Product Update and Looking Backward

Audrey Yap

May 21, 2006

Abstract

The motivation behind this paper is to look at temporal information in models of BMS product update. That is, it may be useful to look at models produced through taking products with action models, as being structurally similar to game trees. So a state in a product can be seen as encoding a history of actions, along with an original world. Given that, it seems useful to add the ability to express information about past states. For this purpose, we can combine a temporal modality with product update. This involves adding a new modality to the language allowing us to form statements similar to, "Before you did c, I didn't know whether ϕ was true, but now I know that ϕ is false."

1 Product Update

1.1 Language and Models

A belief epistemic model M is a tuple

$$M = (W, \{\sim_j : j \in G\}, V, w_0).$$

1. W is a set of possible worlds, called the states of the model.
2. G is a set of agents.
3. $\sim_j$ is an equivalence relation defined on W for each agent j. The intended interpretation is that $s \sim_j t$ whenever j cannot differentiate between worlds s and t.
4. V is a valuation.
5. w_0 is the world corresponding to the actual world.

The language for these static models is simply the language of dynamic epistemic logic for our static models, which will be extended with operators expressing what is true after updates, and expressing what is true in the past.

$$\mathcal{L}_{St}\ \phi := p, q, ... \mid \neg\phi \mid \phi \wedge \psi \mid K_i\phi$$

The semantics for the propositional part are standard. However, for a belief epistemic model M and a world w, the semantics for $K_j\phi$ are as follows:

$$M, w \models K_j\phi \text{ iff for all } v \text{ s.t. } w \sim_j v,\ M, v \models \phi.$$

Now, we can define our epistemic action model

$$A = (\Sigma, \{\sim_j : j \in G\}, \{\textsc{Pre}_\sigma : \sigma \in \Sigma\}, \sigma_0).$$

1. Σ is the set of simple actions.

2. $\sim_j$ is an equivalence relation which is defined on Σ for each agent j. The intended interpretation is that $\sigma \sim_j \tau$ whenever j cannot differentiate between actions σ and τ.

3. For each simple action σ, $\textsc{Pre}_\sigma$ defines the preconditions which must be true at a world in order for σ to be performed at that world.

4. σ_0 is the actual action in our update.

We then define $M \times A$ as the epistemic action model

$$(W \times \Sigma, \{\sim'_j : j \in G\}, V', w'_0).$$

1. $W \times \Sigma = \{(w, \sigma) : M, w \models \textsc{Pre}_\sigma\}$. So the update model is the product of the two previous models, restricted only by the condition that a world must satisfy the preconditions for an action for that action to be performed there.

2. We define $\sim'_j$ such that $(w_1, \sigma) \sim'_j (w_2, \tau)$ iff $w_1 \sim_j w_2$ and $\sigma \sim_j \tau$. So j is only uncertain between two updated states if he could not previously tell the difference between the worlds, and the actions performed are also indistinguishable.

3. V' is essentially the old valuation on worlds, such that $(w_1, \sigma) \in V'(p)$ iff $w_1 \in V(p)$.

4. $w'_0 = (w_0, \sigma_0)$. The new actual world is the product of the previous actual world with the actual action performed.

So now $M \times A$ is a new state model, which can be used in further product updates with any action model whose preconditions are in the same language. However, one thing which we might notice is that with subsequent updates, we can see that the worlds in $M \times A$ encode their history. That is, after we take the product by A n many times, the worlds in the resulting model can be seen as $n+1$-tuples, such that each world is of the form $(w, \sigma, \tau, ..., \sigma_n)$, where w is a world in the original state model M, and each σ_i is an action in Σ. So in that sense, we can see a world as encoding its history, where the history is the set of actions which led us there. This might lead us to view product models as trees of a sort, where each subsequent update adds a layer.

Furthermore, the logic of public announcement can be seen as a special case of product update, where there is only one action. The action model corresponding to the announcement of ϕ is as follows:

$$A = (\{\phi!\}, \{\langle \phi!, \phi! \rangle_j\}_{j \in G}, \text{Pre}_{\phi!} \equiv \phi, \phi!).$$

In other words, there is only one action, the $\sim_j$ relation is just a reflexive loop for each agent $j \in G$, and the only precondition for announcing ϕ at a world is that ϕ holds at that world.

Now that we have dynamic BMS models, we can extend the language to reflect this.

$$\mathcal{L}_{BMS}\ \phi := p, q, ... \mid \neg\phi \mid \phi \wedge \psi \mid K_j\phi \mid [\sigma]\phi$$

The semantics for $[\sigma]\phi$ are as follows:

$$M, w \models [\sigma]\phi \text{ iff } M, w \models \text{Pre}_\sigma \text{ implies } M \times A, (w, a) \models \phi.$$

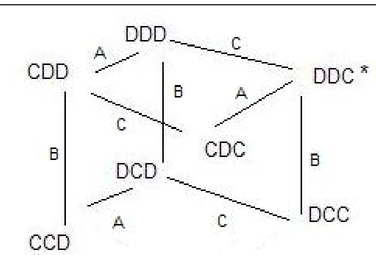
So it is then clear how the public announcement operator $[\phi!]\psi$ is is just a special case of the $[\sigma]$ operator in product update, where our action model A is just this one-action model described above.

