Playing to Learn, or to Keep Secret: Alternating-Time Logic Meets Information Theory

Masoud Tabatabaei
Interdisciplinary Centre for Security and Trust, SnT, University of Luxembourg
masoud.tabatabaei@uni.lu

Wojciech Jamroga
Institute of Computer Science, Polish Academy of Sciences and SnT, University of Luxembourg
wojciech.jamroga@uni.lu

ABSTRACT

Many important properties of multi-agent systems refer to the participants' ability to achieve a given goal, or to prevent the system from an undesirable event. Among intelligent agents, the goals are often of epistemic nature, i.e., concern the ability to obtain knowledge about an important fact φ. Such properties can be e.g. expressed in **ATLK**, that is, alternating-time temporal logic **ATL** extended with epistemic operators. In many realistic scenarios, however, players do not need to fully learn the truth value of φ. They may be almost as well off by gaining *some* knowledge; in other words, by reducing their uncertainty about φ. Similarly, in order to keep φ secret, it is often insufficient that the intruder never fully learns its truth value. Instead, one needs to require that his uncertainty about φ never drops below a reasonable threshold.

With this motivation in mind, we introduce the logic **ATLH**, extending **ATL** with quantitative modalities based on the Hartley measure of uncertainty. The new logic enables to specify agents' abilities w.r.t. the uncertainty of a given player about a given set of statements. It turns out that **ATLH** has the same expressivity and model checking complexity as **ATLK**. However, the new logic is exponentially more succinct than **ATLK**, which is the main technical result of this paper.

KEYWORDS

Multiagent Systems, Knowledge Representation, Uncertainty

ACM Reference Format:
Masoud Tabatabaei and Wojciech Jamroga. 2023. Playing to Learn, or to Keep Secret: Alternating-Time Logic Meets Information Theory. In *Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), London, United Kingdom, May 29 – June 2, 2023*, IFAAMAS, 9 pages.

1 INTRODUCTION

Many important properties of multi-agent systems refer to *strategic abilities* of agents and their groups [3, 17]. They can be formalized in logics of strategic ability, such as alternating-time temporal logic **ATL** [8, 50] or strategy logic **SL** [22, 46]. For example, the **ATL** formula $\langle\langle taxi\rangle\rangle F\mathsf{destination}$, built upon the strategic operator $\langle\langle A\rangle\rangle\varphi$ for "there is a strategy for A such that φ holds" and the temporal modality F ("eventually"), can be used to express that the autonomous cab can deliver the passenger to his/her destination. Similarly, $\langle\langle passg\rangle\rangle(\neg\mathsf{dead})\,\mathcal{U}\,\mathsf{destination}$ says that the passenger has the ability to survive the ride alive. Such statements allow to express important functionality and safety requirements in a simple and intuitive way. Moreover, they provide input to algorithms and tools for verification, that have been in constant development for over 20 years [5, 6, 12, 12, 19–21, 23, 31, 34, 39, 41–44, 47].

Knowledge and information has always been an important aspect of interaction, but it has become even more important with the emergence of Internet and, more recently, social networks. Information is an important resource on which strategies are built, e.g., it is widely acknowledged that executable strategies must comply with so called uniformity constraints [37, 50]. More and more often, information becomes also the *goal* of the interaction. Agents may play to *learn* about a particular subject. People strive to know what the state of the economy is, what is the latest clothing fashion, and whether the coffee machine at their workplace serves good espresso or not. Using strategic-epistemic specifications that involve the knowledge operator K_a, the latter kind of ability can be expressed by $\langle\langle worker\rangle\rangle F(K_{worker}\mathsf{good} \vee K_{worker}\neg\mathsf{good})$. Dually, the user of a social network may want to post a message for their friends only, in which case no outsider should learn the content of the message. This kind of ability can be captured by $\langle\langle user\rangle\rangle G\neg(K_{outsider}\mathsf{post} = \mathsf{m} \vee K_{outsider}\mathsf{post} \neq \mathsf{m})$.

In many cases, however, strategic-epistemic specifications are too coarse. It is great if the worker can obtain full knowledge about the quality of workplace espresso, but being *almost sure* is almost as good. Dually, leaking *some* information about the social network post can be damaging, even if the intruder does not learn its exact content. With this motivation in mind, we propose to extend alternating-time temporal logic with new, information-theoretic modalities $\mathcal{H}$, based on the Hartley measure of uncertainty [29]. We also demonstrate the usefulness of the framework on a real-life voting scenario.

In terms of technical results, we prove that the resulting logic has the same expressive power and model checking complexity as strategic-epistemic specifications; however, it is exponentially more succinct. This is an important result, as it shows that the verification of a given *property* with uncertainty operators can take exponentially less time than when one uses knowledge modalities.

Related work. Strategic-epistemic reasoning has been intensively studied in the early 2000s, especially within the framework of ATEL [2, 32, 54–56] and Dynamic Epistemic Logic [4, 57]. Dynamic epistemic planning [13] is a particularly relevant example. Still, we are not aware of any works combining logical formalizations of strategic reasoning with information-theoretic properties. The paper [36] comes closest, as it discusses the relation between a variant of resource-bounded temporal-epistemic logic and Hartley measure. Moreover, our proposal is directly inspired by information-theoretic notions of security, cf. [40] for an introduction.

Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), A. Ricci, W. Yeoh, N. Agmon, B. An (eds.), May 29 – June 2, 2023, London, United Kingdom. © 2023 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). All rights reserved.

Another strand of related works concerns quantitative specification and verification of MAS due to stochastic interaction [24, 30], graded [9, 26] and fuzzy strategic modalities [11, 14], or probabilistic beliefs about the opponents' response [18]. Those papers considered neither knowledge nor information-theoretic properties, though [26] leaned in that direction by including a count over the accessible imperfect worlds.

Succinctness of logical representations has been studied since early 1970s [51]. In particular, the relative succinctness of branching-time logics was investigated in [1, 45, 59], and [15] studied the succinctness of the strategic logic **ATL*** with past-time operators. The methodology of proving succinctness by means of *formula size games* was proposed in [1], and later generalized in [27]. We adapt the latter approach to obtain our main technical result here.

2 LOGICS OF STRATEGIC ABILITY

We first recapitulate the logical foundations that we chose for our approach.

2.1 Alternating-Time Logic ATL

Alternating-time temporal logic **ATL** [7, 8, 50] generalizes the branching-time temporal logic **CTL** [25] by replacing the path quantifiers E, A with *strategic modalities* $\langle\langle A\rangle\rangle$. Informally, $\langle\langle A\rangle\rangle\gamma$ says that a group of agents A has a collective strategy to enforce temporal property γ. **ATL** formulas can include temporal operators: "X" ("in the next state"), "G" ("always from now on"), "F" ("now or sometime in the future"), and $\mathcal{U}$ (strong "until").

Syntax. Formally, let *Agt* be a finite set of agents, and *Prop* a countable set of atomic propositions. The language of **ATL** is defined as follows:

$$\varphi ::= \mathrm{p} \mid \neg\varphi \mid \varphi \wedge \varphi \mid \langle\langle A\rangle\rangle X\varphi \mid \langle\langle A\rangle\rangle G\varphi \mid \langle\langle A\rangle\rangle \varphi\, \mathcal{U}\, \varphi.$$

where $A \subseteq Agt$ and $\mathrm{p} \in Prop$. Derived boolean connectives and constants ($\vee, \top, \bot$) are defined as usual. "Sometime" is defined as $F\varphi \equiv \top\, \mathcal{U}\, \varphi$.

2.2 Semantics of ATL

Models. The semantics of **ATL** is defined over a variant of synchronous multi-agent transition systems. Let $S = \{Agt, St, Act, d, out\}$ be a concurrent game structure (CGS) such that: $Agt = \{1, ..., k\}$ is a set of agents (or players), St is the set of states of the system, act is a set of actions, $d : Agt \times St \to 2^{Act} \setminus \{\emptyset\}$ shows what actions are available for each player in each state, and $out : St \times Act^1 \times ... \times Act^k \to St$ is the transition function which, given a state and one action from each player in that state, returns the resulting sate. A CGS together with a set of atomic propositions Pv and a valuation function $V : Pv \to 2^{St}$ is called a concurrent game model (CGM). A *pointed CGM* is a pair (M, q_0) consisting of a concurrent game model M and an initial state q_0 in M.

Strategies and their outcomes. Given a CGS, we define the strategies and their outcomes as follows. A *strategy* for $a \in Agt$ is a function $s_a : St \to Act$ such that $s_a(q) \in d(a, q)$.[1] The set of such strategies is sometimes denoted by Σ_a^{Ir}, with the capital "I" referring to perfect **I**nformation, and the lowercase "r" for possibly imperfect **r**ecall. A *collective strategy* for a group of agents $A = \{a_1, \ldots, a_r\}$ is a tuple of individual strategies $s_A = \langle s_{a_1}, \ldots, s_{a_r}\rangle$. The set of such strategies is denoted by Σ_A^{Ir}.

