We study dynamic changes of agents’ observational power in logics of knowledge and time. We consider $\text{CTL}^*\text{K}$, the extension of $\text{CTL}^*$ 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 $\text{CTL}^*\text{K}$. We reduce the model-checking problem for our logic to that for $\text{CTL}^*\text{K}$, 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

**ABSTRACT**

We study dynamic changes of agents’ observational power in logics of knowledge and time. We consider $\text{CTL}^*\text{K}$, the extension of $\text{CTL}^*$ 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 $\text{CTL}^*\text{K}$. We reduce the model-checking problem for our logic to that for $\text{CTL}^*\text{K}$, 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.

**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, $\Delta\sigma$, that represents changes of observation power, and is read "the agent changes her observation power to $\sigma$". For instance, the formula $\Delta^ao\text{ AF} (\Delta^o1 (Kp \lor K\neg p))$ expresses that "For an agent with initial observation power $\sigma_1$, in all possible futures there exists a point where, if the agent updates her observation power to $\sigma_2$, 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 $\text{CTL}$, $\text{LTL}$ and $\text{CTL}^*$ 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 $\text{LTL}$, $\text{CTL}$ and $\text{CTL}^*$ [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 $\text{CTLK}$ [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 $\text{CTL}^*\text{K}$ (which subsumes $\text{CTLK}$ and $\text{LTL}$) with observation-change operators $\Delta\sigma$. 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 $\Delta^a\sigma$ for each agent $a$, and we extend our approach to solve its model-checking problem.

---

*An extended abstract of this work was published in [5].*
problem. Next, we study the expressivity of our logic, so 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*KA

In this section we define the logic CTL*KA, 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. Σω). The length of a finite word w = w0 . . . wn is |w| = n + 1,
and we let last(w) = wn. Given a finite (resp. infinite) word w and
0 ≤ i < |w| (resp. i ∈ N), we let wi be the letter at position i in w,
w(≤ i) is the prefix of w that ends at position i, and w(≥ i) is the suffix
that starts at position i. We write w ≪ w′ if w is a prefix of w′.

2.2 Syntax

We fix a countably infinite set of atomic propositions, 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:

φ ::= p | ¬φ | φ ∧ ψ | Aψ | Kφ | Δφ
ψ ::= φ | ψ | ψ ∧ ψ | Xψ | ψUψ,

where p ∈ AP and o ∈ O.

CTL*KA 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 Kφ reads as “the agent knows
that φ is true”. Our new observation change operator, Δφ, reads as
the “agent now observes the system with observational power φ”.

As usual, we define T = p ∨ ¬p, φ ∨ φ′ = ¬(φ ∧ ¬φ), φ → φ′ = ¬φ ∨ φ′, as well as the temporal operators finally (F) and always
(G): Fφ = T Uφ, and Gφ = ¬F¬φ.

2.3 Semantics

Models of CTL*KA are Kripke structures equipped with one relation
∽ on states for each observation o.

Definition 2.2 (Models). A Kripke structure with observations is a
structure M = (AP, S, T, V, (∼o)oceO, s′, o′), where

• AP ⊆ AP is a finite subset of atomic propositions,
• S is a set of states,
• T ⊆ S × S is a left-total1 transition relation,
• V : S → 2AP is a valuation function,
• ∼o ⊆ S × S is an equivalence relation, for each o ∈ O,
• s′ ⊆ S is an initial state, and
• o′ ∈ O is the initial observation.

A path is an infinite sequence of states π = s0s1 . . . such that for
all i ≥ 0, siTs(i+1), and a history h is a finite prefix of a path. For I ⊆ S,
we write T(I) = {s′ | ∃s ∈ I. sTs′} for the set of successors of
states in I. Finally, for o ∈ O and s ∈ S, we let [s]o = {s′ | s ∼o s′}
be the equivalence class of s for relation ∼o.

Remark 1. We model agents’ information via indistinguishability relations
∽, where s ∼ 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 indistin-
guishable 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 ob-
ervation 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 × N,
i.e., r ∈ (O × N)*.