1.2 Examples and Problems

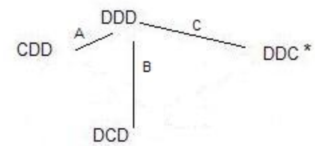
We can do some examples of this; for instance, the familiar Muddy Children example can be modeled using this formalism.

In this example, we suppose that we have three children, who we will name A, B, and C. The children have been playing outside, and some of them have dirty faces. For simplicity's sake, suppose that it is common knowledge that at least one child has a dirty face. This gives us seven possible states of affairs. Suppose that A and B are dirty, and C is clean. The parents then ask if the children know whether or not they are dirty. They answer simultaneously, and must attempt to determine their own state based only on these facts.

We can model the information in a table:

World	A	B	C	$\sim_A$	$\sim_B$	$\sim_C$	
w_1	clean	clean	dirty	w_1, w_5	w_1, w_3	w_1	
w_2	clean	dirty	clean	w_2, w_6	w_2	w_2, w_3	
w_3	clean	dirty	dirty	w_3, w_7	w_1, w_3	w_2, w_3	
w_4	dirty	clean	clean	w_4	w_4, w_6	w_4, w_5	
w_5	dirty	clean	dirty	w_1, w_5	w_5, w_7	w_4, w_5	
w_6✓	dirty	dirty	clean	w_2, w_6	w_4, w_6	w_6, w_7	
w_7	dirty	dirty	dirty	w_3, w_7	w_5, w_7	w_6, w_7	

Then the first action is the simultaneous announcement by all of the children, that they do not know if they are dirty. This action cannot be performed at worlds w_1, w_2, and w_4, since at each of these worlds at least one of the children knows the actual state of affairs. In other words, this announcement can only be made truthfully if at least two children are dirty.

World	A	B	C	$\sim_A$	$\sim_B$	$\sim_C$	
w_3'	clean	dirty	dirty	w_3', w_7'	w_3'	w_3'	
w_5'	dirty	clean	dirty	w_5'	w_5', w_7'	w_5'	
w_6'✓	dirty	dirty	clean	w_6'	w_6'	w_6', w_7'	
w_7'	dirty	dirty	dirty	w_3', w_7'	w_5', w_7'	w_6', w_7'	

However, in the actual world, it is now the case that two children know their state, so the next announcement τ is that A and B know their state, but the only world at which this holds is w_6, the actual world. So this next action eliminates all the worlds except the actual world, and we have only the following:

World	A	B	C	$\sim_A$	$\sim_B$	$\sim_C$
w_6''✓	dirty	dirty	clean	w_6''	w_6''	w_6''

So we can see that product update can model public announcements, and furthermore, with each subsequent update, our worlds encode the actions which brought us to that world. Now, it is not obvious that this history is important in our public announcement logic, since each action model consists of only a single action. Yet the history might be useful nonetheless, for it might be useful to be able to refer to past states.

Before we turn to an example of that, we can see a fairly simple example of product update in which different actions are possible. Suppose we take an example of secret communication, where there are three players in a card game. Let the situation be such that Player 1 has either a red card or a blue card, but the other players don't know which. We'll consider only the uncertainties of the third player, for the sake of simplicity. Now consider three actions. First, Player 1 does nothing. Second, Player 1 secretly shows his card to Player 2. For player 3, the first two actions are indistinguishable. And the third possible action is the one where Player 1 openly shows his card to Player 2.

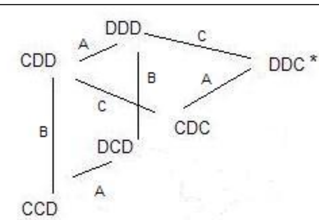
The update can be seen in the following two tables:

World	Cards	Player 3
w_1✓	rbw	w_1, w_2
w_2	rwb	w_1, w_2

World	Cards, Action	Player 3
v_1✓	rbw, 1 shows (secret)	v_1, v_2, v_4, v_5
v_2	rbw, null	v_1, v_2, v_4, v_5
v_3	rbw, 1 shows (public)	v_3, v_6
v_4	rwb, 1 shows (secret)	v_1, v_2, v_4, v_5
v_5	rwb, null	v_1, v_2, v_4, v_5
v_6	rwb, 1 shows (public)	v_3, v_6

So this is an example of the way in which product update works, when there are several different possible actions, some of which are differentiable from others. The example we will consider next will show how the history can become important.

Now consider a variation on the Muddy Children game, where instead of making simultaneous announcements, the children answer in turn whether or not they know their state. A goes first and announces that he does not know whether or not he is dirty. This eliminates w_4, since that is the only world at which A knows his state.