A *path* $\lambda = q_0 q_1 q_2 \ldots$ in a CGS is an infinite sequence of states such that there is a transition between each q_i and q_{i+1}. $\lambda[i]$ denotes the ith position on λ (starting from $i = 0$) and $\lambda[i, \infty]$ the suffix of λ starting with i. The "outcome" function $out(q, s_A)$ returns the set of all paths that can occur when agents A execute strategy s_A from state q onward, defined as follows:

$out(q, s_A) = \{\lambda = q_0, q_1, q_2 \ldots \mid q_0 = q$ and for each $i = 0, 1, \ldots$ there exists $\langle \alpha^i_{a_1}, \ldots, \alpha^i_{a_k}\rangle$ such that $\alpha^i_a \in d_a(q_i)$ for every $a \in Agt$, and $\alpha^i_a = s_A[a](q_i)$ for every $a \in A$, and $q_{i+1} = o(q_i, \alpha^i_{a_1}, \ldots, \alpha^i_{a_k})\}$.

Semantic clauses. The semantics of **ATL** is defined by the following clauses:

- $M, q, \models \mathrm{p}$ iff $q \in V(\mathrm{p})$, for $\mathrm{p} \in Prop$;
- $M, q, \models \neg\varphi$ iff $M,, \not\models \varphi$;
- $M, q, \models \varphi_1 \wedge \varphi_2$ iff $M, q \models \varphi_1$ and $M, q \models \varphi_2$;
- $M, q \models \langle\langle A\rangle\rangle X\varphi$ iff there is a strategy $s_A \in \Sigma_A^{Ir}$ such that, for each path $\lambda \in out(q, s_A)$, we have $M, \lambda[1] \models \varphi$.
- $M, q \models \langle\langle A\rangle\rangle G\varphi$ iff there is a strategy $s_A \in \Sigma_A^{Ir}$ such that, for each path $\lambda \in out(q, s_A)$, we have $M, \lambda[i] \models \varphi$ for all $i \geq 0$.
- $M, q \models \langle\langle A\rangle\rangle \varphi_1\, \mathcal{U}\, \varphi_2$ iff there is a strategy $s_A \in \Sigma_A^{Ir}$ such that, for each path $\lambda \in out(q, s_A)$, we have $M, \lambda[i] \models \varphi_2$ for some $i \geq 0$ and $M, \lambda[j] \models \varphi_1$ for all $0 \leq j < i$.

2.3 Imperfect Information and Knowledge

Realistic multi-agent interaction always includes some degree of limited observability. Here, we use the classical variant of "**ATL** with imperfect information", defined as follows:

We extend concurrent game structures with indistinguishability relations $\sim_a$, for each $a \in Agt$. The resulting structure $S = \{Agt, St, \{\sim_a \mid a \in Agt\}, Act, d, out\}$ is called a concurrent epistemic game structure (CEGS). A CEGS together with a set of atomic propositions Pv and a valuation function $V : Pv \to 2^{St}$ is called a concurrent epistemic game model CEGM.

Strategies of agents must specify identical choices in indistinguishable situations. That is, strategies with imperfect information (*ir* strategies, for short) are functions $s_a : St \to Act$ such that (1) $s_a(q) \in d(a, q)$, and (2) if $q \sim_a q'$ then $s_a(q) = s_a(q')$. [2] As before, collective strategies for $A \subseteq Agt$ are tuples of individual strategies for $a \in A$. We denote the set of A's imperfect information strategies by Σ_A^{ir} (with the lowercase "i" for imperfect **i**nformation).

The semantics of "**ATL** with imperfect information" ($\mathbf{ATL}_{ir}$) differs from the one presented in Section 2.1 only in that the strategies are taken from Σ_A^{ir} instead of Σ_A^{Ir}. In other words, the agents in A should have an executable strategy which enforces φ from all the states that at least one member of the coalition considers possible.

Alternating-time temporal epistemic logic **ATLK** adds the knowledge modality of the *multi-agent epistemic logic* to **ATL** with imperfect information. In multi-agent epistemic logic, expressing the *knowledge* of the agents is formalised by epistemic formulae of type $\mathcal{K}_a\varphi$, stating that agent *a knows that* φ *holds*, with the following semantics:

[1] This corresponds to the notion of *memoryless* or *positional* strategies. In other words, we assume that the memory of agents is explicitly defined by the states of the model.

[2] Again, we consider only positional strategies here.

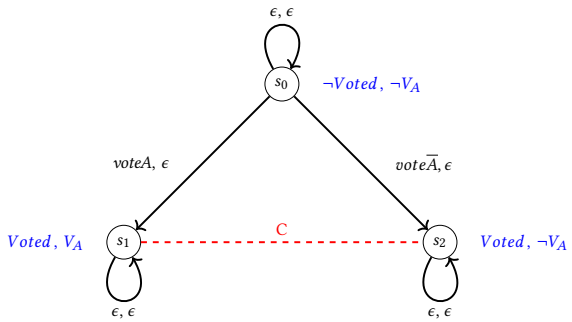


Figure 1: Single issue referendum with one voter and one coercer

$M, q \models \mathcal{K}_a\varphi$ iff, for every state q' such that $q \sim_a q'$, we have that $M, q' \models \varphi$.

The formula stating mutual knowledge $E_A\varphi$ (read "everybody in A knows that φ") is defined as:

$$M, q \models E_A\varphi \text{ iff } M, q \models \mathcal{K}_a\varphi, \text{ for all } a \in A.$$

3 MOTIVATING EXAMPLE

In this section we show, using an example, how our proposed logic can express more refined epistemic goals of agents using considerably more concise formulas. As we will see, not only the new formulation of these epistemic properties will be significantly shorter; the interpretation of the formulas will also be easier to understand in comparison to their analogous formulas in **ATLK**.

3.1 Coercion in Referendums

We consider a very simple scenario of an election with a single voter and a single coercer. The election is a referendum, in the sense that each voter has to either vote for an issue in question or to vote against it. We consider two variants. In the first one there is only one issue put for referendum (we call it proposal A). The model consists of two agents, the voter v and the coercer c. The set of possible actions for the coercer in the model is $\{\epsilon\}$ and for the voter is $\{voteA, vote\overline{A}, \epsilon\}$, where ϵ represents a *null* action (meaning the action of doing nothing). $voteA$ and $vote\overline{A}$ respectively represent voting for and against the proposal A. The atomic proposition V_A states that the vote is cast in favor of proposal A, while the atomic proposition $Voted$ shows that the vote is already cast. The game model is depicted in Figure 1.

The valuations of atomic propositions are depicted in blue, and the red dashed line between s_1 and s_2 shows that the these two states are indistinguishable for player c. In this simple model, we might express the property of coercion resistance in **ATLK** as follows:

$$M, s_0 \models \langle\langle v\rangle\rangle F(Voted \wedge V_A \wedge G\neg(\mathcal{K}_c V_A \vee \mathcal{K}_c\neg V_A)) \\ \wedge\langle\langle v\rangle\rangle F(Voted \wedge \neg V_A \wedge G\neg(\mathcal{K}_c V_A \vee \mathcal{K}_c\neg V_A))$$

The formula states that the voter has a strategy to vote for the proposal A or against it, in a way that in either case the coercer does not know the value of the vote.

3.2 Referendum with Multiple Proposals

Consider a more sophisticated variant in which the voter participates in a double referendum, i.e., votes on two proposals A and B on a single ballot. The set of atomic propositions in this scenario is $\{Voted, V_A, V_B\}$ and the set of actions for the voter is $\{voteAB, vote\overline{AB}, voteA\overline{B}, vote\overline{B}A\}$. For expressing the property of coercion resistance in this scenario, a seemingly reasonable way is to extend the above formulas as below:

$$\begin{aligned} M, s_0 \models\ & \langle\langle v\rangle\rangle F(Voted \wedge V_A \wedge V_B \wedge \\ & G\neg(\mathcal{K}_c V_A \vee \mathcal{K}_c\neg V_A \vee \mathcal{K}_c V_B \vee \mathcal{K}_c\neg V_B)) \\ \wedge\ & \langle\langle v\rangle\rangle F(Voted \wedge V_A \wedge \neg V_B \wedge \\ & G\neg(\mathcal{K}_c V_A \vee \mathcal{K}_c\neg V_A \vee \mathcal{K}_c V_B \vee \mathcal{K}_c\neg V_B)) \\ \wedge\ & \langle\langle v\rangle\rangle F(Voted \wedge \neg V_A \wedge V_B \wedge \\ & G\neg(\mathcal{K}_c V_A \vee \mathcal{K}_c\neg V_A \vee \mathcal{K}_c V_B \vee \mathcal{K}_c\neg V_B)) \\ \wedge\ & \langle\langle v\rangle\rangle F(Voted \wedge \neg V_A \wedge \neg V_B \wedge \\ & G\neg(\mathcal{K}_c V_A \vee \mathcal{K}_c\neg V_A \vee \mathcal{K}_c V_B \vee \mathcal{K}_c\neg V_B)) \end{aligned}$$

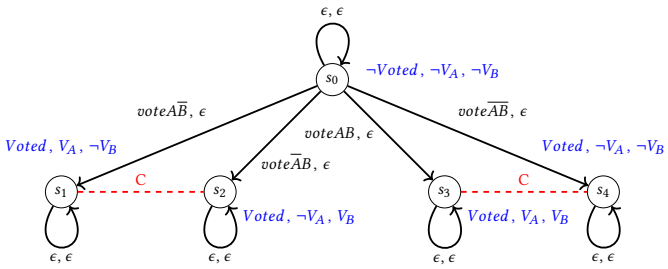
The formula states that the voter can vote in any combination, for or against A or B, without the coercer knowing the value of any single vote. At the first glance these security properties seem to be strong enough for capturing the desirable property of coercion resistance. However, if we look at the two models in Figure 2, both of them satisfy the property above. On the other hand, we would consider model M_1 less secure than M_2. There are 4 possible combinations of the valuations of V_A and V_B. In M_2, the coercer considers all 4 of them as plausible, but in M_1 he could narrow that down to only two possible combinations. In other words, the uncertainty of the coercer about propositions V_A and V_B is higher in M_2 than in M_1. In fact, as we shall see later, it is possible to write a formula in the language of **ATLK** that keeps the above property and yet distinguishes M_1 and M_2. But if we want to write a security property in **ATLK** that rejects all the models where the coercer has more distinguishing power over states s_1 to s_4 than the model M_2, then the length of that formula would be very large – in the worst case, even exponential in the number of distinguishing properties.