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

Example 2.4. Consider a model M with initial observation o′, a
history h = s0 . . . s4 and an observation record r = (o1, 0) · (o2, 3) ·
(o3, 3). The agent first observes state s0 with observation o′. The
observation record shows that at time 0, thus before the first transition,
the agent changed for observation o1. She then observed state
s0 again, but this time with observation o1. Then the system goes
through states s1 and s2 and reaches s3, all of which she observes
with observation o1. At time 3, the agent changes to observation
o2, and thus observes state s3 again, but this time with observation
o2, and finally she switches to observation o3 and thus observes
s3 once more, with observation o3. Finally, the system goes to state
s4, which the agent observes with observation o3.

We write r · (o, n) for the observation record obtained by ap-
pending (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[n] 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.
Observations at time \( t \). We let \( o(r, n) \) 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 \( n \):

\[
o(r, 0) = o' \cdot o_1 \cdot \ldots \cdot o_k, \\
o(r, n + 1) = \text{last}(o(r, n)) \cdot o_1 \cdot \ldots \cdot o_k,
\]

if \( r(0) = (o_1, 0) \cdot \ldots \cdot (o_k, 0) \), and

Observe that \( o(r, n) \) is never empty: if no observation change occurs at time \( n \), \( o(r, n) \) only contains the last observation taken by the agent. If \( r \) is empty, the latter is the initial observation \( o_0 \).

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

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 \( h \sim_r h' \), if \( |h| = |h'| \) and \( \forall i < |h|, \forall o \in o(r, i), h_i \sim_o h'_i \). We now define the natural semantics of CTL*KA.

Definition 2.7 (Natural semantics). Fix a model \( M \). A history formula \( \phi \) is evaluated in a history \( h \) and an observation record \( r \). A path formula \( \psi \) is interpreted on a run \( \pi \), 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):

\[
h, r \models p \quad \text{if} \quad p \in V(\text{last}(h))
\]

\[
h, r \models A\psi \quad \text{if} \quad \forall \pi \text{ s.t. } h < \pi, \forall o_i, |h| - 1, r \models \psi
\]

\[
h, r \models K\psi \quad \text{if} \quad \forall h', s.t. h' \sim_r h, h', r \models \psi
\]

\[
h, r \models \Delta^\psi \quad \text{if} \quad \forall h, r \cdot (o, |h| - 1) \models \psi
\]

\[
\pi, n, r \models \phi \quad \text{if} \quad \forall \pi \leq n, r \models \phi
\]

\[
\pi, n, r \models X\psi \quad \text{if} \quad \forall \pi, (n + 1), r \models \psi
\]

\[
\pi, n, r \models \psi_1 U \psi_2 \quad \text{if} \quad \exists m \geq n \text{ s.t. } \forall \pi, m, r \models \psi_2 \text{ and } \forall k \text{ s.t. } n \leq k < m, \pi, k, r \models \psi_1
\]

We say that a model \( M \) with initial state \( s' \) satisfies a CTL*KA formula \( \phi \), written \( M \models \phi \), if \( s' \cdot \emptyset \models \phi \).

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 \( \Delta^\psi \) and history \( h \). By definition, \( h, r \models \Delta^\psi \) if \( h, r \cdot (o', |h| - 1) \models \psi \). 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 \( o(r, |h| - 1) = o \). After switching to \( o' \), the agent considers possible all histories \( h' \) such that \( i ) h \models o' h' \) (they were considered possible before the change of observation) and \( ii) \text{last} h' \models o' \text{last} h' \) (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 \( o' \) instead of \( o \). 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 \( \Delta^o \). 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 \( \rho \) from the users. A desirable property is thus expressed by the formula \( (\Delta^{o_1} AG\neg K\rho) \land (\Delta^{o_2} AG\neg K\rho) \), which means that a user using either \( o_1 \) or \( o_2 \) will never know that \( \rho \) holds. Model \( M \) from Figure 1 satisfies this formula.

