstract

We introduce ATLES – a variant of ATL with explicit names for strategies in the object language. ATLES makes it possible to refer to the same strategy in different occurrences of path quantifiers, and, as a consequence, it possible to express in ATLES some properties that cannot be expressed even in ATL*. We present a complete axiomatic system for ATLES. Moreover, we show that satisfiability problem for ATLES is no more complex than for ATL: it is EX-PTIME-complete. We identify two variants of the model-checking problem for ATLES and investigate their computational complexity. Finally, we show how ATLES can be used to reason about extensive games.

1 Introduction and Preliminaries

Alternating-time Temporal Logic (ATL) is a logic in which one can reason about the strategic abilities of players [1]. It provides an elegant and popular representation formalism to specify game-like, multi-agent systems. The key construct in ATL is of the form $\langle\!\langle A \rangle\!\rangle \varphi$, expressing that coalition A has a strategy so as to ensure that the temporal property φ holds.

It has been long recognized ([9, 2]) that the semantics of ATL is richer than its language: whereas in the semantics one can explicitly reason about strategies, in the language one can only quantify over them. The aim of this paper is to present *Alternating-time Logic with Explicit Strategies* (ATLES), a logic intended to bridge this gap. This logic extends both CATL [9] and the 'Action Logic' (AL) of [2]. After introducing ATLES in this section, in Section 2, we investigate its expressivity, while, in Section 3, we formulate two variants of the model-checking problem for ATLES, establish their computational complexity, and also give an axiomatization that is complete with respect to ATLES semantics. We also settle the complexity of the satisfiability problem. Section 4 rounds off with conclusions, related work, and an outlook.

We now introduce ATL with explicit names for strategies. The key idea is to replace the cooperation modalities $\langle\!\langle A \rangle\!\rangle$ of ATL with constructions of the form $\langle\!\langle A \rangle\!\rangle_\rho$, where ρ is a *commitment function*. A commitment function, ρ, is a partial function mapping agents to strategy terms. That is, each agent b for which ρ is defined (i.e., $b \in \mathrm{dom}(\rho)$) commits to the strategy $\rho(b)$. Then $\langle\!\langle A \rangle\!\rangle_\rho \varphi$ means that 'while the agents in $\mathrm{dom}(\rho)$ act according to their commitments, the coalition A can cooperate to ensure φ as an outcome'.

Our work can be seen as a revised version of the logic CATL [9], which included commitment operators of the form $C_a(\varrho, \varphi)$, meaning 'under the assumption that a commits to strategy ϱ, then φ holds'. The operators were interpreted using an *update semantics*, which had the effect that, once an agent committed to a strategy, she could not change that commitment. Here, we use a different semantics, which does not make this assumption of irrevocable commitment. Moreover, in the strategies, we allow for agents with perfect information, i.e., making a choice depends on the full history of previously traversed states. Finally, CATL was defined for a fixed set of agents and a fixed set of strategy terms [9]. In this work, however, (as

in [11] for ATL), we do not fix the agents and strategy terms in advance.

1.1 ATLES

Definition 1.1 [ATLES Syntax] Let $\mathbf{\Pi}$ be a countable infinite set of *atomic propositions*, $\mathbf{\Sigma}$ a countable infinite set of *agents*, and $\mathbf{\Upsilon}$ a set of *strategy terms* with $\mathbf{\Upsilon} = \bigcup_{a \in \mathbf{\Sigma}} \mathbf{\Upsilon}_a$, where $\mathbf{\Upsilon}_a$ is a countable infinite set of *strategy terms for agent a*. A *coalition* is a finite set $A \subset \mathbf{\Sigma}$ of agents. A *commitment function* is a partial function $\rho : \mathbf{\Sigma} \to \mathbf{\Upsilon}$ with a finite domain, which maps an agent $a \in \mathbf{\Sigma}$ to a strategy term $\rho(a) \in \mathbf{\Upsilon}_a$ for a. The set of ATLES-formulas is generated by the following grammar, where $p \in \mathbf{\Pi}$, A ranges over coalitions, ρ over commitment functions and φ over ATLES-formulas:

$$\varphi := \quad p \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle\!\langle A \rangle\!\rangle_\rho \bigcirc \varphi \mid \langle\!\langle A \rangle\!\rangle_\rho \Box \varphi \mid$$
$$\langle\!\langle A \rangle\!\rangle_\rho \varphi \mathcal{U} \varphi.$$

Logical truth ($\top$), the Boolean connectives ($\wedge$, $\to$, and $\leftrightarrow$) and $\diamondsuit$ are defined as usual. Observe that, in ATLES, the operators $\langle\!\langle A \rangle\!\rangle_\rho \Box$ are explicitly defined in the syntax, since $\langle\!\langle A \rangle\!\rangle_\rho \Box$ cannot be expressed in terms of $\langle\!\langle A \rangle\!\rangle_\rho$ and $\mathcal{U}$ within ATLES. ATL is the fragment of ATLES only allowing for commitment functions that are undefined for all agents.

As semantic structures, an extension of alternating transition systems were suggested in [9] that explicitly account for actions and action pre-conditions, so-called action-based alternating transition systems. In this paper, however, we confine ourselves to a variant of the alternating transition systems introduced in [1] extended with strategy terms and a denotation function mapping strategy terms to strategies. These transition systems can easily be seen to be equivalent to the action-based structures of [9].

Definition 1.2 [Alternating Transition System] Let $\Sigma = \{1, \dots, n\} \subset \mathbf{\Sigma}$, with $n \geq 1$, be a finite set of agents and, for each agent $i \in \Sigma$, $\Upsilon_i \subset \mathbf{\Upsilon}_i$ a finite set of i-strategy terms. An *alternating transition system (ATS) for* Σ is a tuple $\mathcal{S} = \langle \Pi, \Sigma, Q, \Upsilon_1, \dots, \Upsilon_n, \pi, \delta, ||\cdot|| \rangle$ where

- $\Pi \subseteq \mathbf{\Pi}$ is a finite, non-empty set of *atomic propositions*,

- Q is a finite, non-empty set of *states*,

- $\pi : Q \to 2^\Pi$ is a *valuation function* which assigns to every state a set of atomic propositions that are true there,

- $\delta : Q \times \Sigma \to 2^{2^Q}$ is a *transition function* which maps a state $q \in Q$ and an agent $a \in \Sigma$ to a non-empty set of *choices* $\delta(q, a)$ available to a at q such that the following condition is satisfied: for every state $q \in Q$ and every set $Q_1, \dots, Q_n$, where n is the number of agents, of choices $Q_i \in \delta(q, i)$, $1 \leq i \leq n$, the intersection $Q_1 \cap \cdots \cap Q_n$ is a singleton set, and

- $||\cdot|| : \Upsilon \to (Q^+ \to 2^Q)$ is a *denotation function*, where $\Upsilon = \bigcup_{i \in \Sigma} \Upsilon_i$, which, for each agent $a \in \Sigma$, maps an a-strategy term to an a-strategy (to be defined below).

