University of Groningen

Semantic results for ontic and epistemic change

van Ditmarsch, Hans; Kooi, Barteld

Published in:
Logic and the foundations of game and decision theory (LOFT 7)

IMPORTANT NOTE: You are advised to consult the publisher's version (publisher's PDF) if you wish to cite from it. Please check the document version below.

Document Version
Publisher's PDF, also known as Version of record

Publication date:
2008

[Link to publication in University of Groningen/UMCG research database](#)

Semantic Results for Ontic and Epistemic Change

Hans van Ditmarsch[1,2]
Barteld Kooi[3]

[1] Department of Computer Science
University of Otago
PO Box 56
Dunedin 9054, New Zealand

[2] Institut de Recherche en Informatique de Toulouse
Université Paul Sabatier
118 Route de Narbonne
31062 Toulouse Cedex 9, France

[3] Faculteit Wijsbegeerte
Rijksuniversiteit Groningen
Oude Boteringestraat 52
9712 GL Groningen, The Netherlands

hans@cs.otago.ac.nz, B.P.Kooi@rug.nl

Abstract

We present an epistemic logic incorporating dynamic operators to describe information changing events. Such events include epistemic changes, where agents become more informed about the non-changing state of the world, and ontic changes, wherein the world changes. The events are executed in information states that are modelled as pointed Kripke models. Our contribution consists of three semantic results. (i) Every consistent formula can be made true in every information state by the execution of an event. (ii) Every event corresponds to an event with assignments to true and false only. (iii) Every event corresponds to a sequence of events with assignments of a single atom only. We apply the logic to model dynamics in a multi-agent setting involving card players.

1 Introduction

In dynamic epistemic logics [32, 23, 9, 4, 17] one does not merely describe the static (knowledge and) beliefs of agents but also dynamic features: how does belief change as a result of events taking place. The main focus of such logics has been change of *only* belief, whereas the facts describing the world remain the same. Change of belief is known as *epistemic* change. One can also model change of facts, and the resulting consequences of such factual

changes for the beliefs of the agents. Change of facts is also known as *ontic change* (change of the real world, so to speak).[1] In this contribution we use 'event' to denote *any* sort of information change, both epistemic and ontic. Let us begin by a simple example involving various events.

Example 1.1. Given are two players Anne and Bill. Anne shakes a cup containing a single coin and deposits the cup upside down on the table (there are no opportunities for cheating). Heads or tails? Initially, we have a situation wherein both Anne (a) and Bill (b) are uncertain about the truth of that proposition. A player may observe whether the coin is heads or tails, and/or flip the coin, and with or without the other player noticing that. Four example events are as follows.

1. Anne lifts the cup and looks at the coin. Bill observes this but is not able to see the coin. All the previous is common knowledge to Anne and Bill.

2. Anne lifts the cup and looks at the coin without Bill noticing that.

3. Anne lifts the cup, looks at the coin, and ensures that it is tails (by some sleight of hand). Bill observes Anne looking but is not able to see the coin, and he considers it possible that Anne has flipped the coin to tails (and this is common knowledge).

4. Bill flips the coin (without seeing it). Anne considers that possible (and this is common knowledge).

Events 1, 3, and 4 are all public in the sense that the actual event is considered possible by both agents, and that both agents know that, and know that they know that, etc.; whereas event 2 is private: Bill is unaware of the event; the event is private to Anne. Events 3 and 4 involve ontic change, whereas events 1 and 2 only involve epistemic change. Flipping a coin is ontic change: the value of the atomic proposition 'the coin is heads' changes from false to true, or from true to false, because of that. But in the case of events 1 and 2 that value, whether true or false, remains unchanged. What changes instead, is how informed the agents are about that value, or about how informed the other agent is. In 1 and 2, Anne still learns whether the coin is heads or tails. In 1, Bill 'only' learns that Anne has learnt whether the coin is heads or tails: he has not gained factual information at all. In Example 2.5, later, we will formalize these descriptions.

---

[1] In the areas known as 'artificial intelligence' and 'belief revision', epistemic and ontic change are called, respectively, *belief revision* [1] and *belief update* [27]. We will not use that terminology.

Various logics have been proposed to model such events. A well-known setting is that of interpreted systems by Fagin et al. [21]. Each agent has a local state; the local states of all agents together with a state of the environment form a global state; belief of an agent is modelled as uncertainty to distinguish between global states wherein the agent has the same local state, and change of belief is modelled as a transition from one global state to another one, i.e., as a next step in a run through the system. In an interpreted system the treatment of epistemic and ontic change is similar— either way it is just a next step in a run, and how the valuation between different points changes is not essential to define or describe the transition. There is a long tradition in such research [30, 21].

The shorter history of dynamic epistemic logic started by focusing on epistemic change [32, 23, 9, 4, 17]. In that community, how to model ontic change was first mentioned by Baltag, Moss, and Solecki as a possible extension to their *action model logic* for epistemic change [5]. More detailed proposals for ontic change are far more recent [20, 16, 15, 10, 28, 33, 25, 26]. The literature will be discussed in more detail in Section 5.

Section 2 contains logical preliminaries, including detailed examples. Section 3 contains the semantic results that we have achieved for the logic. This is our original contribution to the area. These results are that: *(i)* for all finite models and for all consistent formulas we can construct an event that 'realizes' the formula, i.e., the formula becomes true after execution of the event; that: *(ii)* every event (with assignments of form $p := \varphi$, for φ in the language) corresponds to an event with assignments to true and false only; and also that: *(iii)* every event corresponds to a sequence of events with assignments for a single atom only. Section 4 applies the logic to model the dynamics of card players. Section 5 discusses related work in detail.

2   A logic of ontic and epistemic change

We separately introduce the logical language, the relevant structures, and the semantics of the language on those structures. The syntax and semantics *appear* to overlap: updates are structures that come with preconditions that are formulas. In fact it is properly covered by the double induction used in the language definition, as explained after Definition 2.4 below. For a detailed treatment of the logic without ontic change, and many examples, we recommend [17]; for more examples of the logic involving ontic change, see [15, 16, 20].

2.1   Language

We use the style of notation from propositional dynamic logic (PDL) for modal operators which is also used in [10].

Definition 2.1 (Language). Let a finite set of agents A and a countable set of propositional variables P be given. The language $\mathscr{L}$ is given by the following BNF:

$$\varphi \quad ::= \quad p \mid \neg\varphi \mid \varphi \wedge \varphi \mid [\alpha]\varphi$$
$$\alpha \quad ::= \quad a \mid B^* \mid (\mathsf{U}, \mathsf{e})$$

where $p \in P$, $a \in A$, $B \subseteq A$ (the dynamic operator $[B^*]$ is associated with 'common knowledge among the agents in B'), and where (U, e) is an *update* as (simultaneously) defined below.

We use the usual abbreviations, and conventions for deleting parentheses. In particular, $[B]\varphi$ stands for $\bigwedge_{a \in B}[a]\varphi$, and (the diamond form) $\langle\alpha\rangle\varphi$ is equivalent to $\neg[\alpha]\neg\varphi$. Non-deterministic updates are introduced by abbreviation: $[(\mathsf{U}, \mathsf{e}) \cup (\mathsf{U}', \mathsf{f})]\varphi$ is by definition $[\mathsf{U}, \mathsf{e}]\varphi \wedge [\mathsf{U}', \mathsf{f}]\varphi$.

2.2   Structures

Epistemic model. The models which adequately present an information state in a multi-agent environment are Kripke models from epistemic logic. The set of states together with the accessibility relations represent the information the agents have. If one state s has access to another state t for an agent a, this means that, if the actual situation is s, then according to a's information it is possible that t is the actual situation.

Definition 2.2 (Epistemic model). Let a finite set of agents A and a countable set of propositional variables P be given. An epistemic model is a triple $M = (S, R, V)$ such that

- *domain* S is a non-empty set of possible states,

- $R : A \rightarrow \wp(S \times S)$ assigns an *accessibility relation* to each agent a,

- $V : P \rightarrow \wp(S)$ assigns a set of states to each propositional variable; this is the *valuation* of that variable.

A pair (S, R) is called an *epistemic frame*. A pair (M, s), with $s \in S$, is called an *epistemic state*.

A well-known notion of sameness of epistemic models is 'bisimulation'. Several of our results produce models that are bisimilar: they correspond in the sense that even when not identical (isomorphic), they still cannot be distinguished in the language.

Definition 2.3 (Bisimulation). Let two models $M = (S, R, V)$ and $M' = (S', R', V')$ be given. A non-empty relation $\mathfrak{R} \subseteq S \times S'$ is a bisimulation iff for all $s \in S$ and $s' \in S'$ with $(s, s') \in \mathfrak{R}$:

atoms for all $p \in P$: $s \in V(p)$ iff $s' \in V'(p)$;

forth for all $a \in A$ and all $t \in S$: if $(s, t) \in R(a)$, then there is a $t' \in S'$ such that $(s', t') \in R'(a)$ and $(t, t') \in \mathfrak{R}$;

back for all $a \in A$ and all $t' \in S'$: if $(s', t') \in R'(a)$, then there is a $t \in S$ such that $(s, t) \in R(a)$ and $(t, t') \in \mathfrak{R}$.

We write $(M, s) \leftrightarrow (M', s')$, iff there is a bisimulation between M and M' linking s and s', and we then call (M, s) and (M', s') bisimilar. A model such that all bisimilar states are identical is called a *bisimulation contraction* (also known as a *strongly extensional model*).

Update model. An epistemic model represents the information of the agents. *Information change* is modelled as changes of such a model. There are three variables. One can change the set of states, the accessibility relations and the valuation. It may be difficult to find the exact change of these variables that matches a certain description of an information changing event. It is often easier to think of such an event separately. One can model an information changing event in the same way as an information state, namely as some kind of Kripke model: there are various possible events, which the agents may not be able to distinguish. This is the domain of the model. Rather than a valuation, a *precondition* captures the conditions under which such events may occur, and *postconditions* also determine what epistemic models may evolve into. Such a Kripke model for events is called an *update model*, which were first studied by Baltag, Moss and Solecki, and extended with simultaneous substitutions by van Eijck [5, 20].[2] Here we use van Eijck's definition.

Definition 2.4 (Update model). An *update model* for a finite set of agents A and a language $\mathscr{L}$ is a quadruple $\mathsf{U} = (\mathsf{E}, \mathsf{R}, \mathsf{pre}, \mathsf{post})$ where

- *domain* E is a finite non-empty set of events,

- $\mathsf{R} : A \to \wp(\mathsf{E} \times \mathsf{E})$ assigns an *accessibility relation* to each agent,

- $\mathsf{pre} : \mathsf{E} \to \mathscr{L}$ assigns to each event a *precondition*,

- $\mathsf{post} : \mathsf{E} \to (P \to \mathscr{L})$ assigns to each event a *postcondition* for each atom. Each $\mathsf{post}(e)$ is required to be only finitely different from the identity id; the finite difference is called its *domain* $\mathrm{dom}(\mathsf{post}(e))$.

