Reasoning About Factual Games Using Information Updates

Prakash Panangaden[1], Caitlin Phillips[1], Doina Precup[1], and Mehrnoosh Sadrzadeh[2]

[1] Reasoning and Learning Lab, School of Computer Science, McGill University, Montreal, Canada
[2] Computing Laboratory, Oxford University, Oxford, UK
{prakash,cphill, dprecup}@cs.mcgill.ca & mehrs@comlab.ox.ac.uk

Abstract. Dynamic epistemic logic plays a key role in reasoning about multi-agent systems. Past approaches to dynamic epistemic logic have typically been focused on actions whose primary purpose is to communicate information from one agent to another. These actions are unable to alter the valuation of any proposition within the system. In fields such as security, it is easy to imagine situations in which this sort of action would be insufficient. We expand the algebraic framework presented in [16] to include both communication actions and dynamic actions that change the state of the system. Furthermore, we propose a new modality that captures both epistemic and propositional changes resulting from the agents' actions.

1 Introduction

As the applications for epistemic logic and dynamic epistemic logic grow more numerous and more diverse, we are faced with the challenge of developing logics rich enough to model these applications but also flexible enough that they are not limited to one particular application. Although it is unlikely that a one-size-fits-all logic will work for every application, logics that incorporate more algebraic structure make it easier to model a variety of situations without having to be too explicit in the description of the situation.

Epistemic Logic, the branch of logic dealing with knowledge and belief, was first introduced by Hintikka in [10]. Hintikka gave a semantics for epistemic logic that is a simple variation of Kripke's semantics (see [12]), which was then extended to model multi-agent systems by using accessibility relations for each agent (see [5, 8, 7]). However, neither of these logics dealt with situations in which the agents' knowledge changes over time.

Since then, many different approaches have been proposed to formalize the dynamics of knowledge in multi-agent systems. In some cases dynamic modalities (see [9]) are used in conjunction with model restriction (see [18, 6]) to alter the structural properties of the Kripke model, and thus the agents' knowledge. In particular, a lot of focus has been put on logics in which actions take the form of public announcements of propositions [18, 15]. While a lot can be done using logics of this type, they have one major drawback: actions are necessarily idempotent, meaning that announcing the same proposition twice will have the exact same effect as announcing it once. However, the repetition of a statement may actually convey information; an example of problem of this type is the *Muddy Children Puzzle* (see [8]).

In [3, 16], the authors view actions as resources. This helps overcome the issue of idempotent actions, thus opening the door to modelling more complex scenarios. In [16], the dynamics of knowledge is represented by a proposition set and action set with more algebraic structure than in previous logics.

In this paper, we expand the work from [16] and [3] by broadening the algebraic framework they introduced to model knowledge in multi-agent systems. In particular, previous work focused predominantly on communication actions (see [16, 2, 5]). Our goal is to also model dynamic actions, which may change the state of the system (and hence, the valuations of the propositions within the system), in addition to conveying information. Such actions are important in multi-agent systems in which planning has to occur.

Regarding related work, dynamic epistemic logic has been extended with *assignments* and *post-conditions*, e.g. see [11], to be able to also reason about learning after fact-changing actions. However, the reduction axiom introduced there for these assignments cannot derive the knowledge properties we are interested in. This may be because the actions of our protocols, which for instance change the location of the system, are different from their fact-changing actions, which for instance change the status of a child in the muddy children puzzle from dirty to clean via washing. Also, since we use backward actions, there might be connections to a dynamic epistemic logic with converse actions, e.g. see [1]. Although, again it seems that their logical system cannot help in deriving the properties we are interested in here. A further exploration of these connections constitutes future work.

The paper is organized as follows. In Sec. 2.1, we review Kripke structures and how they are used to model agents' knowledge. In Sec. 2.2 we describe in detail the algebraic structures used to model multi-agent systems in [16]. In Sec. 3 we define systems with dynamic actions and present an example showing why the model proposed in [16] does not work for such systems. In Sec. 4, we extend the model presented in Sec. 2.2 to accommodate such situations, and revisit the example. Finally in Sec. 5, we conclude and discuss possible extensions of this work.

For a more theoretical version of the approach of this paper, which develops full algebraic semantics for the new knowledge modality and backward actions discussed here, proves that the old setting embeds in the new one, and includes more examples of protocols, see [14].

2 Preliminaries

2.1 Kripke Structures

Kripke structures are a common way of formalizing and connecting epistemic concepts in multi-agent systems. They allow for the model to keep track of the underlying state of the system, while also modelling which states (or worlds) each agent deems to be possible.