World	A	B	C	$\sim_A$	$\sim_B$	$\sim_C$	
w_1	clean	clean	dirty	w_1, w_5	w_1, w_3	w_1	
w_2	clean	dirty	clean	w_2, w_6	w_2	w_2, w_3	
w_3	clean	dirty	dirty	w_3, w_7	w_1, w_3	w_2, w_3	
w_5	dirty	clean	dirty	w_1, w_5	w_5, w_7	w_5	
w_6✓	dirty	dirty	clean	w_2, w_6	w_6	w_6, w_7	
w_7	dirty	dirty	dirty	w_3, w_7	w_5, w_7	w_6, w_7	

But then notice that in the actual world, B now does know his state. So when B answers the question, he says that he does know. And in fact, this eliminates every world except w_6, the actual world, and w_2, and A never finds out whether or not he is dirty. What the problem is here is that A does not know whether B discovered his state because of A's announcement (which is the case at w_6, or whether B knew that all along (which is the case at w_2.

2 Product Update + History

2.1 Language and Models

We need to redefine $M \times A$ in such a way that our models encode previous states. First, we redefine belief epistemic models such that a model M is now a tuple

$$M = (W, \{\sim_j : j \in G\}, R, V, w_0).$$

We define $M \times A$ as the epistemic action model

$$(W \cup (W \times \Sigma), \{\sim'_j : j \in G\}, R \cup \{R_\sigma : \sigma \in \Sigma\}, V', w'_0).$$

1. $W \cup (W \times \Sigma) = W \cup \{(w, \sigma) : M, w \models \text{PRE}_\sigma\}$. So the update model is the original model, together with the product of the two previous models, restricted only by the condition that a world must satisfy the preconditions for an action for that action to be performed there. So in our new product update models, we keep the old worlds around.

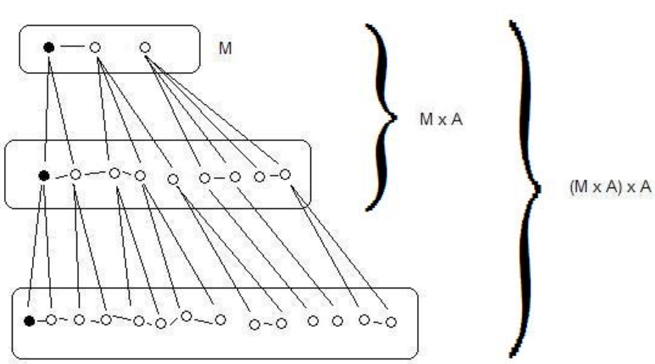
2. We define $\sim'_j$ separately for $W \times \Sigma$ and for W. We will never have agents uncertain between worlds in W and worlds in $W \times \Sigma$. For $w_1, w_2 \in W$, $w_1 \sim'_j w_2$ iff $w_1 \sim_j w_2$. For $W \times \Sigma$, $\sim'_j$ is defined such that $(w_1, \sigma) \sim'_j (w_2, \tau)$ iff $w_1 \sim_j w_2$ and $\sigma \sim_j \tau$. So j is only uncertain between two updated states if he could not previously tell the difference between the worlds, and the actions performed are also indistinguishable.

3. The R_σ relations are a new addition. For each action σ, and world of the form (w, σ), let $R_\sigma((w, \sigma), w)$. In other words, each world in a product model points to its ancestor. So when we take a product, we keep all the old R-relations, and add a new arrow for every world in $W \times \Sigma$, pointing to its ancestor.

4. V' is essentially the old valuation on worlds, though to be more precise, we can define it separately for W and $W \times \Sigma$. For $w \in W$, and p a proposition letter, $w \in V'(p)$ iff $w \in V'(p)$. And for $W \times \Sigma$, we say that $(w_1, \sigma) \in V'(p)$ iff $w_1 \in V(p)$.

5. $w'_0 = (w_0, \sigma_0)$. The new actual world is the product of the previous actual world with the actual action performed.

Action models remain the same, in spite of the new update mechanism. One way to picture our updates, though, is as adding subsequent layers to a tree. Considering only the $[\sigma]$ and P_σ modalities, we do in fact have a tree. And indeed, the way the uncertainties are defined, agents are only uncertain between worlds at the same "level". So the intuitive picture looks something like this:



And we now can extend the language once more.

$$\mathcal{L}_{BMS+H}\ \phi := p, q, ... \mid \neg\phi \mid \phi \wedge \psi \mid K_i\phi \mid [\sigma]\phi \mid P_\sigma\phi$$

The semantics for P_σ are as follows:

$$M, w \models P_\sigma\phi \text{ iff } \exists v \text{ such that } R_\sigma(w, v) \text{ and } M, v \models \phi.$$

Alternatively, we could drop the indexing condition and simply have $P\phi$, with the following semantics:

$$M, w \models P\phi \text{ iff } \exists v, \sigma \text{ such that } R_\sigma(w, v) \text{ and } M, v \models \phi.$$

Furthermore, if we chose to drop the indexing condition, we could even include an iterated past operator $P^*\phi$, such that

$$M, w \models P^*\phi \text{ iff } \exists v_1, ..., v_n, \sigma_1, ..., \sigma_n \text{ such that } R_{\sigma_1}(w, v_1), ..., R_{\sigma_n}(v_{n-1}, v_n) \text{ and } M, v_n \models \phi$$

(assuming only finitely many updates)

This simply states that $P^*\phi$ is true if ϕ is true at some point in the transitive closure of the backward-pointing arrows. Since our models index each arrow by an action, the definition is more complicated, but the intention is to mirror the operation of the Kleene-$*$ operator in PDL.