Now consider formula \( \phi = \Delta^{o_1} EFA \Delta^{o_2} K\rho \), 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 \( \phi \) and thus that users should not be allowed to change security level. Consider history \( h = s_0s_3s_5 \) in \( M \) with initial observation \( o_1 \). At time \( o \) 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 \( s_4 \). 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 \( \rho \) 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 \( \Delta^{o_{sc}} AG((Kc_1 \lor Kc_2)) \) holds, where \( o_{sc} \) is the observation corresponding to sensor configuration \( sc \).
We define an alternative semantics for $\mathcal{M}$, a history
of information sets, a classic notion in games with imperfect information.

### 3 ALTERNATIVE SEMANTICS

We define an alternative semantics for CTL$^*\mathsf{K}\Lambda$. 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 $M = (\mathcal{AP}, S, T, V, (\cdot \circ)_o \in O \cdot s', o')$. Functions $U_T$ and $U_\Lambda$ are defined as follows, for all $I \subseteq S$, all $s', s \in S$ and $o, o' \in O$.

$$U_T(I, s', o) = T(I) \cap [s']_o$$

$$U_\Lambda(I, s, o') = I \cap [s]_{o'}.$$

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 $U_\Lambda(I, s, o')$, 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, $O(h, r) = o_h$ if $o_h(|h| - 1) = o_1 \cdots o_{|h|}$. The following result establishes that the functions $U_T$ and $U_\Lambda$ 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

$$I(h \cdot s, r) = U_T(I(h, r), s, O(h, r)), \quad \text{and}$$

$$I(h, r \cdot (o, |h| - 1)) = U_\Lambda(I(h, r), \text{last}(h), o).$$

We can now define our alternative semantics for CTL$^*\mathsf{K}\Lambda$.

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

$$s, I, o \models_{\pi} p \quad \text{if} \quad p \in V(s)$$

$$s, I, o \models_{\pi} A\psi \quad \text{if} \quad \forall \pi \text{ s.t. } \pi_0 = s, \pi, I, o \models_{\pi} \psi$$

$$s, I, o \models_{\pi} K\phi \quad \text{if} \quad \exists \psi' \in I, \pi', I, o \models_{\pi'} \phi$$

$$s, I, o \models_{\pi} \Delta^o\phi \quad \text{if} \quad s, U_\Lambda(I, s, o'), o' \models_{\pi} \phi$$

$$\pi, I, o \models_{\pi} X\psi \quad \text{if} \quad \pi_2, U_T(I, \pi, o), o \models_{\pi_2} \psi$$

$$\pi, I, o \models_{\pi} \psi_3 U\psi_2 \quad \text{if} \quad \exists n \geq 0 \text{ such that}$$

$$\pi_{2n}, U^m_T(I, \pi, o), o \models_{\pi_2} \psi_2 \quad \text{and} \quad \forall m \text{ such that } 0 \leq m < n,$$

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

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

Using Proposition 3.3, one can prove that the natural semantics $\models_{\pi}$ and the information semantics $\models_{\pi}$ are equivalent.

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

$$h, r \models \phi \quad \text{iff} \quad \text{last}(h), I(h, r), O(h, r) \models_{\pi} \phi.$$
4 MODEL CHECKING CTL*KA

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*KA is in EXPTIME.

Augmented model. Given a model $M$, we define an augmented model $\tilde{M}$ in which the states are tuples $(s, l, o)$ consisting of a state $s$ of $M$, an information set $l$ 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 $M = (AP, S, T, V, (\sim_o)_{o \in O}, s', o')$. We define the Kripke structure $\tilde{M} = (S', T', V', s')$, where:

- $S' = S \times 2^O \times O$,
- $(s, l, o) T' (s', l', o)$ if $s T s'$ and $l' = U_T (l, s', o)$,
- $V'(s, l, o) = V(s')$, and
- $s' = (s', [s']_I, o')$.

We call $\tilde{M}$ the augmented model, and we write $\tilde{M}_0$ the Kripke structure obtained by restricting $\tilde{M}$ to states of the form $(s, l, o')$ where $o' = o$. Note that the different $\tilde{M}_0$ are disjoint with regards to $T'$.