Definition 1. *A **Kripke structure** M for a set of agents $\mathcal{A}$ over a set of propositions Φ is a tuple $M = (W, V, \{R_A\}_{A\in\mathcal{A}})$ [5] where W is a set of states (also called possible worlds), R_A defines a binary relation on W (referred to as the **accessibility relation**) for each $A \in \mathcal{A}$, and V : is a valuation mapping $\Phi \to \mathcal{P}(W)$.*

A proposition p is satisfied at a state (possible world) w in a Kripke structure M (written as $(M, w) \vDash p$) if and only if $w \in V(p)$. In addition, $(M, w) \vDash \top, \forall w$ and $(M, w) \nvDash \bot, \forall w$. More complex propositions (conjunctions, disjunctions, negation and implication) can be evaluated using propositional logic [5].

In Kripke structures, knowledge is modelled using the accessibility relation R_A defined over $(W \times W)$. For each agent, if two worlds are related by the agent's accessibility relation, it means the agent is unable to tell them apart. More specifically, if world w_1 is related to world w_2 by agent A's modality (denoted by $w_1 R_A w_2$), then when w_1 is the case, A thinks that w_2 is possible. In layman's terms, an agent "knows" a formula is true if the formula holds in all worlds it thinks might be possible. Formally, $(M, w) \vDash K_A\varphi$ iff $\forall w'$ s.t. $wR_Aw', (M, w') \vDash \varphi$. For any model M and agent A, we typically require that the knowledge modality to satisfy the following axioms [5]:

A0 $M \vDash \varphi$, then $M \vDash K_A\varphi$ (Knowledge generalization)
K $M \vDash (K_A\varphi \wedge K_A(\varphi \Rightarrow \psi)) \Rightarrow K_A\psi$ (Distribution)
T $M \vDash K_A\varphi \Rightarrow \varphi$ (Truth or knowledge axiom)
D $M \vDash K_A\varphi \Rightarrow \neg K_A\neg\varphi$ (Consistency axiom)
4 $M \vDash K_A\varphi \Rightarrow K_AK_A\varphi$ (Positive Introspection)
5 $M \vDash \neg K_A\varphi \Rightarrow K_A\neg K_A\varphi$ (Negative Introspection)

Depending on the applications, the axioms that the knowledge modality is required to satisfy might change. Remarkably, almost all of these axioms correspond to a single, specific property of the accessibility (R_A) relations. The properties of interest include reflexivity, symmetry, transitivity, Euclidity, and seriality. Table 1 presents this correspondence.

For many purposes, the correct choice is to require that R_A be an equivalence relation, meaning that R_A must be symmetric, transitive and reflexive (and consequently serial and Euclidean). In structures of this kind, two worlds are indistinguishable if and only if the agent has the same information about each world [5].

	Axiom	Name	Property
T	$M \models K_i\varphi \Rightarrow \varphi$	Truth or knowledge axiom	Reflexive
D	$M \models K_i\varphi \Rightarrow \neg K_i\neg\varphi$	Consistency axiom	Serial
4	$M \models K_i\varphi \Rightarrow K_iK_i\varphi$	Positive Introspection	Transitive
5	$M \models \neg K_i\varphi \Rightarrow K_i\neg K_i\varphi$	Negative Introspection	Euclidean

Table 1. Knowledge axioms and their corresponding properties

2.2 Intuitionistic Dynamic Epistemic Action Logic

Intuitionistic Dynamic Epistemic Action Logic (*IDEAL*) [16] is an algebraic approach to dynamic epistemic logic, in which states are described entirely by the propositions they satisfy. By eliminating the state space and using the algebra of the set of propositions, a much richer structure than just a set with a binary relation becomes available, and with this added structure come new and useful properties. In order for these properties to hold though, we must first equip the set of propositions with sufficient algebraic structure. In particular, we will construct a complete algebraic lattice.

At this point, it is useful to distinguish between the concepts of formulas and propositions. To do this we define two sets: P (whose elements are denoted by lower-case letters $(p, q, ..)$) is a finite set of atomic propositions, and Φ a set of formulas built from Boolean combinations $(\wedge, \vee, \neg, ...)$ of propositions in P. The only requirement placed on Φ is that it must contain all propositions in P and be closed under conjunction. Other entailment axioms may be added as needed.