A pair (U, e) with a distinguished actual event $\mathsf{e} \in \mathsf{E}$ is called an *update*. A pair (U, E) with $\mathsf{E}' \subseteq \mathsf{E}$ and $|\mathsf{E}'| > 1$ is a *multi-pointed update*, first introduced in [19]. The event e with $\mathsf{pre}(\mathsf{e}) = \top$ and $\mathsf{post}(\mathsf{e}) = \mathsf{id}$ we name

---

[2] In the literature update models are also called action models. Here we follow [10] and call them update models, since no agency seems to be involved.

skip. An update with a singleton domain, accessible to all agents, and pre-condition $\top$, is a *public assignment*. An update with a singleton domain, accessible to all agents, and identity postcondition, is a *public announce-ment*.

Instead of

$$\mathsf{pre}(\mathsf{e}) = \varphi \text{ and } \mathsf{post}(\mathsf{e})(p_1) = \psi_1, \dots, \text{ and } \mathsf{post}(\mathsf{e})(p_n) = \psi_n$$

we also write[3]

$$\text{for event } \mathsf{e}\text{: if } \varphi, \text{ then } p_1 := \psi_1, \dots, \text{ and } p_n := \psi_n$$

The event skip stands for: nothing happens except a tick of the clock.

To see an update as part of the language we observe that: an update (U, e) is an inductive construct of type α that is built the frame underlying U (we can assume a set enumerating such frames) and from simpler constructs of type φ, namely the preconditions and postconditions for the events of which the update consists. This means that there should be a finite number of preconditions and a finite number of postconditions only, otherwise the update would be an infinitary construct. A finite number of preconditions is guaranteed by restricting ourselves in the language to *finite* update models. A finite number of postconditions is guaranteed by (as well) restricting ourselves to *finite* domain for postconditions. This situation is similar to the case of automata-PDL [24, Chapter 10, Section 3].

If in case of nondeterministic updates the underlying models are the same, we can also see this as executing a multi-pointed update. For example, $(\mathsf{U}, \mathsf{e}) \cup (\mathsf{U}, \mathsf{f})$ can be equated with $(\mathsf{U}, \{\mathsf{e}, \mathsf{f}\})$.

Example 2.5. Consider again the scenario of Example 1.1 on page 88. Let atomic proposition p stand for 'the coin lands heads'. The initial information state is represented by a two-state epistemic model with domain $\{1, 0\}$, with universal access for a and b, and with $V(p) = \{1\}$. We further assume that the actual state is 1. This epistemic state is depicted in the top-left corner of Figure 1. The events in Example 1.1 can be visualized as the following updates. The actual state is underlined.

1. Anne lifts the cup and looks at the coin. Bill observes this but is not able to see the coin. All the previous is common knowledge to Anne and Bill.

$$a, b \quad \underline{\mathsf{p}} \longleftarrow b \longrightarrow \mathsf{np} \quad a, b$$

---
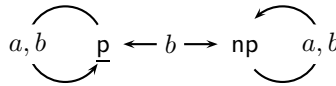
[3] The notation is reminiscent of that for a *knowledge-based program* in the interpreted systems tradition. We discuss the correspondence in Section 5.

Here, $\mathsf{pre}(\mathsf{p}) = p$, $\mathsf{post}(\mathsf{p}) = \mathsf{id}$, $\mathsf{pre}(\mathsf{np}) = \neg p$, $\mathsf{post}(\mathsf{np}) = \mathsf{id}$. The update model consists of two events. The event p corresponds to Anne seeing heads, and the event np to Anne seeing tails; Anne is aware of that: thus the reflexive arrows (identity relation). Bill cannot distinguish them from one another: thus the universal relation. The aspect of common knowledge, or common awareness, is also present in this dynamic structure: the reflexive arrow for Anne also encodes that Anne knows that she lifts the cup and that Bill observes that; similarly for Bill, and for iterations of either awareness.

2. Anne lifts the cup and looks at the coin without Bill noticing that.
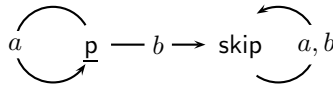
Event p is as in the previous item and skip is as above. In this update, there is no common awareness of what is going on: Anne observes heads knowing that Bill is unaware of that, whereas Bill does not consider the actual event; the b-arrow points to the other event only.

3. Anne lifts the cup, looks at the coin, and ensures that it is tails (by some sleight of hand). Bill observes Anne looking but is not able to see the coin, and he considers it possible that Anne has flipped the coin to tails (and this is common knowledge).
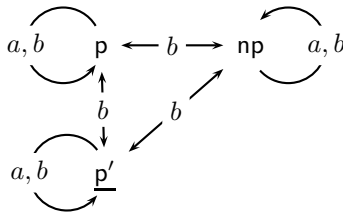
Events p and np are as before, whereas $\mathsf{pre}(\mathsf{p}') = \top$, $\mathsf{post}(\mathsf{p}')(p) = \bot$. The event p' may take place both when the coin is heads and when the coin is tails, in the first case atom p is set to false (tails), and in the second it remains false.

4. Bill flips the coin (without seeing it). Anne considers that possible (and this is common knowledge).
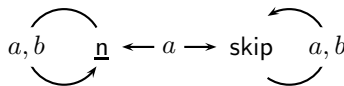
Here, $\mathsf{pre}(\mathsf{n}) = \top$, $\mathsf{post}(\mathsf{n})(p) = \neg p$, and skip is as before. For models where all accessibility relations are equivalence relations we will also use a simplified visualization that merely links states in the same equivalence class. E.g., this final event is also depicted as:

$$\underline{\mathsf{n}} \text{ — } a \text{ — } \mathsf{skip}$$

2.3   Semantics

The semantics of this language is standard for epistemic logic and based on the product construction for the execution of update models from the previous section. Below, $R(B)^*$ is the transitive and reflexive closure of the union of all accessibility relations $R(a)$ for agents $a \in B$. Definitions 2.6 and 2.7 are supposed to be defined simultaneously.

Definition 2.6 (Semantics). Let a model (M, s) with $M = (S, R, V)$ be given. Let $a \in A$, $B \subseteq A$, and $\varphi, \psi \in \mathscr{L}$.

$$
\begin{array}{lll}
(M, s) \models p & \text{iff} & s \in V(p) \\
(M, s) \models \neg\varphi & \text{iff} & (M, s) \not\models \varphi \\
(M, s) \models \varphi \wedge \psi & \text{iff} & (M, s) \models \varphi \text{ and } (M, s) \models \psi \\
(M, s) \models [a]\varphi & \text{iff} & (M, t) \models \varphi \text{ for all } t \text{ such that } (s, t) \in R(a) \\
(M, s) \models [B^*]\varphi & \text{iff} & (M, t) \models \varphi \text{ for all } t \text{ such that } (s, t) \in R(B)^* \\
(M, s) \models [\mathsf{U}, \mathsf{e}]\varphi & \text{iff} & (M, s) \models \mathsf{pre}(\mathsf{e}) \text{ implies } (M \otimes \mathsf{U}, (s, \mathsf{e})) \models \varphi
\end{array}
$$

We now define the effect of an update on an epistemic state—Figure 1 gives an example of such update execution.

Definition 2.7 (Execution). Given are an epistemic model $M = (S, R, V)$, a state $s \in S$, an update model $\mathsf{U} = (\mathsf{E}, \mathsf{R}, \mathsf{pre}, \mathsf{post})$, and an event $\mathsf{e} \in \mathsf{E}$ with $(M, s) \models \mathsf{pre}(\mathsf{e})$. The result of executing (U, e) in (M, s) is the model $(M \otimes \mathsf{U}, (s, \mathsf{e})) = ((S', R', V'), (s, \mathsf{e}))$ where

- $S' = \{(t, \mathsf{f}) \mid (M, t) \models \mathsf{pre}(\mathsf{f})\}$,

- $R'(a) = \{((t, \mathsf{f}), (u, \mathsf{g})) \mid (t, \mathsf{f}), (u, \mathsf{g}) \in S' \text{ and } (t, u) \in R(a) \text{ and } (\mathsf{f}, \mathsf{g}) \in \mathsf{R}(a)\}$,

- $V'(p) = \{(t, \mathsf{f}) \mid (M, t) \models \mathsf{post}(\mathsf{f})(p)\}$.

Definition 2.8 (Composition of update models). Let update models $\mathsf{U} = (\mathsf{E}, \mathsf{R}, \mathsf{pre}, \mathsf{post})$ and $\mathsf{U}' = (\mathsf{E}', \mathsf{R}', \mathsf{pre}', \mathsf{post}')$ and events $\mathsf{e} \in \mathsf{E}$ and $\mathsf{e}' \in \mathsf{E}'$ be given. The composition $(\mathsf{U}, \mathsf{e}) \,;\, (\mathsf{U}', \mathsf{e}')$ of these update models is $(\mathsf{U}'', \mathsf{e}'')$ where $\mathsf{U}'' = (\mathsf{E}'', \mathsf{R}'', \mathsf{pre}'', \mathsf{post}'')$ is defined as

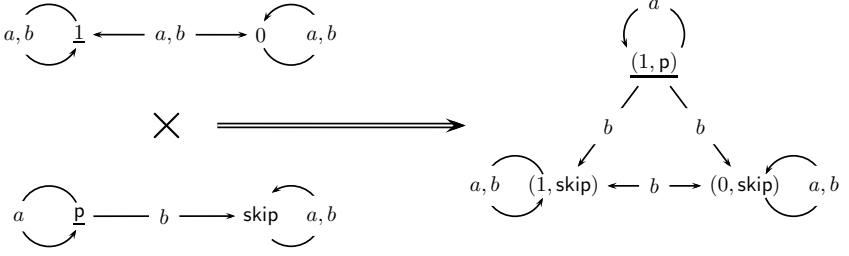- $\mathsf{E}'' = \mathsf{E} \times \mathsf{E}'$,

FIGURE 1. In an epistemic state where Anne and Bill are uncertain about the truth of p (heads or tails), and wherein p is true, Anne looks at the coin without Bill noticing it. The actual states and events are underlined.