Intuitively, $\delta(q, a)$ describes the a-*choices* available in q: when in state q, agent a chooses a set from $\delta(q, a)$ to ensure that the "next state" will be among those in the chosen set. This notion of a-choices is generalized to A-*choices* for coalitions A of agents as follows: Given an ATS $\mathcal{S} = \langle \Pi, \Sigma, Q, \Upsilon_1, \dots, \Upsilon_n, \pi, \delta, ||\cdot|| \rangle$, for each state $q \in Q$ and each coalition $A \subseteq \Sigma$, set $\delta(q, A)$ to be the set $\{Q_A \subseteq Q \mid Q_A = \bigcap_{a \in A} Q_a \text{ where } Q_a \in \delta(q, a)\}$ if $A \neq \emptyset$, or otherwise, i.e., if $A = \emptyset$, the set $\{\bigcup \delta(q, \Sigma)\}$. When in state q, the agents in coalition A collectively choose a set from $\delta(q, A)$ to ensure that the "next state" is from this set. Note that $\delta(q, A)$ is non-empty for each state q and coalition A, and $\delta(q, \Sigma)$ is a set of singletons. The states in the singleton sets of $\delta(q, \Sigma)$ are the *successors* of q, i.e., the system is completely determined when all the agents have made their choice. Since the empty coalition cannot influence the behavior of the system, $\delta(q, \emptyset)$ is set to $\bigcup \delta(q, \Sigma)$, the set of all possible successors of q.

An infinite sequence $\lambda = q_0 q_1 q_2 \cdots \in Q^\omega$ of states is a *computation* if, for all positions $i \geq 0$, there is a choice $\{q_{i+1}\} \in \delta(q_i, \Sigma)$. Denote with $\lambda[i]$ the i-th component q_i in λ, and with $\lambda[0, i]$ the initial sequence $q_0 \cdots q_i$ of λ.

A *strategy* for an agent $a \in \Sigma$ is a function $\sigma_a : Q^+ \to 2^Q$ that maps all finite sequences $\lambda \cdot q \in Q^+$ of states to a choice $\sigma_a(\lambda \cdot q) \in \delta(q, a)$ available to agent a at q. Note that $\lambda \cdot q$ denotes the concatenation of the finite sequence λ with the state q. A *strategy for a coalition A* is a set of strategies $\sigma_A = \{\sigma_a \mid a \in A\}$,

one for each agent in A. Given a commitment function ρ, we augment the notion of strategies for A to that of a *ρ-strategy for A*. This is a set of strategies containing for each committed agent in $\mathsf{dom}(\rho)$, the strategy she committed to and, for each free agent in $A \setminus \mathsf{dom}(\rho)$, an arbitrary strategy for this agent. Formally, a *ρ-strategy for A* is a strategy $\sigma_{A \cup \mathsf{dom}(\rho)}$ for the agents in $A \cup \mathsf{dom}(\rho)$ such that for all agents $a \in \mathsf{dom}(\rho)$, the strategy σ_a for a in $\sigma_{A \cup \mathsf{dom}(\rho)}$ is such that $\sigma_a = ||\rho(a)||$.

The set $\mathsf{out}(q, \sigma_A)$ of *outcomes* of a strategy σ_A for the agents in A starting at a state q is the set of all computations $\lambda = q_0 q_1 q_2 \cdots \in Q^\omega$ such that $q_0 = q$ and $q_{i+1} \in \bigcap_{\sigma_a \in \sigma_A} \sigma_a(\lambda[0, i])$ for all $i \geq 0$.

Now we can be more precise about the meaning of $\langle\!\langle A \rangle\!\rangle \varphi$:

> $\langle\!\langle A \rangle\!\rangle_\rho \varphi$ means that, given the commitments of the agents $b \in \mathsf{dom}(\rho)$ to use strategy $\rho(b)$, the agents $a \in A \setminus \mathsf{dom}(\rho)$ have a strategy such that, no matter what the agents $c \in \Sigma \setminus (\mathsf{dom}(\rho) \cup A)$ will do, φ will result.

Definition 1.3 [ATLES Semantics] Given an ATS $\mathcal{S} = \langle \Pi, \Sigma, Q, \Upsilon_1, \ldots, \Upsilon_n, \pi, \delta, ||\cdot|| \rangle$, the satisfaction relation $\models$ is inductively defined as follows, where we omit the Boolean cases. Validity and satisfiability can be defined in a standard way.

- $\mathcal{S}, q \models \langle\!\langle A \rangle\!\rangle_\rho \bigcirc \varphi$ iff there is a ρ-strategy $\sigma_{A \cup \mathsf{dom}(\rho)}$ for the agents in $A \cup \mathsf{dom}(\rho)$ such that for all computations $\lambda \in \mathsf{out}(q, \sigma_{A \cup \mathsf{dom}(\rho)})$, it holds that $\mathcal{S}, \lambda[1] \models \varphi$;

- $\mathcal{S}, q \models \langle\!\langle A \rangle\!\rangle_\rho \Box \varphi$ iff there is a ρ-strategy $\sigma_{A \cup \mathsf{dom}(\rho)}$ for the agents in $A \cup \mathsf{dom}(\rho)$ such that for all computations $\lambda \in \mathsf{out}(q, \sigma_{A \cup \mathsf{dom}(\rho)})$, it holds that $\mathcal{S}, \lambda[i] \models \varphi$ for all positions $i \geq 0$;

- $\mathcal{S}, q \models \langle\!\langle A \rangle\!\rangle_\rho \psi \mathcal{U} \varphi$ iff there is a ρ-strategy $\sigma_{A \cup \mathsf{dom}(\rho)}$ for the agents in $A \cup \mathsf{dom}(\rho)$ such that for all computations $\lambda \in \mathsf{out}(q, \sigma_{A \cup \mathsf{dom}(\rho)})$, there is a position $i \geq 0$ such that $\mathcal{S}, \lambda[i] \models \varphi$ and $\mathcal{S}, \lambda[j] \models \psi$ for all positions j with $0 \leq j < i$.

1.2 Axiomatic system for ATLES

In this section, we present an axiomatic system for ATLES: Figure 1 contains the axioms and inference rules. The notions of ATLES-provability and consistency are defined as usual.

The axioms and the inference rules were inspired by the axiomatization of Coalition Logic in [7] and of ATL in [6] and extended to ATLES. In particular, we added three new axioms (C1) to (C3) that characterize the expressivity of the commitment function. Intuitively, these three axioms express the following:

> (C1): given a commitment ρ, a coalition A can still ensure φ at the next state after an agent outside of A commits to a strategy;

> (C2): given a commitment ρ', a coalition A can still ensure φ at the next state after a member of A dismisses her commitment;

> (C3): given a commitment ρ, after a committed agent a has left the coalition $A \cup \{a\}$, the remaining agents in A are still able to ensure φ at the next state.