We will build the lattice on the set Φ, because its nature suggests an obvious, non-trivial notion of order: entailment. Entailment is a preorder on Φ, such that for $\varphi, \psi \in \Phi$, $\varphi \sqsubseteq \psi$ if and only if φ entails ψ. It is clear that the properties of a preorder (reflexivity and transitivity) are satisfied by $\sqsubseteq$. We also include a least and greatest element in Φ ($\bot$ and $\top$ respectively) to ensure that all pairs of elements (and thus all sets) have a least upperbound and a greatest lowerbound.

Recall that $(\Phi, \sqsubseteq)$ is only a preorder and thus may not be anti-symmetric (a requirement if we are building a lattice). To resolve this issue, we use filters to construct a complete algebraic lattice of formulas $(M, \leq)$, which preserves the natural entailment relation on the propositions but is now equipped with the structure of a complete algebraic lattice. Now we describe how actions interact with this lattice, and eventually how it relates to knowledge.

Unlike Kripke structures (where the valuation V is fixed), the sorts of systems we wish to model are those in which the actions serve two purposes: first, they modify the underlying valuation functions of the system and second, they allow agents to communicate information to one another. We will address the communication aspect later, as it must be considered from the perspective of a particular agent.

The idea that actions can modify the valuation of propositions can be expressed in terms of operators on the lattice. We define the action set Q to be a monoid (it is labelled Q rather than A to avoid confusion with agent names), where ; is concatenation. In some cases it makes sense to equip this monoid with a partial-order structure, in which case it becomes a *quantale* [16]. However, in this paper we consider the action set Q to be a monoid without any underlying order. Together, the action monoid and the lattice of formulas are referred to as a system.

Definition 2. *A* **system** *is a pair* (M, Q) *where* M *is a lattice and* Q *is a monoid acting on* M *[16]. Each element* $q \in Q$ *defines a mapping* $q : M \to M$ *such that :*

1. $q(\bigvee_i m_i) = \bigvee_i (q(m_i))$, $\forall m_i \in M$ *(q preserves joins)*
2. $\epsilon(m) = m, \forall m \in M$ *(the unit of the monoid is the identity operator on M)*
3. $(q_1; q_2)(m) = q_1(q_2(m))$ *(the mapping is associative over the binary operation of the monoid).*

Since each action $q \in Q$ is an join-preserving endomorphism of a complete algebraic lattice (M), it must have a right adjoint. This adjoint, denoted by $[q] : M \to M$, is defined as: $[q]m = \bigvee\{m' \in M | q(m') \leq m\}$, i.e., the join of all formulas m' which, when acted on by q, result in m being true. In other words, m' is the *weakest precondition*, or *dynamic modality* [16]. Formally, the dynamic modality adjoint i as follows:

$$\frac{q(m') \leq m}{m' \leq [q]m} \tag{1}$$

If the action set is a quantale, it is possible to define other adjoints as well [16], but the dynamic modality is the one most frequently used in defining and reasoning about epistemic systems.

In Kripke structures (as defined in Sec. 2.1), the agents' uncertainty as to which propositions are true was reflected in the fact that agents found certain sets of states indistinguishable. In epistemic systems, we approach uncertainty in a different manner, using the complete algebraic lattice of formulas instead of accessibility relations. For example, if an agent A is unable to distinguish between the times when m and m' hold (for some $m, m' \in M$), we say that when m is true, it *appears* to the agent that $m \vee m'$ is true. This notion is formalized as an appearance map.

Definition 3. *An* **appearance map** *for an agent* A *is an endomorphism* $f_A : M \to M$ *with the following properties:*

1. f_A *is increasing:* $m \leq f_A(m), \forall m \in M$.
2. f_A *is monotone:* $m_1 \leq m_2 \Rightarrow f_A(m_1) \leq f_A(m_2)$.
3. f_A *is idempotent:* $f_A(f_A(m)) = f_A(m), \forall m \in M$.
4. f_A *is join-preserving:* $f_A(m_1 \vee m_2) = f_A(m_1) \vee f_A(m_2)$.

The appearance map f_A is defined with respect to a specific agent, A. Furthermore, properties 1-3 make f_A a closure operator. In the context of epistemic systems, these properties have very specific meanings. Property 1 says that f_A is obscuring information in some way, as $f_A(m)$ is *at most* as informative as

m (recall that M is ordered by entailment). Property 2 says that f_A is order-preserving: if m_1 was more informative than m_2 ($m_1 \leq m_2$), then when f_A is applied to both propositions, $f_A(m_1)$ will be more informative than $f_A(m_2)$. Property 3 says that no additional information can be gained (or lost) by applying f_A repeatedly. Finally, property 4, together with property 2 and the fact that f_A is an endomorphism of a complete lattice, tells us that f_A must have a right-adjoint. In the framework presented here, we define this adjoint to be knowledge and therefore denote it by K_A.

