Reasoning about Changes of Observational Power in Logics of Knowledge and Time*

Aurèle Barrière
ENS Rennes

Bastien Maubert
Università degli Studi di Napoli "Federico II"

Aniello Murano
Università degli Studi di Napoli "Federico II"

Sasha Rubin
Università degli Studi di Napoli "Federico II"

ABSTRACT

We study dynamic changes of agents' observational power in logics of knowledge and time. We consider CTL*K, the extension of CTL* with knowledge operators, and enrich it with a new operator that models a change in an agent's way of observing the system. We extend the classic semantics of knowledge for agents with perfect recall to account for changes of observational power, and we show that this new operator increases the expressivity of CTL*K. We reduce the model-checking problem for our logic to that for CTL*K, which is known to be decidable. This provides a solution to the model-checking problem for our logic, but it is not optimal, and we provide a direct model-checking procedure with better complexity.

KEYWORDS

Model checking; Knowledge and time; Epistemic temporal logics

ACM Reference Format:
Aurèle Barrière, Bastien Maubert, Aniello Murano, and Sasha Rubin. 2019. Reasoning about Changes of Observational Power in Logics of Knowledge and Time. In *Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), Montreal, Canada, May 13–17, 2019,* IFAAMAS, 9 pages.

1 INTRODUCTION

In multi-agent systems, agents usually have partial information about the state of the system [38]. This has led to the development of epistemic logics, often combined with temporal logics, to reason about how agents' knowledge evolve over time. Such formalisms have been applied to the analysis of, e.g., distributed protocols [17, 27] or information flow and cryptographic protocols [19, 41].

In these frameworks, an agent's view of a particular state of the system is given by an observation of that state. In all the cited settings, an agent's observation of a given state does not change over time. In other words, these frameworks have no primitive for reasoning about agents whose observation power can change. Because this phenomenon occurs in real scenarios, for instance when a user of a system is granted access to previously hidden data, we propose here to tackle this problem. Precisely, we extend classic epistemic temporal logics with a new unary operator, Δ^{o}, that represents changes of observation power, and is read "the agent changes her observation power to o". For instance, the formula $\Delta^{o_1} AF(\Delta^{o_2}(Kp \vee K\neg p))$ expresses that "For an agent with initial observation power o_1, in all possible futures there exists a point where, if the agent updates her observation power to o_2, she learns whether or not the proposition p holds". If in this example o_1 and o_2 represent different "security levels" and p is sensitive information, then the formula expresses a possible avenue for attack. The present work provides means to express and evaluate such properties.

Related work. There is a rich history of epistemic logic in AI, including the static and temporal [17, 18, 20, 21, 33], dynamic [2, 4, 10, 14, 28, 42, 44] and strategic [6, 7, 13, 23, 38] settings. The most common logics of knowledge and time are CTLK, LTLK and CTL*K, which extend the classic temporal logics CTL, LTL and CTL* with epistemic operators. Satisfiability and axiomatization have been studied in depth in [20, 21]. Model checking has also been studied, for agents with either no memory or perfect recall. For memoryless agents, knowledge operators do not add to the complexity of model checking with regards to purely temporal logics LTL, CTL and CTL* [24, 36]. For agents with perfect recall however, introducing knowledge makes the model-checking problem nonelementary, with k-Exptime upper-bound for formulas with at most k nested knowledge operators [3, 11, 15, 40], and $(k-1)$-Expspace for CTLK [1]. While it is known that no elementary procedure exists, these bounds are not known to be tight.

Two recent works involve dynamic changes of observation power. The first one [8] studies an imperfect-information extension of Strategy Logic [30, 31] in which agents can change observation power when changing strategies, but the logic does not allow reasoning about knowledge. The second [29] extends the latter with knowledge operators, and solves the model-checking problem for a fragment related to the notion of hierarchical information [25, 34, 35]. In these two works, the focus is on strategic aspects. In the present work, instead, we intend to study in depth how the possibility to reason about change of observational power affects the semantics, expressive power, and model checking of epistemic temporal logics.

Contributions. We extend CTL*K (which subsumes CTLK and LTLK) with observation-change operators Δ^{o}. For agents with perfect recall, which we study in this work, extending the classic semantics of knowledge requires to store past observations of agents, which we do thanks to the introduction of *observation records*. Starting with the mono-agent case, we solve the model-checking problem by first defining an alternative semantics which, unlike the natural one, is based on a bounded amount of information. Once the two semantics are proven to be equivalent, designing a model-checking algorithm is almost straightforward. We then extend the logic to the multi-agent case, introducing operators Δ^{o}_{a} for each agent a, and we extend our approach to solve its model-checking

*An extended abstract of this work was published in [5].

Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), N. Agmon, M. E. Taylor, E. Elkind, M. Veloso (eds.), May 13–17, 2019, Montreal, Canada.

problem. Next, we study the expressivity of our logic, showing that the observation-change operator increases expressivity. We finally provide a reduction to CTL*K which removes observation-change operators at the cost of a blow-up in the size of the model. We show that going through this reduction and using known model-checking algorithms for CTL*K is more costly than our direct approach.

2 CTL*KΔ

In this section we define the logic CTL*KΔ, which corresponds to the case of one agent. We generalize to multiple agents in Section 5.

2.1 Notation

A *finite* (resp. *infinite*) *word* over some alphabet Σ is an element of Σ^* (resp. Σ^ω). The *length* of a finite word $w = w_0 \dots w_n$ is $|w| = n + 1$, and we let $\text{last}(w) = w_n$. Given a finite (resp. infinite) word w and $0 \le i < |w|$ (resp. $i \in \mathbb{N}$), we let w_i be the letter at position i in w, $w_{\le i}$ is the prefix of w that ends at position i, and $w_{\ge i}$ is the suffix that starts at position i. We write $w \preccurlyeq w'$ if w is a prefix of w'.

2.2 Syntax

We fix a countably infinite set of atomic propositions, $\mathcal{AP}$, and a finite set of *observations* O, that represent possible observational powers of the agent. Note that in this work, "observation" does not refer to a punctual observation of a system's state, but rather a way of observing the system, or "observational power" of an agent.

As for state and path formulas in CTL*, we distinguish between *history formulas* and *path formulas* (the terminology *history formula* reflects the perfect-recall semantics we consider, for which the truth of epistemic formulas depends on the whole history).

Definition 2.1 (Syntax). The sets of history formulas φ and path formulas ψ are defined by the following grammar:

$$\begin{array}{lll}\varphi & ::= & p \mid \neg\varphi \mid \varphi \wedge \varphi \mid A\psi \mid K\varphi \mid \Delta^o\varphi \\ \psi & ::= & \varphi \mid \neg\psi \mid \psi \wedge \psi \mid X\psi \mid \psi U \psi,\end{array}$$

where $p \in \mathcal{AP}$ and $o \in O$.

CTL*KΔ formulas are all history formulas. Operators X and U are the *next* and *until* operators of temporal logics, and A is the path quantifier from branching-time temporal logics. K is the knowledge operator from epistemic logics, and $K\varphi$ reads as "the agent knows that φ is true". Our new *observation change* operator, Δ^o, reads as "the agent now observes the system with observational power o".

As usual, we define $\top = p \vee \neg p$, $\varphi \vee \varphi' = \neg(\varphi \wedge \neg\varphi)$, $\varphi \to \varphi' = \neg\varphi \vee \varphi'$, as well as the temporal operators *finally* (F) and *always* (G): $F\varphi = \top U \varphi$, and $G\varphi = \neg F \neg\varphi$.

2.3 Semantics

Models of CTL*KΔ are Kripke structures equipped with one relation $\sim_o$ on states for each observation o.

Definition 2.2 (Models). A *Kripke structure with observations* is a structure $M = (\text{AP}, S, T, V, \{\sim_o\}_{o \in O}, s^\iota, o^\iota)$, where

- $\text{AP} \subset \mathcal{AP}$ is a finite subset of atomic propositions,
- S is a set of states,
- $T \subseteq S \times S$ is a left-total[1] transition relation,
- $V : S \to 2^{\text{AP}}$ is a valuation function,
- $\sim_o \subseteq S \times S$ is an equivalence relation, for each $o \in O$,
- $s^\iota \subseteq S$ is an initial state, and
- $o^\iota \in O$ is the initial observation.

A *path* is an infinite sequence of states $\pi = s_0 s_1 \dots$ such that for all $i \ge 0$, $s_i T s_{i+1}$, and a *history* h is a finite prefix of a path. For $I \subseteq S$, we write $T(I) = \{s' \mid \exists s \in I \text{ s.t. } sTs'\}$ for the set of successors of states in I. Finally, for $o \in O$ and $s \in S$, we let $[s]_o = \{s' \mid s \sim_o s'\}$ be the equivalence class of s for relation $\sim_o$.

