

ATEL with Common and Distributed Knowledge is EXPTIME-Complete

Dirk Walther
University of Liverpool
dirk@csc.liv.ac.uk

Abstract

Alternating-time Temporal Logic (ATL) of Alur, Henzinger, and Kupferman is increasingly widely applied in formal system specification and verification of distributed systems and game-like multi-agent systems. The extension Alternating-time Temporal Epistemic Logic (ATEL) of van der Hoek and Wooldridge of ATL with knowledge modalities widens the scope of reasoning over multi-agent systems with epistemic notions. In this paper, the complexity of the satisfiability problem for ATEL with epistemic operators for common and distributed knowledge is investigated and settled at ExpTime-complete. Thus ATEL has the same complexity as ATL.

1 Introduction

Alternating-time Temporal Logic (ATL) introduced by Alur, Henzinger, and Kupferman [1] has become a successful tool for formal system specification and verification of distributed systems and game-like multi-agent systems. ATL provides a *path quantifier* $\langle\langle \rangle\rangle$ that is parametrized by a coalition A of agents and *temporal operators* $\bigcirc$ (‘next’), $\Box$ (‘always’), and $\mathcal{U}$ (‘until’) where, e.g., $\langle\langle A \rangle\rangle \Box \psi$ states that the agents in coalition A can cooperate to ensure that always ψ . ATL generalizes the Computational Tree Logic (CTL) since the path quantifiers A (“on all paths”) and E (“on some path”) of CTL can be expressed by the ATL path quantifiers $\langle\langle \emptyset \rangle\rangle$ (“the empty coalition can cooperate”) and $\langle\langle \Sigma \rangle\rangle$ (“the grand coalition of all agents can cooperate”).

ATL satisfiability was investigated in [11, 13]: [11] proves an EXPTIME upper bound for ATL satisfiability under the assumption that the set of available agents is fixed in advance; [13] shows containment in EXPTIME even when the set of agents occurring in formulas is drawn from an arbitrary (possibly infinite) set. The EXPTIME lower bound easily follows from CTL.

Soon after its introduction, ATL was extended with knowledge modalities to Alternating-time Temporal Epistemic Logic (ATEL) by van der Hoek and Wooldridge [14]. For an illustration of ATEL’s expressiveness, consider the following formula that states: if agent a knows a secret, then she can keep it to herself, i.e., a can make sure that her secret will never be known by another

agent b :

$$K_a secret \wedge \neg K_b secret \rightarrow \langle\langle a \rangle\rangle \Box \neg K_b secret.$$

The expressivity of ATEL depends on the *epistemic operators* available. In this paper, we focus on the epistemic operators of the very expressive language $S5^{C,D}$ from [8] which contains epistemic operators K_a (“agent a knows”), E_A (“every agent in A knows”), C_A (“the agents in A have *common knowledge*”), and D_A (“the agents in A have *distributed knowledge*”) for each agent a and coalition A of agents. Intuitively, a coalition A of agents has common knowledge of a fact φ if every agent in A knows that every agent in A knows . . . that φ is true; A has distributed knowledge of φ if the combined individual knowledge of all agents in A implies φ . In what follows, we denote with ATEL the logic ATL extended with epistemic operators K_a , E_A , C_A , and D_A for each agent a and coalition A of agents.

In [6], Goranko, Jamroga, and van Drimmelen axiomatized ATEL without operators for distributed knowledge and conjectured an axiomatization of ATEL with distributed knowledge operators by extending the axiomatic system for ATL [7]. From this axiomatization, it becomes clear that there is no interaction between the temporal and epistemic modalities in ATEL, i.e., ATEL is an independent fusion in the sense of [5] of the logics ATL and $S5^{C,D}$. Although both logics, ATL [11, 13] and $S5^{C,D}$ [8], are EXPTIME-complete, it does not follow from general results on fusions of modal logics that ATEL is EXPTIME-complete as well. For instance, an increase in computational complexity when taking fusions can be observed with the fusion of S5 with itself: while S5 is in NP [8], the fusion is PSpace-hard [5]. From general results on complexity transfer for fusions [3, 2], it only follows that ATEL is in 2-EXPTIME. So far, it was an open problem whether satisfiability for ATEL is also in EXPTIME. In fact, because of the different character of ATL and $S5^{C,D}$, it seemed that their fusion is not in EXPTIME. In this paper, however, we show that satisfiability of ATEL (with common and distributed knowledge) is EXPTIME-complete using a type elimination construction along the lines of [13] combined with a technique of [8]. This also shows that adding epistemic operators for common knowledge and distributed knowledge to ATL does not yield an increase in computational complexity. The result is presented in a compact way due to space limitations; more details and full proofs can be found in the accompanying technical report [12].

When formulating satisfiability problems for ATEL some care is needed as it was analyzed in [13]. 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 [10, 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 [13] 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:

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

This paper is organized as follows. Section 2 introduces ATEL by giving its syntax, defining alternating epistemic transition systems as semantic structures, and, by giving the semantics, stating how formulas are interpreted over these transition systems. In Section 3, we show that ATEL is EXPTIME-complete. This is done by defining a decision procedure for ATEL that runs in exponential time using a combination of a type elimination construction introduced in [13] with the technique of [8]. The final section states some concluding remarks and gives ideas for further research.

2 ATEL

In this section, Alternating-time Temporal Epistemic Logic [14] with epistemic operators for common and distributed knowledge is defined. First consider the syntax.