3.3 Reasoning about Uncertainty

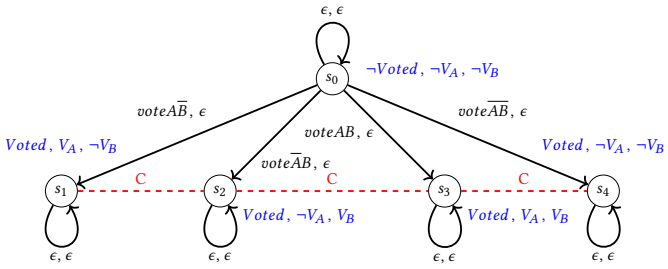
One way of looking at the above situation is that, when reaching any of the states s_1 to s_4, we want the coercer to have the least amount of information, or in other words the maximum uncertainty about the possible values of V_A and V_B. To express this concept, we can use one of the well known quantitative measures of uncertainty. Two measures that come to mind are Shannon entropy and Hartley measure. Choosing Shannon entropy would be meaningful only if we knew the intrinsic probabilities of each state. However, in the models that we are using, and in the scenarios similar to the one above, what we are interested is the uncertainty of the agents about different possible outcomes of a set of properties (here V_A and V_B). We recall the definition of Hartley measure below:

Definition 3.1 (Hartley measure of uncertainty [29]). If A is a set of possible outcomes, then its Hartley measure is defined by $H(A) = \log |A|$.

The Hartley function coincides with Shannon entropy when ignorance can be modeled by the uniform probability distribution.



(a) M_1



(b) M_2

Figure 2: Double referendum with one voter and one coercer. Model M_1 depicts a scenario which is less secure than $M2$

Using this measure, what we want to specify as a security property in the example above is that the uncertainty of the coercer about the values of V_A and V_B should be maximal. The set of properties of interest $\{V_A, V_B\}$ could have $2^2 = 4$ different combinations of values. Therefore if we want that the coercer considers all of these combinations as possible, the Hartley measure of uncertainty of the coercer would be $\log 4 = 2$ bits. To express this, we add a new operator $\mathcal{H}$, and write the formula:

$$\langle\langle v \rangle\rangle F(Voted \wedge \mathcal{H}_c^{\geq 2}\{V_A, V_B\})$$

The formula states that the voter has a strategic ability to eventually cast her vote, while keeping the uncertainty of the coercer about the valuations of V_A and V_B at the level of at least 2 bits. Intuitively, the formula holds in state s_1 of model M_2, but not M_1.

In the next section, we use this idea to formalize the syntax and semantics of the logic **ATLH**.

4 ATL WITH UNCERTAINTY

In this section we define the syntax and semantics of the logic of strategic abilities with uncertainty operator **ATLH**. The logic is based on the idea of using the Hartley measure to quantify the uncertainty of agents about the possible valuations of a set of formulas. Similarly to **ATLK**, the semantics of **ATLH** is also defined with respect to concurrent epistemic game models (CEGM).

4.1 Syntax

The syntax of **ATLH** is given as follows:

$$\varphi ::= \mathrm{p} \mid \neg\varphi \mid \varphi \wedge \varphi \mid \langle\langle A \rangle\rangle X\varphi \mid \langle\langle A \rangle\rangle G\varphi \mid \langle\langle A \rangle\rangle \varphi\, \mathcal{U}\, \varphi \mid \mathcal{H}_a^{\otimes m}\beta.$$

where $A \in 2^{Agt}$ is a set of players, $\beta \in 2^{\varphi} \setminus \{\emptyset\}$ is a set of formulas, $a \in Agt$ is a player, and $\otimes \in \{<, \leq, >, \geq, =\}$ is a comparison operator. For instance, the formula $\mathcal{H}_a^{>m}\beta$ states that the the uncertainty of agent a about the set of formulas β is higher than m.

4.2 Semantics

Let $[q]_{\sim_a} = \{q' \in St \mid q' \sim_a q\}$ denote the abstraction class of state $q \in St$ with respect to relation $\sim_a$, i.e., the epistemic neighbourhood of q from the perspective of agent $a \in Agt$. For a given formula φ, we define relation $\sim^{\varphi} \in St \times St$ that connects states with the same valuation of φ:

$$q_1 \sim^{\varphi} q_2 \text{ iff } M, q_1 \models \varphi \Leftrightarrow M, q_2 \models \varphi.$$

If $\beta = \{\varphi_1, ..., \varphi_n\}$ is a set of formulas and $a \in Agt$, then we define

$$\sim_a^{\beta} = \sim_a \cap \bigcap_{i=1}^{n} \sim^{\varphi_i}$$

I.e., $q_1 \sim_a^{\beta} q_2$ iff q_1, q_2 look the same to a and cannot be discerned by any formula in β. Note that $\sim_a^{\beta}$ is an equivalence. We define

$$R_{a,q}(\beta) = \{[q']_{\sim_a^{\beta}} \mid q' \sim_a q\}$$

for the set of equivalence classes of $\sim_a^{\beta}$ contained in the epistemic neighbourhood of state q. Then, the truth value of the statement "agent a's uncertainty about the set of formulas β is in $\otimes$ relation to value $m \in \mathbb{R}$" can be defined as follows:

$$M, q \models \mathcal{H}_a^{\otimes m}\beta, \text{ iff } \log|R_{a,q}(\beta)| \otimes m$$

Some straightforward validities of **ATLH** are:

(1) $\mathcal{H}_a^{\substack{=m \\ \geq m}}\beta \rightarrow \mathcal{H}_a^{\substack{\geq m \\ \geq m}}\beta'$, for all $\beta \subseteq \beta'$;

(2) $\mathcal{H}_a^{\substack{=m \\ \leq m}}\beta \rightarrow \mathcal{H}_a^{\substack{<m \\ \leq m}}\beta'$, for all $\beta \subseteq \beta'$.

Also, if $|St|$ is the number of states in the model, then it holds that $M, q \models \mathcal{H}_a^{<\min(|\beta|, \log(|St|))}\beta$.

4.3 Model Checking

In this section, we discuss model checking for **ATLH**. The following results have long been known in the literature:

- Model checking of epistemic logic is in **P** with respect to the size of the model and the length of the formula [28].
- Model checking of **ATLK** for agents with *ir* strategies is $\Delta_2^{\mathbf{P}}$-complete with respect to size of the model and the length of the formula [16]. This is a direct consequence of the fact that model checking of $\mathbf{ATL}_{ir}$ is $\Delta_2^{\mathbf{P}}$-complete [33, 50].

In the following, we show that model checking of **ATLH** is also $\Delta_2^{\mathbf{P}}$-complete. To this end, it suffices to show that model checking of the uncertainty part of the language is in **P**.

Proposition 4.1. *If φ is an **ATLH** formula without strategic and temporal operators and M is a CEGM, then checking if φ is satisfied in a state q of M can be done in polynomial time with respect to $|\varphi|$ and $|M|$, where $|M|$ is the total number of states, transitions, and epistemic relation pairs in M.*

Proof. Let $\varphi_1, \varphi_2, ...\varphi_k$ be the subformulas of φ (which incrementally generate φ) listed in order of length. We can see that $k \leq |\varphi|$, as there are at most $|\varphi|$ subformulas of φ. We start labeling each state in M in increasing order of i, with labels φ_i or $\neg\varphi_i$, depending

on whether φ_i is true in that state or not. It is easy to see that we can do this in at most $\mathbf{O}(k|M|)$ labeling step. If the formula φ_i is a propositional formula with respect to it's subformulas, then it can be labeled in in each state in constant time. In cases where φ_i is of the form $\mathcal{H}_a^{\otimes m}\beta$ where $\otimes \in \{<, >, =\}$ and $\beta = \{\alpha_1, ...\alpha_{k'}\}$, we have that each α_j is a subformula of φ_i. Therefore for labeling φ_i we construct the set of equivalence classes $R_{a,q}(\beta)$ by checking the k' labels of formulas in β in all the states q' where $q' \sim_a q$. Then we calculate $\log|R_{a,q}(\beta)|$ and compare it with m in order to label φ_i. This procedure can be done in at most $\mathbf{O}(k'|M|)$ steps. Therefore the whole process of checking whether φ is satisfied in a state q or not can be done in at most $\mathbf{O}(|\varphi|^2|M|^2)$. □

Proposition 4.2. *Model checking of **ATLH** for agents with ir strategies is $\Delta_2^{\mathbf{P}}$-complete with respect to size of the model and the length of the formula.*

Proof. The lower bound follows from the fact that **ATLH** subsumes $\mathbf{ATL}_{ir}$, and model checking $\mathbf{ATL}_{ir}$ is $\Delta_2^{\mathbf{P}}$-hard. The upper bound is straightforward from Proposition 4.1 and the fact that model checking $\mathbf{ATL}_{ir}$ is in $\Delta_2^{\mathbf{P}}$. □

5 EXPRESSIVE POWER OF ATLH

In this section we show that **ATLH** and **ATLK** have the same expressive power. We start by recalling the semantic definition of comparative expressivity [58].