Remark 1. *We model agents' information via indistinguishability relations $\sim_o$, where $s \sim_o s'$ means that s and s' are indistinguishable for an agent who has observation power o. Other approaches exist. One is via* observation functions *(see, e.g., [40]), that map states to atomic observations, and where two states are indistinguishable if they have the same image. Another consists in seeing states as tuples of* local states, *one for each agent, two global states being indistinguishable for an agent if her local state is the same in both (see, e.g., [24]). All these formalisms are essentially equivalent with respect to epistemic temporal logics [32]. In these alternative formalisms, change of observation power would correspond to, respectively, changing observation function, and changing the local states inside each global state. We find that indistinguishability relations are convenient to study theoretical aspects of our logic.*

Observation records. To define which histories the agent cannot distinguish, we need to keep track of how she observed the system at each point in time. To do so, we record each observation change as a pair (o, n), where o is the new observation and n is the time when this change occurs.

Definition 2.3. An *observation record* r is a finite word over $O \times \mathbb{N}$, i.e., $r \in (O \times \mathbb{N})^*$.

Observation records, which represent changes of observational ability, do not contain the initial observation (which is given in the model). We write $\emptyset$ for the empty observation record.

Example 2.4. Consider a model M with initial observation o^ι, a history $h = s_0 \dots s_4$ and an observation record $r = (o_1, 0) \cdot (o_2, 3) \cdot (o_3, 3)$. The agent first observes state s_0 with observation o^ι. The observation record shows that at time 0, thus before the first transition, the agent changed for observation o_1. She then observed state s_0 again, but this time with observation o_1. Then the system goes through states s_1 and s_2 and reaches s_3, all of which she observes with observation o_1. At time 3, the agent changes to observation o_2, and thus observes state s_3 again, but this time with observation o_2, and finally she switches to observation o_3 and thus observes s_3 once more, with observation o_3. Finally, the system goes to state s_4, which the agent observes with observation o_3.

We write $r \cdot (o, n)$ for the observation record obtained by appending (o, n) to the observation record r, and $r[n]$ for the record consisting of all pairs (o, m) in r such that $m = n$. We say that an observation record r *stops at* n if $r[m]$ is empty for all $m > n$, and r *stops at history* h if it stops at $|h| - 1$. Unless otherwise specified, when we consider an observation record r together with a history h, it is understood that r stops at h.

[1] i.e., for every $s \in S$ there exists $s' \in S$ such that sTs'. This cosmetic restriction is made to avoid having to deal with finite runs ending in deadlocks.

Observations at time n. We let $ol(r, n)$ be the list of observations used by the agent at time n. It consists of the observation that the agent has when the n-th transition is taken, plus those of observation changes that occur before the next transition. It is defined by induction on n:

$$ol(r,0) = o^\iota \cdot o_1 \cdot \ldots \cdot o_k, \quad \text{if } r[0] = (o_1,0)\cdot \ldots \cdot (o_k,0), \text{ and}$$

$$ol(r,n+1) = \text{last}(ol(r,n)) \cdot o_1 \cdot \ldots \cdot o_k, \quad \text{if } r[n+1] = (o_1,n+1)\cdot \ldots \cdot (o_k,n+1).$$

Observe that $ol(r, n)$ is never empty: if no observation change occurs at time n, $ol(r, n)$ only contains the last observation taken by the agent. If r is empty, the latter is the initial observation o_ι.

Example 2.5. If $r = (o_1, 0) \cdot (o_2, 3) \cdot (o_3, 3)$, then $ol(r, 0) = o^\iota \cdot o_1$, $ol(r, 1) = ol(r, 2) = o_1$, $ol(r, 3) = o_1 \cdot o_2 \cdot o_3$, and $ol(r, 4) = o_3$.

Synchronous perfect recall. The usual definition of synchronous perfect recall states that for an agent with observation o, histories h and h' are indistinguishable if they have the same length and are point-wise indistinguishable, i.e., $|h| = |h'|$ and for each $i < |h|$, $h_i \sim_o h'_i$. We adapt this definition to changing observations: two histories are indistinguishable if, at each point in time, the states are indistinguishable for all observations used at that time.

Definition 2.6 (Dynamic synchronous perfect recall). Given an observation record r, two histories h and h' are equivalent, written $h \approx^r h'$, if $|h| = |h'|$ *and* $\forall i < |h|$, $\forall o \in ol(r, i)$, $h_i \sim_o h'_i$.

We now define the natural semantics of CTL*KΔ.

Definition 2.7 (Natural semantics). Fix a model M. A history formula φ is evaluated in a history h and an observation record r. A path formula ψ is interpreted on a run π, a point in time $n \in \mathbb{N}$ and an observation record. The semantics is defined by induction on formulas (we omit the obvious boolean cases):

$$\begin{array}{lll}
h, r \models p & \text{if} & p \in V(last(h)) \\
h, r \models A\psi & \text{if} & \forall \pi \text{ s.t. } h \preccurlyeq \pi,\ \pi, |h|-1, r \models \psi \\
h, r \models K\varphi & \text{if} & \forall h' \text{ s.t. } h' \approx^r h,\ h', r \models \varphi \\
h, r \models \Delta^o\varphi & \text{if} & h, r\cdot(o, |h|-1) \models \varphi \\
\pi, n, r \models \varphi & \text{if} & \pi_{\leq n}, r \models \varphi \\
\pi, n, r \models X\psi & \text{if} & \pi, (n+1), r \models \psi \\
\pi, n, r \models \psi_1 U \psi_2 & \text{if} & \exists m \geq n \text{ s.t. } \pi, m, r \models \psi_2 \text{ and} \\
 & & \forall k \text{ s.t. } n \leq k < m,\ \pi, k, r \models \psi_1
\end{array}$$

We say that a model M with initial state s^ι *satisfies* a CTL*KΔ formula φ, written $M \models \varphi$, if $s^\iota, \emptyset \models \varphi$.

We first discuss a subtlety of our semantics, which is that an agent can observe the same state consecutively with several observations.

Remark 2. *Consider the formula $\Delta^{o'}\varphi$ and history h. By definition, $h, r \models \Delta^{o'}\varphi$ iff $h, r \cdot (o', |h| - 1) \models \varphi$. Although the history does not change (it is still h), the observation record is extended by the observation o' at time $|h|-1$, with the following consequence. Suppose that $ol(r, |h|-1) = o$. After switching to o', the agent considers possible all histories h' such that i) $h \approx^r h'$ (they were considered possible before the change of observation) and ii) $last(h) \sim_{o'} last(h')$ (they are still considered possible after the change of observation). This means that by changing observation from o to o', the agent's information is refined by o', and it is as though the agent at time $|h| - 1$ observed the system with observation $o \cap o'$. At later times, her observation is simply o', until another change of observation occurs.*

2.4 Examples of observation change

We now illustrate that observation change is natural and relevant.

Example 2.8. A logic of accumulative knowledge (and resource bounds) is introduced in [22]. It studies agents that can perform successive *observations* to improve their knowledge of the situation, each observation refining their current view of the world. In their framework, an observation models a yes/no question about the current situation; if the answer is 'yes', the agent can eliminate all possible worlds for which the answer is 'no', and vice versa. Formally, an observation is a binary partition of the possible states, and the agent learns in which partition is the current state. Such observations are particular cases of our models' indistinguishability relations, and the semantics of an agent performing an observation o is exactly captured by the semantics of our operator Δ^o. Similarly, performing sequence of observations $o_1 \ldots o_n$ corresponds to the successive application of operators $\Delta^{o_1} \ldots \Delta^{o_n}$. As an example, [22] shows how to model a medical diagnosis in which the disease is narrowed down by performing a series of successive tests.

Our logic is incomparable with the one discussed in the previous example: in the latter observations have a cost, but no temporal aspect is considered, while in this work we do not consider costs, but we study the evolution of knowledge through time in addition to dynamic observation change. We now illustrate how both interact.

Example 2.9 (Security scenario). Consider a system with two possible levels of security clearance, modelled by observations o_1 and o_2, which define what information users have access to. In this scenario, we want to hide a secret p from the users. A desirable property is thus expressed by the formula $(\Delta^{o_1} AG\neg Kp) \wedge (\Delta^{o_2} AG\neg Kp)$, which means that a user using either o_1 or o_2 will never know that p holds. Model M from Figure 1 satisfies this formula.