The following lemma formalizes the relationship between next formulas with different coalitions and commitment functions by combining the axioms (C1) to (C3). It characterizes the conditions under which the power of one coalition A, given some commitments ρ, can be "transferred" to the power of B, by assuming commitments ξ. That is, it characterizes when a formula of the form $\langle\!\langle A \rangle\!\rangle_\rho \bigcirc \psi \rightarrow \langle\!\langle B \rangle\!\rangle_\xi \bigcirc \psi$ can be derived. Intuitively, the implication in Lemma 1.4 can be seen as a course of action where agents join or leave the coalition and agents take or dismiss commitments.

Lemma 1.4 *Let A, B be two coalitions of agents and ρ, ξ two commitment functions. Then, for all ATLES-formulas ψ, it holds that*

$$\vdash \langle\!\langle A \rangle\!\rangle_\rho \bigcirc \psi \rightarrow \langle\!\langle B \rangle\!\rangle_\xi \bigcirc \psi$$

if, and only if, the following four conditions are satisfied:

(P0) $\rho(a) = \xi(a)$, *for all* $a \in \mathsf{dom}(\rho) \cap \mathsf{dom}(\xi)$;

(P1) $(\mathsf{dom}(\xi) \setminus \mathsf{dom}(\rho)) \cap A = \emptyset$;

(P2) $\mathsf{dom}(\rho) \setminus \mathsf{dom}(\xi) \subseteq B$;

(P3) $A \setminus B \subseteq \mathsf{dom}(\rho) \cap \mathsf{dom}(\xi)$.

(TAUT)	Propositional tautologies
($\bot$)	$\neg\langle\!\langle A\rangle\!\rangle_\rho\bigcirc\bot$
($\top$)	$\langle\!\langle A\rangle\!\rangle_\rho\bigcirc\top$
(S)	$\langle\!\langle A\rangle\!\rangle_\rho\bigcirc\varphi \wedge \langle\!\langle B\rangle\!\rangle_\rho\bigcirc\psi \rightarrow \langle\!\langle A\cup B\rangle\!\rangle_\rho\bigcirc(\varphi\wedge\psi)$ where $A\cap B\subseteq \mathsf{dom}(\rho)$
(C1)	$\langle\!\langle A\rangle\!\rangle_\rho\bigcirc\varphi \rightarrow \langle\!\langle A\rangle\!\rangle_{\rho'}\bigcirc\varphi$ where $\rho' = \rho\cup\{a\mapsto\varrho_a\}, a\notin A, \varrho_a\in\Upsilon_a$
(C2)	$\langle\!\langle A\rangle\!\rangle_{\rho'}\bigcirc\varphi \rightarrow \langle\!\langle A\rangle\!\rangle_\rho\bigcirc\varphi$ where $\rho' = \rho\cup\{a\mapsto\varrho_a\}, a\in A, \varrho_a\in\Upsilon_a$
(C3)	$\langle\!\langle A\cup\{a\}\rangle\!\rangle_\rho\bigcirc\varphi \rightarrow \langle\!\langle A\rangle\!\rangle_\rho\bigcirc\varphi$ where $a\in\mathsf{dom}(\rho)$
(FP$_\Box$)	$\langle\!\langle A\rangle\!\rangle_\rho\Box\varphi \leftrightarrow \varphi \wedge \langle\!\langle A\rangle\!\rangle_\rho\bigcirc\langle\!\langle A\rangle\!\rangle_\rho\Box\varphi$
(GFP$_\Box$)	$\langle\!\langle\emptyset\rangle\!\rangle_\rho\Box(\theta\rightarrow(\varphi\wedge\langle\!\langle A\rangle\!\rangle_\rho\bigcirc\theta)) \rightarrow \langle\!\langle\emptyset\rangle\!\rangle_\rho\Box(\theta\rightarrow\langle\!\langle A\rangle\!\rangle_\rho\Box\varphi)$
(FP$_\mathcal{U}$)	$\langle\!\langle A\rangle\!\rangle_\rho\psi\,\mathcal{U}\,\varphi \leftrightarrow \varphi \vee (\psi\wedge\langle\!\langle A\rangle\!\rangle_\rho\bigcirc\langle\!\langle A\rangle\!\rangle_\rho\psi\,\mathcal{U}\,\varphi)$
(LFP$_\mathcal{U}$)	$\langle\!\langle\emptyset\rangle\!\rangle_\rho\Box((\varphi\vee(\psi\wedge\langle\!\langle A\rangle\!\rangle_\rho\bigcirc\theta))\rightarrow\theta) \rightarrow \langle\!\langle\emptyset\rangle\!\rangle_\rho\Box(\langle\!\langle A\rangle\!\rangle_\rho\psi\,\mathcal{U}\,\varphi\rightarrow\theta)$

(Modus Ponens) $\dfrac{\varphi,\ \varphi\rightarrow\psi}{\psi}$ ($\langle\!\langle\emptyset\rangle\!\rangle_\rho\Box$-Necessitation) $\dfrac{\varphi}{\langle\!\langle\emptyset\rangle\!\rangle_\rho\Box\varphi}$

($\langle\!\langle A\rangle\!\rangle_\rho\bigcirc$-Monotonicity) $\dfrac{\varphi\rightarrow\psi}{\langle\!\langle A\rangle\!\rangle_\rho\bigcirc\varphi\rightarrow\langle\!\langle A\rangle\!\rangle_\rho\bigcirc\psi}$

Figure 1: Axioms and Inference Rules for ATLES.

2 Expressivity

ATLES seems more expressive than its fragment ATL, but exactly how does its expressivity compare to that of ATL*? Let us first consider some examples. Consider the ATS $\mathcal{S}$ in Figure 2. In ATL, one cannot distinguish between using general strategies as we defined them (functions from finite sequences to choices), and *memoryless strategies* $\sigma : Q \rightarrow 2^Q$ (cf. [1, 8]). Hence, one immediately sees that in $\mathcal{S}$, our agent has no strategy to ensure that both $p \wedge q$ and $(\neg p \wedge \neg q)$ will be visited infinitely often. Such a liveness condition can be expressed in ATLES by $\langle\!\langle a\rangle\!\rangle_\rho\Box(\langle\!\langle a\rangle\!\rangle_\rho\Diamond(p\wedge q) \wedge \langle\!\langle a\rangle\!\rangle_\rho\Diamond(\neg p\wedge\neg q))$, for a suitable ρ. This example also shows that ATLES *is* expressive enough to distinguish memoryless strategies from general ones.