- $\mathsf{R}''(a) = \{((\mathsf{f}, \mathsf{f}'), (\mathsf{g}, \mathsf{g}')) \mid (\mathsf{f}, \mathsf{g}) \in \mathsf{R}(a) \text{ and } (\mathsf{f}', \mathsf{g}') \in \mathsf{R}'(a)\}$,

- $\mathsf{pre}''(\mathsf{f}, \mathsf{f}') = \mathsf{pre}(\mathsf{f}) \wedge [\mathsf{U}, \mathsf{f}]\mathsf{pre}'(\mathsf{f}')$,

- $\mathrm{dom}(\mathsf{post}''(\mathsf{f}, \mathsf{f}')) = \mathrm{dom}(\mathsf{post}(\mathsf{f})) \cup \mathrm{dom}(\mathsf{post}'(\mathsf{f}'))$ and
  if $p \in \mathrm{dom}(\mathsf{post}''(\mathsf{f}, \mathsf{f}'))$, then

$$\mathsf{post}''(\mathsf{f}, \mathsf{f}')(p) = \begin{cases} \mathsf{post}(\mathsf{f})(p) & \text{if } p \notin \mathrm{dom}(\mathsf{post}'(\mathsf{f}')), \\ [\mathsf{U}, \mathsf{f}]\mathsf{post}'(\mathsf{f}')(p) & \text{otherwise.} \end{cases}$$

The reason for $[\mathsf{U}, \mathsf{f}]\mathsf{post}'(\mathsf{f}')(p)$ in the final clause will become clear from the proof detail shown for Proposition 2.9.

Proposition 2.9. $\models [\mathsf{U}, \mathsf{e}][\mathsf{U}', \mathsf{e}']\varphi \leftrightarrow [(\mathsf{U}, \mathsf{e}) \; ; \; (\mathsf{U}', \mathsf{e}')]\varphi$

Proof. Let (M, t) be arbitrary. To show that $(M, t) \models [(\mathsf{U}, \mathsf{e}) \; ; \; (\mathsf{U}', \mathsf{e}')]\varphi$ if and only if $(M, t) \models [\mathsf{U}, \mathsf{e}][\mathsf{U}', \mathsf{e}']\varphi$, it suffices to show that $M \otimes (\mathsf{U} \; ; \; \mathsf{U}')$ is isomorphic to $(M \otimes \mathsf{U}) \otimes \mathsf{U}'$. A detailed proof (for purely epistemic updates) is found in [17]. The postconditions (only) play a part in the proof that the valuations correspond:

For the valuation of facts p in the domain of post'' we distinguish the cases $(p \in \mathrm{dom}(\mathsf{post}(\mathsf{e}))$ but $p \notin \mathrm{dom}(\mathsf{post}'(\mathsf{e}')))$ *(i)*, and otherwise *(ii)*. The valuation of a *i*-atom in a triple $(t, (\mathsf{e}, \mathsf{e}'))$ is $\mathsf{post}(\mathsf{e})(p)$ according to the definition of updates composition; and the valuation of a *ii*-atom is $[\mathsf{U}, \mathsf{e}]\mathsf{post}'(\mathsf{e}')(p)$. Consider the corresponding triple $((t, \mathsf{e}), \mathsf{e}')$. The valuation of an *i*-atom in (t, e) is $\mathsf{post}(\mathsf{e})(p)$, and because p does not occur in $\mathrm{dom}(\mathsf{post}'(\mathsf{e}'))$ its value in the triple $((t, \mathsf{e}), \mathsf{e}')$ will remain the same. For a *ii*-atom, its final value will be determined by evaluating $\mathsf{post}'(\mathsf{e}')(p)$ in $((M \otimes \mathsf{U}), (t, \mathsf{e}))$. This corresponds to evaluating $[\mathsf{U}, \mathsf{e}]\mathsf{post}'(\mathsf{e}')(p)$ in (M, t). \hfill Q.E.D.

2.4   Proof system

A proof system **UM** for the logic is given in Table 1. The proof system is a lot like the proof system for the logic of epistemic actions (i.e., for the logic *without* postconditions to model valuation change) in [5]. There are two differences that makes it worthwhile to present this system. The axiom 'atomic permanence' in [5]—$[\mathsf{U}, \mathsf{e}]p \leftrightarrow (\mathsf{pre}(\mathsf{e}) \to p)$—is now instead an axiom expressing when atoms are *not* permanent, namely how the value of an atom can change, according to the postcondition for that atom:

$$[\mathsf{U}, \mathsf{e}]p \leftrightarrow (\mathsf{pre}(\mathsf{e}) \to \mathsf{post}(\mathsf{e})(p)) \qquad\qquad \text{update and atoms}$$

The second difference is not apparent from Table 1. The axiom

$$[\mathsf{U}, \mathsf{e}][\mathsf{U}', \mathsf{e}']\varphi \leftrightarrow [(\mathsf{U}, \mathsf{e}) \; ; \; (\mathsf{U}', \mathsf{e}')]\varphi \qquad\qquad \text{update composition}$$

also occurs in [5]. But Definition 2.8 to compute that composition is in our case a more complex construction than the composition of update models with only preconditions, because it also involves resetting the postconditions. We find it remarkable that these are the only differences: the interaction between *postconditions* for an atom and the logical operators, *only* occurs in the axiom where that atom is mentioned, or implicitly, whereas the interaction between *preconditions* and the logical operators appears in several axioms and rules.

The proof system is sound and complete. The soundness of the 'update and atoms' axiom is evident. The soundness of the 'update composition' axiom was established in Proposition 2.9. We proved completeness of the logic as a modification of the completeness proof for the logic without ontic change—action model logic—as found in [17], which in turn is a simplified version of the original proof for that logic as found in [5]. We do not consider the modified proof of sufficient original interest to report on in detail.

3   Semantic results

We now present some semantic peculiarities of the logic. These we deem our contribution to the area. The results help to relate different approaches combining ontic and epistemic change (see Section 5). The various 'normal forms' for update models that we define are also intended to facilitate future tool development. Finally, they are relevant when modelling AGM belief revision [1] in a dynamic epistemic setting.

3.1   Arbitrary belief change

Let (M, s) and (M', s') be arbitrary finite epistemic states for the same set of atoms and agents, with $M = (S, R, V)$ and $M' = (S', R', V')$. Surprisingly enough, there is almost always an update transforming the former into the

All instantiations of propositional tautologies
$[\alpha](\varphi \rightarrow \psi) \rightarrow ([\alpha]\varphi \rightarrow [\alpha]\psi)$ distribution
From φ and $\varphi \rightarrow \psi$, infer ψ modus ponens
From φ, infer $[\alpha]\varphi$ necessitation
$[\mathsf{U}, \mathsf{e}]p \leftrightarrow (\mathsf{pre}(\mathsf{e}) \rightarrow \mathsf{post}(\mathsf{e})(p))$ update and atoms
$[\mathsf{U}, \mathsf{e}]\neg\varphi \leftrightarrow (\mathsf{pre}(\mathsf{e}) \rightarrow \neg[\mathsf{U}, \mathsf{e}]\varphi)$ update and negation
$[\mathsf{U}, \mathsf{e}](\varphi \wedge \psi) \leftrightarrow ([\mathsf{U}, \mathsf{e}]\varphi \wedge [\mathsf{U}, \mathsf{e}]\psi)$ update and conjunction
$[\mathsf{U}, \mathsf{e}][a]\varphi \leftrightarrow (\mathsf{pre}(\mathsf{e}) \rightarrow \bigwedge_{(\mathsf{e},\mathsf{f})\in\mathsf{R}(a)}[a][\mathsf{U}, \mathsf{f}]\varphi)$ update and knowledge
$[\mathsf{U}, \mathsf{e}][\mathsf{U}', \mathsf{e}']\varphi \leftrightarrow [(\mathsf{U}, \mathsf{e}); (\mathsf{U}', \mathsf{e}')]\varphi$ update composition
$[B^*]\varphi \rightarrow (\varphi \wedge [B][B^*]\varphi)$ mix
$[B^*](\varphi \rightarrow [B]\varphi) \rightarrow (\varphi \rightarrow [B^*]\varphi)$ induction axiom

Let (U, e) be an update model and let a set   updates and common
of formulas χ_f for every f such that $(\mathsf{e}, \mathsf{f}) \in$ knowledge
$\mathsf{R}(B)^*$ be given. From $\chi_\mathsf{f} \rightarrow [\mathsf{U}, \mathsf{f}]\varphi$ and
$(\chi_\mathsf{f} \wedge \mathsf{pre}(\mathsf{f})) \rightarrow [a]\chi_\mathsf{g}$ for every $\mathsf{f} \in \mathsf{E}$ such
that $(\mathsf{e}, \mathsf{f}) \in \mathsf{R}(B)^*$, $a \in B$ and $(\mathsf{f}, \mathsf{g}) \in$
$\mathsf{R}(a)$, infer $\chi_\mathsf{e} \rightarrow [\mathsf{U}, \mathsf{e}][B^*]\varphi$.

TABLE 1. The proof system **UM**.

latter. There are two restrictions. Both restrictions are technical and not conceptual. Firstly, for the set of agents with non-empty access in M' there must be a submodel of M containing actual state s that is serial for those agents. In other words, if an agent initially has empty access and therefore believes everything ('is crazy') you cannot change his beliefs, but otherwise you can. This seems reasonable. Secondly, models M and M' should only differ in the value of a finite number of atoms; more precisely, if we define that

> an atom is *relevant* in a model iff its valuation is neither empty nor the entire domain,

then the requirement is that only a finite number of atoms is relevant in $M \cup M'$. This is required, because we can only change the value of a finite number of atoms in the postconditions. This also seems reasonable: as both models are finite, the agents can only be uncertain about the value of a *finite* number of atoms (in the combined models M and M'); in other words, they are 'not interested' in the value of the remaining atoms.

For expository purposes we initially assume that all agents consider the actual state s of M a possibility (as in all S5 models, such as Kripke models representing interpreted systems), thus satisfying the first of the two restrictions above: the serial submodel required is then the singleton model

consisting of s, accessible to all agents. The update transforming (M, s) into (M', s') can be seen as the composition of two intuitively more appealing updates. That will make clear how we can also describe the required update in one stroke.

In the first step we get rid of the structure of M. As the epistemic state (M, s) is finite, it has a characteristic formula $\delta_{(M,s)}$ [6, 8].[4] We let the agents publicly learn that characteristic formula. This event is represented by the singleton update (U, e) defined as

$$((\{\mathsf{e}\}, \mathsf{R}, \mathsf{pre}, \mathsf{post}), \mathsf{e}) \text{ with } \begin{aligned} &\mathsf{pre}(\mathsf{e}) = \delta_{(M,s)}, \\ &\text{for all } a : (\mathsf{e}, \mathsf{e}) \in \mathsf{R}(a), \\ &\text{and } \mathsf{post}(\mathsf{e}) = \mathsf{id} \end{aligned}$$

In other words, the structure of the current epistemic state is being publicly announced. The resulting epistemic state is, of course, also singleton, or bisimilar to a singleton epistemic state, as $\delta_{(M,s)}$ holds in all states in M bisimilar to s. Without loss of generality assume that it is singleton. Its domain is $\{(s, \mathsf{e})\}$. This pair (s, e) is accessible to itself because for all agents, $(s, s) \in R(a)$ (all agents consider the actual state s a possibility), and $(\mathsf{e}, \mathsf{e}) \in \mathsf{R}(a)$. The valuation of propositional variables in this intermediate state are those of state s in M. What the value is does not matter: we will not use that valuation.