Now consider formula $\varphi = \Delta^{o_1} EF \Delta^{o_2} Kp$, which means that if the user starts with observation o_1, there exists a path and a moment when changing observation lets her discover the secret. We show that M satisfies φ and thus that users should not be allowed to change security level. Consider history $h = s_0 s_2 s_5$ in M with initial observation o_1. At time 0 the user knows that the current state is s_0. After going to s_2, she does not know if the current state is s_2 or s_1, as they are indistinguishable by o_1. At time 2, at first the user does not know whether the system is in s_4 or s_5. Now, if she changes to observation o_2, she sees that the system is either in state s_5 or s_6. Refining her previous knowledge that the system is either in state s_4 or s_5, she deduces that the current state is s_5, and that p holds.

Example 2.10 (Fault-Tolerant Diagnosability). Diagnosability is a property of systems which states that every failure is eventually detected [37]. In the setting considered in [9], the system is monitored through a set of sensors, and a *diagnosability condition* is a pair (c_1, c_2) of disjoint sets of states that the system should always be able to tell apart. The problem of finding minimal sets of sensors that ensure diagnosability is studied, that is, finding a minimal sensor configuration sc such that $\Delta^{o_{\text{sc}}} AG(Kc_1 \vee Kc_2)$ holds, where o_{sc} is the observation corresponding to sensor configuration sc.

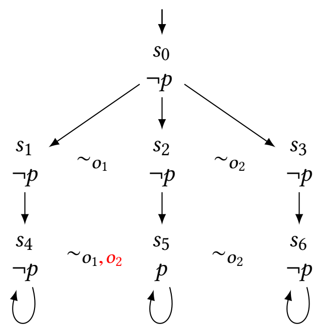

Figure 1: Model M in Example 2.9, and its variant M'

In CTL*KΔ one can express and model check a stronger notion of diagnosability that we call *fault-tolerant diagnosability*, where the system must remain diagnosable even after the loss of a sensor. For a given diagnosability condition (c_1, c_2) and sensor configuration sc, we write o the original observation (with every sensor in sc), o_i the observation where sensor i failed, and p_i is a proposition indicating the failure of sensor i. The following formula expresses that sensor configuration sc ensures fault-tolerant diagnosability:

$$\Phi_{\text{diag}} = \Delta^{o} AG((Kc_1 \lor Kc_2) \land (p_i \rightarrow \Delta^{o_i} AG(Kc_1 \lor Kc_2))).$$

Observe that it is possible for a system to satisfy Φ_{diag} but not $\Delta^{o_i} AG(Kc_1 \lor Kc_2)$ if sensor i, before failing, brings some piece of information that is crucial for diagnosis.

2.5 Model-checking problem

The model checking-problem for CTL*KΔ consists in, given a model M and a formula φ, deciding whether $M \models \varphi$.

Model-checking approach. Perfect-recall semantics refers to histories of unbounded length, but it is well known that in many situations it is possible to maintain a bounded amount of information that is sufficient to deal with perfect recall. We show that it is also the case for our logic, by generalising the classic approach. Intuitively, it is enough to know the current state, the current observational power and the set of states that the agent believes the system might be in. The latter is usually called *information set* in epistemic temporal logics and games with imperfect information. We define an alternative semantics based on information sets instead of histories and records, and we prove that this semantics is equivalent to the natural one presented in this section. Because information sets are of bounded size, it is then easy to build from this alternative semantics a model-checking algorithm for CTL*KΔ.

3 ALTERNATIVE SEMANTICS

We define an alternative semantics for CTL*KΔ. It is based on information sets, a classic notion in games with imperfect information [43], whose definition we now adapt to our setting.

Definition 3.1. Given a model M, the *information set* $I(h, r)$ after a history h and an observation record r is defined as follows:

$$I(h, r) = \{s \in S \mid \exists h', h' \approx^r h \text{ and } \text{last}(h') = s\}.$$

This information is sufficient to evaluate epistemic formulas for one agent. We now describe how to maintain this information along the evaluation of a formula. To do so, we define two update functions for information sets: one reflects changes of observational power, and the other captures transitions taken in the system.

Definition 3.2. Fix a model $M = (\text{AP}, S, T, V, \{\sim_o\}_{o \in O}, s^\iota, o^\iota)$. Functions U_T and U_Δ are defined as follows, for all $I \subseteq S$, all $s, s' \in S$ and $o, o' \in O$.

$$U_T(I, s', o) = T(I) \cap [s']_o$$
$$U_\Delta(I, s, o') = I \cap [s]_{o'}$$

When the agent has observational power o and information set I, and the model takes a transition to a state s', the new information set is $U_T(I, s', o)$, which consists of all successors of her previous information set I that are $\sim_o$-indistinguishable with the new state s'. When the agent is in state s with information set I, and she changes for observational power o', her new information set is $U_\Delta(I, s, o')$, i.e., all states that she considered possible before and that she still considers possible after switching to o'.

We let $O(h, r)$ be the last observation taken by the agent after history h, according to r. Formally, $O(h, r) = o_n$ if $ol(r, |h| - 1) = o_1 \cdot \ldots \cdot o_n$. The following result establishes that the functions U_Δ and U_T correctly update information sets. It is proved by simple application of the definitions.

Proposition 3.3. *For every history $h \cdot s$, observation record r that stops at h and observation o, it holds that*

$$\begin{aligned} I(h \cdot s, r) &= U_T(I(h, r), s, O(h, r)), \quad \textit{and} \\ I(h, r \cdot (o, |h| - 1)) &= U_\Delta(I(h, r), \textit{last}(h), o). \end{aligned}$$

We can now define our alternative semantics for CTL*KΔ.

Definition 3.4 (Alternative semantics). Fix a model M. A history formula φ is evaluated in a state s, an information set I and an observation o. A path formula ψ is interpreted on a run π, an information set I and an observation o. The semantic relation $\models_I$ is defined by induction on formulas (we omit the obvious boolean cases):

$$\begin{array}{lll}
s, I, o \models_I p & \text{if} & p \in V(s) \\
s, I, o \models_I A\psi & \text{if} & \forall \pi \text{ s.t. } \pi_0 = s,\ \pi, I, o \models_I \psi \\
s, I, o \models_I K\varphi & \text{if} & \forall s' \in I,\ s', I, o \models_I \varphi \\
s, I, o \models_I \Delta^{o'}\varphi & \text{if} & s, U_\Delta(I, s, o'), o' \models_I \varphi \\
\pi, I, o \models_I \varphi & \text{if} & \pi_0, I, o \models_I \varphi \\
\pi, I, o \models_I X\psi & \text{if} & \pi_{\geq 1}, U_T(I, \pi_1, o), o \models_I \psi \\
\pi, I, o \models_I \psi_1 U \psi_2 & \text{if} & \exists n \geq 0 \text{ such that} \\
 & & \pi_{\geq n},\ U_T^n(I, \pi, o), o \models_I \psi_2 \text{ and} \\
 & & \forall m \text{ such that } 0 \leq m < n, \\
 & & \pi_{\geq m}, U_T^m(I, \pi, o), o \models_I \psi_1,
\end{array}$$

where $U_T^n(I, \pi, o)$ is the iteration of the temporal update, defined inductively as follows:

- $U_T^0(I, \pi, o) = I$, and
- $U_T^{n+1}(I, \pi, o) = U_T(U_T^n(I, \pi, o), \pi_{n+1}, o)$.

Using Proposition 3.3, one can prove that the natural semantics $\models$ and the information semantics $\models_I$ are equivalent.

Theorem 3.5. *For every history formula φ, model M, history h and observation record r that stops at h,*

$$h, r \models \varphi \quad \textit{iff} \quad \textit{last}(h), I(h, r), o(h, r) \models_I \varphi.$$

4 MODEL CHECKING CTL*KΔ

In this section we devise a model-checking procedure based on the equivalence between the natural and alternative semantics (Theorem 3.5), and we prove the following result.

Theorem 4.1. *Model checking* CTL*KΔ *is in* Exptime.

Augmented model. Given a model M, we define an augmented model $\hat{M}$ in which the states are tuples (s, I, o) consisting of a state s of M, an information set I and an observation o. According to Theorem 3.5, history formulas can be viewed on this model as state formulas, and a model checking procedure can be devised by merely following the definition of the alternative semantics.

Let $M = (\mathrm{AP}, S, T, V, \{\sim_o\}_{o\in O}, s^\iota, o^\iota)$. We define the Kripke structure $\hat{M} = (S', T', V', s^{\iota\prime})$, where:

- $S' = S \times 2^S \times O$,
- $(s, I, o)\ T'\ (s', I', o)$ if $s\ T\ s'$ and $I' = U_T(I, s', o)$,
- $V'(s, I, o) = V(s)$, and
- $s^{\iota\prime} = (s^\iota, [s^\iota]_{o^\iota}, o^\iota)$.