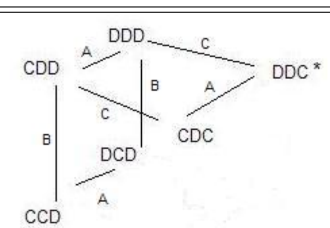
2.2 Expressive Power

2.2.1 Examples

We can now look at several instances of the increase in expressive power with this new modality. For instance, we have the expressive power to model statements such as, "Even before you did σ, I knew that ϕ was true." This simply becomes $P_\sigma K_i\phi$. We can see by the semantics that this formula is true at w exactly when we did obtain w from a world v through a σ-action, so $w = (v, \sigma)$, and ϕ is true at every world which i cannot distinguish from v.

Example 1: References to the past give us a way of getting around the problem with consecutive announcements in Muddy Children. Recall that after A announces that he does not know his state, this is the situation:

World	A	B	C	$\sim_A$	$\sim_B$	$\sim_C$
w_1	clean	clean	dirty	w_1, w_5	w_1, w_3	w_1
w_2	clean	dirty	clean	w_2, w_6	w_2	w_2, w_3
w_3	clean	dirty	dirty	w_3, w_7	w_1, w_3	w_2, w_3
w_5	dirty	clean	dirty	w_1, w_5	w_5, w_7	w_5
w_6✓	dirty	dirty	clean	w_2, w_6	w_6	w_6, w_7
w_7	dirty	dirty	dirty	w_3, w_7	w_5, w_7	w_6, w_7



Previously, the problem was that there was no way for B to make an announcement of the acceptable form which would allow A to guess his own state. But suppose that instead, the following dialogue takes place:

A: "I do not know whether or not I am dirty."
B: "I didn't know that. But I do know whether nor not *I* am dirty."

In this case, B's announcement eliminates all worlds except for the actual world, so C does not even have to make an announcement for all of the children to know their state. However, B's announcement refers back to a past state. So in order to allow for statements of this kind, we need to extend the expressive power of the language in order to be able to make statements about previous states of affairs. Furthermore, there is no statement of the kind the children are allowed to make which can differentiate between the two worlds once B's announcement has been made.

For after B announces that he does know whether or not he is dirty, this eliminates every world except w_6, the actual world, and w_2. A is uncertain between these two worlds, and is clean in one and dirty in the other one. B and C however, know what the actual state of affairs is. So in both w_2 and w_6, B and C know their state, but A does not. But since the children were only allowed to make statements about whether or not they know their own state, no further permissible announcements can differentiate between w_2 and w_6. So the extension of the language to allow for a past modality is a way of looking at what interesting new statements could be permitted which would allow A to learn his state.

Now, we model A's statement, as in the original language, as

$$\neg(K_A D_A \vee K_A \neg D_A).$$

A does not know that he is dirty, and he does not know that he is not dirty. And we now have a way to model B's statements. B's second statement, that he does know whether or not he is dirty, is what we would expect:

$$(K_B D_B \vee K_B \neg D_B).$$

However, his first statement, that he did not know what A just said, is expressed by the following formula:

$$P \neg K_B \neg(K_A D_A \vee K_A \neg D_A).$$

What this formula expresses is that, at the state before the action took place, B did not know the proposition

$$\neg(K_A D_A \vee K_A \neg D_A),$$

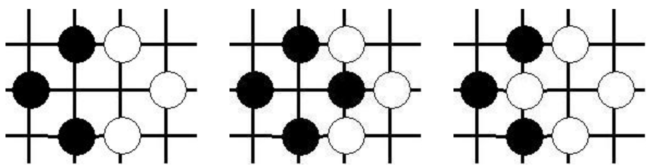
which is exactly A's statement. And combined with his statement that he also does not know his own state, this has the same effect on A's knowledge as a simultaneous announcement by A and B that they do not know their respective states.

Example 2: A simpler example is that we could now introduce preconditions which refer to the past. For instance, this would allow us to model the idea that some actions cannot be performed twice in a row. We could have

$$\textsc{Pre}_\sigma = \neg P_\sigma \top.$$

In other words, σ can only be performed at states which are not of the form (s, σ).

For instance, in the game Go, there are certain moves which do have such past preconditions. There is a rule which states that a player cannot make a move if doing so would result in the same board state as existed after his previous turn. In particular, note the following sequence of turns:



If we want to model the fact that it is illegal for black to play his next stone in the middle of whites stones, then we need to refer to the past. If we call that move σ, whites subsequent move τ and let Φ stand for a conjunction of other rules of the game, then we have

$$\textsc{Pre}_\sigma \equiv \Phi \wedge \neg P_\tau P_\sigma \top.$$

So black can only perform that move if it was not his last move.