Now proceed to the second step. In the epistemic state wherein the agents have common knowledge of the facts in s, the agents learn the structure of the resulting epistemic state $M' = (S', R', V')$ and their part in it by executing update (U', s') defined as

$$((S', R', \mathsf{pre}', \mathsf{post}'), s') \text{ with } \begin{aligned} &\text{for all } t' \in S' : \mathsf{pre}'(t') = \top \text{ and} \\ &\text{for relevant } p : \mathsf{post}'(t')(p) = \top \text{ iff } t' \in V'(p) \end{aligned}$$

Note that the domain S' and the accessibility relation R' of U' are precisely those of M', the resulting final epistemic model. The postcondition post' is well-defined, as only a finite number of atoms (the relevant atoms) is considered. Because we execute this update in a singleton model with public access, and because it is executable for every event t', the resulting epistemic state has the same structure as the update: it returns S' and R' again. The postcondition delivers the required valuation of atoms in the final model: for each *event* t' in U' and for all relevant atoms p, p become true in t' if p is true in *state* t' in M' ($\mathsf{post}(t')(p) = \top$), else p becomes false. The value of irrelevant atoms remains the same.

---

[4] A characteristic formula φ for a state (M, s) satisfies that for all ψ, $(M, s) \models \varphi$ iff $\varphi \models \psi$. In fact, for the construction we only need formulas that can distinguish all states in the domain from one another, modulo bisimilarity. Characteristic formulas satisfy that requirement.

We combine these two updates into one by requiring the precondition of the first and the postcondition of the second. Consider U'_r that is exactly as U' except that in all events t' in its domain the precondition is not $\top$ but $\delta_{(M,s)}$: the characteristic formula of *the point s* of (M, s). Update (U'_r, s') does the job: epistemic state $(M \otimes \mathsf{U}'_r, (s, s'))$ is isomorphic to (M', s'). This will be Corollary 3.3 of our more general result, to follow.

Now consider the more general case that the agents with non-empty access in M' are serial in a submodel M^{ser} of M that contains s, with domain S^{ser}. In other words: at least all agents who finally have consistent beliefs in some states, initially have consistent beliefs in all states. The construction above will no longer work: if the actual state is not considered possible by an agent, then that agent has empty access in actual state (s, s') of $(M \otimes \mathsf{U}'_r)$, but not in s' in M'. But if we relax the precondition $\delta_{(M,s)}$, for the point s of (M, s), to the disjunction $\bigvee_{u \in S^{\mathsf{ser}}} \delta_{(M,u)}$, that will now carry along the serial subdomain S^{ser}, the construction will work because an agent can then always imagine *some* state wherein the update has been executed, even it that is not the actual state. This indeed completes the construction.

Definition 3.1 (Update for arbitrary change)**.** Given finite epistemic models $M = (S, R, V)$ and $M' = (S', R', V')$ for the same sets of agents and atoms. Assume that all agents with non-empty access in M' are serial in M^{ser} containing s. The update for arbitrary change $(\mathsf{U}'', (s, s')) = ((\mathsf{E}'', \mathsf{R}'', \mathsf{pre}'', \mathsf{post}''), (s, s'))$ is defined as (for arbitrary agents a and arbitrary relevant atoms p):

$$
\begin{aligned}
\mathsf{E}'' &= S' \\
(t', u') \in \mathsf{R}''(a) \quad &\text{iff} \quad (t', u') \in R'(a) \\
\mathsf{pre}''(t') &= \bigvee_{u \in S^{\mathsf{ser}}} \delta_{(M,u)} \\
\mathsf{post}''(t')(p) &= \begin{cases} \top & \text{if } t' \in V'(p) \\ \bot & \text{otherwise} \end{cases}
\end{aligned}
$$

The epistemic state $(M \otimes \mathsf{U}'', (s, s'))$ is bisimilar to (M', s'), which is the desired result. It will typically not be isomorphic: $M \otimes \mathsf{U}''$ can be seen as consisting of a number of copies of M' (namely $|S^{\mathsf{ser}}|$ copies) 'with the accessibility relations just right to establish the bisimulation'. One copy may not be enough, namely when the state t in M to which that copy corresponds, lacks access for some agents. This access will then also be 'missing' between the states of $(\{t\} \times S')$. But because of seriality one of

the other M' copies will now make up for this lack: there is a $u \in S^{\mathsf{ser}}$ such that $(t, u) \in R(a)$, which will establish access when required, as in the proof of the following proposition.

Proposition 3.2. Given (M, s), (M', s'), and U'' as in Definition 3.1. Then $\mathfrak{R} : ((M \otimes \mathsf{U}''), (s, s')) \leftrightarrow (M', s')$ by way of, for all $t \in S^{\mathsf{ser}}$ and $t' \in S'$: $\mathfrak{R}(t, t') = t'$.

Proof. Let $R^{\otimes}$ be the accessibility relation and $V^{\otimes}$ the valuation in $(M \otimes \mathsf{U}'')$.

   atoms: For an arbitrary relevant atom p: $(t, t') \in V^{\otimes}(p)$ iff $(M, t) \models \mathsf{post}''(t')(p)$, and by definition of post'' we have that $(M, t) \models \mathsf{post}''(t')(p)$ iff $t' \in V'(p)$. Irrelevant atoms do not change value.

   forth: Let $((t_1, t'_1), (t_2, t'_2)) \in R^{\otimes}(a)$ and $((t_1, t'_1), t'_1) \in \mathfrak{R}$. From $((t_1, t'_1), (t_2, t'_2)) \in R^{\otimes}(a)$ follows $(t'_1, t'_2) \in R'(a)$. By definition of $\mathfrak{R}$ we also have $((t_2, t'_2), t'_2) \in \mathfrak{R}$.

   back: Let $((t_1, t'_1), t'_1) \in \mathfrak{R}$ and $(t'_1, t'_2) \in R'(a)$. As M^{ser} is serial for a, and $t_1 \in S^{\mathsf{ser}}$, there must be a t_2 such that $(t_1, t_2) \in R(a)$. As $(M, t_2) \models \bigvee_{t \in \mathrm{dom}(M^{\mathsf{ser}})} \delta_{(M,t)}$ (because t_2 is one of those t) we have that $(t_2, t'_2) \in \mathrm{dom}(M \otimes \mathsf{U}'')$. From that, $(t_1, t_2) \in R(a)$, and $(t'_1, t'_2) \in R'(a)$, follows that $((t_1, t'_1), (t_2, t'_2)) \in R^{\otimes}(a)$. By definition of $\mathfrak{R}$ we also have $((t_2, t'_2), t'_2) \in \mathfrak{R}$.
                                                                                            Q.E.D.

   Note that we keep the states outside the serial submodel M^{ser} out of the bisimulation. Without the seriality constraint the 'back' condition of the bisimilarity cannot be shown: given a $((t_1, t'_1), t'_1) \in \mathfrak{R}$ and $(t'_1, t'_2) \in R'(a)$, but where t_1 has no outgoing arrow for a, the required a-accessible pair from (t_1, t'_1) does not exist. A special case of Proposition 3.2 is the corollary already referred to during the initial two-step construction, that achieves even isomorphy:

Corollary 3.3. Given (M, s), (M', s'), and U'_r as above. Assume that M is a bisimulation contraction. Then $(M \otimes \mathsf{U}'_r) \cong M'$.

Proof. In this special case we have that $(t, t') \in \mathrm{dom}(M \otimes \mathsf{U}'_r)$ iff $(M, t) \models \mathsf{pre}'(t')$ iff $(M, t) \models \delta_{(M,s)}$ for the point s of (M, s). As the last is only the case when $t = s$ (as M is a bisimulation contraction), we end up with a domain consisting of all pairs (s, t') for all $t' \in S'$, a 1-1-correspondence. The bisimulation $\mathfrak{R}$ above becomes the isomorphism $\mathfrak{I}(s, t') = t'$.   Q.E.D.

   A different wording of Proposition 3.2 is that for arbitrary finite epistemic states (M, s) and (M', s') also satisfying the serial submodel constraint, there is an update (U, e) transforming the first into the second. A final appealing way to formulate this result is:

Corollary 3.4. Given are a finite epistemic state (M, s) and a satisfiable formula φ. If all agents occurring in φ have non-trivial beliefs in state s of M, then there is an update *realizing* φ, i.e., there is a (U, e) such that $(M, s) \models \langle \mathsf{U}, \mathsf{e} \rangle \varphi$.

Using completeness of the logic, this further implies that all consistent formulas can be realized in any given finite model. We find this result both weak and strong: it is strong because any conceivable (i.e., using the same propositional letters and set of agents) formal specification can be made true whatever the initial information state. At the same time, it is weak: the current information state does apparently not give any constraints on future developments of the system, or, in the opposite direction, any clue on the sequence of events resulting in it; the ability to change the value of atomic propositions arbitrarily gives too much freedom. Of course, if one restricts the events to *specific protocols* (such as legal game moves [15], and for a more general treatment see [11]), the amount of change is constrained.

AGM belief revision and belief update. Our results on arbitrary belief change seem related to the postulate of success in AGM belief revision [1]. AGM belief revision corresponds to epistemic change, and AGM (in their terminology) belief update [27] corresponds to ontic change (unfortunately, in the AGM community 'update' means something far more specific than what we mean by that term). Given this correspondence we can achieve only expansion by epistemic change, and not proper revision; and the combination of ontic and epistemic change can be seen as a way to make belief update result in belief revision. Apart from this obvious interpretation of epistemic and ontic change, one can also view our result that all consistent formulas can be realized, differently: a consequence of this is that for arbitrary consistent φ and ψ there is an update (U, e) such that $[a]\varphi \rightarrow \langle \mathsf{U}, \mathsf{e} \rangle [a]\psi$. In AGM terms: if φ is believed, then there is a way to revise that into belief of ψ, regardless of whether $\varphi \wedge \psi$ is consistent or not. In other words: revision with ψ is always successful. That suggests that our way of achieving that result by combining epistemic and ontic change might somehow simulate standard AGM belief revision. Unfortunately it is immediately clear that we allow far too much freedom for the other AGM postulates to be fulfilled. It is clearly not a *minimal* change, for example. So walking further down this road seems infeasible.