Definition 4. Knowledge *is the right-adjoint of the appearance map, and is defined as follows:*

$$\frac{f_A(m_1) \leq m_2}{m_1 \leq K_A m_2} \tag{2}$$

Additionally, the following holds: $m \leq K_A f_A(m)$

We claim that the (f_A, K_A) adjoint pair is the connection between how the world appears to an agent and what the agent knows. Indeed, one can interpret the first equation in Def. 4 as saying: If agent A's view of m_1 entails m_2, then whenever m_1 holds in reality, it follows that agent A knows m_2. Because the properties of f_A induce analogous properties of K_A, we can capture some of the properties of knowledge (as described in Section 2.1). For example, positive introspection ($K_A m \Rightarrow K_A K_A m$) follows from the fact that $\leq$ is reflexive ($K_A m \leq K_A m$) and f_A is the idempotent left adjoint to K_A [16]. In systems where the action set is a quantale, it is possible to define an appearance map on the actions as well [16]. For the purpose of this work though, we assume that all actions are visible to all agents (the appearance map on the action set is the identity mapping). We can now modify the definition of a system to include appearance maps, which are key to modelling epistemic situations.

Definition 5. *An* **epistemic system** *is a tuple* $(M, Q, \{f_A\}_{A \in \mathcal{A}})$ *where* (M, Q) *is a system as defined in Def, 2) and* $\{f_A\}_{A \in \mathcal{A}}$ *is a set of appearance maps for each agent* $A \in \mathcal{A}$ *[16].*

Embedded within this epistemic system are two modalities in the form of Galois adjoints: the dynamic modality $(q, [q])$, and the knowledge modality (f_A, K_A). It remains to be shown how these two concepts interact. More specifically, since actions are not necessarily increasing on M, we do not know for any arbitrary $q \in Q, m \in M$ how $q(m)$ relates to m. Because of this, it is not possible to derive how $f_A(m)$ relates to $f_A(q(m))$. In [16], the following update inequality was defined to specify this connection:

$$f_A(q(m)) \leq q(f_A(m)) \tag{3}$$

Intuitively this means that for an agent, observing the execution of an action q should be at least as informative as imagining the outcome of the action.

3 Games with Fact-Changing Actions

Automata games are a concept derived from automata theory, involving one or more agents and an automaton (a finite-state transition system). This system can be deterministic or not. In the deterministic case, the system is defined by a triple $(S, Act, \{\tau_q\}_{q \in Act})$ where S is the state set and Act is the action set. The dynamics of the system is defined by mappings $\tau_a : S \to S$, such that $\tau_q(s) = t$ if and only if taking action q from state s causes a transitions to state t. τ_q does not need to be total (some actions may be disabled in certain states).

These systems can be the setting of a variety of epistemic tasks. Even when the underlying structure of the system is common knowledge, there are still epistemic tasks that can be studied, starting with the localization task (finding the current state), which can be extended to learning in which state the agent started, or "steering" the system to reach or avoid certain states. All of these tasks are useful in multi-agent systems as well, highlighting the need to study these systems from an epistemic perspective.

Given a transition system $(S, Act, \{\tau_q\}_{q \in Act})$, we begin by defining the proposition set. In the case of localization, we take the atomic proposition set P to be $S \cup \{\top, \bot\}$. From here we build our module M using filters as described before (see Sec. 2.2). For the action set, we use again the structure inherited from the transition system and define Act to be the set of atomic actions. From Act we can define a monoid $Q = Act^*$, equipped with an identity element ϵ and an composition function $; : Q \to Q$ such that for $q_1, q_2 \in Q$, $q_1; q_2 = q_1 q_2$ (the concatenation of the two action sequences).

Next we establish the manner in which the action set Q acts on the module of propositions, M. At an atomic level, this is already defined by the transition function τ. To make τ_q a total function, we simply extend it such that if the action q is not enabled at a state s, then $\tau_q(s) = \bot$. Furthermore, note that from Def. 2, the way sequences of actions (i.e. non-atomic actions) act on propositions is defined entirely by the composition of the atomic actions in the sequence. Hence, to characterize the effects of the action monoid Q, it is sufficient to define the effects of atomic actions on the proposition set.