2.2.2 Model-Theoretic Considerations

One thing we ought to remark upon is that we no longer have stabilization under bisimulation. For in BMS product update, it is possible to arrive at a state in which subsequent models produced by the update are bisimilar to previous ones. That is, we can have $M \times A$ bisimilar to M, or even more complicated cases of "looping", in which $(M \times A) \times A$ is bisimilar to M, but not to $(M \times A)$. This is treated in detail in [3]. However, our temporal modality is sufficiently expressive to distinguish between worlds and their ancestors, so there will never be a case in which $M \times A$ is bisimilar to M. In fact, if our models have a tree structure with one world at the root, it is possible to define each world uniquely by a temporal formula tracing the path.

However, we can still use the notion of bisimulation to apply to our new kinds of models, since we could have bisimulation instead between entire trees. Viewed structurally, the P_σ modality is simply a diamond modality, so a product model in our new sense can just be seen as a multimodal frame, for which bisimulation is perfectly well defined.

However, one interesting addition in expressive power occurs if we allow for common knowledge in our language. For a group of agents G, let $R(G)^*$ be the reflexive transitive closure of all the $\sim_i$ relations for $i \in G$. Then the semantics for $C_G\phi$ are as follows:

$$M, w \models C_G\phi \text{ iff for all } v \text{ s.t. } \langle w, v\rangle \in R(G)^*,\ M, v \models \phi.$$

However, the typical problem with common knowledge in dynamic epistemic logic, is that there is no reduction axiom for formulas for the form $[\sigma]C_G\phi$. In order to provide a reduction axiom, instead of using the ordinary common knowledge operator, we can instead use a relativized common knowledge operator $C_G(\phi, \psi)$, which expresses that every G-path which consists exclusively of ϕ worlds ends in a ψ world. Then our ordinary common knowledge operator $C_G\phi$ is simply expressed as a special case of the relativized version, by $C_G(\top, \phi)$. Let us define $[\![\phi]\!]$ as

$$[\![\phi]\!] = \{w \in W : M, w \models \phi\}$$

and then we have the semantics for $C_G(\phi, \psi)$:

$$M, w \models C_G(\phi, \psi) \text{ iff for all } v \text{ s.t. } \langle w, v\rangle \in (R(G) \cap [\![\phi]\!])^*,\ M, v \models \phi.$$

A natural language paraphrase of this operator is "If ϕ were announced, it would be common knowledge among G that ψ was the case before the announcement." And this is in fact a statement about what agents knew in the past, so it seems quite natural to model it using our past modality. Then it is perhaps not surprising that if you add P to the language of public announcement with common knowledge, then in fact, relativized common knowledge is definable, since we have the following equivalence:

$$C_G(\phi, \psi) \equiv [\phi!]C_G P\psi.$$

So, in a sense, the past modality operator captures something quite natural about relativized common knowledge. This operator is introduced and discussed in more detail in [5]. However, as it turns out, adding C_G and P to the language cannot replace $C_G(\phi, \psi)$ in many respects, because the past operator is simply too expressive. The main purpose of the relativized common knowledge operator is to obtain reduction axioms for C_G. But we will later show that no reduction axioms for P are possible.

However, even though the past operator does not make relativized common knowledge redundant, it does shed some insight on our intuitions about common knowledge and public announcement. For instance, we might intuitively consider the formula

$$[\phi!]C_G\phi$$

to be true. After all, it seems as though when ϕ is announced, it then becomes common knowledge. However, there fairly simple counterexamples, such as $\phi \equiv p \wedge \neg K_i p$, expressing something like "p is true, and you don't know it." And clearly the following formula is false:

$$[(p \wedge \neg K_i p)!]K_i(p \wedge \neg K_i p)$$

So after the announcement, ϕ is not common knowledge, since ϕ's being announced makes it false. However, perhaps our intuition about common knowledge is better captured by the following formula:

$$[\phi!]C_G P_{\phi!}\phi$$

In other words, what *actually* becomes common knowledge is not ϕ, but that ϕ was true just before the announcement. So the problem of expressing just what is learned through a public announcement of ϕ can be dealt with by using the P modality.

2.3 About the Logic

2.3.1 Axiomatizing the Language

The BMS axiomatization of product update is complete, and gives a reduction to epistemic logic. To some extent, we can follow this treatment for our past modality. Below are the axioms for the forward-looking portion of our language.