Model-checking procedure. We define function CHECKCTL*KA which evaluates a history formula in a state of $\tilde{M}$:

CHECKCTL*KA $(\tilde{M}, (s_c, l_c, o_c), \phi)$ returns true if $M, s_c, l_c, o_c \models_I \phi$ and false otherwise, and is defined as follows: if $\phi$ is a CTL* formula, we evaluate it using a classic model-checking procedure for CTL*. Otherwise, $\phi$ contains a subformula of the form $\psi = K\varphi_1$ or $\psi = \Delta o' \varphi_1$ where $\varphi_1 \in CTL*$. We evaluate $\varphi_1$ in every state of every component $\tilde{M}_0$ (recall that the different $\tilde{M}_0$ are disjoint), and mark those that satisfy $\varphi_1$ with a fresh atomic proposition $p_{\varphi_1}$. Then, if $\phi = K\varphi_1$, we mark with a fresh atomic proposition $p_{\phi}$ every state $(s, l, o)$ of $\tilde{M}$ such that for every $s' \in I$, $(s', l, o)$ is marked with $p_{\varphi_1}$. Else, $\phi = \Delta o' \varphi_1$ and we mark with a fresh proposition $p_{\phi}$ every state $(s, l, o)$ such that $(s, U_I (l, s, o'), o')$ is marked with $p_{\varphi_1}$. Finally, we recursively call function CHECKCTL*KA on the marked model and formula $\phi'$ obtained by replacing $\phi$ with $p_{\phi}$ in $\phi$.

To model check a formula $\phi$ in a model $M$, we build $\tilde{M}$ and call CHECKCTL*KA $(\tilde{M}, (s_c, [s']_I, o_c), \phi)$.

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

- For each formula $K\varphi_1$ chosen by the algorithm, $p_{\phi} = V'(s, l, o) \iff M, s, l, o \models_I K\varphi_1$.
- For each formula $\Delta o' \varphi_1$ chosen by the algorithm, $p_{\phi} = V'(s, l, o) \iff M, s, l, o \models_I \Delta o' \varphi_1$.

Complexity analysis. Let $|M|$ be the number of states in model $M$. Model checking a CTL* formula $\phi$ on a model $M$ with state-set $S$ can be done in time $2^{|O|||O||S|||}$ [16, 26]. Our procedure, for a CTL*KA formula $\phi$ and a model $M$, calls the CTL* model-checking procedure for at most $|\phi|$ formulas of size at most $|\phi|$, on each state of $\tilde{M}$. The latter is of size $2^{|O|||O|| |O||}$. But each call to the CTL* model-checking procedure is performed on a disjoint component $\tilde{M}_0$ of size $2^{|O|||O|| |O||}$. Our overall procedure thus runs in time $|O| \times 2^{|O|||O|| |O||}$.

5 MULTI-AGENT SETTING

We now extend CTL*KA to the multi-agent setting. We fix $Ag = \{a_1, \ldots, a_m\}$ a finite set of agents and define the logic CTL*KA$_m$. This logic contains, for each agent $a$ and observation $o$, an operator $\Delta^o_a$ 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 $K\varphi \rightarrow \varphi$.

5.1 Syntax and natural semantics

We first extend the syntax, with knowledge operators $K_a$ and observation change operators $\Delta^o_a$ for each agent.

Definition 5.1 (Syntax). The sets of history formulas $\varphi$ and path formulas $\psi$ are defined by the following grammar:

$\varphi ::= p \mid \varphi \land \varphi \mid A\psi \mid K\varphi \mid \Delta^o_a \varphi$

$\psi ::= \psi \mid \psi \land \varphi \mid X\psi \mid \psi U\psi$.

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

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

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

Definition 5.2 (Multiagent models). A multiagent Kripke structure with observations is a structure $M = (AP, S, T, V, (\sim_o)_{o \in O}, s', o')$, where all components are as in Definition 2.2, except for $s'$ and $o'$ in $O$ Ag, 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 $r$ for a tuple $\{r_a\}_{a \in Ag}$. Given a tuple $r = \{r_a\}_{a \in Ag}$ and $a \in Ag$, we write $r_a$ for $r$, and for an observation $o$ and time $n$ we let $r \cdot (o, n)_a$ be the record tuple $r$ where $r_a$ is replaced with $r_a \cdot (o, n)$. Finally, for $i \in \{1, \ldots, m\}$, $r_i$ refers to $r_{a_{i-1}}$.