First, we define the identity element $\epsilon \in Q$, such that $\epsilon(m) = m$, $\forall m \in M$. Then, for any atomic action $q \in Act$, we have the following:

- $q(\top) = \bigvee_{s \in S} q(s)$ (Recall that the proposition set $P = S \cup \{\top, \bot\}$).
- $q(\bot) = \bot$.
- $q(s) = \tau_q(s)$ where s is an atomic proposition ($s \in S \subset P$).
- $q(m_1 \vee m_2) = q(m_1) \vee q(m_2)$ where $m_1, m_2 \in M$.
- $q(m_1 \wedge m_2) = q(m_1) \wedge q(m_2)$ where $m_1, m_2 \in M$.

Finally, we have to define appearance maps and knowledge. For the purpose of the following example, we assume that the underlying structure of the transition system is common knowledge to all agents. Thus, by observing the actions available to it at any given time, an agent will be able to rule out certain states. To formalize this concept, we define a function $en : S \to \mathcal{P}(Act)$ defined by

$$en(s) = \{q \in Act | \tau_q(s) \neq \bot\}. \tag{4}$$

$en(s)$ gives the set of actions which are enabled at state s. Now we can use the *enabled* function to define appearance maps for atomic (state) propositions:

$$f_A(s) = \bigvee\{t \in S | en(s) = en(t)\}. \tag{5}$$

That is, when the agent is actually in the state s (and thus proposition s holds), it appears to the agent as though it might be in any state in which the enabled actions match those enabled in s. We also have that $f_A(\bot) = \bot$ (if an illegal action occurs, everyone sees it) and $f_A(\top) = \top$ (this follows from the fact that appearance maps are increasing). Since appearance maps are join preserving as well, we can define how they act on disjunctions of state propositions entirely by specifying their behaviour on atomic propositions: $f_A(s \vee t) = f_A(s) \vee f_A(t)$ for any $s, t \in S$.

As was the case with the actions, each appearance map f_A has a right adjoint ($f_A \dashv K_A$) which models the agent's knowledge:

$$\frac{f_A(m) \leq m'}{m \leq K_A m'}. \tag{6}$$

3.1 Example

To illustrate the epistemic nature of automata games, and the way IDEAL works, we present a simple example, to which we will return throughout the rest of the paper. Consider the automaton in Fig. 1, in which there is only one agent trying to learn its location in the system. For simplicity, we omit the subscripts on the appearance maps and refer to the single agent's appearance map and knowledge operator as f and K respectively.

As depicted in Fig. 1, the environment is a simple four-state world ($S = \{s_1, s_2, s_3, s_4\}$) with two possible actions ($Act = \{a, b\}$). However, these two actions are not enabled in every state. In particular, states s_3 and s_4 are dead states, meaning they have no outgoing transition arrows. In other words, $en(s_3) = en(s_4) = \emptyset$. On the other hand, states s_1 and s_2 have both actions enabled, thus $en(s_1) = en(s_2) = \{a, b\}$. Note that although both actions are enabled in both states, they do not have identical outcomes. In particular, taking an a action from state s_1 leads to a dead state (s_4), whereas taking an a action from state s_2 leads to state s_1. There is no way to know the exact outcome of an action before it has been taken. This is where the appearance maps come in. Recall that we defined appearance maps for automata games as $f(s) = \bigvee\{t \in S | en(t) = en(s)\}$. Applying this definition to the example, we see that $f(s_1) = f(s_2) = s_1 \vee s_2$, and $f(s_3) = f(s_4) = s_3 \vee s_4$. Prior to taking an action, the agent can only base its knowledge on the actions available to it, and thus cannot distinguish between states s_1 and s_2 even though they behave differently under both actions.

Now let us investigate what happens *after* an a action is taken from state s_1. Recall that prior to taking the action, the agent could not know it was in state s_1 (because $f(s_1) = s_1 \vee s_2$ and applying the knowledge adjoint, $s_1 \leq K(s_1 \vee s_2) \nleq Ks_1$). To determine exactly where in the system it is, the agent must use the

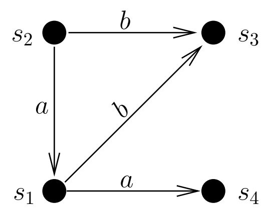

Fig. 1. The transition system for an automaton game

actions available to it. Since the structure of the system is common knowledge, it is easy to enumerate all possible results of an a action. In this case, if the agent believes it may be in state s_1 or s_2, it knows an a action will lead to state s_4 (if s_1 was the initial state) or state s_1 (if s_2 was the initial state). What we have just calculated is $a(f(s_1))$. However, it is only after the action has been executed that the agent will learn which of the two scenarios have occurred. If the agent truly started in state s_1, it ends up in a dead state after an a transition and ascertains (by means of the appearance map) that the current state must be s_3 or s_4 (this is the calculation for $f(a(s_1))$). To state formally what was just described:

- $a(f(s_1)) = a(s_1 \vee s_2) = a(s_1) \vee a(s_2) = \tau_a(s_1) \vee \tau_a(s_2) = s_4 \vee s_1$
- $f(a(s_1)) = f(s_3) = s_3 \vee s_4$

First and foremost, note that these elements of M are incomparable. That is, $s_4 \vee s_1 \nleq s_3 \vee s_4$, and likewise $s_3 \vee s_4 \nleq s_4 \vee s_1$. Hence the update inequality (Eqn. 3) is not applicable to this situation. Even more curious though, is the fact that neither of these propositions tells the whole story. Indeed, if a human were put in this position, he or she would be able to put these two pieces together and determine the exact location, ruling out s_3 since it cannot be reached on an a transition.

This sort of knowledge cannot be obtained by "imagining" the outcome of an action ($a(f(s_1))$), or by "forgetting" that the action occurred and looking only at the resulting proposition ($f(a(s_1))$). Instead, these two concepts need to be combined and actions must be remembered in some way. In order to model this situation effectively, appearance maps *must be allowed to change* as a result of the agents' actions.

4 Dynamic Appearance Maps

We would like to combine the rich algebraic structure of epistemic systems with the versatility of Kripke structures. The reason Kripke structures are able to model complex epistemic situations is that the equivalence relations upon which the agents' knowledge is based change over time. This means that even if the state of the system remains constant, the states that each agent believes possible (and hence the agent's knowledge) change as the agent observes the game.

Applying this idea to our framework means that even if the execution of an action q does not alter the valuation of a proposition m, we do not require the agent's appearance map of m to be fixed as well. In other words, the actions, in addition to acting on propositions, also alter appearance maps. This is analogous to the way equivalence relations in Kripke structures change over time but also allows us to preserve the algebraic structure of epistemic systems.

We will now introduce an extension of the epistemic system that captures the idea that agents' knowledge can change even when the underlying propositions do not. Let $\mathcal{F} : M \to M$ be the set of all possible appearance maps.

Definition 6. *An* **extended epistemic system** *for a set* $\mathcal{A}$ *of agents is a tuple* $(M, Q, \{f_A | A \in \mathcal{A}\}, \{\hat{q} | q \in Q\})$*, where* $(M, Q, \{f_A | A \in \mathcal{A}\})$ *is an epistemic system as defined previously and* $\hat{q}$ $(q \in Q)$ *defines a mapping* $\mathcal{F} \to \mathcal{F}$ *describing how an agent's appearance map changes after a* q *action is executed.*

We must now address the issue of knowledge. Because the definition of knowledge modality rests on that of the appearance map, a change in the appearance map will result in a change in knowledge. Previously it was sufficient to say that $m \le K_A m'$, meaning that whenever m held, agent A knew m' to be true. Clearly this is no longer the case as illustrated by the example. An agent's knowledge now depends not only on the propositions that hold, but also on the actions which have occurred, and this must somehow be reflected in the knowledge modality. In order to do this, we introduce a new adjoint relationship:

$$\frac{\hat{q}(f_A)(m) \le m'}{m \le K_A^q m'} \tag{7}$$

First, note that $\hat{q}(f_A) \in \mathcal{F}$ is itself an appearance map and thus has a right adjoint. This adjoint, K_A^q, looks very similar to the initial knowledge modality but depends on the actions that have occurred.The next step is to define exactly *how* these actions modify the agents appearance maps. To do this, we introduce a new concept, *backward actions*, and use it to define an update inequality analogous to Eqn. 3. Backwards actions are operators acting on the module of propositions. They serve the purpose of allowing the agent to reason retroactively. Every atomic action $q \in Act$ has a corresponding backwards action, denoted by $\backslash_q$. Non-atomic actions, that is, concatenations of two or more atomic actions, also have corresponding backwards actions, which are defined inductively:

- $\backslash_\epsilon = \epsilon$ (The backwards action corresponding to the empty action sequence is the empty action sequence.)
- For any action sequence $\alpha = \beta; q$ such that $\alpha, \beta \in Q, q \in Act$, $\backslash_\alpha = \backslash_q; \backslash_\beta$ (Note that the order is of the actions is reversed: the last action in the original sequence is the first in the backward action sequence).