Basic Axioms

All propositional tautologies	
(K-normality)	$K_i(\phi \rightarrow \psi) \rightarrow (K_i\phi \rightarrow K_i\psi)$
($[\sigma]$-normality)	$[\sigma](\phi \rightarrow \psi) \rightarrow ([\sigma]\phi \rightarrow [\sigma]\psi)$

Action Axioms

(Atomic Permanence)	$[\sigma]p \leftrightarrow (\textsc{Pre}_\sigma \rightarrow p)$
(Partial Functionality)	$[\sigma]\neg\phi \leftrightarrow (\textsc{Pre}_\sigma \rightarrow \neg[\sigma]\phi)$
(Action-Knowledge)	$[\sigma]K_i\phi \leftrightarrow (\textsc{Pre}_\sigma \rightarrow \bigwedge\{K_i[\tau]\phi : \sigma \sim_i \tau\})$
(Interaction with Past)	$[\sigma]P_\tau\phi \leftrightarrow (\textsc{Pre}_\sigma \rightarrow \sigma = \tau \wedge \phi)$

Modal Rules

(Modus Ponens)	From $\vdash \phi$ and $\vdash \phi \rightarrow \psi$, infer $\vdash \psi$
(K-necessitation)	From $\vdash \phi$, infer $\vdash K_i\phi$
($[\sigma]$-necessitation)	From $\vdash \phi$, infer $\vdash [\sigma]\phi$

So we have four reduction axioms for $[\sigma]$, and we might want to look for four corresponding axioms for P_σ. We first note that we need something playing the same role as $\textsc{Pre}_\sigma$, since these define when it is possible to perform an action at a world. This means we want something like a $\textsc{Post}_\sigma$, which defines when an world has been attained by the performance of an action. For this, we can simply use $P_\sigma\top$. Given this, the first few reduction axioms are straightforward:

Past Looking Axioms

(Atomic Permanence)	$P_\sigma q \leftrightarrow P_\sigma\top \wedge q$
($\neg$-Reduction)	$P_\sigma\neg\phi \leftrightarrow P_\sigma\top \wedge \neg P_\sigma\phi$
($\wedge$-Reduction)	$P_\sigma(\phi \wedge \psi) \leftrightarrow P_\sigma\phi \wedge P_\sigma\psi$
(Interaction with $[\sigma]$)	$P_\sigma[\sigma]\phi \leftrightarrow P_\sigma\top \wedge \phi$

Note, of course, that we do not need an axiom to deal with the formula $P_\tau[\sigma]\phi$ in general, since that involves taking a step back in the history, and then a step forward, but when $\sigma \neq \tau$, the two steps do not lead back to the same world.

We could also include some optional axioms, based on PDL-type considerations, since we have already considered a Kleene-$*$ version of the P operator. For instance, we could also allow for composition ; of past steps, since the Kleene-$*$ can be seen as a generalization of that.

PDL Style Axioms

(Composition Axiom) $P_{\sigma;\tau}\phi \leftrightarrow P_\sigma P_\tau \phi$

(Kleene-$*$ Axiom) $P^*\phi \leftrightarrow \phi \vee PP^*\phi$

2.3.2 Past Reduction Axioms

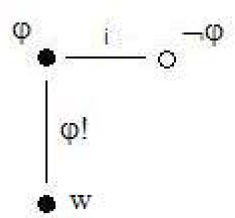
Yet one reduction axiom is still conspicuously absent, and it is this one which poses the most significant difficulty. What about $P_\sigma K_i \phi$? We at the very least have one direction of a reduction axiom for this, which corresponds to a principle of perfect recall.

$$P_\sigma K_i \phi \rightarrow (P_\sigma \top \wedge K_i \bigvee \{P_\tau \phi : \sigma \sim_i \tau\}).$$

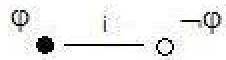
However, the other direction of this implication is not sound, and this implies the following claim.

Claim. No reduction axioms are possible for the P operator.

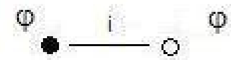
First, to show that the other direction of the implication is not sound, we can use a simple counterexample in public announcement logic. Suppose we begin with two worlds, w a ϕ-world (the actual world) and v a $\neg\phi$-world, between which i cannot differentiate. Then, consider our state after an announcement of ϕ. At $(w, \phi!)$, we have it true that $P_{\phi!}$, and also that $K_i P_{\phi!}\phi$. So the right hand side is satisfied. However, it is false that $P_{\phi!}K_i\phi$, since before the announcement, i could not distinguish w and v. We can illustrate this by a picture:



For insight into the more general reasons for why there are no reduction axioms, consider the following two models:

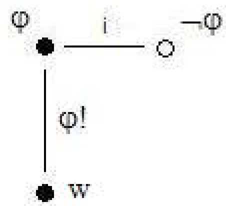


M_1

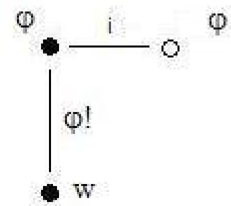


M_2

After a public announcement of ϕ in the actual world, we obtain the following two updated models:



M_1



M_2

Now, consider the formula

$$\neg P_{\phi!} K_i \phi.$$

In other words, i didn't know ϕ before it was announced. Clearly, we have $M_1, w \models \neg P_{\phi!} K_i \phi$, but $M_2, w \not\models \neg P_{\phi!} K_i \phi$. But there is no formula in $\mathcal{L}_{BMS}$ allowing us to distinguish between the two worlds. Since $\neg P_{\phi!} K_i \phi$ is not equivalent to any formula in $\mathcal{L}_{BMS}$, there can be no reduction axioms for $\mathcal{L}_{BMS+H}$.