3.2 Postconditions true and false only

The postconditions for propositional atoms can be entirely simulated by the postconditions true or false for propositional atoms. For a simple example, the public assignment $p := \varphi$ can be simulated by a two-point update $\mathsf{e} \text{———} A \text{———} \mathsf{f}$ (i.e., a nondeterministic event where all agents in A cannot distinguish e from f) such that $\mathsf{pre}(\mathsf{e}) = \varphi$, $\mathsf{post}(\mathsf{e})(p) = \top$, $\mathsf{pre}(\mathsf{f}) = \neg\varphi$,

$\mathsf{post}(\mathsf{f})(p) = \bot$. In the public assignment $(p := \varphi, q := \psi)$ to two atoms p and q we would need a *four*-point update to simulate it, to distinguish all *four* ways to combine the values of two independent atoms.

The general construction consists of doing likewise in every event e of an update model. For each e we make as many copies as the cardinality of the powerset of the range of the postcondition associated with that event. Below, the set $\{0,1\}^{\mathrm{dom}(\mathsf{post}(\mathsf{e}))}$ represents that powerset.

Definition 3.5 (Update model $\mathsf{U}^{\top\bot}$). Given is an update model $\mathsf{U} = (\mathsf{E}, \mathsf{R}, \mathsf{pre}, \mathsf{post})$. Then $\mathsf{U}^{\top\bot} = (\mathsf{E}^{\top\bot}, \mathsf{R}^{\top\bot}, \mathsf{pre}^{\top\bot}, \mathsf{post}^{\top\bot})$ is a *normal update model* with

- $\mathsf{E}^{\top\bot} = \bigcup_{\mathsf{e} \in \mathsf{E}} \{(\mathsf{e}, f) \mid f \in \{0,1\}^{\mathrm{dom}(\mathsf{post}(\mathsf{e}))}\}$

- $((\mathsf{e}, f), (\mathsf{e}', f')) \in \mathsf{R}^{\top\bot}(a)$ iff $(\mathsf{e}, \mathsf{e}') \in \mathsf{R}(a)$

- $\mathsf{pre}^{\top\bot}(\mathsf{e}, f) = \mathsf{pre}(\mathsf{e}) \wedge \bigwedge_{f(p)=1} \mathsf{post}(\mathsf{e})(p) \wedge \bigwedge_{f(p)=0} \neg\mathsf{post}(\mathsf{e})(p)$

- $\mathsf{post}^{\top\bot}(\mathsf{e}, f)(p) = \begin{cases} \top & \text{if } f(p) = 1 \\ \bot & \text{if } f(p) = 0 \end{cases}$

Proposition 3.6. Given an epistemic model $M = (S, R, V)$ and an update model $\mathsf{U} = (\mathsf{E}, \mathsf{R}, \mathsf{pre}, \mathsf{post})$ with normal update model $\mathsf{U}^{\top\bot}$ defined as above. Then $(M \otimes \mathsf{U}) \underline{\leftrightarrow} (M \otimes \mathsf{U}^{\top\bot})$.

Proof. We show that the relation $\mathfrak{R} : (M \otimes \mathsf{U}) \underline{\leftrightarrow} (M \otimes \mathsf{U}^{\top\bot})$ defined as

$$((s, \mathsf{e}), (s, \mathsf{e}, f)) \in \mathfrak{R} \text{ iff } (M, s) \models \mathsf{pre}^{\top\bot}(\mathsf{e}, f)$$

is a bisimulation. Below, the accessibility relations in $(M \otimes \mathsf{U})$ and $(M \otimes \mathsf{U}^{\top\bot})$ are also written as $R(a)$.

atoms
Let (s, e, f) be a state in the domain of $(M \otimes \mathsf{U}^{\top\bot})$. We have to show that for all atoms p, $(M, s) \models \mathsf{post}(\mathsf{e})(p) \leftrightarrow \mathsf{post}^{\top\bot}(\mathsf{e}, f)(p)$. From the definition of $\mathsf{post}^{\top\bot}$ it follows that

$$\mathsf{post}^{\top\bot}(\mathsf{e}, f)(p) \text{ iff } f(p) = 1 \ .$$

From $(M, s) \models \mathsf{pre}^{\top\bot}(\mathsf{e}, f)$ and the definition of $\mathsf{pre}^{\top\bot}$ follows that

$$(M, s) \models \mathsf{post}(\mathsf{e})(p) \text{ iff } f(p) = 1 \ .$$

Therefore

$$(M, s) \models \mathsf{post}(\mathsf{e})(p) \leftrightarrow \mathsf{post}^{\top\bot}(\mathsf{e}, f)(p) \ .$$

forth

Assume that $((s, \mathsf{e}), (s', \mathsf{e}')) \in R(a)$ and that $((s, \mathsf{e}), (s, \mathsf{e}, f)) \in \mathfrak{R}$. Let $f' : \mathrm{dom}(\mathsf{post}(\mathsf{e}')) \to \{0, 1\}$ be the function such that

$$f'(p) = \begin{cases} 1 & \text{if } (M, s') \models \mathsf{post}(\mathsf{e}')(p) \\ 0 & \text{otherwise} \end{cases}$$

Therefore $(M, s') \models \mathsf{pre}^{\top\bot}(\mathsf{e}', f')$. Therefore $((s', \mathsf{e}'), (s', \mathsf{e}', f')) \in \mathfrak{R}$. From $((s, \mathsf{e}), (s', \mathsf{e}')) \in R^{\top\bot}(a)$ follows $(s, s') \in R(a)$ and $(\mathsf{e}, \mathsf{e}') \in \mathsf{R}(a)$. From $(\mathsf{e}, \mathsf{e}') \in \mathsf{R}(a)$ and the definition of access $R^{\top\bot}$ follows $((\mathsf{e}, f), (\mathsf{e}', f')) \in R^{\top\bot}(a)$. From $(s, s') \in R(a)$ and $((\mathsf{e}, f), (\mathsf{e}', f')) \in R^{\top\bot}(a)$ follows $((s, \mathsf{e}, f), (s', \mathsf{e}', f')) \in R(a)$.

back

Suppose $((s, \mathsf{e}), (s, \mathsf{e}, f)) \in \mathfrak{R}$ and $(s, \mathsf{e}, f), (s', \mathsf{e}', f') \in R(a)$. From the last follows $(s, s') \in R(a)$ and $((\mathsf{e}, f), (\mathsf{e}', f')) \in R^{\top\bot}(a)$, therefore also $(\mathsf{e}, \mathsf{e}') \in \mathsf{R}(a)$. Therefore $((s, \mathsf{e}), (s, \mathsf{e}')) \in R(a)$. Just as in the case of **forth** it is established that $((s', \mathsf{e}'), (s', \mathsf{e}', f')) \in \mathfrak{R}$.

Q.E.D.

Corollary 3.7. The logic of change with postconditions true and false only is equally expressive as the logic of change with arbitrary postconditions.

Although it is therefore possible to use postconditions true and false only, this is highly unpractical in modelling actual situations: the descriptions of updates become cumbersome and lengthy, and lack intuitive appeal.

A transformation result similar to that in Proposition 3.6 can *not* be established for the logic with only singleton update models, i.e., the logic of public announcements and public assignments (as in [28]). If public assignments could only be to true and to false, then updates with assignments always result in models wherein the assigned atoms are true *throughout* the model, or false *throughout* the model. Therefore, there is no transformation of, e.g., p———$\neg p$ into p———$\neg p$ using public assignments and public announcements only. The construction above results in a *two-event* update model, that is not a singleton.

A transformation result as in Proposition 3.6 immediately gives an expressivity result as in Corollary 3.7 for the languages concerned. It is also tempting to see such a transformation result as a different kind of expressivity result. In two-sorted languages such as the one we consider in this paper one can then distinguish between the expressivity of two kinds of syntactic objects. A formula (φ) corresponds to a class of models that satisfy that formula, and a modality (α) corresponds to a relation on the class of models. The result is stated in terms of the expressivity of formulas, but

it is also a result about the expressivity of modalities. These two kinds of expressivity are not necessarily linked. One can have logics with the same expressivity of formulas, that have different expressivity of modalities and vice versa.

3.3   Single assignments only

Consider the update model $\mathsf{e}\text{——}a\text{——}\mathsf{f}$ for a single agent a and for two atoms p_1 and p_2 such that in e, if φ_1 then $p_1 := \varphi_2$ and $p_2 := \varphi_3$, and in f, if φ_4 then $p_1 := \varphi_5$ and $p_2 := \varphi_6$. Can we also do the assignments one by one? In other words, does this update correspond to a sequence of updates consisting of events g in which at most one atom is assigned a value: the cardinality of $\mathrm{dom}(\mathsf{g})$ is at most 1. This is possible! *First* we 'store' the value (in a given model (M, s) wherein this update is executed) of all preconditions and postconditions in fresh atomic variables, by public assignments. This can be in arbitrary order, so we do it in the order of the φ_i. This is the sequence of six public assignments $q_1 := \varphi_1$, $q_2 := \varphi_2$, $q_3 := \varphi_3$, $q_4 := \varphi_4$, $q_5 := \varphi_5$, and $q_6 := \varphi_6$. Note that such public assignments do not change the structure of the underlying model. *Next* we execute the original update but without postconditions. This is $\mathsf{e}'\text{——}a\text{——}\mathsf{f}'$ with $\mathsf{pre}(\mathsf{e}') = \mathsf{pre}(\mathsf{e}) = \varphi_1$ and $\mathsf{pre}(\mathsf{f}') = \mathsf{pre}(\mathsf{f}) = \varphi_4$ and with $\mathsf{post}(\mathsf{e}') = \mathsf{post}(\mathsf{f}') = \mathsf{id}$. Note that q_1 remains true whenever e' was executed, because q_1 was set to be true whenever φ_1 was true, the precondition of both e and e'. Similarly, q_4 remains true whenever f' was executed. We have now arrived at the final structure of the model, just not at the proper valuations of atoms.

Finally, the postconditions are set to their required value, *conditional* to the execution of the event with which they are associated. Agent a must not be aware of those conditions (the agent cannot distinguish between e' and f'). Therefore we cannot model this as a public action. The way out of our predicament is a number of two-event update models, namely one for each postcondition of each event in the original update. One of these two events has as its precondition the fresh atom associated with an event in the original update, and the other event its negation, and agent a cannot distinguish between both. The four required updates are

- $\mathsf{e}_1\text{——}a\text{——}\mathsf{e}'_1$ with in e_1, if q_1 then $p_2 := q_2$ and in e'_1, if $\neg q_1$ then id

- $\mathsf{e}_2\text{——}a\text{——}\mathsf{e}'_2$ with in e_2, if q_1 then $p_3 := q_3$ and in e'_2, if $\neg q_1$ then id

- $\mathsf{e}_3\text{——}a\text{——}\mathsf{e}'_3$ with in e_3, if q_4 then $p_5 := q_5$ and in e'_3, if $\neg q_4$ then id

- $\mathsf{e}_4\text{——}a\text{——}\mathsf{e}'_4$ with in e_4, if q_4 then $p_6 := q_6$ and in e'_4, if $\neg q_4$ then id

Now, we are done. These four final updates do not change the structure of the model, when executed. Therefore, now having set the postconditions right, the composition of all these constructs is *isomorphic* to the original

update model! The general construction is very much as in this simple
example.