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

- $o_{a}(r_0) = o_{a_{i}} \cdot o_{a_{i-1}} \cdots o_{a_{0}}$.
- $o_{a}(r_{n+1}) = o_{a_{n}} \cdot o_{a_{n-1}} \cdots o_{a_{0}}$.

Definition 5.3 (Dynamic synchronous perfect recall). Given a record tuple $r$, two histories $h$ and $h'$ are equivalent for agent $a$, written $h \equiv_a h'$, if $|h| = |h'|$ and $\forall i \in [h]$, $\forall o \in o_{a}(r, n)$, $h_i \sim_o h_i'$.

Definition 5.4 (Natural semantics). Let $M$ be a model, $h$ a history and $r$ 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):

$h, r \models K_a \varphi$ if $\forall h' s.t. h' \equiv_a h', r \models \varphi$.

$h, r \models \Delta^o_a \varphi$ if $h, r \cdot (o, |h|) \models \varphi$. 

975
A model $M$ with initial state $s^t$ satisfies a CTL$^k\Delta_m$ formula $\varphi$, written $M \models \varphi$, if $s^t, \emptyset \models \varphi$, where $\emptyset$ 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 $K_k\Delta_k K_k p$, 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 $M = \langle \mathcal{AP}, S, T, V, (\sim_\alpha)_{\alpha \in \mathcal{O}}, s^t, o^t \rangle$. Intuitively, a $k$-tree over $M$ is a structure of the form $(s, I_1, \ldots, I_m)$, where $s \in S$ is the current state of the system, and for each $i \in \{1, \ldots, m\}$, $I_i$ 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 $r$, we define by induction on $k$ the $k$-tree $I^k(h, r)$ as follows:

$$I^0(h, r) = (\text{last}(h), \emptyset, \ldots, \emptyset)$$

$$I^{k+1}(h, r) = (\text{last}(h), I_1, \ldots, I_m),$$

where for each $i$, $I_i = (I^k(h, r) | h^t \sim^k_n h)$.

For a $k$-tree $I^k = (s, I_1, \ldots, I_m)$, we call $s$ the root of $I^k$, and write $\text{root}(I^k)$. We also write $I^k(a)$ for $I_i$, where $a = a_i$, and let $T^k$ 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 $U^0$ and $U^1$ (Definition 3.2) to update $k$-trees. We first define, by induction on $k$, the function $U^k_{\Delta}$ that updates $k$-trees when a transition is taken.

$$U^k_{\Delta}(s, \emptyset, \ldots, o, s', \emptyset) = (s', \emptyset, \ldots, o),$$

$$U^{k+1}_{\Delta}(s, I_1, \ldots, I_m, s', \emptyset) = (s', I_1', \ldots, I_m'),$$

where for each $i$,

$$I_i' = \{U^k_{\Delta}(I^k, s', o) | I^k \in I_i, t \sim \alpha_i, s' \text{ and } \text{root}(I^k)T^s\}$$

$U^k_{\Delta}$ takes the current $k$-tree $(s, I_1, \ldots, I_m)$, the new state $s'$ and the current observation $o$ for each agent, and returns the new $k$-tree after the transition.

We now define the second update function $U^k_{\Delta}$, which is used when agent $a_i$ changes observation for some $o'$.

$$U^k_{\Delta}(s, \emptyset, \ldots, o, a_i, o) = (s, \emptyset, \ldots, o),$$

$$U^{k+1}_{\Delta}(s, I_1, \ldots, I_m, a_i, o) = (s, I_1', \ldots, I_m'),$$

where for each $i \\neq i$,

$$I_i' = \{U_{\Delta}^k(I^k, o', a_i) | I^k \in I_i \},$$

$$I_i' = \{U_{\Delta}^k(I^k, o', a_i) | I^k \in I_i \},$$

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 $O(h, r)$ be the tuple of last observations taken by each agent after history $h$, according to $r$. For each $a \in \mathcal{A}_g$, $O(h, r)(h, a) = o_a$ if $a_o(h, a, r)(h, a) = o_a \epsilon \{0, \ldots, n_a\}$. The following proposition establishes that functions $U^k_{\Delta}$ and $U^k_{\Delta}$ correctly update $k$-trees.