Definition 5.1 (Expressivity). Let $L_1 = \langle \Phi_1, \models_1, \mathbb{M}\rangle$ and $L_1 = \langle \Phi_2, \models_2, \mathbb{M}\rangle$ represents two logics, such that Φ_1 and Φ_2 are the set of formulas defined in these logics, $\mathbb{M}$ is a nonempty class of models (or pointed models) over which the logics are defined, and $\models_1$ and $\models_2$ are the truth relations of these logics, such that $\models_1 \subseteq \mathbb{M} \times \Phi_1$ and $\models_2 \subseteq \mathbb{M} \times \Phi_2$. We say that L_2 is at least as expressive as L_1 on the class of models $\mathbb{M}$, iff for every formula $\varphi_1 \in \Phi_1$, there exists a formula $\varphi_2 \in \Phi_2$ such that for every $M \in \mathbb{M}$ we have $M \models_1 \varphi_1$ iff $M \models_2 \varphi_2$. We will write it as $L_1 \leqslant_{\mathbb{M}} L_2$.

If both $L_1 \leqslant_{\mathbb{M}} L_2$ and $L_2 \leqslant_{\mathbb{M}} L_1$, then we say that L_1 and L_2 are equally expressive on $\mathbb{M}$, and write $L_1 =_{\mathbb{M}} L_2$.

In the following, we use $\models_{\mathcal{K}}$ and $\models_{\mathcal{H}}$ to denote the semantic relation of **ATLK** and **ATLH**, respectively, whenever it might not be clear from the context.

5.1 Knowledge as Uncertainty

Theorem 5.2. ***ATLH** is at least as expressive as **ATLK**.*

Proof. Because the set of formulas defined in **ATLH** includes all the formulas defined in **ATLK** except the formulas including $\mathcal{K}$ operator, and the semantics of the common formulas are similar in both logics, it suffices to prove that for any formula of type $\varphi_1 = \mathcal{K}_a\varphi$ in **ATLK** there is a formula φ_2 in **ATLH** such that for every M,

$$M, q \models \mathcal{K}_a\varphi \;\Leftrightarrow\; M, q \models_{\mathcal{H}} \varphi_2$$

We claim that we can construct such φ_2 from $\mathcal{K}_a\varphi$ to be $\varphi_2 = \varphi \wedge \mathcal{H}_a^{=0}\{\varphi\}$. Therefore we need to prove that:

$$M, q \models_{\mathcal{K}} \mathcal{K}_a\varphi \;\Leftrightarrow\; M, q \models_{\mathcal{H}} \varphi \wedge \mathcal{H}_a^{=0}\{\varphi\}$$

We have that $M, q \models_{\mathcal{K}} \mathcal{K}_a\varphi$ if and only if φ holds in all the indistinguishable states from q for a, which includes state q itself. This means that φ holds in q and $|R_{a,q}(\varphi)| = 1$, which in **ATLH** would be expressed as $M, q \models_{\mathcal{H}} \varphi \wedge \mathcal{H}_a^{=0}\{\varphi\}$. □

5.2 Uncertainty as Knowledge

Theorem 5.3. ***ATLK** is at least as expressive as **ATLH**.*

The proof proceeds by translating every occurrence of $\mathcal{H}_a^{\otimes m}\beta$ to a Boolean combination of epistemic formulas that express the knowledge of agent a with respect to the *indistinguishability classes* of the formulas in β, defined as follows:

Definition 5.4 (Indistinguishability class of a formula). For a given model M, if $q \in St$, $a \in Agt$ and φ is a formula, then we define the indistinguishability class of φ with respect to q and a as follows:

$$[\varphi]_a^q = [\varphi] \cap [q]_{\sim_a},$$

where $[q]_{\sim_a}$ denotes the set of states that are indistinguishable from q for a, and $[\varphi]$ is the set of states $q' \in St$ were $M, q' \models \varphi$.

The full proof is technical and rather tedious; it can be found in the extended version of the paper [52]. Here, we present how the translation works on an example. Let φ_1 and φ_2 be two formulas that do not contain any $\mathcal{H}$ operators. We would like to find an **ATLK** formula $P(\varphi_1, \varphi_2)$, such that:

$$M, q \models \mathcal{H}_a^{=\log 3}\{\varphi_1, \varphi_2\} \;\Leftrightarrow\; M, q \models_{\mathcal{K}} P(\varphi_1, \varphi_2)$$

First we define new formulas A, B, C and D as follows: $A = \varphi_1 \wedge \varphi_2$, $B = \varphi_1 \wedge \neg\varphi_2$, $C = \neg\varphi_1 \wedge \varphi_2$ and $D = \neg\varphi_1 \wedge \neg\varphi_2$. It is clear that the sets of states $[A]_a^q$, $[B]_a^q$, $[C]_a^q$ and $[D]_a^q$ are mutually exclusive, and moreover they partition $[q]_{\sim_a}$. Because the truth value of each one of A, B, C, D corresponds to the truth value of exactly one of four possible different valuation combinations of φ_1 and φ_2, so they are distinct.

If $M, q \models \mathcal{H}_a^{=\log 3}\{\varphi_1, \varphi_2\}$, then exactly one of $[A]_a^q$, $[B]_a^q$, $[C]_a^q$ or $[D]_a^q$ has to be empty. Because these sets are mutually disjoint, if all are non-empty then we should have at least four different states in $[q]_{\sim_a}$ with the four different valuation combinations for the formulas φ_1 and φ_2. This would mean that $M, q \models \mathcal{H}_a^{=\log 4}\{\varphi_1, \varphi_2\}$ which contradicts $M, q \models \mathcal{H}_a^{=\log 3}\{\varphi_1, \varphi_2\}$. Similarly if more than one of $[A]_a^q$, $[B]_a^q$, $[C]_a^q$ or $[D]_a^q$ are empty, then it means that only two or less possible valuation combinations of φ_1 and φ_2 exist in $[q]_{\sim_a}$. This entails that $M, q \models \mathcal{H}_a^{<\log 3}\{\varphi_1, \varphi_2\}$, which is again a contradiction. The converse is also true: if exactly three of the sets $[A]_a^q$, $[B]_a^q$, $[C]_a^q$ or $[D]_a^q$ are non-empty, then there are exactly three valuation combinations of φ_1 and φ_2 in $[q]_{\sim_a}$, which follows that $M, q \models \mathcal{H}_a^{=\log 3}\{\varphi_1, \varphi_2\}$. So the formula $M, q \models \mathcal{H}_a^{=\log 3}\{\varphi_1, \varphi_2\}$ holds if and only if exactly one of $[A]_a^q$, $[B]_a^q$, $[C]_a^q$ or $[D]_a^q$ is empty. This can happen in four different ways (one corresponding to each of $[A]_a^q$, $[B]_a^q$, $[C]_a^q$ or $[D]_a^q$ being empty).