Definition 2.1 [ATEL Syntax] Let Π be a countable infinite set of *atomic propositions* and Σ a countable infinite set of *agents*. A *coalition* is a finite set $A \subseteq \Sigma$ of agents. The set of ATEL-formulas is the smallest set containing the following:

- p for $p \in \Pi$;
- $\neg\varphi$, $\varphi \vee \psi$ for ATEL-formulas φ and ψ ;
- $\langle\langle A \rangle\rangle\bigcirc\varphi$, $\langle\langle A \rangle\rangle\Box\varphi$, $\langle\langle A \rangle\rangle\varphi\mathcal{U}\psi$ for a coalition A and ATEL-formulas φ and ψ ;
- $K_a\varphi$ for $a \in \Sigma$ and ATEL-formula φ ;
- $E_A\varphi$, $C_A\varphi$, $D_A\varphi$ for a coalition A and ATEL-formula φ .

◁

Logical truth ($\top$) and the Boolean connectives ($\wedge$, $\rightarrow$, and $\leftrightarrow$) are defined as usual. The operator $\langle\langle A \rangle\rangle \Diamond \varphi$ is defined as $\langle\langle A \rangle\rangle \top \mathcal{U} \varphi$. Observe that, in ATEL, the operator $\langle\langle A \rangle\rangle \Box$ cannot be expressed in terms of $\langle\langle A \rangle\rangle$ and $\mathcal{U}$ since formulas of the form $\langle\langle A \rangle\rangle \neg(\top \mathcal{U} \neg \varphi)$, which are equivalent to $\langle\langle A \rangle\rangle \Box \varphi$, are not according to the syntax. ATL is the fragment of ATEL without epistemic operators.

Now consider the semantic structures that are epistemic extensions of alternating transition systems from [1].

Definition 2.2 [Alternating Epistemic Transition System] Given a finite set $\Sigma = \{1, \dots, n\} \subset \mathbf{\Sigma}$ of agents with $n \geq 1$, an *alternating epistemic transition system (AETS)* for Σ is a tuple $\mathcal{S} = \langle \Pi, \Sigma, Q, \sim_1, \dots, \sim_n, \pi, \delta \rangle$ where

- $\Pi \subseteq \mathbf{\Pi}$ is a finite, non-empty set of *atomic propositions*,
- Q is a finite, non-empty set of *states*,
- $\sim_a \subseteq Q \times Q$ is an *epistemic accessibility relation*, one for each agent $a \in \Sigma$, which is required to be an equivalence relation,
- $\pi : Q \rightarrow 2^\Pi$ is a *valuation function* which assigns to every state a set of atomic propositions that are true there, and
- $\delta : Q \times \Sigma \rightarrow 2^{2^Q}$ 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 $Q_1, \dots, Q_n$ of choices $Q_i \in \delta(q, i)$, $1 \leq i \leq n$, the intersection $Q_1 \cap \dots \cap Q_n$ is a singleton set.

◁

For each coalition A , the following abbreviations for epistemic relations corresponding to epistemic operators E_A , C_A , and D_A are used:

$$\begin{aligned} \sim_A^E &= \bigcup_{a \in A} \sim_a \text{ .. the union of the accessibility relations of the agents in } A; \\ \sim_A^C &= (\sim_A^E)^+ \text{ .. the transitive closure of the relation } \sim_A^E; \\ \sim_A^D &= \bigcap_{a \in A} \sim_a \text{ .. the intersection of the accessibility relations of the agents in } A. \end{aligned}$$

Below the relations $\sim_A^E$, $\sim_A^C$, and $\sim_A^D$ will be used to give a Kripke-style semantics to the notions “every agent in A knows”, “the agents in A have common knowledge”, and “the agents in A have distributed knowledge”.

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 AETS $\mathcal{S} = \langle \Pi, \Sigma, Q, \sim_1, \dots, \sim_n, \pi, \delta \rangle$, for each state $q \in Q$ and each coalition $A \subseteq \Sigma$, set

$$\delta(q, A) := \begin{cases} \{Q_A \subseteq Q \mid Q_A = \bigcap_{a \in A} Q_a \text{ where } Q_a \in \delta(q, a)\} & \text{if } A \neq \emptyset \\ \{\bigcup \delta(q, \Sigma)\} & \text{if } A = \emptyset \end{cases}.$$

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 behaviour of the system, $\delta(q, \emptyset)$ is set to $\bigcup \delta(q, \Sigma)$, the set of all possible successors of q .

An infinite sequence $\lambda = q_0 q_1 q_2 \dots \in Q^\omega$ 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 $q_0 \dots q_i$ of λ .

A *strategy* for an agent $a \in \Sigma$ is a mapping $f_a : Q^+ \rightarrow 2^Q$ such that for all $\lambda \in Q^*$ and all $q \in Q$, $f_a(\lambda \cdot q) \in \delta(q, a)$ where $\lambda \cdot q$ denotes the concatenation of the finite sequence λ with the state q . Note that a strategy f_a maps each finite sequence $\lambda \cdot q$ of states to an choice in $\delta(q, a)$ available to agent a at the state q . A *strategy* for a coalition A is a set of strategies $F_A = \{f_a \mid a \in A\}$, one for each agent in A .

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

Formulas of ATEL are interpreted over alternating epistemic transition systems, cf. [14].