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

$$I^k(h \cdot s, r) = U^k_{\Delta}(I^k(h, r), s, o(h, r)), \text{ and}$$

$$I^k(h, r \cdot o, h, r) = U^k_{\Delta}(I^k(h, r), o, a).$$

We now define the alternative semantics for CTL$^k\Delta_m$.

**Definition 5.6 (Alternative semantics).** The semantics of a history formula $\varphi$ of knowledge depth $k$ is defined inductively on a $k$-tree $I^k$ and a tuple of current observations $o$ (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.

$$I^k, o \models I p \text{ iff } p \in V(\text{root}(I^k))$$

$$I^k, o \models I A \varphi \text{ iff } \forall s. t. s = \text{root}(I^k), \pi, I^k, o \models \varphi$$

$$I^k, o \models I K_\alpha \varphi \text{ iff } \forall t \in I^k(a), I^{k-1}, o \models \varphi$$

$$I^k, o \models I A^k o \varphi \text{ iff } U^k_{\Delta}(I^k, o', a), o(a \leftarrow o') \models \varphi$$

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 $\varphi$ of knowledge depth $k$, each model $M$, history $h$ and tuple of records $r$,

$$h, r \models \varphi \text{ iff } I^k(h, r), o(h, r) \models \varphi$$

6 MODEL CHECKING CTL$^k\Delta_m$

Like in the mono-agent case, it is rather easy to devise from this alternative semantics a model-checking algorithm for CTL$^k\Delta_m$, 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$^k\Delta_m$ 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 $(I^k, o)$ consisting of a $k$-tree $I^k$ and an observation for each agent, $o$.

Let $M = \langle \mathcal{AP}, S, T, V, (\sim_\alpha)_{\alpha \in \mathcal{O}}, s^t, o^t \rangle$. We define the Kripke structure $\hat{M} = (S', T', V', s')$, where:

- $S' = T^k \times O^g$
- $(I^k, o) T' (I^k, o)$ if $T' s'$ and $I^k' = U^k_{\Delta}(I^k, s', o)$, where $s = \text{root}(I^k)$ and $s' = \text{root}(I^k')$
- $V'(I^k, o) = V(\text{root}(I^k))$, and
- $s' = (I^k(s', o), o)$.

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

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

$\text{CheckCTL}^k\Delta_m(M, (I^k, o), \Phi) \text{ returns true if } M, I^k, o \models \varphi$ and false otherwise, and is defined as follows: if $\Phi$ is a $\text{CTL}^k$ formula, we evaluate it using a classic model-checking procedure for $\text{CTL}^k$. Otherwise, $\Phi$ contains a subformula of the form $\varphi = K_a o'$ or $\varphi = \text{true$.}
where $\Delta^\varphi_{\psi} \varphi'$ where $\varphi' \in \text{CTL}^k$. We evaluate $\varphi'$ in every state of $M$, and mark those that satisfy $\varphi'$ with a fresh atom $p_{\varphi'}$. Then, if $\varphi = K_{\Delta^\varphi_{\psi}}$, we mark with a fresh atomic proposition $p_{\varphi'}$ every state $(t^k, o)$ of $M$ such that for every $k-1 \in I^k(a)$, $(k-1, o)$ is marked with $p_{\varphi'}$. Else, $\varphi = \Delta^\varphi_{\psi} \varphi'$ and we mark with a fresh proposition $p_{\varphi'}$ every state $(t^k, o)$ such that $(U^k_{t^k}, o', a, a(a \leftarrow o'))$ is marked with $p_{\varphi'}$. Finally, we recursively call checkCTL*KA$^m$ on the marked model and formula $\Phi'$ obtained by replacing $\varphi$ with $p_{\varphi}$.