First consider the case where $[C]_a^q$ is empty. Then:

$$\begin{aligned}
&\nexists q' \; s.t \;\; q' \sim_a q \; and \; M, q' \models C \\
\Leftrightarrow &(\forall q', \; q' \sim_a q \implies M, q' \not\models C) \\
\Leftrightarrow &(\forall q', \; q' \sim_a q \implies M, q' \models \neg C) \\
\Leftrightarrow &M, q \models \mathcal{K}_a\neg C
\end{aligned}$$

In a similar way we can show that $[A]^q_a$ is non-empty iff $M, q \models \neg\mathcal{K}_a\neg A$. The same goes for $[B]^q_a$, and $[D]^q_a$. Therefore among $[A]^q_a$, $[B]^q_a$, $[C]^q_a$ and $[D]^q_a$, only $[C]^q_a$ is empty iff:

$$M, q \models \neg\mathcal{K}_a\neg A \wedge \neg\mathcal{K}_a\neg B \wedge \mathcal{K}_a\neg C \wedge \neg\mathcal{K}_a\neg D$$

We got this result by assuming that only $[C]^q_a$ is empty. Given that $M, q \models \mathcal{H}_a^{=\log 3}\{\varphi_1, \varphi_2\}$ iff exactly one of $[A]^q_a$, $[B]^q_a$, $[C]^q_a$ or $[D]^q_a$ is empty, and knowing that we have four possible choices for which one is to be empty, we get that:

$$M, q \models \mathcal{H}_a^{=\log 3}\{\varphi_1, \varphi_2\} \Leftrightarrow M, q \models_{\mathcal{K}} P(\varphi_1, \varphi_2)$$

where $P(\varphi_1, \varphi_2)$ is defined as:

$$\begin{aligned}
P(\varphi_1, \varphi_2) = \quad & (\mathcal{K}_a\neg(\varphi_1 \wedge \varphi_2) \wedge \neg\mathcal{K}_a\neg(\varphi_1 \wedge \neg\varphi_2) \\
& \quad \wedge \neg\mathcal{K}_a\neg(\neg\varphi_1 \wedge \varphi_2) \wedge \neg\mathcal{K}_a\neg(\neg\varphi_1 \wedge \neg\varphi_2)) \\
& \vee(\neg\mathcal{K}_a\neg(\varphi_1 \wedge \varphi_2) \wedge \mathcal{K}_a\neg(\varphi_1 \wedge \neg\varphi_2) \\
& \quad \wedge \neg\mathcal{K}_a\neg(\neg\varphi_1 \wedge \varphi_2) \wedge \neg\mathcal{K}_a\neg(\neg\varphi_1 \wedge \neg\varphi_2)) \\
& \vee(\neg\mathcal{K}_a\neg(\varphi_1 \wedge \varphi_2) \wedge \neg\mathcal{K}_a\neg(\varphi_1 \wedge \neg\varphi_2) \\
& \quad \wedge \mathcal{K}_a\neg(\neg\varphi_1 \wedge \varphi_2) \wedge \neg\mathcal{K}_a\neg(\neg\varphi_1 \wedge \neg\varphi_2)) \\
& \vee(\neg\mathcal{K}_a\neg(\varphi_1 \wedge \varphi_2) \wedge \neg\mathcal{K}_a\neg(\varphi_1 \wedge \neg\varphi_2) \\
& \quad \wedge \neg\mathcal{K}_a\neg(\neg\varphi_1 \wedge \varphi_2) \wedge \mathcal{K}_a\neg(\neg\varphi_1 \wedge \neg\varphi_2)).
\end{aligned}$$

6 UNCERTAINTY IS EXPONENTIALLY MORE SUCCINCT THAN KNOWLEDGE

The notion of succinctness [1, 51, 59] is a refinement of the notion of expressivity. Assume that one particular property can be expressed in both languages L_1 and L_2, with formulas φ_1 and φ_2 respectively. When comparing the representational succinctness of these two languages, we are interested in whether there is a significant difference in the lengths of φ_1 and φ_2. Similar to analysis of complexity, what we consider *significant* is at least exponential growth of the size of a formula in one of the languages comparing to the equivalent formula in the other language. In this section, we prove that the language of **ATLH** is exponentially more succinct than **ATLK**. We use the so called *formula size games (FSG)* from [27] to construct the proof. In brief, we will show that for any $n \in \mathbb{N}$, there is a formula φ_n of size linear to n in **ATLH**, such that for any formula φ'_n in **ATLK** with the same set of satisfying models as φ_n, the parse tree of φ'_n will have at least 2^n distinct nodes, and therefore the size of φ'_n is at least exponential in n.

6.1 Succinctness and Formula Size Games

Before showing that **ATL** with uncertainty is exponentially more succinct than **ATL** with knowledge, we summarize the basic terminology.

*Definition 6.1 (Length of formulas in **ATLH**).* The length of formula φ is denoted by $|\varphi|$, recursively defined as follows:
$|p| = 1$,
$|(\varphi_1 \vee \varphi_2)| = |\varphi_1 \,\mathcal{U}\, \varphi_2| = |\varphi_1| + |\varphi_2| + 1$,
$|\neg\varphi| = |X\varphi| = |G\varphi| = 1 + |\varphi|$
$|\langle\langle A\rangle\rangle\varphi| = |A| + |\varphi|$,
$|\mathcal{H}_a^{\otimes m}\beta| = 1 + \sum_{\varphi_i \in \beta} |\varphi_i|$.

Definition 6.2 (Succinctness). Let $L_1 = \langle \Phi_1, \models_1, M\rangle$ and $L_2 = \langle \Phi_2, \models_2, M\rangle$ be two logics such that $L_1 \leqslant_M L_2$ and let $f(x) = O(g(n))$ be a strictly increasing function. If for every $n \in (N)$ there are two formulas $\alpha_n \in \Phi_1$ and $\beta_n \in \Phi_2$ where:

- $|\alpha_n| = f(n)$
- $|\beta_n| = 2^{g(n)}$
- β_n is the shortest formula on Φ_2 that is equivalent to α_n on M,

then we say that L_1 is exponentially more succinct than L_2 on M and write it as: $L_1 \not\leqslant^{subexp}_M L_2$.

In the following, for a set of pointed models A, we use the term $A \models \varphi$ to mean that $\forall M \in A \,.\, M \models \varphi$.

Definition 6.3 (FSG). One-person formula size game (FSG) on two sets of pointed models A and B is played as follows: during the course of the game, a game tree is constructed such that each node is labeled with pair $\langle C \circ D\rangle$ of sets of pointed models. The possible moves for the player (called *the spoiler*) on each node of the tree are $\{p \in Pv, \neg, \vee, \mathcal{K}_i\}$, where $i \in Agt$. A node can be open or closed. Once a node is closed, no further move can be played there. The condition and consequences of each of possible moves are as below:

- $p \in Pv$ (*Atomic* move): the spoiler chooses $p \in Pv$ such that $C \models p$ and $D \models \neg p$. Then the node is declared closed.
- $\neg$ (*Not* move): A new node $\langle C \circ D\rangle$ is added to the tree.
- $\vee$ (*Or* move): two nodes $\langle C_1 \circ D\rangle$ and $\langle C_2 \circ D\rangle$ are added to the tree such that $C_1 \cup C_2 = C$.
- K_i (*Knows* move), where $i \in Agt$: For each $(M, s) \in D$ the spoiler chooses a pointed model (M, s') such that $s \sim_a s'$. If for some $(M, s) \in D$ such (M, s') does not exist, then the spoiler cannot play this move. All such chosen pointed models are collected in D'. Moreover, for each $(M, s) \in C$, all possible pointed models (M, s') such that $s \sim_a s'$ are added to C'. Then a new node $\langle C' \circ D'\rangle$ is added to the tree.

We say that the spoiler wins FSG starting at $\langle A \circ B\rangle$ in n moves iff there is a game tree T with root $\langle A \circ B\rangle$ and precisely n nodes such that every leaf of T is closed.

Theorem 6.4 ([27]). *The spoiler can win the FSG starting at $\langle A \circ B\rangle$ in less than k moves iff there is some $n < k$ and a formula $\varphi \in \Phi_{MEL}$ such that $A \models_{MEL} \varphi$, $B \models_{MEL} \neg\varphi$ and $|\varphi| = n$, where Φ_{MEL} is the set of formulas defined in Multiagent epistemic logic and $\models_{MEL}$ shows truth relation in it.*

The game tree through which the spoiler wins the FSG is the parse three of formula φ in the language of *Multiagent epistemic logic*. For any $\langle A \circ B\rangle$, the set of all closed game trees with root $\langle A \circ B\rangle$ is denoted by $T(\langle A \circ B\rangle)$. Consequently, the set of closed trees represents also the set of all formulas φ that could distinguish the set of pointed models A from the set of pointed models B via the truth relation $\models_{MEL}$.

6.2 ATLH Is More Succinct than ATLK

Theorem 6.4 allows us to use FSG for proving the succinctness of our new logic **ATLH** with respect to **ATLK**.

Theorem 6.5. *The logic **ATLH** is exponentially more succinct than the logic **ATLK**.*

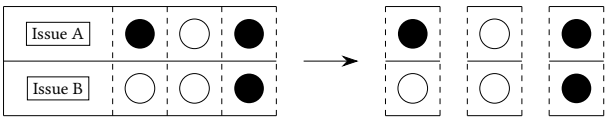


Figure 3: Example ThreeBallot vote. Here the voter has voted for issue A and against issue B. Thus, the value of the vote is $A\overline{B}$. The voter fills two fields in the row of issue A and one space in the row of issue B, and then she separates the three ballots. The resulting ballot set is $\{FB, BB, FF\}$.