We call $\hat{M}$ the *augmented model*, and we write $\hat{M}_o$ the Kripke structure obtained by restricting $\hat{M}$ to states of the form (s, I, o') where $o' = o$. Note that the different $\hat{M}_o$ are disjoint with regards to T'.

Model-checking procedure. We define function CheckCTL*KΔ which evaluates a history formula in a state of $\hat{M}$:

CheckCTL*KΔ $(\hat{M}, (s_c, I_c, o_c), \Phi)$ returns *true* if $M, s_c, I_c, o_c \models_I \varphi$ and *false* otherwise, and is defined as follows: if Φ is a CTL* formula, we evaluate it using a classic model-checking procedure for CTL*. Otherwise, Φ contains a subformula of the form $\varphi = K\varphi_1$ or $\varphi = \Delta^{o'}\varphi_1$ where $\varphi_1 \in$ CTL*. We evaluate φ_1 in every state of every component $\hat{M}_o$ (recall that the different $\hat{M}_o$ are disjoint), and mark those that satisfy φ_1 with a fresh atomic proposition p_{φ_1}. Then, if $\varphi = K\varphi_1$, we mark with a fresh atomic proposition p_φ every state (s, I, o) of $\hat{M}$ such that for every $s' \in I$, (s', I, o) is marked with p_{φ_1}. Else, $\varphi = \Delta^{o'}\varphi_1$ and we mark with a fresh proposition p_φ every state (s, I, o) such that $(s, U_\Delta(I, s, o'), o')$ is marked with p_{φ_1}. Finally, we recursively call function CheckCTL*KΔ on the marked model and formula Φ' obtained by replacing φ with p_φ in Φ.

To model check a formula φ in a model M, we build $\hat{M}$ and call CheckCTL*KΔ $(\hat{M}, (s_\iota, [s_\iota]_{o_\iota}, o_\iota), \varphi)$.

Algorithm correctness. The correctness of the algorithm follows from the following properties:

- For each formula $K\varphi_1$ chosen by the algorithm, $p_\varphi \in V'(s, I, o)$ *iff* $M, s, I, o \models_I K\varphi_1$
- For each formula $\Delta^{o'}\varphi_1$ chosen by the algorithm, $p_\varphi \in V'(s, I, o)$ *iff* $M, s, I, o \models_I \Delta^{o'}\varphi_1$

Complexity analysis. Let $|M|$ be the number of states in model M. Model checking a CTL* formula φ on a model M with state-set S can be done in time $2^{O(|\varphi|)}O(|S|)$ [16, 26]. Our procedure, for a CTL*KΔ formula φ and a model M, calls the CTL* model-checking procedure for at most $|\varphi|$ formulas of size at most $|\varphi|$, on each state of $\hat{M}$. The latter is of size $2^{O(|M|)} \times |O|$, but each call to the CTL* model-checking procedure is performed on a disjoint component $\hat{M}_o$ of size $2^{O(|M|)}$. Our overall procedure thus runs in time $|O| \times 2^{O(|\varphi|+|M|)}$.

5 MULTI-AGENT SETTING

We now extend CTL*KΔ to the multi-agent setting. We fix $Ag = \{a_1, \ldots, a_m\}$ a finite set of agents and define the logic CTL*KΔ$_m$. This logic contains, for each agent a and observation o, an operator Δ_a^o which reads as "agent a changes for observation o". We consider that these observation changes are public in the sense that all agents are aware of them. The reason is that if agent a changes observation without agent b knowing it, agent b may entertain false beliefs about what agent a knows. This would not be consistent with the S5 semantics of knowledge that we consider in this work, where false beliefs are ruled out by the Truth axiom $K\varphi \to \varphi$.

5.1 Syntax and natural semantics

We first extend the syntax, with knowledge operators K_a and observation change operators Δ_a^o for each agent.

Definition 5.1 (Syntax). The sets of history formulas φ and path formulas ψ are defined by the following grammar:

$$\begin{array}{rcl}\varphi & ::= & p \mid \neg\varphi \mid \varphi \wedge \varphi \mid A\psi \mid K_a\varphi \mid \Delta_a^o\varphi \\ \psi & ::= & \varphi \mid \neg\psi \mid \psi \wedge \psi \mid X\psi \mid \psi U \psi,\end{array}$$

where $p \in \mathcal{AP}$, $a \in Ag$ and $o \in O$.

Formulas of CTL*KΔ$_m$ are all history formulas.

The models of CTL*KΔ$_m$ are as for the one-agent case, except that we assign one initial observation to each agent. We write $\boldsymbol{o}$ for a tuple $\{o_a\}_{a\in Ag}$, $\boldsymbol{o}_a$ for o_a, and $\boldsymbol{o}[a \leftarrow o]$ for the tuple $\boldsymbol{o}$ where $\boldsymbol{o}_a$ is replaced by o. Finally, for $1 \le i \le m$, $\boldsymbol{o}_i$ refers to $\boldsymbol{o}_{a_i}$.

Definition 5.2 (Multiagent models). A *multiagent Kripke structure with observations* is a structure $M = (\mathrm{AP}, S, T, V, \{\sim_o\}_{o\in O}, s^\iota, \boldsymbol{o}^\iota)$, where all components are as in Definition 2.2, except for $\boldsymbol{o}^\iota \in O^{Ag}$, the initial observation for each agent.

We now adapt some definitions to the multi-agent setting.

Records tuples. We now need one observation record for each agent. We shall write $\boldsymbol{r}$ for a tuple $\{r_a\}_{a\in Ag}$. Given a tuple $\boldsymbol{r} = \{r_a\}_{a\in Ag}$ and $a \in Ag$ we write $\boldsymbol{r}_a$ for r_a, and for an observation o and time n we let $\boldsymbol{r} \cdot (o, n)_a$ be the record tuple $\boldsymbol{r}$ where $\boldsymbol{r}_a$ is replaced with $\boldsymbol{r}_a \cdot (o, n)$. Finally, for $i \in \{1, \ldots, m\}$, $\boldsymbol{r}_i$ refers to $\boldsymbol{r}_{a_i}$.

Observations at time n. We let $ol_a(\boldsymbol{r}, n)$ be the list of observations used by agent a at time n:

$$\begin{aligned} ol_a(\boldsymbol{r}, 0) &= \boldsymbol{o}_a^\iota \cdot o_1 \cdot \ldots \cdot o_k, \\ &\qquad \text{if } \boldsymbol{r}_a[0] = (o_1, 0) \cdot \ldots \cdot (o_k, 0), \text{ and} \\ ol_a(\boldsymbol{r}, n+1) &= \mathrm{last}(ol_a(\boldsymbol{r}, n)) \cdot o_1 \cdot \ldots \cdot o_k, \\ &\qquad \text{if } \boldsymbol{r}_a[n+1] = (o_1, n+1) \cdot \ldots \cdot (o_k, n+1). \end{aligned}$$

Definition 5.3 (Dynamic synchronous perfect recall). Given a record tuple $\boldsymbol{r}$, two histories h and h' are equivalent for agent a, written $h \approx_a^{\boldsymbol{r}} h'$, if $|h| = |h'|$ *and* $\forall i < |h|,\ \forall o \in ol_a(\boldsymbol{r}, i),\ h_i \sim_o h'_i$.

Definition 5.4 (Natural semantics). Let M be a model, h a history and $\boldsymbol{r}$ a record tuple. We define the semantics for the following inductive cases, the remaining ones are straightforwardly adapted from the one-agent case (Definition 2.7).

$$\begin{array}{lll} h, \boldsymbol{r} \models K_a\varphi & \text{if} & \forall h' \text{ s.t. } h' \approx_a^{\boldsymbol{r}} h,\ h', \boldsymbol{r} \models \varphi \\ h, \boldsymbol{r} \models \Delta_a^o\varphi & \text{if} & h, \boldsymbol{r} \cdot (o, |h| - 1)_a \models \varphi \end{array}$$

A model M with initial state s^ι *satisfies* a CTL*KΔ_m formula φ, written $M \models \varphi$, if $s^\iota, \boldsymbol{\emptyset} \models \varphi$, where $\boldsymbol{\emptyset}$ is the tuple where each agent has empty observation record.