To model check a formula $\varphi$ in a model $M$, we build $M$ and call checkCTL*KA$^m$ $(\hat{M}, (t^k(s^i, \theta), o^i), \varphi)$.

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

- For each formula $K_{\Delta^\varphi_{\psi}} \varphi$ chosen by the algorithm, $p_{\varphi} \in V^*(t^k, o)$ iff $M, t^k, o \models \varphi$.
- For each formula $\Delta^\varphi_{\psi} \varphi$ chosen by the algorithm, $p_{\varphi} \in V^*(t^k, o)$ iff $M, t^k, o \models \Delta^\varphi_{\psi} \varphi$.

**Complexity analysis.** The number of different $k$-trees for $m$ agents and a model with $l$ states is no greater than $C_k = (m \times l, k)/m$, where $\exp(a, b)$ is defined as $\exp(a, 0) = a$ and $\exp(a, b+1) = a2^{\exp(a, b)}$ [4]. The size of the augmented model $M$ is thus bounded by $\exp(m \times l, k)/m \times |\phi|^{|A|}$, and it can be computed in time $\exp(|\phi|, k) \times |\phi|^{|A|}$.

Model checking a CTL* formula $\varphi$ on a model $M$ with state-set $S$ can be done in time $2^{\exp(|\phi|, k)} \times |O|(|S|)$ [16, 26]. For a CTL*KA$^m$ formula $\varphi$ of knowledge depth at most $k$ and a model $M$ with $l$ states, our procedure calls the CTL* model-checking procedure for at most $|\phi|$ formulas of size at most $|\phi|$, on each state of the augmented model $\hat{M}$, which has size $\exp(m \times l, k)/m \times |\phi|^m$. Each recursive call (for each subformula and state of $M$) performs on a disjoint component $M_0$ of size at most $\exp(m \times l, k)/m$, and thus takes time $2^{\exp(|\phi|, k)} \times |O|$ (for $m \times l, k)/m \times |\phi|^m$ of them. Our overall procedure thus runs in time $|\phi|^m \times 2^{|\phi|} \times \exp(|\phi|, k)$, which we rewrite as $|\phi|^{|A|} \times 2^{|\phi|} \times \exp(|\phi|, |A| \times |M|, k)$.

Note that, as described in [39, 40], the $k$-trees machinery can be refined to deal with formulas of alteration depth $k$. Theorem 4.1 would then become the instantiation 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*KA$\Delta_m$ with that of CTL*KA$^m$ [12, 20], which is the syntactic fragment of CTL*KA$\Delta_m$ obtained by removing the observation-change operator. Our semantics for CTL*KA$^m$ generalises that of CTL*KA$^m$, with which it coincides on CTL*KA$^m$ formulas. Note that our multi-agent models (Definition 5.2) are more general than usual models for CTL*KA$^m$, as they may contain observation relations that are not initially assigned to any agent, but such relations are mute in the evaluation of CTL*KA 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 $\mathcal{L} \leq \mathcal{L}'$, 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 $\mathcal{L} < \mathcal{L}'$, if $\mathcal{L} \leq \mathcal{L}'$ and $\mathcal{L}' \not\leq \mathcal{L}$.

Finally, $\mathcal{L}$ and $\mathcal{L}'$ are equiexpressive, written $\mathcal{L} \equiv \mathcal{L}'$, if $\mathcal{L} \leq \mathcal{L}'$ and $\mathcal{L}' \leq \mathcal{L}$. First, since CTL*KA$^m$ extends CTL*KA$^m$, we have that:

**Proposition 7.1.** For all $m \geq 1$, $\text{CTL}^k\text{KA}_m \leq \text{CTL}^k\text{KA}_m$.

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*KA$^m$ has no more expressive than CTL*KA$^m$.

**Proposition 7.2.** For $|O| = 1$, $\text{CTL}^k\text{KA}_m \equiv \text{CTL}^k\text{KA}_m$.

**Proof.** We show that for $|O| = 1$, CTL*KA$\Delta_m$ and CTL*KA$^m$, 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*KA$^m$ formula $\varphi$, define the CTL*KA$^m$ formula $\varphi'$ by removing all observation-change operators $\Delta^\varphi_a$ from $\varphi$. 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 $\text{CTL}^k\text{KA} \not\equiv \text{CTL}^k$.