Definition 2.3 [ATEL Semantics] Given an AETS $\mathcal{S} = \langle \Pi, \Sigma, Q, \sim_1, \dots, \sim_n, \pi, \delta \rangle$, the satisfaction relation $\models$ is inductively defined as follows: For all states q of $\mathcal{S}$, coalitions of agents $A \subseteq \Sigma$, agents $a \in \Sigma$, and ATEL-formulas φ , φ_1 , and φ_2 , it holds that

- $\mathcal{S}, q \models p$ iff $p \in \pi(q)$, for all atomic propositions $p \in \Pi$;
- $\mathcal{S}, q \models \neg \varphi$ iff $\mathcal{S}, q \not\models \varphi$;
- $\mathcal{S}, q \models \varphi_1 \vee \varphi_2$ iff $\mathcal{S}, q \models \varphi_1$ or $\mathcal{S}, q \models \varphi_2$;
- $\mathcal{S}, q \models \langle \langle A \rangle \rangle \bigcirc \varphi$ iff there is a strategy F_A such that for all computations $\lambda \in \text{out}(q, F_A)$, it holds that $\mathcal{S}, \lambda[1] \models \varphi$;
- $\mathcal{S}, q \models \langle \langle A \rangle \rangle \Box \varphi$ iff there is a strategy F_A such that for all computations $\lambda \in \text{out}(q, F_A)$, it holds that $\mathcal{S}, \lambda[i] \models \varphi$ for all positions $i \geq 0$;
- $\mathcal{S}, q \models \langle \langle A \rangle \rangle \varphi_1 \mathcal{U} \varphi_2$ iff there is an strategy F_A such that for all computations $\lambda \in \text{out}(q, F_A)$, there is a position $i \geq 0$ such that $\mathcal{S}, \lambda[i] \models \varphi_2$ and $\mathcal{S}, \lambda[j] \models \varphi_1$ for all positions j with $0 \leq j < i$;

- $\mathcal{S}, q \models K_a \varphi$ iff $\mathcal{S}, q' \models \varphi$ for all q' with $q \sim_a q'$;
- $\mathcal{S}, q \models E_A \varphi$ iff $\mathcal{S}, q' \models \varphi$ for all q' with $q \sim_A^E q'$;
- $\mathcal{S}, q \models C_A \varphi$ iff $\mathcal{S}, q' \models \varphi$ for all q' with $q \sim_A^C q'$;
- $\mathcal{S}, q \models D_A \varphi$ iff $\mathcal{S}, q' \models \varphi$ for all q' with $q \sim_A^D q'$.

If for some state q of some AETS $\mathcal{S}$ it holds that $\mathcal{S}, q \models \varphi$, then the ATEL-formula φ is *true* at q , and $\mathcal{S}$ is called a *model* of φ . An ATEL-formula is *satisfiable* if it has a model. $\triangleleft$

3 The Main Result

In this section, ATEL satisfiability is shown to be EXPTIME-complete. This is done by extending the EXPTIME-completeness result of ATL satisfiability [13] to ATEL which shows that adding epistemic operators for common knowledge and distributed knowledge to ATL does not yield an increase in computational complexity.

The following theorem states the main result of this paper. Note that we consider the satisfiability problem for ATEL as it was presented in the introduction above.

Theorem 3.1 *The satisfiability problem for ATEL is EXPTIME-complete.*

The lower complexity bound carries over from the EXPTIME-hard fragment ATL; see [11, 13]. For the upper bound, it is to show that ATEL satisfiability is in EXPTIME. The rest of the paper is devoted to presenting a decision procedure for ATEL that runs in exponential time and thus shows containment of ATEL satisfiability in EXPTIME. Correctness of the decision procedure follows from Lemma 3.8 while Lemma 3.9 proves its exponential running time. Due to space limitations, the presentation is rather compact and the proofs are sketched only. More details and full proofs, however, can be found in the accompanying technical report [12] and the paper [13] on which this work is based on.

The decision procedure implements an extended version of the type elimination construction for ATL from [13] that additionally accounts for ATEL's epistemic operators K_a , E_A , C_A , and D_A for each agent a and coalition A of agents where the epistemic operators are dealt with similarly to [8]. Start with the definition of an extended closure that contains all formulas that are relevant for deciding the input formula. This and the following notion of types as specific subsets of the extended closure are adapted from [13] to additionally account for epistemic operators.

Definition 3.2 [Extended Closure] Let φ be an ATEL-formula and n the number of agents occurring in φ . The *extended closure* $\text{ecl}(\varphi)$ of φ is the smallest set which is closed under the following conditions:

- $\varphi \in \text{ecl}(\varphi)$;
- $\text{ecl}(\varphi)$ is closed under subformulas;
- $\text{ecl}(\varphi)$ is closed under single negation;
- if $\langle\langle A \rangle\rangle \Box \psi \in \text{ecl}(\varphi)$, then $\langle\langle A \rangle\rangle \bigcirc \langle\langle A \rangle\rangle \Box \psi \in \text{ecl}(\varphi)$;
- if $\langle\langle A \rangle\rangle \psi \mathcal{U} \vartheta \in \text{ecl}(\varphi)$, then $\langle\langle A \rangle\rangle \bigcirc \langle\langle A \rangle\rangle \psi \mathcal{U} \vartheta \in \text{ecl}(\varphi)$;
- if $E_A \psi \in \text{ecl}(\varphi)$ with $A = \{a_1, \dots, a_\ell\}$, then $\neg(\neg K_{a_1} \psi \vee \dots \vee \neg K_{a_\ell} \psi) \in \text{ecl}(\varphi)$;
- if $C_A \psi \in \text{ecl}(\varphi)$, then $E_A C_A \psi \in \text{ecl}(\varphi)$;
- if $K_a \psi \in \text{ecl}(\varphi)$ and $D_A \vartheta \in \text{ecl}(\varphi)$ with $a \in A$, then $D_A \psi \in \text{ecl}(\varphi)$;
- if $D_A \psi \in \text{ecl}(\varphi)$ and $D_B \vartheta \in \text{ecl}(\varphi)$ with $A \subseteq B$, then $D_B \psi \in \text{ecl}(\varphi)$.