5.2 Alternative semantics

As in the one-agent case, we define an alternative semantics that we prove equivalent to the natural one and upon which we build our model-checking algorithm. The main difference here is that we need richer structures than information sets to represent an epistemic situation of a system with multiple agents. For instance, to evaluate formula $K_a K_b K_c p$, we need to know what agent a knows about agent b's knowledge of agent c's knowledge of the system's state. To do so we use the k-trees introduced in [39, 40] in the setting of static observations, and which contain enough information to evaluate formulas of knowledge depth k.

k-trees. Fix a model $M = (\mathrm{AP}, S, T, V, \{\sim_o\}_{o\in O}, s^\iota, \boldsymbol{o}^\iota)$. Intuitively, a k-tree over M is a structure of the form $\langle s, \mathcal{I}_1, \dots, \mathcal{I}_m\rangle$, where $s \in S$ is the current state of the system, and for each $i \in \{1, \dots, m\}$, $\mathcal{I}_i$ is a set of $(k-1)$-trees that represents the state of knowledge (of depth $k-1$) of agent a_i. Formally, for every history h and record tuple $\boldsymbol{r}$ we define by induction on k the k-tree $I^k(h, \boldsymbol{r})$ as follows:

$$I^0(h, \boldsymbol{r}) = \langle \mathrm{last}(h), \emptyset, \dots, \emptyset\rangle$$

$$I^{k+1}(h, \boldsymbol{r}) = \langle \mathrm{last}(h), \mathcal{I}_1, \dots, \mathcal{I}_m\rangle,$$

where for each i, $\mathcal{I}_i = \{I^k(h', \boldsymbol{r}) \mid h' \approx^{\boldsymbol{r}}_{a_i} h\}$.

For a k-tree $I^k = \langle s, \mathcal{I}_1, \dots, \mathcal{I}_m\rangle$, we call s the *root* of I^k, and write it $\mathrm{root}(I^k)$. We also write $I^k(a)$ for $\mathcal{I}_i$, where $a = a_i$, and we let $\mathcal{T}^k$ be the set of k-trees for M. Observe that for one agent ($m = 1$), a 1-tree is an information set together with the current state.

Updating k-trees. We generalise our update functions U_Δ and U_T (Definition 3.2) to update k-trees. We first define, by induction on k, the function U^k_T that updates k-trees when a transition is taken.

$$U^0_T(\langle s, \emptyset, \dots, \emptyset\rangle, s', \boldsymbol{o}) = \langle s', \emptyset, \dots, \emptyset\rangle$$

$$U^{k+1}_T(\langle s, \mathcal{I}_1, \dots, \mathcal{I}_m\rangle, s', \boldsymbol{o}) = \langle s', \mathcal{I}'_1, \dots, \mathcal{I}'_m\rangle,$$

where for each i,

$$\mathcal{I}'_i = \{U^k_T(I^k, s'', \boldsymbol{o}) \mid I^k \in \mathcal{I}_i, s'' \sim_{\boldsymbol{o}_i} s' \text{ and } \mathrm{root}(I^k) T s''\}.$$

U^k_T takes the current k-tree $\langle s, \mathcal{I}_1, \dots, \mathcal{I}_m\rangle$, the new state s' and the current observation $\boldsymbol{o}$ for each agent, and returns the new k-tree after the transition.

We now define the second update function U^k_Δ, which is used when an agent a_i changes observation for some o'.

$$U^0_\Delta(\langle s, \emptyset, \dots, \emptyset\rangle, o, a_i) = \langle s, \emptyset, \dots, \emptyset\rangle$$

$$U^{k+1}_\Delta(\langle s, \mathcal{I}_1, \dots, \mathcal{I}_m\rangle, o, a_i) = \langle s, \mathcal{I}'_1, \dots, \mathcal{I}'_m\rangle,$$

where for each $j \neq i$,

$$\mathcal{I}'_j = \{U^k_\Delta(I^k, o', a_i) \mid I^k \in \mathcal{I}_j\}, \text{ and}$$

$$\mathcal{I}'_i = \{U^k_\Delta(I^k, o', a_i) \mid I^k \in \mathcal{I}_i \text{ and } \mathrm{root}(I^k) \sim_{o'} s\}.$$

Intuitively, when agent a_i changes observation for o', in every place of the k-tree that refers to agent a_i's knowledge, we remove possible states (and corresponding subtrees) that are no longer equivalent to the current possible state for a_i's new observation o'.

We let $\boldsymbol{O}(h, \boldsymbol{r})$ be the tuple of last observations taken by each agent after history h, according to $\boldsymbol{r}$. For each $a \in Ag$, $\boldsymbol{O}(h, \boldsymbol{r})_a = o_n$ if $ol_a(\boldsymbol{r}, |h|-1) = o_1 \cdot \ldots \cdot o_n$. The following proposition establishes that functions U^k_T and U^k_Δ correctly update k-trees.

Proposition 5.5. *For every history $h \cdot s$, record tuple $\boldsymbol{r}$ that stops at h, observation tuple $\boldsymbol{o}$ and integer k, it holds that*

$$\begin{aligned} I^k(h \cdot s, \boldsymbol{r}) &= U^k_T(I^k(h, \boldsymbol{r}), s, \boldsymbol{o}(h, \boldsymbol{r})), \text{ and} \\ I^k(h, \boldsymbol{r} \cdot (o, |h|-1)_a) &= U^k_\Delta(I^k(h, \boldsymbol{r}), o, a). \end{aligned}$$

We now define the alternative semantics for CTL*KΔ_m.

Definition 5.6 (Alternative semantics). The semantics of a history formula φ of knowledge depth k is defined inductively on a k-tree I^k and a tuple of current observations $\boldsymbol{o}$ (note that the current state is the root of the k-tree). We only give the following inductive cases, the others are simply adapted from Definition 3.4.

$$\begin{aligned}
I^k, \boldsymbol{o} &\models_I p && \text{if} \quad p \in V(\mathrm{root}(I^k)) \\
I^k, \boldsymbol{o} &\models_I A\psi && \text{if} \quad \forall \pi \text{ s.t. } \pi_0 = \mathrm{root}(I^k),\ \pi, I^k, \boldsymbol{o} \models_I \psi \\
I^k, \boldsymbol{o} &\models_I K_a\varphi && \text{if} \quad \forall I^{k-1} \in I^k(a),\ I^{k-1}, \boldsymbol{o} \models_I \varphi \\
I^k, \boldsymbol{o} &\models_I \Delta^{o'}_a\varphi && \text{if} \quad U^k_\Delta(I^k, o', a), \boldsymbol{o}[a \leftarrow o'] \models_I \varphi
\end{aligned}$$

The following theorem can be proved similarly to Theorem 3.5, using Proposition 5.5 instead of Proposition 3.3.

Theorem 5.7. *For every history formula φ of knowledge depth k, each model M, history h and tuple of records $\boldsymbol{r}$,*

$$h, \boldsymbol{r} \models \varphi \quad \textit{iff} \quad I^k(h, \boldsymbol{r}), \boldsymbol{o}(h, r) \models_I \varphi.$$

6 MODEL CHECKING CTL*KΔ_m

Like in the mono-agent case, it is rather easy to devise from this alternative semantics a model-checking algorithm for CTL*KΔ_m, the main difference being that the states of the augmented model are now k-trees. We prove the following result.

Theorem 6.1. *The model-checking problem for* CTL*KΔ_m *is in k-EXPTIME for formulas of knowledge depth at most k.*

Augmented model. Given a model M, we define an augmented model $\hat{M}$ in which the states are pairs $(I^k, \boldsymbol{o})$ consisting of a k-tree I^k and an observation for each agent, $\boldsymbol{o}$.

Let $M = (\mathrm{AP}, S, T, V, \{\sim_o\}_{o\in O}, s^\iota, \boldsymbol{o}^\iota)$. We define the Kripke structure $\hat{M} = (S', T', V', s^{\iota\prime})$, where:

- $S' = \mathcal{T}^k \times O^{Ag}$,
- $(I^k, \boldsymbol{o})\ T'\ (I^{k\prime}, \boldsymbol{o})$ if $s\ T\ s'$ and $I^{k\prime} = U^k_T(I^k, s', \boldsymbol{o})$, where $s = \mathrm{root}(I^k)$ and $s' = \mathrm{root}(I^{k\prime})$,
- $V'(I^k, \boldsymbol{o}) = V(\mathrm{root}(I^k))$, and
- $s^{\iota\prime} = (I^k(s^\iota, \boldsymbol{\emptyset}), \boldsymbol{o}^\iota)$.

We call $\hat{M}$ the *augmented model*, and we write $\hat{M}_{\boldsymbol{o}}$ the Kripke structure obtained by restricting $\hat{M}$ to states of the form $(I^k, \boldsymbol{o}')$ where $\boldsymbol{o}' = \boldsymbol{o}$. Again, the different $\hat{M}_{\boldsymbol{o}}$ are disjoint with regards to T'.