Definition 3.8 (Update model U^{one}). Given an update model $U = (E, R, pre, post)$, update model U^{one} is the composition of the following update models: *First* perform $\Sigma_{e \in E}|dom(post(e)) + 1|$ public assignments for fresh variables $q, \ldots,$ namely for each $e \in E$, $q_0^e := pre(e)$, and for all $p_1, \ldots, p_n \in dom(post(e))$, $q_1^e := post(e)(p_1), \ldots, q_n^e := post(e)(p_n)$. *Then* execute U but with identity ('trivial') postconditions, i.e., execute $U' = (E, R, pre, post')$ with $post'(e) = id$ for all $e \in E$. *Finally*, execute $\Sigma_{e \in E}|dom(post(e))|$ two-event update models with universal access for all agents wherein for each event just one of its postconditions is set to its required value, by way of the auxiliary atoms. For example, for $e \in E$ as above the first executed update is $e_1 \text{——} A \text{——} e_2$ with in e_1, if q_0^e, then $p_1 := q_1^e$, and in e_2, if $\neg q_0^e$ then id.

The following proposition will be clear without proof:

Proposition 3.9. Given epistemic model M and update model U executable in M. Then U^{one} is isomorphic to U, and $(M \otimes U^{one})$ is isomorphic to $(M \otimes U)$.

This result brings our logic closer to the proposals in [5, 16] wherein only one atom is simultaneously assigned a value. The relation to other proposals will be discussed in Section 5.

4 Card game actions

In this section we apply the logic to model multi-agent system dynamics in the general settings of various game actions for card players, such as showing, drawing, and swapping cards. The precise description of card game dynamics is a prerequisite to compute optimal strategies to play such games [13, 18]. Card deals are also frequently used as standard representation for cryptographic bit exchange protocols [22, 34], where communications/transmissions should from our current perspective be seen as public announcements and/or assignments.

Consider a deck of two Wheat, two Flax, and two Rye cards (w, x, y). Wheat, Flax and Rye are called the *commodities*. Three players Anne, Bill, and Cath (a, b, and c) each draw two cards from the stack. Initially, given a deal of cards, it is common knowledge what the deck of cards is, that all players hold two cards, and that all players (only) know their own cards. For the card deal where Anne holds a Wheat and a Flax card, Bill a Wheat and a Rye card, and Cath a Flax and a Rye card, we write $wx.wy.xy$, and so on. As the cards in one's hand are unordered, $wx.wy.xy$ is the same deal of cards as $xw.wy.xy$, but for improved readability we will always list cards in a hand in alphabetical order. There are certain game actions that result
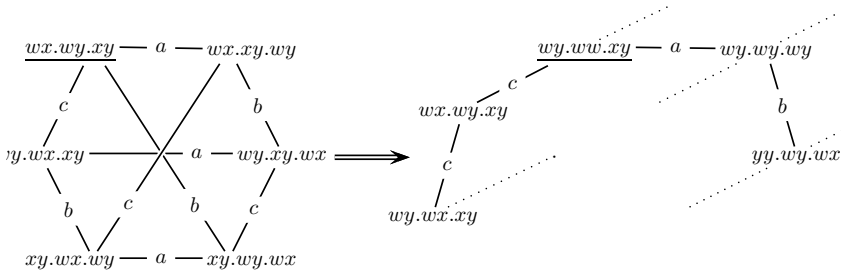
FIGURE 2. On the left is the game state after the cards have been dealt and Anne received Wheat and Flax, Bill received Wheat and Rye, and Cath received Flax and Rye. On the right is part of the game state that results if Anne trades her Wheat card for Bill's Rye card: only states resulting from trading Wheat for Wheat, and (what really happened) Wheat for Rye, are present. The actual deal of cards is underlined. In the figures, assume reflexivity and transitivity of access. The dotted lines suggest that some states are indistinguishable for Cath from yet other states but not present in the picture.

in players exchanging cards. This is called *trading* of the corresponding commodities. Players attempt to get two cards of the same suit. That is called establishing a *corner* in a commodity. Subject to game rules that are non-essential for our exposition, the first player to declare a corner in any commodity, wins the game. For example, given deal $wx.wy.xy$, after Anne swaps her Wheat card for Bill's Rye card, Bill achieves a corner in Wheat, and wins. Of course, players can already achieve a corner when the cards are dealt. This six-card scenario is a simplification of the 'Pit' card game that simulates the trading pit of a stock exchange [31, 13, 15]; the full game consists of 74 cards: 8 commodities of each 9 cards, and two special cards.

An initial game state wherein players only know their own cards and nobody has won the game yet, can be modelled as an epistemic state. There are six such card deals. Assume that the actual deal is $wx.wy.xy$. The (hexagonally shaped) epistemic state (Pit, $wx.wy.xy$) in Figure 2 pictures what players know about each other. All six card deals have to occur to describe the epistemic information present, e.g., Anne cannot distinguish actual deal $wx.wy.xy$ from deal $wx.xy.wy$, the other deal wherein Anne holds Wheat and Flax. But if the deal had been $wx.xy.wy$, Bill could not have distinguished *that* deal from $wy.xy.wx$, wherein Anne holds Wheat and Rye. Therefore, Anne considers it possible that Bill considers it possible that she holds Rye, even though this is actually not the case.

The event wherein Anne and Bill swap one of their cards involves both epistemic and ontic change. Assume that, given deal $wx.wy.xy$,

Anne swaps her Wheat card for Bill's Rye card.

This informal description is not specific enough to be modelled as an update.

In the first place, the role of Cath in the event is not specified. The event where Cath is unaware of the swap taking place, is different from the event where Cath observes the swap and where all agents are commonly aware of this. If Cath had been unaware of the event, she would be mistaken about the actual state of the world. For example, she would incorrectly still believe that neither Anne nor Bill has a corner in a commodity, whereas Bill holds two Wheat cards after the trade (and we assume that he has not yet so declared). It is hard to conceive of such a scenario as a *game*: even in imperfect information games, such as Pit, a basic condition of *fairness* must be fulfilled for players to be able to act rationally. This means that all events should at least be partially observable, and that 'what actually happened' should be considered a possibility for all players. We therefore assume, for now, that Cath learns that Anne and Bill swap one of their cards, but not which card it is. (The 'private swap' will be modelled later, as another example.)

Anne and Bill's roles in the event are also underspecified. Anne may knowingly *choose* a card to hand to Bill (the obvious interpretation), or blindly *draw* one of her cards to hand to Bill. The latter is not obvious, given this description, but becomes more so if we see it as Bill drawing (therefore blindly) one of Anne's cards. For now, assume the obvious. Another specification issue is that we may think of Bill as receiving Anne's Wheat card facedown and only then, in a subsequent action, picking it up. From our modelling perspective, Bill already can be said to *own* the card after he has been handed it, but before he has picked it up he does not yet *know* that he owns it. We first assume that players immediately 'see' the card they are being traded (in this case maybe not the most obvious choice, but the simplest one to model). In other words: Anne and Bill *jointly* learn the new ownership of both cards.

To describe this multi-agent system and its dynamics, assume a propositional language for three agents a, b, c and with atoms u_a^n expressing that Anne holds n cards of suit u. For example, w_a^2 expresses that Anne holds two Wheat cards. In the event where Anne and Bill swap Wheat for Rye, Anne gets one *less* Wheat card, Bill gets one *more* Wheat card, Bill gets one *less* Rye card, and Anne gets one *more* Rye card. In the update model we have to distinguish separate events for each card deal wherein this swap can take place, i.e., corresponding to $wx.wy.xy$, $wx.xy.wy$, and $wy.xy.wx$ (in general this depends on a feature of the local states of the card swapping agents only, namely for both agents on the number of Wheat and Rye cards in their hands, in this specific case that information is sufficient to determine the entire card deal). In case the card deal is $wx.wy.xy$ the precondition

and postcondition are

If $(w_a^1 \wedge y_a^0 \wedge w_b^1 \wedge y_b^1)$, then
$$w_a^1 := \bot \text{ and } w_a^0 := \top \text{ and } w_b^1 := \bot \text{ and } w_b^2 := \top$$
$$\text{and } y_a^0 := \bot \text{ and } y_a^1 := \top \text{ and } y_b^1 := \bot \text{ and } y_b^0 := \top.$$

We name this event $\mathrm{swap}_{ab}^{wx.wy.xy}(w, y)$. If two cards of the same suit are swapped, a simpler description is sufficient. For example, the event wherein Anne and Bill swap Wheat given deal $wx.wy.xy$ is described as $\mathrm{swap}_{ab}^{wx.wy.xy}(w, w)$ with (the same) precondition and (empty) postcondition

$$\text{If } (w_a^1 \wedge y_a^0 \wedge w_b^1 \wedge y_b^1), \text{ then } \varnothing.$$

From the point of view of an actual card deal, there are always four different ways to exchange a single card, namely for each agent either the one or the other card. All of these are clearly different for Anne and Bill, because they either give or receive a different card (we assumed that they know which card they give and see which card they receive). None of these are different for Cath. For different card deals, card swapping events are indistinguishable if those deals were indistinguishable. For example, the event where (Anne and Bill swap Wheat and Rye given $wx.wy.xy$) is indistinguishable for Anne from the event where (Anne and Bill swap Wheat and Rye given $wx.xy.wy$), because card deals $wx.wy.xy$ and $wx.xy.wy$ are the same for Anne.

Therefore, the update model for Anne swapping her Wheat card for Bill's Rye card consists of 24 events. The preconditions and postconditions of the events are as above. The accessibility relations are defined as, for deals $d, d' \in \mathrm{dom}(\mathsf{Pit}) = \{wx.wy.xy, wx.xy.wy, \dots\}$ and cards $q, q', q_1, q_1' \in \{w, x, y\}$, and accessibility relations $R(a), R(b), R(c)$ in the epistemic model Pit:

$$(\mathrm{swap}_{ab}^d(q, q'), \mathrm{swap}_{ab}^{d'}(q_1, q_1')) \in \mathsf{R}(a) \text{ iff } (d, d') \in R(a), q = q_1 \text{ and } q' = q_1'$$
$$(\mathrm{swap}_{ab}^d(q, q'), \mathrm{swap}_{ab}^{d'}(q_1, q_1')) \in \mathsf{R}(b) \text{ iff } (d, d') \in R(b), q = q_1 \text{ and } q' = q_1'$$
$$(\mathrm{swap}_{ab}^d(q, q'), \mathrm{swap}_{ab}^{d'}(q_1, q_1')) \in \mathsf{R}(c) \text{ iff } (d, d') \in R(c)$$

We name the update model *Swap*. The event of Anne and Bill swapping Wheat for Rye has therefore been modelled as update $(Swap, \mathrm{swap}_{ab}^d(w, y))$. The result of executing this update model in epistemic state $(\mathsf{Pit}, wx.wy.xy)$ has the same structure as the update model, as the preconditions are unique for a given state, and as access between events in the update model copies that in the epistemic state. It has been partially visualized in Figure 2. An intuitive way to see the update and the resulting structure in Figure 2, is as a restricted product of the Pit model and *nine* card swapping

events $\text{swap}(q, q_1)$, namely for each combination of the three different cards. Figure 2 then shows just two of those copies, namely for $\text{swap}(w, y)$ and $\text{swap}(w, w)$. For example, the event $\text{swap}(w, y)$ 'stands for' the three events $\text{swap}_{ab}^{wx.wy.xy}(w, y)$, $\text{swap}_{ab}^{wy.xy.wx}(w, y)$, and $\text{swap}_{ab}^{wx.xy.wy}(w, y)$.