◁

Note that the cardinality of $\text{ecl}(\varphi)$ is linear in the length of φ .

Definition 3.3 [Type] Let φ be an ATEL-formula. The set $\Psi \subseteq \text{ecl}(\varphi)$ is a *type for φ* if the following conditions are satisfied:

- (T1) $\psi_1 \vee \psi_2 \in \Psi$ iff $\psi_1 \in \Psi$ or $\psi_2 \in \Psi$, for all $\psi_1 \vee \psi_2 \in \text{ecl}(\varphi)$;
- (T2) $\psi \in \Psi$ iff $\neg \psi \notin \Psi$, for all $\neg \psi \in \text{ecl}(\varphi)$;
- (T3) $\langle\langle A \rangle\rangle \Box \psi \in \Psi$ iff $\{\psi, \langle\langle A \rangle\rangle \bigcirc \langle\langle A \rangle\rangle \Box \psi\} \subseteq \Psi$, for all $\langle\langle A \rangle\rangle \Box \psi \in \text{ecl}(\varphi)$;
- (T4) $\langle\langle A \rangle\rangle \psi \mathcal{U} \vartheta \in \Psi$ iff $\vartheta \in \Psi$ or $\{\psi, \langle\langle A \rangle\rangle \bigcirc \langle\langle A \rangle\rangle \psi \mathcal{U} \vartheta\} \subseteq \Psi$, for all $\langle\langle A \rangle\rangle \psi \mathcal{U} \vartheta \in \text{ecl}(\varphi)$;
- (T5) if $K_a \psi \in \Psi$, then $\psi \in \Psi$;
- (T6) $E_A \psi \in \Psi$ iff $\neg(\neg K_{a_1} \psi \vee \dots \vee \neg K_{a_\ell} \psi) \in \Psi$, for all $E_A \psi \in \text{ecl}(\varphi)$ with $A = \{a_1, \dots, a_\ell\}$;
- (T7) $C_A \psi \in \Psi$ iff $\{\psi, E_A C_A \psi\} \subseteq \Psi$, for all $C_A \psi \in \text{ecl}(\varphi)$;
- (T8) if $D_A \psi \in \Psi$, then $\psi \in \Psi$;
- (T9) if $K_a \psi \in \Psi$, then $D_A \psi \in \Psi$, for all $D_A \psi \in \text{ecl}(\varphi)$ with $a \in A$;
- (T10) if $D_A \psi \in \Psi$, then $D_B \psi \in \Psi$, for all $D_B \psi \in \text{ecl}(\varphi)$ with $A \subseteq B$.

The set of all types for φ is designated by Γ_φ .

◁

Note that the cardinality of Γ_φ is exponential in the length of φ .

Intuitively, a type describes a state q of a transition system by means of formulas from $\text{ecl}(\varphi)$ which are true at q . The decision procedure will use all types for an input formula to determine its decidability. The conditions (T1) to (T10) state restrictions on the formulas in a type: (T1) to (T4) deal with Boolean and temporal formulas; the conditions (T5) to (T10) were added to handle epistemic formulas. In particular, the relation between formulas with epistemic operators K and D are taken care of by (T9) and (T10).

Before proceeding with the construction, introduce some auxiliary notions. Denote with Σ_φ the set of all agents that occur in the formula φ . A *next formula* is a formula of the form $\langle\langle A \rangle\rangle\bigcirc\psi$ or $\neg\langle\langle A \rangle\rangle\bigcirc\psi$. Assume that all next formulas in the extended closure $\text{ecl}(\varphi)$ are linearly ordered such that no negative next formula occurs before a positive one. Denote with $\#_\psi$ the number of the next formula $\psi \in \text{ecl}(\varphi)$ in this ordering; the numbering starts with 0. Since there are as many positive next formulas in $\text{ecl}(\varphi)$ as negative ones, obtain the following enumeration $\psi_1, \dots, \psi_k$ with $\psi_1, \dots, \psi_{k/2-1}$ positive next formulas and $\psi_{k/2}, \dots, \psi_k$ negative next formulas.