At best, we can give a partial reduction axiom, which does not entirely allow us to eliminate the P modality, but isolates the cases in which it is not possible to eliminate it.

$$\begin{array}{lll} P_\sigma K_i \phi & \leftrightarrow & P_\sigma \top \wedge K_i \bigvee \{P_\tau \phi : \sigma \sim_i \tau\} \wedge \\ & & P_\sigma K_i \bigwedge \{\neg \textsc{Pre}_\tau \rightarrow \phi : \sigma \sim_i \tau\} \end{array}$$

What this axiom essentially does is split the past into two cases. The first conjunct deals with knowledge about past worlds which the agent still holds possible. However, product update allows agents to discover that certain worlds which were previously indistinguishable from the actual world, are actually impossible past states of affairs. The means by which this can happen is that, even though certain worlds are indistinguishable to an agent, they satisfy different formulas. And furthermore, one world can satisfy the preconditions for an action τ, which cannot be performed in the other world. Suppose then that no action indistinguishable from τ can be performed at this other world. Then after the action, the agent will know that τ, or an action which cannot be differentiated from τ, was performed. And since this was not possible in one of the two worlds, the agent now knows that this world was never the actual world. Product update allows agents to learn more about their state. So the second conjunct deals with these worlds which the agent, after the action, knows are impossible. It claims that, in these worlds, before any action took place, the agent knew ϕ. And these two conjuncts together, imply $P_\sigma K_i \phi$.

Yet because of these considerations, perhaps a partial reduction axiom is all we really want. For what it would mean to give a full reduction of the language including our past modality to that of dynamic epistemic logic, would be that information about the past is reducible to information about the present. But as these examples of learning have illustrated, there is more to the past, and to what I actually knew in the past, than what I know now about the past. This, in some sense, was the point of introducing the new modal operator to the language, so the fact that no reduction to dynamic epistemic logic can be performed should not be too unexpected. However, we do learn something from our attempt to give reduction axioms, since we see just how far we can go. The particular case for which it is not possible to give a reduction axiom shows us what the addition of a past modality really does for our product update models. It allows us to express an agent's having learned something which he did not previously know.

However, the fact that no reduction to dynamic epistemic logic is possible means that we do not have a completeness result for these kinds of models. It is clear that we do not want a reduction to dynamic epistemic logic; but such reductions are one of the more common ways in which its extensions are proved to be complete. The question of completeness for these kinds of tree models, however, is still an open problem.

3 Applications and Problems

As for applications of this past modality, we can actually use this with a public announcement logic to solve several problems involving statements about past knowledge. The classic example of this kind is probably the Sum and Product problem [6], which runs like this:

There are two integers x, y chosen such that $1 < x < y < 100$. Mr. S is told their sum and Mr. P is told their product. Now we have a conversation:

P: I don't know the numbers.
S: I knew that you didn't know. I don't know either.
P: Now I know the numbers.
S: Now I know them too.

We need a temporal modality to model the statement "I knew that you didn't know." Now, the Sum and Product example is somewhat intractable in terms of presentability, so to show how our past operator can deal with these kinds of problems, we will instead consider a smaller version of this problem, which we can call the Sixteen Cards problem:

There are sixteen cards in a drawer.
Hearts: A, Q, 4
Spades: J, 8, 7, 4, 3, 2

Clubs: K, Q, 6, 5, 4
Diamonds: A, 5

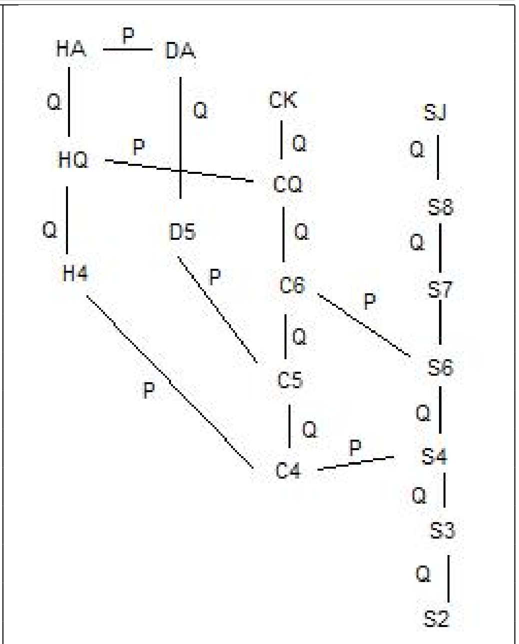
One card is chosen. Mr. P is told the point value of the card, and Mr. Q is told the colour. This fact is common knowledge. Now we have a conversation:

P: I don't know what the card is.
Q: I knew that you didn't know.
P: I know the card now.
Q: I know it too.

Most of the stages of this update can be carried out using the techniques of standard product update (or its more specific version of public announcement update). However, as we saw with the sequential Muddy Children problem, pronouncements such as Q's "I knew that you didn't know" require our temporal modality.