Why did we not define such $\text{swap}(q, q_1)$ as updates in their own right, in the first place? Although intuitive, this is not supported by our modelling language. We would like to say that the postconditions are 'Anne gets *one less* Wheat card, and Bill gets *one more* Wheat card,' and similarly for Rye. But instead, we only *can* demand that in case Bill already had a Wheat card (extra precondition), *then* he now has two, etc. Incidentally, we can also add non-deterministic choice to the update language by notational abbreviation, as $[\alpha \cup \beta]\varphi \leftrightarrow ([\alpha]\varphi \wedge [\beta]\varphi)$ (this corresponds to taking the *union* of the epistemic state transformations induced by α and β). We can then define, in the update language, $\text{swap}(w, y) = \text{swap}_{ab}^{wx.wy.xy}(w, y) \cup \text{swap}_{ab}^{wy.xy.wx}(w, y) \cup \text{swap}_{ab}^{wx.xy.wy}(w, y)$.

The case where Anne does not choose her card but Bill blindly draws one of Anne's can also be modelled as an update. The accessibility for Anne then expresses that she is unaware which of her cards has been drawn:

$$(\text{swap}_{ab}^{d}(q, q'), \text{swap}_{ab}^{d'}(q_1, q_1')) \in \mathsf{R}(a) \quad \text{iff} \quad (d, d') \in R(a) \text{ and } q' = q_1'$$

This is somewhat counterintuitive when we still suppose that Anne observes which card she receives from Bill. (We'd have to imagine Bill blindly drawing one of Anne's cards, Anne putting her remaining card facedown on the table, and receiving the card Bill gives her faceup.) A more realistic setting is then that Bill draws one of Anne's card and 'pushes the card he gives to Anne facedown towards her'. At that stage Anne can already be said to own the card, but not yet to know that. All four swapping actions for a given deal are indistinguishable for Anne (as they were and still are for Cath).

Yet another event is where Anne chooses a card to show to Bill, and receives Bill's card facedown (before she picks it up). Access is now

$$(\text{swap}_{ab}^{d}(q, q'), \text{swap}_{ab}^{d'}(q_1, q_1')) \in \mathsf{R}(a) \quad \text{iff} \quad (d, d') \in R(a) \text{ and } q = q_1$$

Obviously, all these variables can be applied to Bill as well.

Picking up a card. The action of picking up a card after it has been handed to you, has another description (see, using another dialect of the language, [15]). One aspect is interesting to observe in the current context. Imagine that given deal $wx.wy.xy$ after Anne and Bill swapping Wheat and Rye, Bill receives the card facedown in the following way: after having laid down his remaining card (a Wheat card) facedown on the table, Anne puts her Wheat card facedown on top of it or under it in a way that Bill cannot

see that. He now picks up his two cards. How does Bill know which of the two Wheat cards that he then holds, is the received card? He does not know, but he does not care either. By looking at his cards after the swap, he effectively *learns* that he holds two Wheat cards (which was already true after having received the card), and after that event he then *knows* that he holds two Wheat cards. A neat way to express that he learns the suit of the card he received, is to say that there is a suit for which he learns to have one more card than he knows. This makes sense in the general Pit game setting wherein one only is allowed to trade a certain number of cards of the same suit. This is formalized in our setting by an update (U, e) with an event with precondition $\mathsf{pre}(\mathsf{e}) = w_b^2 \wedge \neg[b]w_b^2$ (and empty postcondition), as a result of which Bill then knows to hold two Wheat cards, i.e., $[\mathsf{U}, \mathsf{e}][b]w_b^2$.

Private swap. The event where Anne and Bill swap Wheat for Rye but where Cath is unaware of the event is modelled by a four-event update with events $\mathsf{swap}_{ab}^{wx.wy.xy}(w, y)$, $\mathsf{swap}_{ab}^{wy.wx.xy}(w, y)$, $\mathsf{swap}_{ab}^{wx.wy.xy}(w, y)$ and skip, such that for Anne and Bill access among the swap events is as already discussed (including all variations), but where 'Cath thinks nothing happens': for the deals d in the three swap events: $(\mathsf{swap}_{ab}^d(w, y), \mathsf{skip}) \in R(c)$, and $(\mathsf{skip}, \mathsf{skip}) \in R(a), R(b), R(c)$.

5 Comparison to other approaches

Action model logic. Dynamic modal operators for ontic change, in addition to similar operators for epistemic change, have been suggested in various recent publications. As far as we know it was first mentioned by Baltag, Moss, and Solecki as a possible extension to their *action model logic* (for epistemic change), in [5]. This was by example only and without a language or a logic. A precise quotation of all these authors say on ontic change may be in order:

> *Our second extension concerns the move from actions as we have been working them to actions which change the truth values of atomic sentences. If we make this move, then the axiom of Atomic Permanence[5] is no longer sound. However, it is easy to formulate the relevant axioms. For example, if we have an action α which effects the change $p := p \wedge \neg q$, then we would take an axiom $[\alpha]p \leftrightarrow (PRE(\alpha) \rightarrow p \wedge \neg q)$. Having made these changes, all the rest of the work we have done goes through. In this way, we get a completeness theorem for this logic.* [5, p. 24]

The logic that we present here is a realization of their proposal, and we can confidently confirm that the authors were correct in observing that "all the rest (...) goes through". To obtain such theoretical results the

---

[5] I.e., $[\alpha]p \leftrightarrow (PRE(\alpha) \rightarrow p)$, [5, p. 15].

notion of *simultaneous* postconditions (assignments) for a finite subset of atomic propositional letters is essential; this feature is not present in [5] (but introduced in [20]).

In a later proposal by Baltag [2] a fact changing action $flip P$ is proposed that changes ('flips') the truth value of an atom P, with accompanying axioms (for the proper correspondent action α resembling a single-pointed action model) $[\alpha]p \leftrightarrow (\mathsf{pre}(\alpha) \to \neg p)$ if "p changes value (flips) in α", and otherwise $[\alpha]p \leftrightarrow (\mathsf{pre}(\alpha) \to p)$ [2, p. 29]. The approach is restricted to ontic change where the truth value of atoms flips. In the concluding section of [2], the author defers the relation of this proposal to a general logic of ontic and epistemic change to the future.

Recent work in dynamic epistemics. More recently, in a Multi-Agent Systems application-driven line of research [16, 15] assignments are added to the relational action language of [12] but without providing an axiomatization. In this setting only change of *knowledge* is modelled and not change of belief, i.e., such actions describe transformation of S5 models only, such as Kripke models corresponding to interpreted systems.

A line of research culminating in *Logics of communication and change* [10] also combines epistemic and ontic change. It provides a more expressive setting for logical dynamics than our approach. The logic presented here is a sublogic of LCC. In [10] the focus is on obtaining completeness via so-called reduction axioms for dynamic epistemic logics, by extending the basic modal system to PDL. Our treatment of postconditions, also called substitutions, stems from [20]. In the current paper we focus on specific semantic results, and, as said, we use a dynamic epistemic 'dialect', not full PDL.

A recent contribution on combining *public* ontic and epistemic change, including detailed expressivity results for different combinations of static and dynamic modalities, is found in [28]. Our work uses a similar approach to ontic events but describes more complex than public events: the full generality of arbitrarily complex events involves exchange of cards among subgroups of the public, and other events with a 'private' (as opposed to public) character.

Finally, a general dynamic modal logic is presented in [33], where ontic changes are also studied. The semantics of this logic uses tree-like structures, and fixed points are introduced in the language to be able to reason about updates.

Belief revision. An independent recent line of investigation combining epistemic with ontic change arises from the belief revision community. Modelling *belief revision*, i.e., epistemic change, by dynamic operators is an old idea going back to Van Benthem [7]. In fact, this is one of the two original publications—together with [32]—that starts the area of dynamic epistemic

logic. For an overview of such matters see [3, 14, 17]. But modelling on-
tic change—known as *belief update* [27]—in a similar, dynamic modal, way,
including its interaction with epistemic change, is only a recent focus of on-
going research by Herzig and collaborators and other researchers based at
the *Institut de Recherche en Informatique de Toulouse* (IRIT) [25, 26, 29].
Their work sees the execution of an event as so-called *progression* of infor-
mation, and reasoning from a final information state to a sequence of events
realizing it as *regression*—the last obviously relates to planning. The focus
of progression and regression is the change of the *theory* describing the in-
formation state, i.e., the set of all true, or believed, formulas. As already
mentioned in Section 3, the results for arbitrary belief change in Proposi-
tion 3.2 and following corollaries can potentially be applied to model belief
update in the AGM tradition.

Interpreted systems. In a way, dynamic epistemic logics that combine
epistemic and ontic change reinvent results already obtained in the inter-
preted systems community by way of knowledge-based programs [30, 21]:
in that setting, ontic and epistemic change are integrated. Let us point out
some correspondences and differences, using the setting of van der Mey-
den's [30]. This work investigates the implementation of knowledge-based
programs. The transition induced by an update between epistemic states,
in our approach, corresponds exactly to a step in a run in an interpreted
system that is the implementation of a knowledge-based program; the rela-
tion between both is explicit in van der Meyden's notion of the *progression
structure*. Now the dynamic epistemic approach is both more general and
more restrictive than the interpreted systems approach. It is more restric-
tive because dynamic epistemics assumes perfect recall and synchronicity.
This assumption is implicit: it is merely a consequence of taking a state
transition induced by an update as primitive. But the dynamic epistemic
approach is also somewhat more general: it does not assume that accessibil-
ity relations for agents are equivalence relations, as in interpreted systems.
In other words, it can also be used to model other epistemic notions than
knowledge, such as introspective belief and even weaker notions.