Intuitively, the presented type elimination algorithm eliminates those types that are not relevant for building an AETS $\mathcal{S}$ that is a model for an input formula φ . The states of such an $\mathcal{S}$ are thought of consisting of sequences of *n-tuples* where $n = |\Sigma_\varphi|$. Each component of such an *n-tuple* corresponds to an agent and takes values from $\{0, \dots, k-1\}$ which in turn refer to one of the k next formulas in $\text{ecl}(\varphi)$. The set of all such *n-tuples* is denoted with $[k/n]$, and the states of $\mathcal{S}$ will be from $[k/n]^*$. Supposing $q \in [k/n]^*$ is a state of $\mathcal{S}$, then the set $\{q \cdot \vec{t} \mid \vec{t} \in [k/n]\}$ contains its potential successor states, thus the choices in $\delta(q, a)$ are subsets of this set. Every state q of $\mathcal{S}$ corresponds to some type that was not eliminated and the construction of $\mathcal{S}$ is aiming at satisfying every formula of this type at q . In particular, q will have to satisfy a number of next formulas, i.e., the definition of δ that specifies the choices available at q is crucial. The transition function δ is defined as follows: For each agent a (ranging from 0 to $n-1$), set

$$\delta(q, a) := \{\{q \cdot \vec{t} \in [k/n]^* \mid \vec{t} = (t_0, \dots, t_{n-1}) \text{ and } t_a = p\} \mid p < k/2\}.$$

The idea here is that agent a makes an a -choice by “voting” for a positive next formula that he wants to be satisfied. Such votes are done via a ’s position in tuples $\vec{t} = (t_0, \dots, t_{n-1})$. An A -choice is made by a collective vote of the entire coalition A . This is characterized as follows: a set $S \subseteq [k/n]$ is called an *A-voting set* if

$$S = \{\vec{t} = (t_0, \dots, t_{n-1}) \mid t_a = \tau(a) \text{ for all } a \in A\}$$

for some mapping $\tau : A \rightarrow \{0, \dots, k/2-1\}$. In order for q to satisfy a positive next formula $\langle\langle A \rangle\rangle\bigcirc\psi$, the construction of $\mathcal{S}$ ensures that ψ is satisfied at all states of the A -choice that corresponds to the voting set where coalition A votes solely for this formula.

Satisfying negative next formulas $\neg\langle\langle\Sigma_\varphi\rangle\rangle\bigcirc\psi$ is simple: they merely state that *all* successors of q satisfy $\neg\psi$. For satisfying $\neg\langle\langle A\rangle\rangle\bigcirc\psi$ with $A \subset \Sigma_\varphi$ at q , ψ needs to be refuted at a state in every A -choice. Note that, by picking a state from every A -choice, a state from every A' -choice for all $A' \subseteq A$ is picked as well. But this is fine as $\neg\langle\langle A\rangle\rangle\bigcirc\psi$ implies $\neg\langle\langle A'\rangle\rangle\bigcirc\psi$. However, for any $B \not\subseteq A$, $\neg\langle\langle A\rangle\rangle\bigcirc\psi$ does not imply $\neg\langle\langle B\rangle\rangle\bigcirc\psi$. Thus, in order not to automatically satisfy $\neg\langle\langle B\rangle\rangle\bigcirc\psi$ when picking elements in A -choices, some care is needed such that the picked elements from A -choices are not also elements of B -choices for any such B . Picking the right states that refute ψ is implemented by a “refutation function”; see [13] for details.

The construction of a model is more involved due to the presents of negated box formulas, until formulas and formulas of the form $\neg C_A\psi$. This is because the satisfaction of such formulas needs to be witnessed by some state in a model and such witnesses can be “far away”. Negated box formulas and until formulas can be dealt with analogously to [4] for CTL with the help of so-called witness trees. Since, in ATEL, epistemic and temporal operators do not interact, formulas of the form $\neg C_A\psi$ can be treated separately with a notion of witness paths. For defining witness trees, the following auxiliary notions from [13] are needed.

Definition 3.4 [ϑ -vector, φ -tree, $\bigcirc$ -matching] Let φ be an ATEL-formula, $n = |\Sigma_\varphi|$, and k the number of next-formulas in $\text{ecl}(\varphi)$. For each next-formula $\vartheta \in \text{ecl}(\varphi)$ and vector $\vec{t} = (t_0, \dots, t_{n-1}) \in [k/n]$,

- if $\vartheta = \langle\langle A\rangle\rangle\bigcirc\psi$ and $t_a = \sharp_\vartheta$ for each $a \in A$, then $\vec{t}$ is called a ϑ -vector;
- if $\vartheta = \neg\langle\langle A\rangle\rangle\bigcirc\psi$ with $A \subset \Sigma_\varphi$, then $\vec{t}$ is called a ϑ -vector if the refutation function says so; see [13] for details;
- if $\vartheta = \neg\langle\langle\Sigma_\varphi\rangle\rangle\bigcirc\psi \in \text{ecl}(\varphi)$, then $\vec{t}$ is called a ϑ -vector.

For each type $\Psi \in \Gamma_\varphi$ and each $\vec{t} = (t_0, \dots, t_{n-1}) \in [k/n]$, let $S_\Psi(\vec{t}) \subseteq \text{ecl}(\varphi)$ be the smallest set such that

- (M1) if $\langle\langle A\rangle\rangle\bigcirc\psi \in \Psi$ and $\vec{t}$ is a $\langle\langle A\rangle\rangle\bigcirc\psi$ -vector, then $\psi \in S_\Psi(\vec{t})$, and
- (M2) if $\neg\langle\langle A\rangle\rangle\bigcirc\psi \in \Psi$ and $\vec{t}$ is a $\neg\langle\langle A\rangle\rangle\bigcirc\psi$ -vector, then $\neg\psi \in S_\Psi(\vec{t})$.