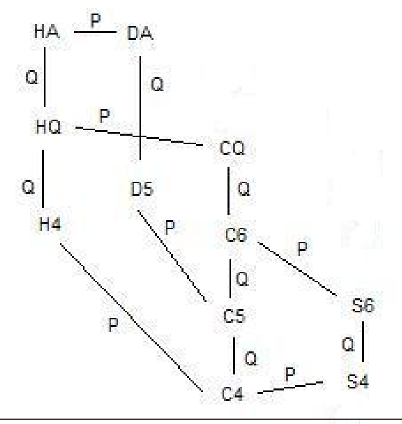
So our initial state model M has sixteen states, one corresponding to each possible card which could have been chosen. For the sake of readability, we will name each state by the card to which it corresponds, and since there is only one possible action (since we have public announcement) in each update, we will drop the reference to the action performed.

World	Mr. P	Mr. Q
HA	HA, DA	HA, HQ, H4
HQ	HQ, CQ	HA, HQ, H4
H4	H4, C4, S4	HA, HQ, H4
DA	HA, DA	DA, D5
D5	D5, C5	DA, D5
CK	CK	CK, CQ, C6, C5, C4
CQ	HQ, CQ	CK, CQ, C6, C5, C4
C6	C6, S6	CK, CQ, C6, C5, C4
C5	D5, C5	CK, CQ, C6, C5, C4
C4	H4, C4, S4	CK, CQ, C6, C5, C4
SJ	SJ	SJ, S8, S7, S6, S4, S3, S2
S8	S8	SJ, S8, S7, S6, S4, S3, S2
S7	S7	SJ, S8, S7, S6, S4, S3, S2
S6	C6, S6	SJ, S8, S7, S6, S4, S3, S2
S4	H4, C4, S4	SJ, S8, S7, S6, S4, S3, S2
S3	S3	SJ, S8, S7, S6, S4, S3, S2
S2	S2	SJ, S8, S7, S6, S4, S3, S2



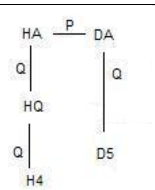
P: I don't know what the card is. ($\neg K_P \phi$)

World	Mr. P	Mr. Q	
HA	HA, DA	HA, HQ, H4	
HQ	HQ, CQ	HA, HQ, H4	
H4	H4, C4, S4	HA, HQ, H4	
DA	HA, DA	DA, D5	
D5	D5, C5	DA, D5	
CQ	HQ, CQ	CQ, C6, C5, C4	
C6	C6, S6	CQ, C6, C5, C4	
C5	D5, C5	CQ, C6, C5, C4	
C4	H4, C4, S4	CQ, C6, C5, C4	
S6	C6, S6	S6, S4	
S4	H4, C4, S4	S6, S4	



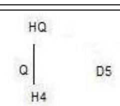
Q: I knew that you didn't know. ($PK_Q\neg K_P\phi$)

World	Mr. P	Mr. Q	
HA	HA, DA	HA, HQ, H4	
HQ	HQ	HA, HQ, H4	
H4	H4	HA, HQ, H4	
DA	HA, DA	DA, D5	
D5	D5	DA, D5	



P: I know the card now. $K_P\phi$

World	Mr. P	Mr. Q	
HQ	HQ	HQ, H4	
H4	H4	HQ, H4	
D5	D5	D5	



Q: I know it too. $K_Q\phi$

World	Mr. P	Mr. Q
D5	D5	D5

So we are left with only the actual world, which we now know is D5.

4 Conclusion

In spite of the open issue of completeness for the system, it seems as though there are several interesting applications for the P modality. By referring to the past, we can model several situations which we could not before, in which information about agents' past epistemic states is relevant to their current states. In fact, what we can now model is agents' learning new information. The BMS framework can model changes in agents' information states, but the language is not expressive enough to talk about agents actually learning new information. This is because what is key in expressing propositions about what agents learn is that we have to be able to express that they now know things which they did not know before. But the BMS language does not allow us to talk about what agents knew before the updates happened, so it does not let us express when agents learned something, since learning suggests knowing something in the present that you did not know in the past. Of course, with the increase in expressive power comes the problem of whether or not the new system is complete, and this is as yet an open problem to be considered in the future.

References

[1] Guillaume Aucher. A combined system for update logic and belief revision. Master's thesis, ILLC Amsterdam, 2003.

[2] Alexandru Baltag, Lawrence Moss, and Slawomir Solecki. The logic of public announcements, common knowledge and private suspicions. In *Proceedings TARK 1998*, 1998.

[3] Tomasz Sadzik. Exploring the iterated update universe. In *Report PP-2006-26*. Institute for Logic, Language, and Compuation, University of Amsterdam, 2006.

[4] Johan van Benthem. One is a lonely number. In *Report PP-2003-07*. Institute for Logic, Language, and Compuation, University of Amsterdam, 2003.

[5] Johan van Benthem, Jan van Eijck, and Barteld Kooi. Common knowledge in update logics. In *Proceedings of the 10th Conference on Theoretical Aspects of Rationality and Knowledge*. 2005.

[6] Hans van Ditmarsch, J. Ruan, and L.C. Verbrugge. Model checking sum and product. In *Proceedings of the 18th Australian Joint Conference on Artificial Intelligence*. 2005.