We can use the same ATS to point at an important difference with CATL: let $free$ denote the property $\langle\!\langle a\rangle\!\rangle\bigcirc p \wedge \langle\!\langle a\rangle\!\rangle\bigcirc\neg p$: agent a can next guarantee p, but also can she guarantee $\neg p$. In CATL, commitment to a strategy ρ is *irrevocable*: one has, in our ATS, $\mathcal{S}, x \models \langle\!\langle a\rangle\!\rangle\bigcirc free \wedge \neg C_a(\sigma,\bigcirc free)$: agent a does
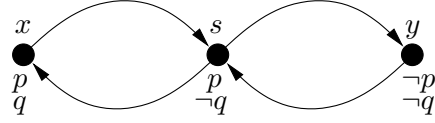

Figure 2: A single agent ATS.

not have a strategy that allows him to still make up his mind in the next state, s. Compare this to ATLES, where we have $\mathcal{S}, x \models \langle\!\langle a\rangle\!\rangle\bigcirc free \wedge \langle\!\langle a\rangle\!\rangle_\rho\bigcirc free$, for every ρ.

Recall that, other than in ATLES, in ATL* we can express Boolean combinations and nesting of temporal operators inside a path quantifier. Some of these formulas, however, can be translated into ATLES in a satisfiability preserving way. For instance, the ATL*-formula $\langle\!\langle a,b\rangle\!\rangle\bigcirc^n\varphi$, $n \geq 0$, with an n-fold nesting of the temporal operator next-time ($\bigcirc$) can be translated into ATLES as:

$$\underbrace{\langle\!\langle a,b\rangle\!\rangle_{\{a\mapsto\varrho_a,b\mapsto\varrho_b\}}\bigcirc\cdots\langle\!\langle a,b\rangle\!\rangle_{\{a\mapsto\varrho_a,b\mapsto\varrho_b\}}\bigcirc}_{n-\text{times}}\varphi$$

where ϱ_a, ϱ_b are fresh strategy terms for the agents a, b, respectively. Note that with the translation, we have an exponential blow-up in the formula size if n is coded in binary.

Here are three more ATL*-formulas with nesting of temporal operators that can be translated into ATLES: $\langle\langle a,b\rangle\rangle[\varphi\,\mathcal{U}(\psi\,\mathcal{U}\,\vartheta)]$, $\langle\langle a,b\rangle\rangle[\varphi\,\mathcal{U}(\bigcirc\psi)]$, and $\langle\langle a,b\rangle\rangle\bigcirc(\varphi\,\mathcal{U}\,\psi)$ can, respectively (in this order), be expressed as

$$\langle\langle a,b\rangle\rangle_{\{a\mapsto\varrho_a,b\mapsto\varrho_b\}}[\varphi\,\mathcal{U}(\langle\langle a,b\rangle\rangle_{\{a\mapsto\varrho_a,b\mapsto\varrho_b\}}(\psi\,\mathcal{U}\,\vartheta))],$$
$$\langle\langle a,b\rangle\rangle_{\{a\mapsto\varrho_a,b\mapsto\varrho_b\}}[\varphi\,\mathcal{U}(\langle\langle a,b\rangle\rangle_{\{a\mapsto\varrho_a,b\mapsto\varrho_b\}}\bigcirc\psi)],$$
$$\langle\langle a,b\rangle\rangle_{\{a\mapsto\varrho_a,b\mapsto\varrho_b\}}\bigcirc\langle\langle a,b\rangle\rangle_{\{a\mapsto\varrho_a,b\mapsto\varrho_b\}}(\varphi\,\mathcal{U}\,\psi).$$

To pinpoint the relationship between ATLES and ATL* more precisely, we now define a translation function $(\cdot)^*$ mapping some ATL*-formulas φ to formulas $(\varphi)^*$ of ATLES such that φ is satisfiable if, and only if, $(\varphi)^*$ is satisfiable. To this end, let us first recall the syntax of ATL* with this EBNF notation (for a complete definition of ATL* see, e.g., [1]):

state formulas: $\varphi ::= \quad p \quad | \quad \varphi\vee\varphi \quad | \quad \neg\varphi \quad | \quad \langle\langle A\rangle\rangle\psi$
path formulas: $\psi ::= \quad \varphi \quad | \quad \psi\vee\psi \mid \psi\wedge\psi \mid \neg\psi \mid$
$\qquad\qquad\qquad \bigcirc\psi \mid \psi\,\mathcal{U}\,\psi$

We restrict the translation function to formulas in *negation normal form*, where negation only occurs in front of propositional variables or path quantifiers. To understand the following definition of the translation function, the distinction between ATL*-state formulas and path formulas is important.

Definition 2.1 We define a partial function $(\cdot)^*_{\langle B,\xi\rangle}$ on ATL* state and path formulas as follows, where B ranges over coalitions and ξ over commitment functions. For all state formulas p, φ, φ' and all path formulas ψ, ψ' in negation normal form:

$$
\begin{aligned}
(p)^*_{\langle B,\xi\rangle} &:= p,\text{ for } p\in\mathbf{\Pi};\\
(\neg\varphi)^*_{\langle B,\xi\rangle} &:= \neg(\varphi)^*_{\langle B,\xi\rangle};\\
(\varphi\vee\varphi')^*_{\langle B,\xi\rangle} &:= (\varphi)^*_{\langle B,\xi\rangle}\vee(\varphi')^*_{\langle B,\xi\rangle};\\
(\langle\langle A\rangle\rangle\psi)^*_{\langle B,\xi\rangle} &:= (\psi)^*_{\langle A,\rho\rangle},\text{ for commitment}\\
&\qquad\text{function }\rho\text{ with }\mathrm{dom}(\rho)=A\\
&\qquad\text{and the range of }\rho\text{ containing}\\
&\qquad\text{only fresh strategy terms;}\\
(\psi\wedge\psi')^*_{\langle B,\xi\rangle} &:= (\psi)^*_{\langle B,\xi\rangle}\wedge(\psi')^*_{\langle B,\xi\rangle};\\
(\bigcirc\psi)^*_{\langle B,\xi\rangle} &:= \langle\langle B\rangle\rangle_\xi\bigcirc(\psi)^*_{\langle B,\xi\rangle};\\
(\neg\bigcirc\psi)^*_{\langle B,\xi\rangle} &:= (\bigcirc\neg\psi)^*_{\langle B,\xi\rangle};\\
(\psi\,\mathcal{U}\,\psi')^*_{\langle B,\xi\rangle} &:= \langle\langle B\rangle\rangle_\xi((\psi)^*_{\langle B,\xi\rangle}\,\mathcal{U}(\psi')^*_{\langle B,\xi\rangle}).
\end{aligned}
$$

The following lemma establishes that the translation function is satisfiability preserving; the proof is left to the reader.