Backwards actions distribute over meets and joins so $(m_1 \vee m_2)\backslash_q = m_1 \backslash_q \vee m_2\backslash_q$ and $(m_1 \wedge m_2)\backslash_q = m_1\backslash_q \wedge m_2\backslash_q, \forall m_1, m_2 \in M$. Also, $\bot\backslash_q = \bot$. It remains to describe how backwards actions act on elements of the atomic proposition set P. This is where the distinction between backwards actions and normal actions becomes apparent:

Definition 7. ***Backwards Actions:***

$$p\backslash_q = \begin{cases} p & \text{if } \exists m \in M, m \neq \bot \text{ s.t. } q(m) \leq p \\ \bot & \text{otherwise} \end{cases} \tag{8}$$

So for any proposition $p \in P$, $p\backslash_q \leq p$ if and only if p holds after taking a q action

Note that backwards actions are *decreasing*, that is, for all $m \in M$, $m\backslash_q \leq m$. This follows from the fact that backwards actions either leave atomic propositions unaltered or result in a contradiction ($\bot$). Using backwards actions, we define a new update inequality which defines how *normal* actions should modify appearance maps in extended epistemic systems.

Definition 8. ***Update Inequality:***

$$\hat{q}(f_A)(m) \leq f_A(q(m))\backslash_q \tag{9}$$

In order to explain this update inequality, we will look at each part separately. First, recall that the left-hand side of the equation is simply the revised appearance map $\hat{q}(f)(m)$. If initially m holds, then $\hat{q}(f)(m)$ specifies how the world appears to the agent after taking a q action. Clearly, $\hat{q}(f)(m) \leq f(q(m))$, that is, after taking a q action when m held initially, the agent should have at least as much information about its environment as it would if it ignored the action itself and simply looked at the resulting proposition, $q(m)$.

However, we can say something even stronger. Recall that backwards actions are decreasing and thus, $f(q(m))\backslash_q \leq f(q(m))$. $f(q(m))\backslash_q$ can be viewed as follows: take the appearance map of the resulting proposition $q(m)$ and apply the backwards action $\backslash_q$. This allows us to eliminate any disjuncts of the formula $f(q(m))$ that are not consistent with the fact that the last action taken was an q. In other words, we are remembering the action and its effects without having to keep track of entire action sequences.

4.1 Example

We now revisit our example from Section 2.2, Fig. 1. Recall that the goal of this example was to be able to prove statements of the form: $s_1 \leq [a]Ks_4$ (if the agent is in state s_1, after an a action, it will know that it is in state s_4). We will do this by using dynamic appearance maps and their resulting knowledge modalities, which allow us to explicitly incorporate the observation of an action. Hence it suffices to show that: $s_1 \leq K^a s_4$

From here we can apply the dynamic knowledge adjoint $(\hat{a}(f), K^a)$ and see that it is enough to prove $\hat{a}(f)(s_1) \leq s_4$. However, we do not know exactly how that $\hat{a}$ action affects the appearance map, as this is not explicitly defined in the formalism. However, we know that it must respect the revised update inequality: $\hat{a}(f)(s_1) \leq f(a(s_1))\backslash_a$.

Thus, it suffices to show that $f(a(s_1))\backslash_a \leq s_4$. To this end, we evaluate the left-hand side of the equation and find that $a(s_1) = \tau_a(s_1) = s_4$. Applying the appearance map f to this result, we get that $f(s_4) = \bigvee\{s \in S | en(s) = en(s_4)\} = s_3 \vee s_4$. since s_3 and s_4 are both dead states with no actions enabled. Then we apply the backwards a-action and find that $(s_3 \vee s_4)\backslash_a = \bot \vee s_4 \leq s_4$. We have now shown that $\hat{a}(f)(s_1) \leq s_4$ by way of the dynamic knowledge adjoint and the revised update in equality. It follows then, that $s_1 \leq K^a s_4$.

5 Conclusions and Future Work

The main contribution of this work is to broaden the algebraic framework developed in [16] and [3] to include actions which are both communicative and dynamic in nature. To understand the effect these actions on agents' knowledge, we introduced the notion of dynamic appearance maps along with a new modality that captures both the epistemic and non-epistemic dynamics of the system.

There are similarities between the backward actions defined in Section 4 and the converse action of dynamic epistemic logic (see [1]). Previously, it was shown in [16] and [3], how one can build an epistemic system from the relational update semantics (in terms of initial and action Kripke models) of dynamic epistemic logic [2]. It would be interesting to build on that and further explore how this construction could extend to backwards action and the newly defined knowledge modality (K^q_A) are in Kripke semantics. Exploring the coalgebraic properties of backwards actions, as the authors of [16, 13] did for the knowledge modality, would also be informative.