Model-checking procedure. We define function CheckCTL*KΔ_m which evaluates a history formula in a state of $\hat{M}$:

CheckCTL*KΔ_m $(\hat{M}, (I^k_c, \boldsymbol{o}_c), \Phi)$ returns *true* if $M, I^k_c, \boldsymbol{o}_c \models_I \varphi$ and *false* otherwise, and is defined as follows: if Φ is a CTL* formula, we evaluate it using a classic model-checking procedure for CTL*. Otherwise, Φ contains a subformula of the form $\varphi = K_a\varphi'$ or $\varphi =$

$\Delta_a^{o'}\varphi'$ where $\varphi' \in$ CTL*. We evaluate φ' in every state of $\hat{M}$, and mark those that satisfy φ' with a fresh atom $p_{\varphi'}$. Then, if $\varphi = K_a\varphi'$, we mark with a fresh atomic proposition p_φ every state $(I^k, \boldsymbol{o})$ of $\hat{M}$ such that for every $I^{k-1} \in I^k(a)$, $(I^{k-1}, \boldsymbol{o})$ is marked with $p_{\varphi'}$. Else, $\varphi = \Delta_a^{o'}\varphi'$ and we mark with a fresh proposition p_φ every state $(I^k, \boldsymbol{o})$ such that $(U_\Delta^k(I^k, o', a), \boldsymbol{o}[a \leftarrow o'])$ is marked with $p_{\varphi'}$. Finally, we recursively call CheckCTL*KΔ_m on the marked model and formula Φ' obtained by replacing φ with p_φ in Φ.

To model check a formula φ in a model M, we build $\hat{M}$ and call CheckCTL*KΔ_m $(\hat{M}, (I^k(s^\iota, \emptyset), \boldsymbol{o}^\iota), \varphi)$.

Algorithm correctness. The correctness of the algorithm follows from the following properties:

- For each formula $K_a\varphi$ chosen by the algorithm, $p_\varphi \in V'(I^k, \boldsymbol{o})$ *iff* $M, I^k, \boldsymbol{o} \models_I K_a\varphi$
- For each formula $\Delta_a^{o'}\varphi$ chosen by the algorithm, $p_\varphi \in V'(I^k, \boldsymbol{o})$ *iff* $M, I^k, \boldsymbol{o} \models_I \Delta_a^{o'}\varphi$

Complexity analysis. The number of different k-trees for m agents and a model with l states is no greater than $C_k = \exp(m \times l, k)/m$, where $\exp(a, b)$ is defined as $\exp(a, 0) = a$ and $\exp(a, b+1) = a2^{\exp(a,b)}$ [40]. The size of the augmented model $\hat{M}$ is thus bounded by $\exp(m \times l, k)/m \times |O|^{|Ag|}$, and it can be computed in time $\exp(O(m \times l), k) \times |O|^{|Ag|}$.

Model checking a CTL* formula φ on a model M with state-set S can be done in time $2^{O(|\varphi|)} \times O(|S|)$ [16, 26]. For a CTL*KΔ_m formula φ of knowledge depth at most k and a model M with l states, our procedure calls the CTL* model-checking procedure for at most $|\varphi|$ formulas of size at most $|\varphi|$, on each state of the augmented model $\hat{M}$ which has size $\exp(m \times l, k)/m \times |O|^m$. Each recursive call (for each subformula and state of $\hat{M}$) is performed on a disjoint component $\hat{M}_{\boldsymbol{o}}$ of size at most $\exp(m \times l, k)/m$, and thus takes time $2^{O(|\varphi|)} \times O(\exp(m \times l, k)/m)$, and there are at most $|\varphi| \times \exp(m \times l, k)/m \times |O|^m$ of them. Our overall procedure thus runs in time $|O|^m \times 2^{O(|\varphi|)} \times \exp(O(m \times l), k)$, which we rewrite as $|O|^{|Ag|} \times 2^{O(|\varphi|)} \times \exp(O(|Ag| \times |M|), k)$.

Note that, as described in [39, 40], the k-trees machinery can be refined to deal with formulas of *alternation depth k*. Theorem 4.1 would then become the instanciation of Theorem 6.1 for one agent and $k = 1$. We do not present this result here for reasons of space.

7 EXPRESSIVITY

In this section we prove that the observation-change operator adds expressive power to epistemic temporal logics. Formally, we compare the expressive power of CTL*KΔ_m with that of CTL*K$_m$ [12, 20], which is the syntactic fragment of CTL*KΔ_m obtained by removing the observation-change operator. Our semantics for CTL*KΔ_m generalises that of CTL*K$_m$, with which it coincides on CTL*K$_m$ formulas. Note that our multi-agent models (Definition 5.2) are more general than usual models for CTL*K$_m$, as they may contain observation relations that are not initially assigned to any agent, but such relations are mute in the evaluation of CTL*K$_m$ formulas.

For two logics $\mathcal{L}$ and $\mathcal{L}'$ over the same models, we say that $\mathcal{L}'$ is *at least as expressive as* $\mathcal{L}$, written $\mathcal{L} \preceq \mathcal{L}'$, if for every formula $\varphi \in \mathcal{L}$ there exists a formula $\varphi' \in \mathcal{L}'$ such that $\varphi \equiv \varphi'$. $\mathcal{L}'$ is *strictly more expressive than* $\mathcal{L}$, written $\mathcal{L} \prec \mathcal{L}'$, if $\mathcal{L} \preceq \mathcal{L}'$ and $\mathcal{L}' \not\preceq \mathcal{L}$. Finally, $\mathcal{L}$ and $\mathcal{L}'$ are *equiexpressive*, written $\mathcal{L} \equiv \mathcal{L}'$, if $\mathcal{L} \preceq \mathcal{L}'$ and $\mathcal{L}' \preceq \mathcal{L}$. First, since CTL*K$\Delta_m$ extends CTL*K$_m$, we have that:

Proposition 7.1. *For all $m \geq 1$,* CTL*K$_m \preceq$ CTL*KΔ_m.

We now point out that when there is only one observation, i.e., $|O| = 1$, the observation-change operator has no effect, and thus CTL*KΔ_m is no more expressive than CTL*K$_m$.

Proposition 7.2. *For $|O| = 1$,* CTL*K$_m \equiv$ CTL*KΔ_m.

Proof. We show that for $|O| = 1$, CTL*K$\Delta_m \preceq$ CTL*K$_m$, which together with Proposition 7.1 provides the result. Observe that when $|O| = 1$, observation change has no effect, and in fact observation records can be omitted in the natural semantics. For every CTL*KΔ_m formula φ, define the CTL*K$_m$ formula φ' by removing all observation-change operators Δ_a^o from φ. Clearly, $\varphi \equiv \varphi'$. ∎

On the other hand, we show that as soon as we have at least two observations, the observation-change operator adds expressivity. We first consider the mono-agent case.

Proposition 7.3. *If $|O| > 1$ then* CTL*K$\Delta \not\preceq$ CTL*K.

Proof. Assume that O contains o_1 and o_2. Consider the model M from Example 2.9 (Figure 1), and define the model M' which is the same as M except that s_4 and s_5 are indistinguishable for both o_1 and o_2, while in M they are only indistinguishable for o_1. In both models, agent a is initially assigned observation o_1. To prove the proposition we exhibit a formula of CTL*KΔ that can distinguish between M and M', and justify that no formula of CTL*K can.

Consider formula $\varphi = EF\Delta^{o_2}K_a p$. As detailed in Example 2.9, we have that $M \models \varphi$. We now show that $M' \not\models \varphi$: The only history in which p holds, and thus where agent a may get to know it, is the path $s_0s_2s_5$. After observing this path with observation o_1, agent a considers that both s_4 and s_5 are possible. She still does after switching to observation o_2, as s_4 and s_5 are o_2-indistinguishable. As a result $M' \not\models \varphi$, and thus φ distinguishes M and M'.

Now to see that no formula of CTL*K can distinguish between these two models, it is enough to see that in both models the only agent a is assigned observation o_1, and thus on these models no operator of CTL*K can refer to observation o_2, which is the only difference between M and M'. ∎

This proof for the mono-agent case relies on the fact that CTL*KΔ can refer to observations that are not initially assigned to any agent, and thus cannot be referred to within CTL*K. This proof can be easily adapted to the multi-agent case, by considering the same models M and M' and assigning the same initial observation o_1 to all agents. We show that in fact, when we have at least two agents, CTL*KΔ_m is strictly more expressive than CTL*K$_m$ even when we assume that all observations are initially assigned to some agent.