Vote and ballot set (BS)	Receipt	Possible information sets of the coercer
Vote = $\overline{AB}$, BS = $\{BB, FB, BF\}$	BB	$\{\overline{AB}\}, \{A\overline{B}, \overline{AB}\}, \{\overline{A}B, \overline{AB}\}, \{A\overline{B}, \overline{A}B, \overline{AB}\}, \{A\overline{B}, \overline{A}B, AB, \overline{AB}\}$
	FB	$\{\overline{AB}\}, \{A\overline{B}, \overline{AB}\}, \{\overline{A}B, \overline{AB}\}, \{A\overline{B}, AB, \overline{AB}\}, \{A\overline{B}, \overline{A}B, AB, \overline{AB}\}$
	BF	$\{\overline{AB}\}, \{A\overline{B}, \overline{AB}\}, \{A\overline{B}, \overline{AB}\}, \{\overline{A}B, AB, \overline{AB}\}, \{A\overline{B}, \overline{A}B, AB, \overline{AB}\}$
Vote = $\overline{AB}$, BS = $\{BB, BB, FF\}$	BB	$\{\overline{AB}\}, \{A\overline{B}, \overline{AB}\}, \{A\overline{B}, \overline{AB}\}, \{AB, \overline{AB}\}, \{A\overline{B}, \overline{A}B, AB, \overline{AB}\}$
	FF	$\{\overline{AB}\}, \{A\overline{B}, \overline{AB}\}, \{A\overline{B}, \overline{AB}\}, \{AB, \overline{AB}\}, \{A\overline{B}, \overline{A}B, AB, \overline{AB}\}$
Vote = $A\overline{B}$, BS = $\{BB, FB, FF\}$	BB	$\{A\overline{B}\}, \{A\overline{B}, AB\}, \{A\overline{B}, \overline{AB}\}, \{A\overline{B}, \overline{A}B, \overline{AB}\}, \{A\overline{B}, \overline{A}B, AB, \overline{AB}\}$
	FB	$\{A\overline{B}\}, \{A\overline{B}, AB\}, \{A\overline{B}, \overline{AB}\}, \{A\overline{B}, AB, \overline{AB}\}, \{A\overline{B}, \overline{A}B, AB, \overline{AB}\}$
	FF	$\{A\overline{B}\}, \{A\overline{B}, AB\}, \{A\overline{B}, \overline{AB}\}, \{A\overline{B}, \overline{A}B, AB\}, \{A\overline{B}, \overline{A}B, AB, \overline{AB}\}$
Vote = $A\overline{B}$, BS = $\{FB, FB, BF\}$	FB	$\{A\overline{B}\}, \{A\overline{B}, \overline{A}B\}, \{A\overline{B}, AB\}, \{A\overline{B}, \overline{AB}\}, \{A\overline{B}, \overline{A}B, AB, \overline{AB}\}$
	BF	$\{A\overline{B}\}, \{A\overline{B}, \overline{A}B\}, \{A\overline{B}, AB\}, \{A\overline{B}, \overline{AB}\}, \{A\overline{B}, \overline{A}B, AB, \overline{AB}\}$
Vote = $\overline{A}B$, BS = $\{BB, BF, FF\}$	BB	$\{\overline{A}B\}, \{\overline{A}B, \overline{AB}\}, \{\overline{A}B, AB\}, \{A\overline{B}, \overline{A}B, \overline{AB}\}, \{A\overline{B}, \overline{A}B, AB, \overline{AB}\}$
	BF	$\{\overline{A}B\}, \{\overline{A}B, AB\}, \{\overline{A}B, \overline{AB}\}, \{\overline{A}B, AB, \overline{AB}\}, \{A\overline{B}, \overline{A}B, AB, \overline{AB}\}$
	FF	$\{\overline{A}B\}, \{\overline{A}B, \overline{AB}\}, \{\overline{A}B, AB\}, \{A\overline{B}, \overline{A}B, AB\}, \{A\overline{B}, \overline{A}B, AB, \overline{AB}\}$
Vote = $\overline{A}B$, BS = $\{FB, BF, BF\}$	FB	$\{\overline{A}B\}, \{\overline{A}B, \overline{AB}\}, \{\overline{A}B, AB\}, \{A\overline{B}, \overline{A}B, \overline{AB}\}, \{A\overline{B}, \overline{A}B, AB, \overline{AB}\}$
	BF	$\{\overline{A}B\}, \{\overline{A}B, \overline{AB}\}, \{\overline{A}B, AB\}, \{A\overline{B}, \overline{A}B, \overline{AB}\}, \{A\overline{B}, \overline{A}B, AB, \overline{AB}\}$
Vote = AB, BS = $\{FB, BF, FF\}$	FB	$\{AB\}, \{A\overline{B}, AB\}, \{\overline{A}B, AB\}, \{A\overline{B}, AB, \overline{AB}\}, \{A\overline{B}, \overline{A}B, AB, \overline{AB}\}$
	BF	$\{AB\}, \{A\overline{B}, AB\}, \{\overline{A}B, AB\}, \{\overline{A}B, AB, \overline{AB}\}, \{A\overline{B}, \overline{A}B, AB, \overline{AB}\}$
	FF	$\{AB\}, \{A\overline{B}, AB\}, \{\overline{A}B, AB\}, \{A\overline{B}, \overline{A}B, AB\}, \{A\overline{B}, \overline{A}B, AB, \overline{AB}\}$
Vote = AB, BS = $\{BB, FF, FF\}$	BB	$\{AB\}, \{AB, \overline{AB}\}, \{A\overline{B}, AB\}, \{\overline{A}B, AB\}, \{A\overline{B}, \overline{A}B, AB, \overline{AB}\}$
	FF	$\{AB\}, \{AB, \overline{AB}\}, \{A\overline{B}, AB\}, \{\overline{A}B, AB\}, \{A\overline{B}, \overline{A}B, AB, \overline{AB}\}$

Table 1: Indistinguishability sets of the coercer in a two voter, two issue referendum with ThreeBallot, based on the selected receipt by the voter. Each row lists the possible epistemic classes $[q]_{\sim_c}$ of the coercer, depending on the voter's vote and the way she filled her ballots (invisible to the coercer), the voter's election receipt (visible to the coercer), and the set of ballots posted on the public bulletin board

The full proof is rather technical; it can be found in the extended version of the paper [52]. Here we explain the sketch of the proof:

Proof sketch. Let $Lang(\mathbf{ATLK})$ and $Lang(\mathbf{ATLH})$ represent the languages of the logics **ATLK** and **ATLH** respectively. For every $n \in \mathbb{N}$, we define a formula $\varphi_n \in Lang(\mathbf{ATLK})$ where $f(n) = n$. Then, we define two sets of pointed models A_n and B_n, such that $A_n \models_{\mathcal{H}} \varphi_n$ and $B_n \models_{\mathcal{H}} \neg\varphi_n$. Because **ATLK** is as expressive as **ATLH**, there exists a formula $\psi_n \in Lang(\mathbf{ATLK})$ which is the shortest formula in $Lang(\mathbf{ATLK})$ equivalent to φ_n. Therefore ψ_n too can distinguish sets A_n and B_n, which means the spoiler can win the FSG game starting at $\langle A_n \circ B_n \rangle$ by playing the formula ψ_n. We then prove that the spoiler cannot win the FSG game starting at $\langle A_n \circ B_n \rangle$ in less than 2^n moves, which means the size of ψ_n is at least 2^n. □

7 CASE STUDY: THREE BALLOT

We demonstrate the usefulness of our proposal on a real-life voting scenario.

7.1 Voting with ThreeBallot

ThreeBallot [48, 49] has been proposed by Rivest as a paper-based end-to-end verifiable voting protocol. Here, we use a simplified version of the protocol, which can be used for multiple-issue referendums. Following the example in Section 3, we consider a two-issue referendum, in which each voter votes to accept or reject two proposals A and B.

In ThreeBallot, the ballots are prepared such that for each issue, there are three empty fields that the voter can fill. For voting to accept the issue, the voter has to fill exactly two of the empty fields, and to vote to reject the issue, the voter has to fill exactly one empty field. However, the exact positions of the filled spaces are up to the voter. After filling the ballots, the voter separates three columns of fields, and in this way creates three separate ballots. The voter can make (and keep) a copy of one of those ballots as the receipt. Finally, she puts all the original ballots in the ballot box. The tally of the votes is done by counting the filled spaces for each issue. The difference between the number of filled spaces for each issue and the number of voters shows the number of votes in favor of that issue. After the tallying, all the ballots are published on a public bulletin board, so that everyone can check the correctness of the result.

One of the main goals of ThreeBallot is *coercion resistance* [38], i.e., the protocol should make it impossible for a third party to successfully coerce or bribe voters into voting in a particular way. Informally, coercion resistance is usually understood as the inability of the coercer to learn how the voter has voted, even if the voter cooperates with the coercer. Interestingly, ThreeBallot was both claimed secure [49] and insecure [10]. It might seem that one of the claims must be wrong, but a closer look reveals that they are based on different concepts of vote privacy. In [49], it is argued that the coercer cannot get to *know how the voter has voted*, which is a strategic-epistemic property. In contrast, [10] argues that the coercer can *gain information* about the value of the vote. We demonstrate the difference in the remainder of this section.

7.2 Model of the Scenario

In our example, the two propositions that determine the vote of the voter are V_A and V_B. We encode a vote against an issue by using *overline*. So, $\overline{A}B$ indicates a vote for issue B and against issue A; in other words, it states that V_A is *false* and V_B is *true*. This way, the set of possible votes from the voter is $Votes = \{AB, \overline{A}B, A\overline{B}, \overline{AB}\}$. Similarly, we encode a filled space by F and a blank (not filled) space by B. For instance, a $\{Blank, Filled\}$ ballot is denoted by BF. Figure 3 depicts an example ThreeBallot card and the resulting ballots.