**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*KA that can distinguish between $M$ and $M'$, and justify that no formula of CTL*KA can.

Consider formula $\varphi = EF\Delta^a_{o_2}K_{o_2}p$. 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 $\varphi$ distinguishes $M$ and $M'$.

Now to see that no formula of CTL*KA 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*KA 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*KA can refer to observations that are not initially assigned to any agent, and thus cannot be referred to within CTL*KA. 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*KA$^m$ is strictly more expressive than CTL*KA$^m$ even when we assume that all observations are initially assigned to some agent.

**Proposition 7.4.** If $|O| > 1$ and $m \geq 2$, $\text{CTL}^k\text{KA}_m \not\equiv \text{CTL}^k\text{KA}_m$ 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 $\varphi = EF\Delta^{o_2}K_{o_2}p$ distinguishes between $M$ and $M'$.
Now to see that no formula of $\mathsf{CTL}^\ast \mathsf{K}_m$ 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 $\mathsf{CTL}^\ast \mathsf{K}_m$ 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 $s_0 S_4 S_4$ and $s_0 S_5 S_5$ respectively; since $S_4$ and $S_5$ are not $o_2$-indistinguishable, and we consider perfect recall, $s_0 S_4 S_4$ and $s_0 S_5 S_5$ are not $o_2$-indistinguishable.

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 $o_1$ if $|h| = |h'|$ and for all $i < |h|$, $h_i \sim_{o_1} h'_i$. It is clear that $\mathsf{CTL}^\ast \mathsf{K}_m$ 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 $\mathsf{CTL}^\ast \mathsf{K}_m$ formula.

**Remark 3.** Unlike $\mathsf{CTL}^\ast \mathsf{K}_m$, $\mathsf{CTL}^\ast \mathsf{K}_A m$ 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 $\Delta o_a$.

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

**Theorem 7.5.** If $|O| > 1$ then $\mathsf{CTL}^\ast \mathsf{K}_m < \mathsf{CTL}^\ast \mathsf{K}_A m$.

### 8 ELIMINATING OBSERVATION CHANGE

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

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

We first define $M'$. For each observation symbol $o \in O$ we create a copy $M_o$ of the original model. 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 $M' = (AP \cup \{p_o \mid o \in O\}, S', T', V', \sim', s')$, where

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

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

$$\mathsf{tr}^o(\Delta o_a \Phi) = \begin{cases} \mathsf{tr}^o(\Phi) & \text{if } o = o' \\ \mathcal{A}X(p_o \rightarrow \mathsf{tr}^o(\Phi)) & \text{otherwise} \end{cases}$$

All other cases simply distribute over operators. We finally let $\Phi' = \mathsf{tr}^0(\Phi)$. 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 $\mathsf{CTL}^\ast \mathsf{K}$, this provides a model-checking procedure for $\mathsf{CTL}^\ast \mathsf{K} A$. However this algorithm does not provide optimal complexity. Indeed, the model $M'$ is of size $|M| \times |O|$, and the best known model-checking algorithm for $\mathsf{CTL}^\ast \mathsf{K}$ 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 $M_o$ of the original model $M$ for each possible assignment $o$ of observations to agents. We get a model $M' o = |M| \times |O|^{|\Delta o_a|}$, and since the best known model-checking procedure for $\mathsf{CTL}^\ast \mathsf{K}_m$ 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 $\mathsf{CTL}^\ast \mathsf{K} A$, 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 $\mathsf{CTL}^\ast \mathsf{K}$, removing the observation-change operator. This yields a model-checking procedure for $\mathsf{CTL}^\ast \mathsf{K} A$, 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 $\mathsf{CTL}^\ast \mathsf{K} A$. We conjecture that it should be the same as for $\mathsf{CTL}^\ast \mathsf{K}$, 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 $\mathsf{CTL}^\ast \mathsf{K}$ 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 axiomatisation 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