Proposition 7.4. *If $|O| > 1$ and $m \geq 2$,* CTL*K$\Delta_m \not\preceq$ CTL*K$_m$ *even on models in which all observations are initially assigned.*

Proof. Assume that O contains o_1 and o_2. We consider two agents a and b; the proof can easily be generalised to more agents. Consider again the models M and M' used in the proof of Proposition 7.3. This time, in both models, agent a is initially assigned observation o_1 and agent b observation o_2. For the same reasons as before, formula $\varphi = EF\Delta^{o_2}K_a p$ distinguishes between M and M'.

Now to see that no formula of $\mathsf{CTL}^*\mathsf{K}_m$ can distinguish these two models, recall that the only difference between M and M' concerns observation o_2, and that agents a and b are bound to observations o_1 and o_2 respectively. Since in $\mathsf{CTL}^*\mathsf{K}_m$ agents cannot change observation, the modification of o_2 between M and M' can only affect the knowledge of agent b, by making her unable to distinguish s_4 and s_5. However this cannot happen. Indeed, these states can only be reached via histories $s_0s_1s_4$ and $s_0s_2s_5$ respectively; since s_1 and s_2 are not o_2-indistinguishable, and we consider perfect recall, $s_0s_1s_4$ and $s_0s_2s_5$ are not o_2-indistinguishable neither.

Formally, define the *perfect-recall unfolding* of a model M as the infinite tree consisting of all possible histories starting in the initial state, in which two nodes h and h' are related for o_i if $|h| = |h'|$ and for all $i < |h|$, $h_i \sim_{o_i} h'_i$. It is clear that $\mathsf{CTL}^*\mathsf{K}_m$ is invariant under perfect-recall unfolding. Now it suffices to notice that the perfect-recall unfoldings of M and M' are the same, and thus cannot be distinguished by any $\mathsf{CTL}^*\mathsf{K}_m$ formula. ∎

Remark 3. *Unlike $\mathsf{CTL}^*\mathsf{K}_m$, $\mathsf{CTL}^*\mathsf{K}\Delta_m$ is not invariant under perfect-recall unfolding. Indeed in these unfoldings observation relations on histories are defined for fixed observations, and thus cannot account for observation changes induced by operators Δ^o_a.*

Putting together Propositions 7.1, 7.3 and 7.4, we obtain:

Theorem 7.5. *If $|\mathcal{O}| > 1$ then $\mathsf{CTL}^*\mathsf{K}_m \prec \mathsf{CTL}^*\mathsf{K}\Delta_m$.*

8 ELIMINATING OBSERVATION CHANGE

In this section we show how to reduce the model-checking problem for $\mathsf{CTL}^*\mathsf{K}\Delta$ to that of $\mathsf{CTL}^*\mathsf{K}$.

Fix an instance (M, Φ) of the model-checking problem for $\mathsf{CTL}^*\mathsf{K}\Delta$, where $M = (\mathrm{AP}, S, T, V, \{\sim_o\}_{o\in\mathcal{O}}, s^\iota, o^\iota)$ is a (mono-agent) model and Φ is a $\mathsf{CTL}^*\mathsf{K}\Delta$ formula. We build an equivalent instance (M', Φ') of the model-checking problem for $\mathsf{CTL}^*\mathsf{K}$; in particular, M' contains a single observation relation, and Φ' does not use operator Δ^o.

We first define M'. For each observation symbol $o \in \mathcal{O}$ we create a copy M_o of the original model M. Moving to copy M_o will simulate switching to observation o. To make this possible, we need to introduce transitions between each state s_o of a copy M_o to state $s_{o'}$ of copy $M_{o'}$, for all $o \neq o'$.

Let $M' = (\mathrm{AP} \cup \{p_o \mid o \in \mathcal{O}\}, S', T', V', \sim', s^{\iota\prime})$, where

- for each $o \in \mathcal{O}$, p_o is a fresh atomic proposition,
- $S' = \bigcup_{o\in\mathcal{O}}\{s_o \mid s \in S\}$,
- $T' = \{(s_o, s'_o) \mid o \in \mathcal{O} \text{ and } (s, s') \in T\}$
  $\cup \{(s_o, s_{o'}) \mid s \in S, o, o' \in \mathcal{O} \text{ and } o \neq o'\}$
- $V'(s_o) = V(s) \cup \{p_o\}$, for all $s \in S$ and $o \in \mathcal{O}$,
- $\sim' = \bigcup_{o\in\mathcal{O}}\{(s_o, s'_o) \mid s \sim_o s'\}$, and
- $s^{\iota\prime} = {s^\iota}_{o^\iota}$.

We now define formula Φ'. The translation tr^o is parameterised with an observation $o \in \mathcal{O}$ and is defined by induction on Φ:

$$\mathrm{tr}^o(\Delta^{o'}\varphi) = \begin{cases} \mathrm{tr}^{o'}(\varphi) & \text{if } o = o' \\ AX(p_{o'} \to \mathrm{tr}^{o'}(\varphi)) & \text{otherwise} \end{cases}$$

$$\mathrm{tr}^o(A\psi) = A(Gp_o \to \mathrm{tr}^o(\psi))$$

All other cases simply distribute over operators. We finally let $\Phi' = \mathrm{tr}^{o^\iota}(\Phi)$. Using the alternative semantics, we see that:

Lemma 8.1. *$M \models \Phi$ if, and only if, $M' \models \Phi'$.*

Since we know how to model-check $\mathsf{CTL}^*\mathsf{K}$, this provides a model-checking procedure for $\mathsf{CTL}^*\mathsf{K}\Delta$. However this algorithm does not provide optimal complexity. Indeed, the model M' is of size $|M| \times |\mathcal{O}|$, and the best known model-checking algorithm for $\mathsf{CTL}^*\mathsf{K}$ runs in time exponential in the size of the model and the formula [11]. Going through this reduction thus yields a procedure that is exponential in the number of observations. Our direct model-checking procedure, which generalises techniques used for the classic case of static observations, provides instead a decision procedure which is only linear in the number of observations (Theorem 4.1).

This reduction can be easily generalised to multiple agents, by creating one copy $M_{\boldsymbol{o}}$ of the original model M for each possible assignment $\boldsymbol{o}$ of observations to agents. We get a model M' of size $|M| \times |\mathcal{O}|^{|Ag|}$, and since the best known model-checking procedure for $\mathsf{CTL}^*\mathsf{K}_m$ is k-exponential in the size of the model [11], this reduction provides a procedure which is k-exponential in the number of observations and $k+1$-exponential in the number of agents.

Again our direct approach does better, as it is only polynomial in the number of observations, exponential in the number of agents, and its combined complexity is k-exponential time (Theorem 6.1).

9 CONCLUSION

Previous works in epistemic temporal logics have treated agents' observation power as a static feature. However, in many scenarios, agents' observation power may change. In this work we introduced $\mathsf{CTL}^*\mathsf{K}\Delta$, a logic that can express such dynamic changes of observation power. We showed that it can express natural properties that are not expressible without this operator, and provided some examples of applications of our logic. We showed that model checking is decidable, and known techniques can be extended to deal with observation change with no additional cost in complexity.

We also showed how to reduce the model-checking problem for our logic to that of $\mathsf{CTL}^*\mathsf{K}$, removing the observation-change operator. This yields a model-checking procedure for $\mathsf{CTL}^*\mathsf{K}\Delta$, but that is not as efficient as the direct algorithm we provide.

As future work we would like to establish the precise complexity of model checking $\mathsf{CTL}^*\mathsf{K}\Delta$. We conjecture that it should be the same as for $\mathsf{CTL}^*\mathsf{K}$, i.e., that adding the possibility to reason about changes of observational power comes for free. However, the exact complexity of model checking classic epistemic temporal logics such as LTLK or $\mathsf{CTL}^*\mathsf{K}$ is a long-standing open problem. It would also be interesting to study the satisfiability problem of epistemic temporal logic with changes of observation power. Finally, developing axiomatization of our logic could provide more insights into how changes of observation power work.

ACKNOWLEDGMENTS

The authors acknowledge the support from the Italian GNCS 2018 project "Metodi formali per la verifica e la sintesi di sistemi discreti e ibridi".

REFERENCES

[1] Rajeev Alur, Pavol Černý, and Swarat Chaudhuri. 2007. Model checking on trees with path equivalences. In *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*. Springer, 664–678.

[2] Guillaume Aucher. 2004. A combined system for update logic and belief revision. In *Pacific Rim International Workshop on Multi-Agents*. Springer, 1–17.

[3] Guillaume Aucher. 2014. Supervisory control theory in epistemic temporal logic. In *AAMAS*. 333–340. http://dl.acm.org/citation.cfm?id=2615787

[4] Alexandru Baltag, Lawrence S Moss, and Sławomir Solecki. 2016. The logic of public announcements, common knowledge, and private suspicions. In *Readings in Formal Epistemology*. Springer, 773–812.