Another possible extension is the development of a proof system. In [16], in order to prove the soundness and completeness of *IDEAL*, the author develops a sequent calculus not unlike that of propositional dynamic logic (see [9]). The sequent calculus makes it possible to formalize the axioms of the logic as rules of inference including the epistemic update. In this way, it is possible to prove the soundness and completeness of the proof system with respect to the algebraic semantics.

The most exciting extensions to this work deal with the introduction of new modalities. The algebraic structure of our logic provides a framework for introducing new modalities without having to rethink the entire system. This is especially important when dealing with security protocols. While several logics [4, 17] have been developed for reasoning about authentication protocols, they tend to be very specialized and often a new protocol requires a new logic. It is hoped that the algebraic structure of our logic will provide a setting in which security protocols can be studied effectively with only slight alterations needed to accommodate each protocol. Lastly, the algebraic structure of our logic provides a good starting point when lifting binary concepts like knowledge to probabilistic concepts like belief, which allow reasoning about probabilistic multi-agent systems.

Acknowledgements

This research was supported in part by NSERC, the Office of Naval Research, a Dean's Excellence fellowship from McGill University, and EPSRC grant EP/F042728/1 from United Kingdom.

References

1. G. Aucher and A. Herzig. From del to edl : Exploring the power of converse events. In *LNCS*, pages 199–209, 2007.
2. Alexandru Baltag and Lawrence S. Moss. Logics for epistemic programs. *Synthese*, 139(2):165–224, 2004.
3. Ru Baltag, Bob Coecke, and Mehrnoosh Sadrzadeh. Epistemic actions as resources. In *Journal of Logic and Computation*, pages 555–585, 2007.
4. Michael Burrows, Martin Abadi, and Roger Needham. A logic of authentication. *ACM Trans. Comput. Syst.*, 8(1):18–36, 1990.
5. Ronald Fagin, Joseph Y. Halpern, Yoram Moses, and Moshe Y. Vardi. *Reasoning About Knowledge*. MIT Press, 1995.
6. Jelle Gerbrandy. Dynamic epistemic logic. In *Logic, Language, and Information 2, Stanford University, CSLI Publication*. CSLI Publications, 1997.
7. Jelle Gerbrandy. Bisimulations on planet kripke. In *University of Amsterdam*, 1999.
8. Joseph Y. Halpern and Moshe Y. Vardi. Model checking vs. theorem proving: A manifesto. In *Artificial intelligence and mathematical theory of computation: papers in honor of John McCarthy*, pages 325–334. Morgan Kaufmann, 1991.
9. David Harel, Dexter Kozen, and Jerzy Tiuryn. Dynamic logic. In *Handbook of Philosophical Logic*, pages 497–604. MIT Press, 1984.
10. Jaakko Hintikka. *Knowledge and belief : an introduction to the logic of the two notions / by Jaakko Hintikka*. Cornell University Press, Ithaca, N. Y. :, 1962.
11. B.P. Kooi H.P. van Ditmarsch, W. van der Hoek. Dynamic epistemic logic with assignment. In *Proceedings of AAMAS*, pages 141–148, 2005.
12. S. A. Kripke. Semantical analysis of modal logic I: Normal modal propositional calculi. *Zeitschrift fr Mathematische Logik und Grundlagen der Mathematik*, 9:67–96, 1963.
13. Lawrence S. Moss. Coalgebraic logic. *Annals of Pure and Applied Logic*, 96(1-3):277–317, 1999.
14. P. Panangaden and M. Sadrzadeh. Learning in a changing world, an algebraic approach. In *LNCS proceedings of Thirteenth International Conference on Algebraic Methodology And Software Technology (AMAST2010)*, 2010.
15. Jan Plaza. Logics of public communications. *Synthese*, 158(2):165–179, 2007.
16. Merhnoosh Sadrzadeh. *Actions And Resources In Epistemic Logic*. PhD thesis, Univerisity of Quebec a Montreal, 2005.

17. Paul Syverson and Iliano Cervesato. The logic of authentication protocols. In *Foundations of Security Analysis and Design, LNCS 2171*, pages 63–136. Springer-Verlag, 2001.
18. Hans van Ditmarsch, Wiebe van der Hoek, and Barteld Kooi. *Dynamic Epistemic Logic*. (Synthese Library). Springer, 1st edition, November 2007.