We consider a scenario with two voters and a two issue referendum with issues A and B. Let us call the first voter *the voter*, as she will be the one that we are focusing on in this example. We call the second voter *the other voter*. During the election, the voter has several choices. The first is what vote she is going to cast. Then for each possible vote, there are various ways that the ballots can

be filled, which will result in different ballot sets. After that, the voter has the choice of which of the ballots to keep a copy as the receipt. In our scenario, there exist a coercer who after the election will force the voter to reveal her receipt. The coercer then tries to infer the actual value of the voter's vote, based on the receipt and the published bulletin board.

Table 1 shows the different ways how the choices of the voter affect the possible indistinguishability set of the coercer about the value of the vote, after the receipt has been revealed. The different indistinguishability sets in each row result from various ways in which the other voter might fill his ballot.

7.3 Analysis: Epistemic Security

The coercion resistance security property is usually framed following the idea that the coercer cannot *get to know* how the voter has voted, even if the voter cooperates with the coercer. In [53], various nuances of coercion resistance are formulated in the logic **ATLK**. In a similar way, we can use **ATLK** to express coercion resistance in our ThreeBallot example as follows:

$$\bigwedge_{V_i \in Votes} \neg\langle\langle v, c\rangle\rangle F(V_1 = V_i \wedge (V_1 \neq V_2 \rightarrow \mathcal{K}_c(V_1 = V_i)))$$

The formula states that for any vote choice, there exists no common strategy for the voter and the coercer, such that the voter selects that vote and given that the choice of the two voters are different (the reason for this condition is explained below), the coercer would know the value of the vote.[3]

It is obvious that in the cases where both voters have voted identically, even without revealing the receipt, the coercer will know the value of the vote just by looking at the bulletin board. This is similar to the case when in an election all the voters vote similarly, in which case the privacy of their votes will be broken after publishing the tally (unless some sort of obfuscation is used [35]). We added the condition $(V_1 \neq V_2)$ to the above formula to account for this. Also in the following we only focus on the cases where the two voters has voted differently.

By looking at Table 1 we can see that the model satisfies the coercion resistance property as formulated in the above **ATLK** expression. This is because there is no row in the table that consists of only one indistinguishability set for the coercer which has only one member (the actual value of the vote). However the voter votes and selects the receipt, there is at least one possible indistinguishability set with more than one member, meaning that the coercer might not get to know the actual vote of the voter.

7.4 Information-Theoretic Security in ATLH

We can alternatively define the coercion resistance property in the information-theoretic sense, namely that the coercer cannot *gain information* on how the voter has voted, even if the voter cooperates with the coercer. Phrasing this differently, we want that no matter the course of actions of the voter and coercer, the coercer has always maximum uncertainty about the actual value of the vote. We can express this property in **ATLH** as follows:

$$\bigwedge_{V_i \in Votes} \neg\langle\langle v, c\rangle\rangle F(V_1 = V_i \wedge (V_1 \neq V_2 \rightarrow \mathcal{H}_c^{=\log(|Votes|)}\{V_A, V_B\}))$$

[3]Note that we use $(V_1 = V_i)$ in the role of an atomic proposition which evaluates to true whenever V_1 is indeed equal to V_i. Condition $(V_1 \neq V_2)$ is treated analogously.

The above formula states that, for any joint strategy of the coercer and the voter, the uncertainty of the coercer will always be at the maximum. Looking at Table 1, we can see that the ThreeBallot protocol does not satisfy this property. This is because in each row there exists a possible indistinguishability set whose size is less than the number of possible votes.

This example shows that, although ThreeBallot could be considered secure with respect to the epistemic notion of coercion resistance expressed in **ATLK**, it is not secure when we define the security requirement as an information-theoretic property, and formalize it in **ATLH**.

8 CONCLUSION

In this work, we introduce the logic **ATLH** which extends alternating-time temporal logic **ATL** with quantitative modalities based on the Hartley measure of uncertainty. As the main technical result, we show that **ATLH** has the same expressive power and the same model checking complexity as **ATLK** (i.e., **ATL** with epistemic modalities), but it is exponentially more succinct.

The succinctness result, together with the model checking complexity, is of major significance. As we have seen in Section 4.3, both **ATLK** and **ATLH** have the same verification complexity with respect to the size of the model and *the length of the formula*. Theorem 6.5 promises that, for some properties, their verification in **ATLH** will be exponentially faster than in **ATLK**. Also, a more succinct language often results in better readability, which in turn helps the designers of a system to make less mistakes in the specification of desired properties. Last but not least, many properties can be expressed in **ATLH** in a much more intuitive way than in **ATLK**. Understanding the information-theoretic intuition of a corresponding **ATLK** formula can be a real challenge.

We suggest the specification of security requirements as an important application area for our proposal. In particular, the framework can be used to expose the logical structure of security claims, for example, the difference between the epistemic and information-theoretic notions of privacy. We demonstrate this on a real-life voting scenario involving the ThreeBallot protocol, which has been both claimed secure and insecure in the past. Indeed, the protocol is secure with respect to an epistemic notion of privacy, but it may fail to obtain the information-theoretic one.

In the future, we plan to implement model checking for **ATLH** as an extension of the STV [42] or MCMAS [43] model checkers.

ACKNOWLEDGMENTS

The work has been supported by NCBR Poland and FNR Luxembourg under the PolLux/FNR-CORE projects STV (POLLUX-VII/1/2019 & C18/IS/12685695/IS/STV/Ryan) and SpaceVote (POLLUX-XI/14/SpaceVote/2023).

REFERENCES

[1] Micah Adler and Neil Immerman. 2001. An n! Lower Bound on Formula Size. In *Proceedings of IEEE Symposium on Logic in Computer Science*. 197–206.
[2] T. Ågotnes. 2006. Action and Knowledge in Alternating-time Temporal Logic. *Synthese* 149, 2 (2006), 377–409.
[3] T. Ågotnes, V. Goranko, W. Jamroga, and M. Wooldridge. 2015. Knowledge and Ability. In *Handbook of Epistemic Logic*. 543–589.
[4] T. Ågotnes and H. van Ditmarsch. 2008. Coalitions and Announcements. In *Proceedings of International Joint Conference on Autonomous Agents and Multiagent*

Systems (AAMAS). 673–680.

[5] R. Alur, L. de Alfaro, R. Grossu, T.A. Henzinger, M. Kang, C.M. Kirsch, R. Majumdar, F.Y.C. Mang, and B.-Y. Wang. 2001. jMocha: A Model-Checking Tool that Exploits Design Structure. In *Proceedings of International Conference on Software Engineering (ICSE)*. 835–836.

[6] R. Alur, T. Henzinger, F. Mang, S. Qadeer, S. Rajamani, and S. Tasiran. 1998. MOCHA: Modularity in Model Checking. In *Proceedings of Computer Aided Verification (CAV) (Lecture Notes in Computer Science, Vol. 1427)*. 521–525.

[7] R. Alur, T. A. Henzinger, and O. Kupferman. 1997. Alternating-Time Temporal Logic. In *Proceedings of the 38th Annual Symposium on Foundations of Computer Science (FOCS)*. 100–109.

[8] R. Alur, T. A. Henzinger, and O. Kupferman. 2002. Alternating-Time Temporal Logic. *J. ACM* 49 (2002), 672–713. https://doi.org/10.1145/585265.585270

[9] Benjamin Aminof, Vadim Malvone, Aniello Murano, and Sasha Rubin. 2018. Graded modalities in Strategy Logic. *Information and Computation* 261 (2018), 634–649.

[10] Andrew W. Appel. 2006. How to defeat Rivest's ThreeBallot voting system. (2006). https://www.cs.princeton.edu/~appel/papers/DefeatingThreeBallot.pdf

[11] Francesco Belardinelli, Wojtek Jamroga, Vadim Malvone, Munyque Mittelmann, Aniello Murano, and Laurent Perrussel. 2022. Reasoning about Human-Friendly Strategies in Repeated Keyword Auctions. In *Proceedings of AAMAS*. 62–71.

[12] F. Belardinelli, A. Lomuscio, A. Murano, and S. Rubin. 2017. Verification of Broadcasting Multi-Agent Systems against an Epistemic Strategy Logic. In *Proceedings of IJCAI*. 91–97.

[13] T. Bolander and M. Birkegaard Andersen. 2011. Epistemic planning for single- and multi-agent systems. *Journal of Applied Non-Classical Logics* 21, 1 (2011), 9–34.

[14] Patricia Bouyer, Orna Kupferman, Nicolas Markey, Bastien Maubert, Aniello Murano, and Giuseppe Perelli. 2019. Reasoning about Quality and Fuzziness of Strategic Behaviours. In *Proceedings of IJCAI*. 1588–1594.

[15] Laura Bozzelli, Aniello Murano, and Loredana Sorrentino. 2020. Alternating-time temporal logics with linear past. *Theor. Comput. Sci.* 813 (2020), 199–217.

[16] N. Bulling, J. Dix, and W. Jamroga. 2010. Model Checking Logics of Strategic Ability: Complexity. In *Specification and Verification of Multi-Agent Systems*. 125–159.