Lemma 2.2 *Let E be the empty coalition and ξ_0 the empty commitment function. For all ATL*-formulas φ in negation normal form such that the translation function $(\cdot)^*_{\langle E,\xi_0\rangle}$ is defined for φ, the following are equivalent:*

(a) φ *is satisfiable wrt. ATL*;*

(b) $(\varphi)^*_{\langle E,\xi_0\rangle}$ *is satisfiable wrt. ATLES.*

Note that $(\cdot)^*$ determines only a fragment of ATL* that can be translated into ATLES in a satisfiability preserving way. This was to be expected since ATLES is not expressive enough to subsume full ATL*. In particular, ATL*-formulas with a disjunction of temporal operators inside a path quantifier cannot be translated. For instance, ATLES cannot express the ATL*-formula

$$\langle\langle A\rangle\rangle(\psi\vee\psi')$$

stating that coalition A has a strategy to ensure computations on which ψ or ψ' holds. In fact, ATLES and ATL* are incomparable with respect to expressivity: some formulae of ATL* cannot be expressed in ATLES, while some formulae of ATLES cannot be expressed in ATL*.

2.1 ATLES **for Extensive Games**

Where CATL seems to provide an appropriate machinery for reasoning about *strategic* games [9], ATLES is well suited to generalize this to *extensive* games. Due to space restrictions, this is only sketched here. Consider the two-player extensive game G in Figure 3. Suppose player b has to move at nodes y_1 and y_2, whereas the remaining non-terminal nodes are a's. Note that a has 16 strategies, while player b has 4 strategies. We write $llrl$ to denote a strategy for agent a where this player will choose 'left' at the nodes x_1, x_2, and x_4, but 'right' at node x_3. We identify a strategy with a term denoting it: let Υ_i be the strategies for player i. Let the denotation function $j\mapsto\alpha; i\mapsto\beta$ assign strategy α to player j, and β to player i. A strategy profile is a pair $\sigma=\langle\alpha,\beta\rangle$ and this uniquely determines an outcome. The vectors at the terminal nodes indicate the payoff for the
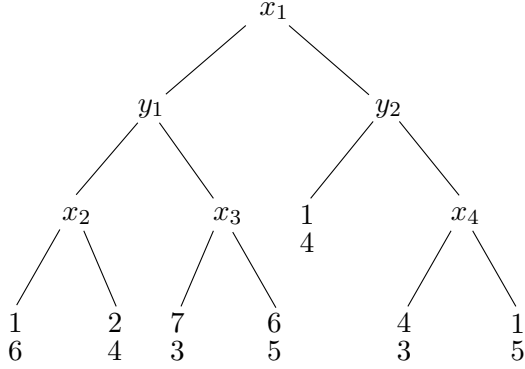
Figure 3: An extensive game.

players: the upper value is the payoff for a and the lower one for b. To reason about the payoffs, let us assume the ATS $\mathcal{S}_G$ that is associated with G interprets atomic propositions $u_i \geq v$, for $i \in \{a, b\}$ and $v \in V = \{0, 1, \ldots, 8\}$, denoting that player i's payoff is at least v. We can express in ATLES that at each terminal state in $\mathcal{S}_G$, the truth of these atoms corresponds with Figure 3, for the other states, they are all false.

We now can explicitly reason about properties of strategies in $\mathcal{S}_G$. In the following, assume $i \neq j$ and that formulas are evaluated in the root x_1 of $\mathcal{S}_G$, unless stated otherwise. A *weakly dominant strategy* for player i is one which is at least as good as any other strategy, no matter what player j does. This notion of weak dominance is captured by

$$WD_i(\alpha) \quad := \bigwedge_{\beta \in \Upsilon_j, v \in V} \left(\langle\!\langle i \rangle\!\rangle_{\{j \mapsto \beta\}} \Diamond (u_i \geq v) \rightarrow \right.$$
$$\left. \langle\!\langle\rangle\!\rangle_{\{i \mapsto \alpha; j \mapsto \beta\}} \Diamond (u_i \geq v) \right).$$

We claim that, in a game G, strategy α is a weakly dominant strategy for player i if, and only if, $WD_i(\alpha)$ is satisfied at x_1 in $\mathcal{S}_G$. A similar correspondence can be given for *dominated strategies*. We move on to characterize Nash Equilibria. The predicate $BR_i(\alpha_j, \beta_i)$ expresses that for player i, given the strategy α_j of player j, β_i is the best response.

$$BR_i(\alpha_j, \beta_i) \quad := \bigwedge_{v \in V} \left(\langle\!\langle i \rangle\!\rangle_{\{j \mapsto \alpha_j\}} \Diamond (u_i \geq v) \rightarrow \right.$$
$$\left. \langle\!\langle\rangle\!\rangle_{\{i \mapsto \beta_i; j \mapsto \alpha_j\}} \Diamond (u_i \geq v) \right)$$

This concept can be used to define that of a *Nash*

Equilibrium, in a straightforward way. We claim that $NE(\langle \alpha, \beta \rangle)$ holds at x_1 in $\mathcal{S}_G$ if, and only if, the strategy profile $\langle \alpha, \beta \rangle$ is a Nash Equilibrium of G:

$$NE(\langle \alpha, \beta \rangle) \quad := \quad BR_1(\beta, \alpha) \wedge BR_2(\alpha, \beta).$$

We can now also characterize a *subgame perfect Nash Equilibrium* as follows:

$$SPNE(\langle \alpha, \beta \rangle) \quad := \quad \langle\!\langle\rangle\!\rangle \Box NE(\langle \alpha, \beta \rangle).$$

In our game G, we then have

$$\mathcal{S}_G, x_1 \quad \models \quad \neg NE(\langle lrlr, LR \rangle)$$
$$\wedge \neg SPNE(\langle lrlr, LR \rangle)$$
$$\wedge SPNE(\langle lrll, LL \rangle).$$

To see that $\mathcal{S}_G, x_1 \models \neg SPNE(\langle lrlr, LR \rangle)$, note that $\mathcal{S}_G, x_3 \models \neg NE(\langle lrlr, LR \rangle)$. In fact, *backward induction*, a procedure to find the subgame perfect Nash Equilibrium of an extensive game, can be modeled in ATLES. Let $rat(i, \alpha)$ denote that α is a *rational* strategy for i, given the current state, and that j is also rational in all next states. Let the variable $term$ be true exactly at all terminal states. We define $rat(i, \alpha)$ as the disjunction of $term$ and the formula

$$\bigvee_{v \in V} \Big[\bigvee_{\beta \in \Upsilon_j} \langle\!\langle\rangle\!\rangle_{\{i \mapsto \alpha\}} \bigcirc (rat(j, \beta) \wedge$$
$$\langle\!\langle\rangle\!\rangle_{\{i \mapsto \alpha; j \mapsto \beta\}} \Diamond (u_i \geq v)) \wedge$$
$$\bigwedge_{\alpha_i \in \Upsilon_i, \beta_j \in \Upsilon_j} \langle\!\langle\rangle\!\rangle_{\{i \mapsto \alpha_i\}} \bigcirc (rat(j, \beta_j) \rightarrow$$
$$\langle\!\langle\rangle\!\rangle_{\{i \mapsto \alpha_i; j \mapsto \beta_j\}} \Box \neg (u_i > v)) \Big].$$

This states that it is rational for i to play α. That is, if we are not in a terminal state, playing α leads to a state, which, assuming rationality of j, gives i a payoff of at least v that can not be improved upon by any other of i's strategies α_i, given that j plays a rational strategy β_j along with it. Note that we have $\mathcal{S}_G, x_4 \models rat(a, \gamma)$ for any a-strategy γ ending in l (it is rational for a to play at x_4 any strategy that dictates 'left' at x_4), whereas, at x_1 in $\mathcal{S}_G$, only strategy $\gamma' = lrll$ qualifies for $\mathcal{S}_G, x_1 \models rat(a, \gamma')$.