Knowledge-based programs consist of joint actions $\langle a_e, a_1, \ldots, a_n \rangle$ where
a_e is an action of the environment and where $a_1, \ldots, a_n$ are *simultaneous*
actions by the agents 1 to n. An agent a acts according to conditions of
the form '*if φ' do a', if φ'' do a'', ...*' etc. Let us overlook the aspect that
conditions φ' have the form of known formulas (by agent a). Still, such
statements *look* familiar to our alternative format for what goes on in an
event, as in '*for event* e*: if φ, then $p_1 := \psi_1$, ..., and $p_n := \psi_n$.*' (see
after Definition 2.4 on page 91). This correspondence is not really there,
but the similar format is still revealing. The different cases in a knowledge-
based program are like the different events in an update model, and they

equally express non-determinism. This is a correspondence. There are also differences. In dynamic epistemics, the condition φ in '*if φ, then $p_1 := \psi_1$,* ...' is both an executability precondition *and* an observation. Inasmuch as it is an observation, it determines agent knowledge. In the interpreted systems approach, observations are (with of course reason) modelled as different from preconditions. The assignments such as $p_1 := \psi_1$ in the 'then' part of event descriptions are merely the ontic part of that event, with the 'if' part describing the epistemic part. Epistemic and ontic features *together* correspond to actions such as a' in '*if φ' do a'*'. In the interpreted systems approach, epistemic and ontic features of information change are therefore not separately modelled, as in our approach.

6   Further research

An unresolved issue is whether updates can be described as compositions of purely epistemic events (preconditions only) and purely ontic events (post-conditions only). In [25] it is shown for public events, for a different logical (more PDL-like) setting. Such a result would be in the line of our other normalization results, and facilitate comparison to related approaches. The result in Section 3.1 seems to suggest this is possible, since a transition from one epistemic model to another is achieved by an epistemic event followed by an ontic event. However, the method described in Section 3.1 is geared towards the original epistemic model. We would like a decomposition that is based solely on the update, which would work regardless of the particular epistemic model to which it is applied.

The logic can be applied to describe cryptographic bit exchange protocols, including protocols where keys change hands or are being sent between agents. The logic is very suitable for the description of protocols for computationally unlimited agents, such as described in the cited [22, 34]. Using dynamic logics may be an advantage given the availability of model checking tools for such logics, as e.g., the very versatile epistemic model checker DEMO [19] by van Eijck. The current version of DEMO only allows epistemic events. But van Eijck and collaborators are in the process of extending DEMO with assignments (postconditions), needed to model events that are both epistemic and ontic.

Our more fine-grained analysis of events may contribute to the description and verification of more complex protocols that also include non-public events. An example that demonstrates the applicability of the logic to the analysis of such protocols is the (solution of the) 'one hundred prisoners and a light bulb' riddle (see e.g., [35]), of which we have a detailed analysis in preparation that we consider of independent interest.

The results for 'arbitrary belief change' suggest yet another possibly promising direction. Under certain conditions arbitrary formulas are real-

izable. What formulas are still realizable if one restricts the events to those considered suitable for specific problem areas, such as forms of multi-agent planning? And given a desirable formula (a 'postcondition' in another sense of the word), what are the initial conditions such that a sequence of events realizes it? This is the relation to AI problems concerning regression as pointed out in the introductory section [25], and also to reasoning given specific protocols, such as always has been the emphasis for knowledge-based programs in the interpreted systems community [21], and as recently investigated in [11] in a dynamic epistemic context.

Acknowledgments

We thank the anonymous reviewers of this contribution to the volume. We also thank Johan van Benthem and Wiebe van der Hoek for their comments and their encouragement.

References

[1] C.E. Alchourrón, P. Gärdenfors & D. Makinson. On the logic of theory change: partial meet contraction and revision functions. *Journal of Symbolic Logic*, 50(2):510–530, 1985.

[2] A. Baltag. A logic for suspicious players: Epistemic actions and belief updates in games. *Bulletin of Economic Research*, 54(1):1–45, 2002.

[3] A. Baltag, H.P. van Ditmarsch & L.S. Moss. Epistemic logic and information update. In J.F.A.K. van Benthem & P. Adriaans, eds., *Handbook on the Philosophy of Information*. Elsevier, Amsterdam. Forthcoming.

[4] A. Baltag & L.S. Moss. Logics for epistemic programs. *Synthese*, 139(2):165–224, 2004.

[5] A. Baltag, L.S. Moss & S. Solecki. The logic of public announcements, common knowledge, and private suspicions. Tech. rep., Centrum voor Wiskunde en Informatica, Amsterdam, 1999. CWI Report SEN-R9922.

[6] J. Barwise & L.S. Moss. *Vicious Circles: On the Mathematics of Non-Wellfounded Phenomena*. CSLI Publications, Stanford, 1996.

[7] J.F.A.K. van Benthem. Semantic parallels in natural language and computation. In H.-D. Ebbinghaus, J. Fernandez-Prida, M. Garrido, D. Lascar & M.R. Artalejo, eds., *Logic Colloquium '87*. North-Holland, Amsterdam, 1989.

[8] J.F.A.K. van Benthem. Dynamic odds and ends. Tech. rep., University of Amsterdam, 1998. *ILLC Publication* ML-1998-08.

[9] J.F.A.K. van Benthem. 'One is a lonely number': on the logic of communication. In Z. Chatzidakis, P. Koepke & W. Pohlers, eds., *Logic Colloquium '02*, vol. 27 of *Lecture Notes in Logic*. Association for Symbolic Logic, 2006.

[10] J.F.A.K. van Benthem, J. van Eijck & B.P. Kooi. Logics of communication and change. *Information and Computation*, 204(11):1620–1662, 2006.

[11] J.F.A.K. van Benthem, J.D. Gerbrandy & E. Pacuit. Merging frameworks for interaction: DEL and ETL. In D. Samet, ed., *Proceedings of the 11th Conference on Theoretical Aspects of Rationality and Knowledge (TARK)*, pp. 72–81. UCL Presses Universitaires de Louvain, 2007.

[12] H.P. van Ditmarsch. Descriptions of game actions. *Journal of Logic, Language and Information*, 11(3):349–365, 2002.

[13] H.P. van Ditmarsch. Some game theory of Pit. In C. Zhang, H.W. Guesgen & W.-K. Yeap, eds., *PRICAI 2004: Trends in Artificial Intelligence, 8th Pacific Rim International Conference on Artificial Intelligence, Auckland, New Zealand, August 9–13, 2004, Proceedings*, vol. 3157 of *Lecture Notes in Computer Science*, pp. 946–947. Springer, 2004.

[14] H.P. van Ditmarsch. Belief change and dynamic logic. In J. Delgrande, J. Lang, H. Rott & J.-M. Tallon, eds., *Belief Change in Rational Agents: Perspectives from Artificial Intelligence, Philosophy, and Economics*, no. 05321 in Dagstuhl Seminar Proceedings. IBFI, Schloss Dagstuhl, 2005.

[15] H.P. van Ditmarsch. The logic of Pit. *Synthese*, 149(2):343–375, 2006.

[16] H.P. van Ditmarsch, W. van der Hoek & B.P. Kooi. Dynamic epistemic logic with assignment. In *Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 05)*, pp. 141–148. ACM Inc., New York, 2005.

[17] H.P. van Ditmarsch, W. van der Hoek & B.P. Kooi. *Dynamic Epistemic Logic*, vol. 337 of *Synthese Library*. Springer, 2007.

[18] S. Druiven. *Knowledge Development in Games of Imperfect Information*. Master's thesis, University of Groningen, 2002.

[19] J. van Eijck. Dynamic epistemic modelling. Tech. rep., Centrum voor Wiskunde en Informatica, Amsterdam, 2004. CWI Report SEN-E0424.

[20] J. van Eijck. Guarded actions. Tech. rep., Centrum voor Wiskunde en Informatica, Amsterdam, 2004. CWI Report SEN-E0425.

[21] R. Fagin, J. Halpern, Y. Moses & M. Vardi. *Reasoning about Knowledge.* MIT Press, Cambridge, MA, 1995.

[22] M. Fischer & R. Wright. Bounds on secret key exchange using a random deal of cards. *Journal of Cryptology*, 9(2):71–99, 1996.

[23] J. Gerbrandy & W. Groeneveld. Reasoning about information change. *Journal of Logic, Language, and Information*, 6(2):147–169, 1997.

[24] D. Harel, D. Kozen & J. Tiuryn. *Dynamic Logic.* MIT Press, Cambridge, MA, 2000. Foundations of Computing Series.

[25] A. Herzig & T. De Lima. Epistemic actions and ontic actions: A unified logical framework. In J.S. Sichman, H. Coelho & S.O. Rezende, eds., *Advances in Artificial Intelligence — IBERAMIA-SBIA 2006, 2nd International Joint Conference, 10th Ibero-American Conference on AI, 18th Brazilian AI Symposium, Ribeirão Preto, Brazil, October 23–27, 2006, Proceedings*, vol. 4140 of *Lecture Notes in Computer Science*, pp. 409–418. Springer, 2006.

[26] A. Herzig, J. Lang & P. Marquis. Action progression and revision in multiagent belief structures, 2005. Manuscript presented at Sixth Workshop on Nonmonotonic Reasoning, Action, and Change (NRAC 2005).

[27] H. Katsuno & A. Mendelzon. On the difference between updating a knowledge base and revising it. In J.F. Allen, R. Fikes & E. Sandewall, eds., *Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning (KR'91). Cambridge, MA, USA, April 22–25, 1991*, pp. 387–394. Morgan Kaufmann, 1991.

[28] B.P. Kooi. Expressivity and completeness for public update logics via reduction axioms. *Journal of Applied Non-Classical Logics*, 17(2):231–253, 2007.

[29] N. Laverny. *Révision, mises à jour et planification en logique doxastique graduelle.* Ph.D. thesis, Institut de Recherche en Informatique de Toulouse (IRIT), Toulouse, France, 2006.

[30] R. van der Meyden. Constructing finite state implementations of knowledge-based programs with perfect recall. In L. Cavedon, A. Rao & W. Wobcke, eds., *Intelligent Agent Systems, Theoretical and Practical Issues (based on PRICAI'96)*, vol. 1209 of *Lecture Notes in Computer Science*, pp. 135–151. Springer, 1997.

[31] Pit game rules. See `http://www.hasbro.com/common/instruct/pit.pdf`.

[32] J. Plaza. Logics of public communications. In M. Emrich, M. Pfeifer, M. Hadzikadic & Z. Ras, eds., *Proceedings of the 4th International Symposium on Methodologies for Intelligent Systems: Poster Session Program*, pp. 201–216. Oak Ridge National Laboratory, 1989. ORNL/DSRD-24.

[33] G.R. Renardel de Lavalette. Changing modalities. *Journal of Logic and Computation*, 14(2):253–278, 2004.

[34] A. Stiglic. Computations with a deck of cards. *Theoretical Computer Science*, 259(1–2):671–678, 2001.

[35] W. Wu. 100 prisoners and a lightbulb, 2001. Manuscript.