Given a set M of labels, a $\langle M, k, n \rangle$ -tree T is a mapping T from a finite prefix-closed subset of $[k/n]^*$ to M . A $\langle \Gamma_\varphi, k, n \rangle$ -tree is called a φ -tree. A φ -tree T is called $\bigcirc$ -matching if, for all $\alpha \in \text{dom}(T)$ and all $\vec{t} \in [k/n]$, $\alpha \cdot \vec{t} \in \text{dom}(T)$ implies $S_{T(\alpha)}(\vec{t}) \subseteq T(\alpha \cdot \vec{t})$. $\triangleleft$

Intuitively, a vector $\vec{t}$ is a $\langle\langle A\rangle\rangle\bigcirc\psi$ -vector if, for all states q satisfying $\langle\langle A\rangle\rangle\bigcirc\psi$, the successor $q \cdot \vec{t}$ has to satisfy ψ . The $\neg\langle\langle A\rangle\rangle\bigcirc\psi$ -vectors can be understood in an analogous way. This intuition is reflected in (M1) and (M2) and in the notion of $\bigcirc$ -matching.

Now, equipped with the notions ϑ -vector, φ -tree, and $\bigcirc$ -matching, define witness trees and witness paths. While witness paths testify satisfaction of formulas of the form $\neg C_A \psi$, witness trees are special φ -trees that affirm satisfaction of negated box formulas and until formulas; cf. [13] for witness trees.

Definition 3.5 [Witness Tree, Witness Path] Let φ be an ATEL-formula, Γ a set of types for φ , and $\Psi \in \Gamma$. A φ -tree T is called a *witness-tree rooted at Ψ in Γ for a formula $\langle\langle A \rangle\rangle\psi\mathcal{U}\vartheta$* if it satisfies the following properties:

1. for all $\alpha \in \text{dom}(T)$, $T(\alpha) \in \Gamma$;
2. T is $\bigcirc$ -matching;
3. $T(\varepsilon) = \Psi$;
4. for all $\alpha \in \text{dom}(T)$, $\langle\langle A \rangle\rangle\psi\mathcal{U}\vartheta \in T(\alpha)$;
5. for all non-leaf nodes α , $\psi \in T(\alpha)$;
6. for all leaf nodes α , $\vartheta \in T(\alpha)$;
7. if $\alpha \in \text{dom}(T)$, $\langle\langle A \rangle\rangle\bigcirc\langle\langle A \rangle\rangle\psi\mathcal{U}\vartheta \in T(\alpha)$, $\vartheta \notin T(\alpha)$, and $\vec{t}$ is a $\langle\langle A \rangle\rangle\bigcirc\langle\langle A \rangle\rangle\psi\mathcal{U}\vartheta$ -vector, then $\alpha \cdot \vec{t} \in \text{dom}(T)$.

T is called a *witness-tree rooted at Ψ in Γ for a formula $\neg\langle\langle A \rangle\rangle\Box\psi$* if it satisfies the following properties:

1. for all $\alpha \in \text{dom}(T)$, $T(\alpha) \in \Gamma$;
2. T is $\bigcirc$ -matching;
3. $T(\varepsilon) = \Psi$;
4. for all $\alpha \in \text{dom}(T)$, $\neg\langle\langle A \rangle\rangle\Box\psi \in T(\alpha)$;
5. for all leaf nodes α , $\neg\psi \in T(\alpha)$;
6. if $\alpha \in \text{dom}(T)$, $\neg\langle\langle A \rangle\rangle\bigcirc\langle\langle A \rangle\rangle\Box\psi \in T(\alpha)$, $\neg\psi \notin T(\alpha)$, and $\vec{t}$ is a $\neg\langle\langle A \rangle\rangle\bigcirc\langle\langle A \rangle\rangle\Box\psi$ -vector, then $\alpha \cdot \vec{t} \in \text{dom}(T)$.

A sequence $\Psi_0 \cdots \Psi_m \in \Gamma^*$, $m \geq 1$, is a *witness path rooted at Ψ in Γ for a formula $\neg C_A \psi$* if the following conditions are satisfied:

1. $\Psi_0 = \Psi$;
2. for all $i < m$, there is an agent $a \in A$ such that for all $K_a \psi' \in \text{ecl}(\varphi)$, $K_a \psi' \in \Psi_i$ iff $K_a \psi' \in \Psi_{i+1}$;
3. $\neg\psi \in \Psi_m$.

$\triangleleft$

The presented decision procedure employs a type elimination algorithm which in turn relies essentially on the notion of *realizability*. This notion from [13] was extended to additionally account for epistemic operators. Intuitively, a type Ψ is realizable in a set of types Γ if it is possible to (a) satisfy all next-formulas in Ψ and all formulas of the form $\neg K_a\psi$ or $\neg D_A\psi$ in Ψ using only types from Γ , (b) construct witness trees for all negated box formulas and until formulas in Ψ , and (c) construct witness paths for all formulas of the form $\neg C_A\psi$ in Ψ using only types from Γ .