A similar analysis can be given for Pareto optimal strategies and other solution concepts. Again, note that, in ATLES, we can express a switch from one strategy to another: a property like $\mathcal{S}_G, x_1 \models \langle\!\langle\rangle\!\rangle_{\{a \mapsto lllr\}} \bigcirc \langle\!\langle a \rangle\!\rangle_{\{\}} (u_a \geq 2)$ says that even if agent a commits to the strategy $lllr$ at x_1 (which only would guarantee her a payoff of 1), in the next step, a can change her mind and still guarantee a payoff of 2.

3 Complexity and Completeness

3.1 ATLES Model-Checking

In this section, we discuss two model-checking algorithms for ATLES. Generally, the *model-checking problem* is, given a formula φ and an ATS $\mathcal{S}$ (a *model*), to compute the set of states of $\mathcal{S}$ that satisfy φ. When model-checking an ATLES-formula, we have to take the strategies into account that come with an ATS. However, it appears to be also an interesting problem to consider the possibility that strategies are not given as part of the input (cf. the 'model-checking as planning' paradigm [5]). Notice that, for model-checking, we confine ourselves to memoryless strategies. With this in mind, we now formulate two variations of the model-checking problem for ATLES:

(a) *Model-checking with given strategies*

Given an ATLES-formula φ, an ATS $\mathcal{S} = \langle \Pi, \Sigma, Q, \Upsilon_1, \ldots, \Upsilon_n, \pi, \delta, ||\cdot|| \rangle$ where the set $\Sigma = \{1, \ldots, n\}$ contains n agents, and a state $q \in Q$, is φ satisfied at q in $\mathcal{S}$?

(b) *Model-checking along with generating strategies*

Given an ATLES-formula φ, an ATS $\mathcal{S} = \langle \Pi, \Sigma, Q, \pi, \delta \rangle$, and a state $q \in Q$, are there strategies, one for each strategy term that occurs in φ, such that φ is satisfied at q in $\mathcal{S}$ augmented with these strategies?

For deciding variant (a) of the problem, we use a modified version of the symbolic model-checking algorithm for ATL from [1]. Figure 4 presents the function ATLES$-eval(\cdots)$ hat solves variant (a) of the model-checking problem for ATLES.

Consider an ATS $\mathcal{S} = \langle \Pi, \Sigma, Q, \Upsilon_1, \ldots, \Upsilon_n, \pi, \delta, ||\cdot|| \rangle$, a state q in $\mathcal{S}$, and an ATLES-formula φ as input. Take Σ to be the set containing all agents occurring in φ, and, for each agent $a \in \Sigma$, let Υ_a be the set containing all a-strategy terms of φ. For suppose otherwise, $\mathcal{S}$ cannot be a model for φ. We denote with Ξ_φ the set of all commitment functions occurring in φ.

The algorithm computes using a bottom-up approach, for each subformula ψ of φ, its extension $[\![\psi]\!]$ in $\mathcal{S}$, a set of states from $\mathcal{S}$ that all satisfy ψ. For computing the extension of formulas of the form $\langle\!\langle A \rangle\!\rangle_\xi \bigcirc \psi$,

$\langle\!\langle A \rangle\!\rangle_\xi \square \psi$, or $\langle\!\langle A \rangle\!\rangle_\xi \psi \, \mathcal{U} \, \vartheta$, we employ a modified pre-image operator Pre' that additionally accounts for the commitments of agents to strategies as specified in ξ. The function $Pre'(\cdots)$ maps a coalition A, a commitment function ξ, and a set Q' of states to the set $Pre'(A, \xi, Q')$ containing states q at which the coalition $A \cup \mathsf{dom}(\xi)$ can make a choice, while respecting the commitments in ξ, to ensure the next state to lie in Q'. More precisely, at q, the agents a in $A \setminus \mathsf{dom}(\xi)$ can select a choice in $\delta(a, q)$, and the agents b in $\mathsf{dom}(\xi)$ select the choice $\sigma_b(q) \in \delta(a, q)$ according to the b-strategy $\sigma_b = ||\xi(b)||$ such that, for all possible choices made by the other agents in $\Sigma \setminus (A \cup \mathsf{dom}(\xi))$, the resulting successor state is in Q'. Formally, $Pre'(\cdots)$ is defined as follows: for all $A \subseteq \Sigma$, all $\xi \in \Xi_\varphi$, all $Q' \subseteq Q$, and all $q \in Q$, we have $q \in Pre'(A, \xi, Q')$ if, and only if, there is a choice $C \in \delta(A \cup \mathsf{dom}(\xi), q)$ such that $C \subseteq Q'$ and $C \subseteq \bigcap_{a \in \mathsf{dom}(\xi)} ||\xi(a)||(q)$.

This modification of the function Pre does not affect the complexity of the model-checking algorithm. Hence, the variant (a) of ATLES model-checking is no more complex than model-checking ATL.

Theorem 3.1 *The variant (a) of the model-checking problem for ATLES is* PTIME-*complete, and can be solved in time* $\mathcal{O}(m \cdot \ell)$ *for an ATS with m transitions and an* ATLES-*formula of length ℓ.*

An algorithm deciding variant (b) of the ATLES model-checking problem, needs to generate the strategies for the strategy terms occurring in the input formula. However, we can make use of the algorithm for variant (a) as follows: we first non-deterministically guess the required strategies with which we augment the model. In the second step, we use the polynomial time algorithm from (a) to model-check the input formula on the augmented model. We obtain the following result.