[5] Aurèle Barrière, Bastien Maubert, Aniello Murano, and Sasha Rubin. 2018. Changing Observations in Epistemic Temporal Logic. In *Sixteenth International Conference on Principles of Knowledge Representation and Reasoning*.

[6] Francesco Belardinelli, Alessio Lomuscio, Aniello Murano, and Sasha Rubin. 2017. Verification of Broadcasting Multi-Agent Systems against an Epistemic Strategy Logic. In *IJCAI*. ijcai.org, 91–97.

[7] Francesco Belardinelli, Alessio Lomuscio, Aniello Murano, and Sasha Rubin. 2017. Verification of Multi-agent Systems with Imperfect Information and Public Actions. In *AAMAS*. ACM, 1268–1276.

[8] Raphaël Berthon, Bastien Maubert, Aniello Murano, Sasha Rubin, and Moshe Y. Vardi. 2017. Strategy logic with imperfect information. In *LICS*. 1–12. https://doi.org/10.1109/LICS.2017.8005136

[9] Benjamin Bittner, Marco Bozzano, Alessandro Cimatti, and Xavier Olive. 2012. Symbolic Synthesis of Observability Requirements for Diagnosability.. In *AAAI*.

[10] Thomas Bolander, Martin Holm Jensen, and François Schwarzentruber. 2015. Complexity Results in Epistemic Planning.. In *IJCAI*. 2791–2797.

[11] Laura Bozzelli, Bastien Maubert, and Sophie Pinchinat. 2015. Uniform strategies, rational relations and jumping automata. *Inf. Comput.* 242 (2015), 80–107. https://doi.org/10.1016/j.ic.2015.03.012

[12] Laura Bozzelli, Bastien Maubert, and Sophie Pinchinat. 2015. Unifying Hyper and Epistemic Temporal Logics. In *FoSSaCS*. 167–182. https://doi.org/10.1007/978-3-662-46678-0_11

[13] Petr Cermák, Alessio Lomuscio, Fabio Mogavero, and Aniello Murano. 2018. Practical verification of multi-agent systems against SLK specifications. *Inf. Comput.* 261, Part (2018), 588–614.

[14] Tristan Charrier, Bastien Maubert, and François Schwarzentruber. 2016. On the Impact of Modal Depth in Epistemic Planning. In *IJCAI*. 1030–1036. http://www.ijcai.org/Abstract/16/150

[15] Cătălin Dima. 2009. Revisiting Satisfiability and Model-Checking for CTLK with Synchrony and Perfect Recall. In *CLIMA IX-2008*. 117–131. https://doi.org/10.1007/978-3-642-02734-5_8

[16] E Allen Emerson and Chin-Laung Lei. 1987. Modalities for model checking: Branching time logic strikes back. *Science of computer programming* 8, 3 (1987), 275–306.

[17] Ronald Fagin, Joseph Y Halpern, Yoram Moses, and Moshe Vardi. 2004. *Reasoning about knowledge*. MIT press.

[18] Peter Gammie and Ron Van Der Meyden. 2004. MCK: Model checking the logic of knowledge. In *International Conference on Computer Aided Verification*. Springer, 479–483.

[19] Joseph Y. Halpern and Kevin R. O'Neill. 2005. Anonymity and information hiding in multiagent systems. *Journal of Computer Security* 13, 3 (2005), 483–512. http://content.iospress.com/articles/journal-of-computer-security/jcs237

[20] Joseph Y. Halpern, Ron van der Meyden, and Moshe Y. Vardi. 2004. Complete Axiomatizations for Reasoning about Knowledge and Time. *SIAM J. Comput.* 33, 3 (2004), 674–703. https://doi.org/10.1137/S0097539797320906

[21] Joseph Y. Halpern and Moshe Y. Vardi. 1989. The complexity of reasoning about knowledge and time. 1. Lower bounds. *J. Comput. System Sci.* 38, 1 (1989), 195–237. https://doi.org/10.1145/12130.12161

[22] Wojciech Jamroga and Masoud Tabatabaei. 2018. Accumulative knowledge under bounded resources. *J. Log. Comput.* 28, 3 (2018), 581–604. https://doi.org/10.1093/logcom/exv003

[23] Wojciech Jamroga and Wiebe van der Hoek. 2004. Agents that Know How to Play. *Fundam. Inform.* 63, 2-3 (2004), 185–219.

[24] Jeremy Kong and Alessio Lomuscio. 2017. Symbolic Model Checking Multi-Agent Systems against CTL*K Specifications. In *AAMAS*. 114–122. http://dl.acm.org/citation.cfm?id=3091147

[25] Orna Kupferman and Moshe Y. Vardi. 2001. Synthesizing distributed systems. In *LICS'01*. 389–398.

[26] Orna Kupferman, Moshe Y Vardi, and Pierre Wolper. 2000. An automata-theoretic approach to branching-time model checking. *Journal of the ACM (JACM)* 47, 2 (2000), 312–360.

[27] Richard E. Ladner and John H. Reif. 1986. The Logic of Distributed Protocols. In *TARK*. 207–222.

[28] Sébastien Lê Cong, Sophie Pinchinat, and François Schwarzentruber. 2018. Small Undecidable Problems in Epistemic Planning. In *IJCAI'18*.

[29] Bastien Maubert and Aniello Murano. 2018. Reasoning about Knowledge and Strategies under Hierarchical Information. In *KR*. 530–540. https://aaai.org/ocs/index.php/KR/KR18/paper/view/17996

[30] Fabio Mogavero, Aniello Murano, Giuseppe Perelli, and Moshe Y. Vardi. 2014. Reasoning About Strategies: On the Model-Checking Problem. *ACM Trans. Comput. Log.* 15, 4 (2014), 34:1–34:47. https://doi.org/10.1145/2631917

[31] Fabio Mogavero, Aniello Murano, Giuseppe Perelli, and Moshe Y. Vardi. 2017. Reasoning about Strategies: on the Satisfiability Problem. *Logical Methods in Computer Science* 13, 1 (2017). https://doi.org/10.23638/LMCS-13(1:9)2017

[32] Eric Pacuit. 2007. Some comments on history based structures. *Journal of Applied Logic* 5, 4 (2007), 613–624.

[33] Wojciech Penczek and Alessio Lomuscio. 2003. Verifying epistemic properties of multi-agent systems via bounded model checking. *Fundamenta Informaticae* 55, 2 (2003), 167–185.

[34] Gary Peterson, John Reif, and Salman Azhar. 2002. Decision algorithms for multiplayer noncooperative games of incomplete information. *CAMWA* 43, 1 (2002), 179–206.

[35] Amir Pnueli and Roni Rosner. 1990. Distributed reactive systems are hard to synthesize. In *FOCS'90*. 746–757.

[36] Franco Raimondi. 2006. *Model checking multi-agent systems*. Ph.D. Dissertation. University of London.

[37] Meera Sampath, Raja Sengupta, Stéphane Lafortune, Kasim Sinnamohideen, and Demosthenis Teneketzis. 1995. Diagnosability of discrete-event systems. *IEEE Transactions on automatic control* 40, 9 (1995), 1555–1575.

[38] Wiebe van der Hoek and Michael Wooldridge. 2003. Cooperation, knowledge, and time: Alternating-time Temporal Epistemic Logic and its applications. *Studia Logica* 75, 1 (2003), 125–157. https://doi.org/10.1023/A:1026185103185

[39] Ron van der Meyden. 1998. Common Knowledge and Update in Finite Environments. *Inf. Comput.* 140, 2 (1998), 115–157. https://doi.org/10.1006/inco.1997.2679

[40] Ron van der Meyden and Nikolay V. Shilov. 1999. Model Checking Knowledge and Time in Systems with Perfect Recall (Extended Abstract). In *FSTTCS*. 432–445.

[41] Ron van der Meyden and Kaile Su. 2004. Symbolic Model Checking the Knowledge of the Dining Cryptographers. In *CSFW-17*. 280–291.

[42] Hans van Ditmarsch, Wiebe Van der Hoek, and Barteld Pieter Kooi. 2007. *Dynamic epistemic logic*. Vol. 337. Springer.

[43] John Von Neumann and Oskar Morgenstern. 2007. *Theory of games and economic behavior (commemorative edition)*. Princeton university press.

[44] Quan Yu, Ximing Wen, and Yongmei Liu. 2013. Multi-Agent Epistemic Explanatory Diagnosis via Reasoning about Actions. In *IJCAI'13*.