Definition 3.6 [Realizability] Let φ be an ATEL-formula and Γ a set of types for φ . A type $\Psi \in \Gamma$ is *realizable in* Γ if the following conditions are satisfied:

1. for all $\vec{t} \in [k/n]$, there is a $\Psi' \in \Gamma$ such that $S_\Psi(\vec{t}) \subseteq \Psi'$;
2. for all $\langle\langle A \rangle\rangle\psi\mathcal{U}\vartheta \in \Psi$, there is a $\langle\langle A \rangle\rangle\psi\mathcal{U}\vartheta$ -witness tree rooted at Ψ in Γ ;
3. for all $\neg\langle\langle A \rangle\rangle\Box\psi \in \Psi$, there is a $\neg\langle\langle A \rangle\rangle\Box\psi$ -witness tree rooted at Ψ in Γ ;
4. for all $\neg K_a\psi \in \Psi$, there is a $\Psi' \in \Gamma$ such that
 - (a) $\neg\psi \in \Psi'$, and
 - (b) $K_a\psi' \in \Psi$ iff $K_a\psi' \in \Psi'$, for all $K_a\psi' \in \text{ecl}(\varphi)$;
5. for all $\neg C_A\psi \in \Psi$, there is $\neg C_A\psi$ -witness path rooted at Ψ in Γ ;
6. for all $\neg D_A\psi \in \Psi$, there is a $\Psi' \in \Gamma$ such that
 - (a) $\neg\psi \in \Psi'$, and
 - (b) $D_A\psi' \in \Psi$ iff $D_A\psi' \in \Psi'$, for all $D_A\psi' \in \text{ecl}(\varphi)$.

◁

Finally the decision procedure for ATEL using type elimination is described. The algorithm starts with the set of all types for an input formula and then repeatedly eliminates types that are not realizable. Since there are only finitely many types to start with, the algorithm terminates with a set of realizable types. The input formula is satisfiable if it is contained in some of the remaining types. Formally, this works as follows: Let φ be an ATEL-formula for which satisfiability is to be decided. Inductively compute a sequence $\Delta_0, \Delta_1, \dots$ of sets of types for φ as follows:

$$\begin{aligned}\Delta_0 &:= \Gamma_\varphi, \\ \Delta_{i+1} &:= \{\Psi \in \Delta_i \mid \Psi \text{ is realizable in } \Delta_i\}.\end{aligned}$$

Stop when $\Delta_{m+1} = \Delta_m$ for some $m \geq 0$. Note that Δ_m is a set of types for φ that are all realizable in Δ_m . The algorithm returns “Yes, φ is satisfiable in an AETS for Σ_φ ” if φ is contained in some type in Δ_m ; otherwise it returns “No, φ is not satisfiable.”

The following lemmas state that the described decision procedure is correct and can be computed within exponential time in the length of the input formula; full proofs can be found in [12].

Lemma 3.7 *Let Γ be a set of types for an ATEL-formula φ . Then the existence of witness trees and paths in Γ can be decided in time exponential in the length of φ .*

Proof (sketch). Suppose it is to check for the existence of a witness tree for $\langle\langle A \rangle\rangle\psi\mathcal{U}\vartheta \in \Psi_0$ rooted at Ψ_0 in Γ ; the cases for $\neg\langle\langle A \rangle\rangle\Box\psi$ and $\neg C_A\psi$ are similar. Start with identifying leaf nodes of possible witness trees by marking all types in Γ which contain $\langle\langle A \rangle\rangle\psi\mathcal{U}\vartheta$ and ϑ . Then identify inner nodes of possible witness trees as follows: For all unmarked types $\Psi \in \Gamma$ such that $\langle\langle A \rangle\rangle\psi\mathcal{U}\vartheta \in \Psi$, mark Ψ if $\psi \in \Psi$ and for every $\langle\langle A \rangle\rangle\Box\langle\langle A \rangle\rangle\psi\mathcal{U}\vartheta$ -vector $\vec{t} \in [k/n]$, there is a type $\Psi' \in \Gamma$ such that:

- (i) $S_\Psi(\vec{t}) \subseteq \Psi'$, and
- (ii) Ψ' is marked.

Repeatedly apply this procedure until no more types in Γ get marked.

It is not hard to see that a witness tree for $\langle\langle A \rangle\rangle\psi\mathcal{U}\vartheta$ exists iff Ψ_0 was marked. Verify that this marking procedure and thus checking for existence of witness trees runs in exponential time. QED

Lemma 3.8 *Let φ be an ATEL-formula. Then the procedure returns “Yes, the input formula φ is satisfiable in an AETS for Σ_φ ” iff it is indeed the case.*

Proof (sketch). Soundness: Assume that the elimination procedure was started on input φ and returns “Yes, the input formula φ is satisfiable”. Let Γ be the computed set of types. Then all types of Γ are realizable in Γ and there is a type $\Psi \in \Gamma$ with $\varphi \in \Psi$. Construct a model $\mathcal{S}$ of φ where states correspond to types in Γ . This is done by constructing witness trees for all until formulas and negated box formulas in $\text{ecl}(\varphi)$ and for each type in Γ , arranging them in a matrix form such that the rows range over the eventualities and the columns over the types. Then we replace all leaf nodes by an edge from the leaf node’s predecessor to the root of some other witness tree in the next row such that this root is labeled by the same type as the replaced leaf. This construction is similar to the one for CTL; cf. [4]. Initially, the epistemic operator D_A is treated as an additional modality and the corresponding epistemic accessibility relation $\sim_A^D$ in $\mathcal{S}$ is explicitly defined, i.e., $\sim_A^D = \bigcap_{a \in A} \sim_a$ for all coalitions A does not necessarily hold. However, a proper model $\mathcal{S}'$ can be constructed with the help of $\mathcal{S}$ by introducing epistemic witness trees on each state in $\mathcal{S}$ and rearranging, for each agent a , its epistemic accessibility relation $\sim_a$ such that temporal and epistemic transitions are kept separate.