Theorem 3.2 *The variant (b) of the model-checking problem for ATLES is* NP-*complete in the number of transitions of the given ATS and in the length of the input formula.*

Proof. The upper bound can easily be seen. Let φ be an ATLES-formula of length ℓ and $\mathcal{S}$ an ATS for the agents occurring in φ. Guessing the strategies can be done in polynomial time in the number of transitions

```
1.  function ATLES−eval(ψ, S = ⟨Π, Σ, Q, {Υ_a}_{a∈Σ}, π, δ, ||·||⟩) returns the extension [[ψ]] in S
2.      case ψ = p:                return π(p)
3.      case ψ = ¬ϑ:               return Q \ ATLES−eval(ϑ, S)
4.      case ψ = ϑ_1 ∨ ϑ_2:        return ATLES−eval(ϑ_1, S) ∪ ATLES−eval(ϑ_2, S)
5.      case ψ = ⟨⟨A⟩⟩_ξ ◯ϑ:       return Pre'(A, ξ, ATLES−eval(ϑ, S))
6.      case ψ = ⟨⟨A⟩⟩_ξ □ϑ:       Δ_1 := Q; Δ_2 = ATLES−eval(ϑ, S)
7.                                 while Δ_1 ⊈ Δ_2 do Δ_1 := Δ_2
8.                                     Δ_2 := Pre'(A, ξ, Δ_1) ∩ ATLES−eval(ϑ, S) od
9.                                 return Δ_1
10.     case ψ = ⟨⟨A⟩⟩_ξ ϑ_1 U ϑ_2: Δ_1 := ∅; Δ_2 = ATLES−eval(ϑ_2, S)
11.                                 while Δ_2 ⊈ Δ_1 do Δ_1 := Δ_1 ∪ Δ_2
12.                                     Δ_2 := Pre'(A, ξ, Δ_1) ∩ ATLES−eval(ϑ_1, S) od
13.                                 return Δ_1
14. end-function
```

Figure 4: ATLES symbolic model-checking (variant (a)).

in S: for at most ℓ strategy terms occurring in φ, we guess one strategy. Notice that we confine ourselves to strategies without history which are of polynomial size in the number of transitions in S: such a strategy specifies one choice at every state of S. Since the second step is in PTIME, we obtain an NP algorithm.

In order to show the lower complexity bound, we reduce the well-known NP-hard satisfiability problem for propositional logic to variant (b) of the model-checking problem for ATLES. For NP-hardness, it is sufficient to consider propositional logic formulas φ in conjunctive normal form only [3]. That is, φ is a conjunction of the form $\psi_1 \wedge \cdots \wedge \psi_m$ where each ψ_i (for $i = 1..m$) is a disjunction the form $\vartheta_1^i \vee \cdots \vee \vartheta_{m_i}^i$. Each ϑ_j^i (for $i = 1..m$ and $j = 1..m_i$) is a literal, i.e. a propositional variable or its negation. Let $p_1, \ldots, p_n$ be an enumeration of the propositional variables occurring in φ. Reserve, for each ϑ_j^i with $1 \le i \le m$ and $1 \le j \le m_i$, a fresh propositional variable $p(i, j)$. We now define the ATS $S_\varphi = \langle \Pi, \Sigma, Q, \pi, \delta \rangle$ with

- $\Pi = \{p(i, j) \mid 1 \le i \le m \text{ and } 1 \le j \le m_i\}$;

- $\Sigma = \{a\}$;

- $Q = \{q_0, q_1, q_1', \ldots q_n, q_n'\}$;

- for all $p(i, j) \in \Pi$, set $\pi(p(i, j)) = \{q_k \mid \vartheta_j^i = p_k\} \cup \{q_k' \mid \vartheta_j^i = \neg p_k\}$;

- $\delta(a, q_0) = \{\{q_1\}, \{q_1'\}\}$, for all i with $1 \le i < n$, set $\delta(a, q_i) = \delta(a, q_i') = \{\{q_{i+1}\}, \{q_{i+1}'\}\}$, $\delta(a, q_n) = \{\{q_n\}\}$, and $\delta(a, q_n') = \{\{q_n'\}\}$.

For an illustration of S_φ, see Figure 5. The number of transitions in S_φ is polynomial in the length of φ. To see that, notice that S_φ contains not more than twice times the length of φ many states with, for each state, at most two a-choices. We define the formula
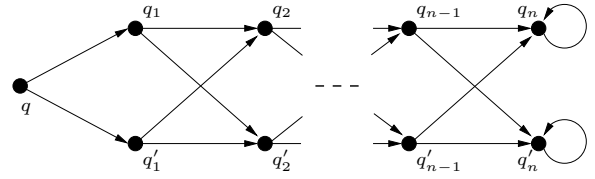

Figure 5: The ATS S_φ for one agent.

$$\varphi_{\text{ATLES}} = \bigwedge_{i=1..m} \left(\bigvee_{j=1..m_i} \langle\langle a \rangle\rangle_{\{a:\varrho_a\}} \Diamond p(i, j) \right).$$

Notice that the length of φ_{ATLES} is polynomial in the length of φ.

The reduction works as follows: φ is satisfiable if, and only if, there is a strategy for agent a such that φ_{ATLES} is satisfied at q_0 in S_φ augmented with that strategy.   QED

3.2 Satisfiability and Completeness

This section investigates the computational complexity of the satisfiability problem for ATLES and completeness of the axiomatic system in Figure 1. The complexity is settled at EXPTIME-complete and thus ATLES is no more complex than ATL. This is done

by adapting a type elimination algorithm for ATLES from [11].

Since, in the definition of ATLES, we do not fix the number of agents in advance, some care is needed when formulating a satisfiability problem for ATLES; cf. a similar consideration for ATL [11]. In particular, the range of semantic structures, over which a formula is to be interpreted, needs to be specified. Such a range can be determined by allowing for a certain number of agents to be present in the semantic structures. To see that allowing for different sets of agents to be present can influence satisfiability, look at the following ATL-formula (adapted from [7, p.47]): $\neg\langle\!\langle a\rangle\!\rangle\bigcirc p \wedge \neg\langle\!\langle a\rangle\!\rangle\bigcirc q \wedge \langle\!\langle a\rangle\!\rangle\bigcirc(p \vee q)$. This formula expresses the fact that, in the next state, agent a cannot make p true, and cannot make q true; but it can make either p or q true. Now the question is whether this formula is satisfiable. The answer is that it is only satisfiable in a semantic structure for more than one agent, and not satisfiable with merely one available agent. Thus the number of agents present in a structure is important for determining satisfiability of a formula in this structure. With these concerns in mind, three variants of the satisfiability problem for ATL were suggested in [11] depending on the possibilities for the number of agents to occur in semantic structures. In this paper, however, we concentrate only on one of these problems; the other two satisfiability problems can be reduced to this one:

> *Satisfiability over formula-defined sets of agents:* Given an ATLES-formula φ, is φ satisfiable in a structure for exactly the agents which occur in φ?

The following theorem states a key result of this paper.

Theorem 3.3 *The satisfiability problem for* CATL *is* EXPTIME-*complete.*

The lower complexity bound carries over from the EXPTIME-hard fragment ATL; see [10, 11]. For the upper bound, a modified version of the type elimination construction from [11] for ATL yields membership in EXPTIME. The main issue that needs to be accounted for is that ATLES allows for commitment of agents to strategies explicitly in its syntax. The details of the construction are rather complex, and we omit them here due to space restrictions[1].