[17] N. Bulling, V. Goranko, and W. Jamroga. 2015. Logics for Reasoning About Strategic Abilities in Multi-Player Games. In *Models of Strategic Reasoning. Logics, Games, and Communities*. Lecture Notes in Computer Science, Vol. 8972. 93–136.

[18] N. Bulling and W. Jamroga. 2009. What Agents Can Probably Enforce. *Fundamenta Informaticae* 93, 1-3 (2009), 81–96.

[19] S. Busard, C. Pecheur, H. Qu, and F. Raimondi. 2014. Improving the Model Checking of Strategies under Partial Observability and Fairness Constraints. In *Formal Methods and Software Engineering*. Lecture Notes in Computer Science, Vol. 8829. 27–42.

[20] P. Cermak, A. Lomuscio, F. Mogavero, and A. Murano. 2014. MCMAS-SLK: A Model Checker for the Verification of Strategy Logic Specifications. In *Proc. of Computer Aided Verification (CAV) (Lecture Notes in Computer Science, Vol. 8559)*. 525–532.

[21] Petr Cermák, Alessio Lomuscio, and Aniello Murano. 2015. Verifying and Synthesising Multi-Agent Systems against One-Goal Strategy Logic Specifications. In *Proceedings of AAAI*. 2038–2044.

[22] K. Chatterjee, T. A. Henzinger, and N. Piterman. 2007. Strategy Logic. In *Proceedings of CONCUR*. 59–73.

[23] T. Chen, V. Forejt, M. Kwiatkowska, D. Parker, and A. Simaitis. 2013. PRISM-games: A Model Checker for Stochastic Multi-Player Games. In *Proceedings of Tools and Algorithms for Construction and Analysis of Systems (TACAS) (Lecture Notes in Computer Science, Vol. 7795)*. 185–191.

[24] Taolue Chen and Jian Lu. 2007. Probabilistic Alternating-time Temporal Logic and Model Checking Algorithm. In *Proceedings of FSKD, Volume 2*. 35–39.

[25] E.A. Emerson. 1990. Temporal and Modal Logic. In *Handbook of Theoretical Computer Science*. Vol. B. 995–1072.

[26] Alessandro Ferrante, Aniello Murano, and Mimmo Parente. 2008. Enriched μ-Calculi Module Checking. *Log. Methods Comput. Sci.* 4, 3 (2008).

[27] Tim French, Wiebe van der Hoek, Petar Iliev, and Barteld P. Kooi. 2013. On the succinctness of some modal logics. *Artificial Intelligence* 197 (2013), 56–85.

[28] Joseph Y Halpern and Moshe Y Vardi. 1991. Model checking vs. theorem proving: a manifesto. *Artificial intelligence and mathematical theory of computation* 212 (1991), 151–176.

[29] R.V.L. Hartley. 1928. Transmission of Information. *Bell Systems Technical Journal* 7 (1928), 535.

[30] X. Huang, K. Su, and C. Zhang. 2012. Probabilistic Alternating-Time Temporal Logic of Incomplete Information and Synchronous Perfect Recall. In *Proceedings of AAAI-12*.

[31] X. Huang and R. van der Meyden. 2014. Symbolic Model Checking Epistemic Strategy Logic. In *Proceedings of AAAI Conference on Artificial Intelligence*. 1426–1432.

[32] W. Jamroga and T. Ågotnes. 2007. Constructive Knowledge: What Agents Can Achieve under Incomplete Information. *Journal of Applied Non-Classical Logics* 17, 4 (2007), 423–475.

[33] W. Jamroga and J. Dix. 2006. Model Checking ATL_{ir} is Indeed Δ_2^P-complete. In *Proceedings of EUMAS (CEUR Workshop Proceedings, Vol. 223)*.

[34] Wojciech Jamroga, Michał Knapik, Damian Kurpiewski, and Łukasz Mikulski. 2019. Approximate Verification of Strategic Abilities under Imperfect Information. *Artificial Intelligence* 277 (2019).

[35] Wojciech Jamroga, Peter B. Rønne, Peter Y. A. Ryan, and Philip B. Stark. 2019. Risk-Limiting Tallies. In *Electronic Voting: Proceedings of E-Vote-ID*. 183–199.

[36] W. Jamroga and M. Tabatabaei. 2013. Accumulative Knowledge Under Bounded Resources. In *Proceedings of CLIMA XIV (Lecture Notes in Computer Science, Vol. 8143)*. 206–222.

[37] W. Jamroga and W. van der Hoek. 2004. Agents that Know how to Play. *Fundamenta Informaticae* 63, 2–3 (2004), 185–219.

[38] A. Juels, D. Catalano, and M. Jakobsson. 2005. Coercion-resistant electronic elections. In *Proceedings of the 2005 ACM workshop on Privacy in the electronic society*. ACM, 61–70.

[39] M. Kacprzak and W. Penczek. 2004. Unbounded Model Checking for Alternating-Time Temporal Logic. In *Proceedings of AAMAS*. 646–653.

[40] Jonathan Katz and Yehuda Lindell. 2020. *Introduction to Modern Cryptography, Third Edition*.

[41] Damian Kurpiewski, Wojciech Jamroga, and Michał Knapik. 2019. STV: Model Checking for Strategies under Imperfect Information. In *Proceedings of the 18th International Conference on Autonomous Agents and Multiagent Systems AAMAS 2019*. 2372–2374.

[42] Damian Kurpiewski, Witold Pazderski, Wojciech Jamroga, and Yan Kim. 2021. STV+Reductions: Towards Practical Verification of Strategic Ability Using Model Reductions. In *Proceedings of AAMAS*. 1770–1772.

[43] A. Lomuscio, H. Qu, and F. Raimondi. 2017. MCMAS: An Open-Source Model Checker for the Verification of Multi-Agent Systems. *International Journal on Software Tools for Technology Transfer* 19, 1 (2017), 9–30.

[44] A. Lomuscio and F. Raimondi. 2006. MCMAS : A Model Checker for Multi-Agent Systems. In *Proceedings of Tools and Algorithms for Construction and Analysis of Systems (TACAS) (Lecture Notes in Computer Science, Vol. 4314)*. 450–454.

[45] Nicolas Markey. 2003. Temporal logic with past is exponentially more succinct. *Bull. EATCS* 79 (2003), 122–128.

[46] F. Mogavero, A. Murano, and M.Y. Vardi. 2010. Reasoning About Strategies. In *Proceedings of FSTTCS*. 133–144.

[47] J. Pilecki, M.A. Bednarczyk, and W. Jamroga. 2014. Synthesis and Verification of Uniform Strategies for Multi-Agent Systems. In *Proceedings of CLIMA XV (Lecture Notes in Computer Science, Vol. 8624)*. 166–182.

[48] R. Rivest. 2006. The ThreeBallot Voting System. (2006). Available online at http://theory.csail.mit.edu/~rivest/Rivest-TheThreeBallotVotingSystem.pdf.

[49] R. Rivest and W. Smith. 2007. Three Voting Protocols: ThreeBallot, VAV, and Twin. In *Proceedings of USENIX/ACCURATE Electronic Voting Technology Workshop (EVT)*.

[50] P. Y. Schobbens. 2004. Alternating-Time Logic with Imperfect Recall. *Electronic Notes in Theoretical Computer Science* 85, 2 (2004), 82–93.

[51] Larry J. Stockmeyer. 1972. *The Complexity of Decision Problems in Automata Theory and Logic*. Ph.D. Dissertation. Massachusetts Institute of Technology.

[52] Masoud Tabatabaei and Wojciech Jamroga. 2023. Playing to Learn, or to Keep Secret: Alternating-Time Logic Meets Information Theory. *CoRR* (2023). https://arxiv.org/abs/2303.00067

[53] M. Tabatabaei, W. Jamroga, and Peter Y. A. Ryan. 2016. Expressing Receipt-Freeness and Coercion-Resistance in Logics of Strategic Ability: Preliminary Attempt. In *Proceedings of the 1st International Workshop on AI for Privacy and Security, PrAISe@ECAI 2016*. 1:1–1:8.

[54] W. van der Hoek and M. Wooldridge. 2002. Tractable Multiagent Planning for Epistemic Goals. In *Proceedings of the First International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS-02)*. 1167–1174.

[55] W. van der Hoek and M. Wooldridge. 2003. Cooperation, Knowledge and Time: Alternating-time Temporal Epistemic Logic and its Applications. *Studia Logica* 75, 1 (2003), 125–157.

[56] W. van der Hoek and M. Wooldridge. 2003. Model Checking Cooperation, Knowledge and Time – A Case Study. *Research in Economics* 57, 3 (2003), 235–265.

[57] H. van Ditmarsch, W. van der Hoek, and B. Kooi. 2007. *Dynamic Epistemic Logic*.

[58] Y. Wang and F. Dechesne. 2009. On expressive power and class invariance. *CoRR* abs/0905.4332 (2009).

[59] Thomas Wilke. 1999. CTL+ is Exponentially more Succinct than CTL. In *Proceedings of FSTTCS (Lecture Notes in Computer Science, Vol. 1738)*. 110–121.