Completeness: Suppose φ is satisfiable in a model $\mathcal{S}$. Let Q be the set of states in $\mathcal{S}$ and denote with $\text{types}(Q)$ the set of all types associated with

some state in Q . It can be shown that all types in $\text{types}(Q)$ are realizable in $\text{types}(Q)$. QED

Lemma 3.9 *The described type elimination procedure runs in exponential time.*

Proof (sketch). Suppose φ is given as input and let $n = |\varphi|$. Recall that the size of the extended closure $\text{ecl}(\varphi)$ is linear in the length of φ , i.e., $|\text{ecl}(\varphi)| = c \cdot n$ for some constant $c \geq 1$. The algorithm computes a sequence $\Delta_0, \dots, \Delta_m$ of sets of types such that $\Delta_0 \supsetneq \Delta_1 \supsetneq \dots \supsetneq \Delta_m$. Since $\Delta_0 = \Gamma_\varphi \subseteq 2^{\text{ecl}(\varphi)}$, this sequence is finite with $m \leq 2^{c \cdot n}$. For each i with $0 < i \leq m$, it holds that $|\Delta_i| < |\Delta_0| \leq 2^{c \cdot n}$. Thus, to compute the set Δ_{i+1} at most $2^{c \cdot n}$ types in Δ_i need to be checked whether they are realizable in Δ_i . Using Lemma 3.7, it can be shown that for a type $\Psi \in \Delta_i$ at most $2^{\mathcal{O}(n^2)}$ steps are needed to check for Ψ 's realizability in Δ_i . Note that checking whether there is a type in Δ_m that contains φ takes not more than $2^{\mathcal{O}(n)}$ steps. Hence the decision procedure runs in time exponential in the size of the input. QED

4 Conclusion

We have investigated the complexity of the satisfiability problem for the epistemic extension ATEL of ATL with common and distributed knowledge which up to now was an open problem. ATEL satisfiability was shown to be EXP-TIME-complete by extending the type elimination construction for ATL from [13]. This result shows that adding epistemic operators for common and distributed knowledge to ATL does not yield an increase in computational complexity.

ATEL, as it is defined in this paper, does not allow for any interaction between knowledge and time. For instance, it is possible for an agent a that at two for a epistemically indistinguishable states a makes different choices. More discussions on the interaction of temporal and epistemic notions in ATEL can be found in, e.g., [14, 9]. For further research, it is interesting to investigate the complexity of variants of ATEL that capture various desired and reasonable interactions between knowledge and time.

References

- [1] Alur, R., T. A. Henzinger and O. Kupferman, *Alternating-time temporal logic*, J. ACM **49** (2002), pp. 672–713.
- [2] Baader, F., S. Ghilardi and C. Tinelli, *A new combination procedure for the word problem that generalizes fusion decidability results in modal logics*, in: D. Basin and M. Rusinowitch, editors, *Proceedings of the 2nd International Joint Conference on Automated Reasoning (IJCAR'04)*, Lecture Notes in Artificial Intelligence (2004), to appear.

- [3] Baader, F., C. Lutz, H. Sturm and F. Wolter, *Fusions of description logics and abstract description systems*, Journal of Artificial Intelligence Research (JAIR) **16** (2002), pp. 1–58.
- [4] Emerson, E. A., *Temporal and modal logic*, in: *Handbook of theoretical computer science (vol. B): formal models and semantics* (1990), pp. 995–1072.
- [5] Gabbay, D., A. Kurucz, F. Wolter and M. Zakharyashev, “Many-Dimensional Modal Logics: Theory and Applications,” Studies in Logic **148**, Elsevier, 2003.
- [6] Goranko, V., W. Jamroga and G. van Drimmelen, *Axiomatic systems for alternating-time temporal epistemic logics (extended abstract)*, in: *6th Conference on Logic and the Foundations of Game and Decision Theory (LOFT 2004), July 16 - 18, 2004, Leipzig Graduate School of Management, Germany, Proceedings*.
- [7] Goranko, V. and G. van Drimmelen, *Decidability and complete axiomatization of the alternating-time temporal logic* (2005), theoretical Computer Science, to appear.
- [8] Halpern, J. Y. and Y. Moses, *A guide to completeness and complexity for modal logics of knowledge and belief*, Artificial Intelligence **54** (1992), pp. 319–380.
- [9] Jamroga, W. and W. van der Hoek, *Agents that know how to play*, Fundamenta Informaticae **62**, 2004, pp. 1–35.
- [10] Pauly, M., “Logic for Social Software,” Ph.D. thesis, University of Amsterdam (2001), ILLC Dissertation Series 2001-10.
- [11] van Drimmelen, G., *Satisfiability in alternating-time temporal logic*, in: *18th IEEE Symposium on Logic in Computer Science (LICS 2003), 22-25 June 2003, Ottawa, Canada, Proceedings*.
- [12] Walther, D., *Satisfiability of ATEL with common and distributed knowledge is ExpTime-complete*, Technical report, University of Liverpool (2005), see <http://www.csc.liv.ac.uk/~dirk/>.
- [13] Walther, D., C. Lutz, F. Wolter and M. Wooldridge, *Atl satisfiability is indeed exptime-complete*, submitted to Journal of Logic and Computation (2005).
- [14] Wooldridge, M. and W. van der Hoek, *Time, knowledge, and cooperation: Alternating-time temporal epistemic logic and its applications*, in: *Proceedings of the 5th International Conference on Coordination Models and Languages* (2002), p. 4.