We now argue that the axiomatic system for ATLES presented in Section 1.2 is sound and complete. For showing completeness we employ the type elimination construction used for Theorem 3.3: the structure of the proof is similar to the completeness proof of the axiomatic system for Computation Tree Logic that can be found in [4]. The following three schemes generate valid and provable implications which state properties of ATLES that are used in the proof of completeness: for all commitment functions ρ, we have

Regularity: $\vdash \langle\!\langle A\rangle\!\rangle_\rho\bigcirc\varphi \rightarrow \neg\langle\!\langle B\rangle\!\rangle_\rho\bigcirc\neg\varphi$, for $A \cap B \subseteq \mathsf{dom}(\rho)$;

Coalition-monotonicity: $\vdash \langle\!\langle A\rangle\!\rangle_\rho\bigcirc\varphi \rightarrow \langle\!\langle B\rangle\!\rangle_\rho\bigcirc\varphi$, for $A \subseteq B$;

Property (P): $\vdash \langle\!\langle A\rangle\!\rangle_\rho\bigcirc\varphi \wedge \neg\langle\!\langle A\rangle\!\rangle_\rho\bigcirc\neg\psi \rightarrow \neg\langle\!\langle\emptyset\rangle\!\rangle_\rho\bigcirc\neg(\varphi \wedge \psi)$.

Instances of Regularity are provable using axioms (S) and ($\bot$), and of Coalition-monotonicity by using (S) and ($\top$). Property (P) can be derived using (S) and the inference rule $\langle\!\langle A\rangle\!\rangle_\rho\bigcirc$-Monotonicity. The axiomatic system for ATL [6] contains the axiom (Σ): $\neg\langle\!\langle\emptyset\rangle\!\rangle_\rho\bigcirc\neg\varphi \rightarrow \langle\!\langle\Sigma\rangle\!\rangle_\rho\bigcirc\varphi$. Notice that ($\Sigma$) is not used in the axiomatic system for ATLES since the grand coalition Σ of all agents is not available in ATLES.

Theorem 3.4 *The deductive system for* ATLES *is sound and complete.*

4 Conclusions and Related Work

We introduced ATLES as an extension of ATL with explicit names for strategies. We showed that ATLES satisfiability is EXPTIME-complete by extending the type elimination construction for ATL from [11]. Moreover, we presented a sound and complete axiomatization for ATLES, the completeness proof of which is based on that type elimination construction. ATLES does neither fix the number of agents nor the available strategy terms in advance and thus contains the logics CATL [9], CL (see below), AL (idem), ATL [11], ATL$_\Sigma$ [6] and ATLES$_{\Sigma,\Upsilon}$ as fragments,

---

[1]An extended version of this paper with full proofs is available from the authors.

where the parameter Σ is a fixed set of agents and the parameter Υ a fixed set of strategy terms. Thus the upper complexity bound for ATLES carries over to these fragments whose validities can, moreover, be derived using the axiomatic system for ATLES.

Pauly's Coalition Logic (CL) can be conceived as the one-step fragment of ATL; see [6]. Interestingly enough, Borgo [2] recently presented his Action Logic (AL) which can be seen as the one-step fragment of ATLES. The basic construct in AL is $\vec{v}\varphi$, where $\vec{v}$ is a vector with for every agent i a place that can be filled with either a constant action term a_i, (meaning that agent i has committed to a_i) a quantifier $\exists x_i$ (agent i has a choice to make) or $\forall y_i$ (for all actions of i). The interpretation of $\vec{c}\varphi$ is that under the "assignment" $\vec{c}$, the formula φ will hold in the next state. One easily sees that this would correspond to our $\langle\!\langle A \rangle\!\rangle_\rho \varphi$, where A is the set of agents with an $\exists x_i$ in $\vec{v}$, the function ρ collects all the pairs (i, a_i) with a_i in $\vec{v}$. Summarizing, these logics relate to one another according to this "rule":

$$\text{CL is to ATL} \quad \text{as} \quad \text{AL is to ATLES.}$$

Various ways are conceivable to extend the language of ATLES, an interesting one of which being to allow for composition of strategies. We can model, e.g., non-deterministic strategies with "disjunctions" of strategies: The formula $\langle\!\langle \rangle\!\rangle_{\{a \mapsto \varrho_1 \vee \varrho_2\}} \Phi$ states that agent a uses the non-deterministic strategy $\varrho_1 \vee \varrho_2$, where, at each state, a chooses either according to strategy ϱ_1 or ϱ_2 to bring about the temporal expression Φ. Thus extending ATLES with "disjunction" of strategies and other operators for composing strategies enables us to succinctly express complex properties about strategic ability of agents. We leave the characterization of the gained expressive power and the investigation of the impact on the computational complexity for future work.

References

[1] R. Alur, T. A. Henzinger, and O. Kupferman. Alternating-time temporal logic. *J. ACM*, 49(5):672–713, 2002.

[2] S. Borgo. Coalitions in action logic. In M. M. Veloso, editor, *Proceedings IJCAI'07*, pages 1822–1827, Hyderabad, India, 2007.

[3] S. A. Cook and D. G. Mitchell. Finding hard instances of the satisfiability problem: A survey. In D. Du, J. Gu, and P. M. Pardalos, editors, *Satisfiability Problem: Theory and Applications*, volume 35 of *DIMACS: Series in Discrete Mathematics and Theoretical Computer Science*, pages 1–17, 1997.

[4] E. A. Emerson. Temporal and modal logic. In *Handbook of Theoretical Computer Science (vol. B): Formal Models and Semantics*, pages 995–1072. MIT Press, Cambridge, MA, USA, 1990.

[5] F. Giunchiglia and P. Traverso. Planning as model checking. In S. Biundo and M. Fox, editors, *Recent Advances in AI Planning (LNAI Volume 1809)*, pages 1–20. Springer-Verlag: Berlin, Germany, 1999.

[6] V. Goranko and G. van Drimmelen. Complete axiomatization and decidability of alternating-time temporal logic. *Theoretical Computer Science*, 353(1-3):93–117, 2006.

[7] M. Pauly. *Logic for Social Software*. PhD thesis, University of Amsterdam, 2001. ILLC Dissertation Series 2001-10.

[8] P. Schobbens. Alternating-time logic with imperfect recall. *Electronic Notes in Computer Science*, 85(2):82–93, 2004.

[9] W. van der Hoek, W. Jamroga, and M. Wooldridge. A logic for strategic reasoning. In *AAMAS '05: Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, pages 157–164, New York, NY, USA, 2005. ACM Press.

[10] G. van Drimmelen. Satisfiability in alternating-time temporal logic. In *LICS '03: Proceedings of the 18th Annual IEEE Symposium on Logic in Computer Science, 22–25 June 2003, Ottawa, Canada*, pages 208–217, Washington, DC, USA, 2003. IEEE Computer Society.

[11] D. Walther, C. Lutz, F. Wolter, and M. Wooldridge. ATL satisfiability is indeed Exptime-complete. *Journal of Logic and Computation*, 16:765–787, 2006.