Product Update and Looking Backward

Audrey Yap

August 1, 2005

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 $K_i\phi$ are as follows:

$$M, w \models K_i\phi \text{ iff for all } v \text{ s.t. } w \sim_i j,\ M, v \models \phi.$$

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 $a \sim_j b$ whenever j cannot differentiate between actions a and b.

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 $(w_1, \sigma_1) \sim'_j (w_2, \sigma_2)$ iff $w_1 \sim_j w_2$ and $\sigma_1 \sim_j \sigma_2$. 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 $(w_1, \sigma_1) \in V'(p)$ 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 $(w, \sigma_1, \sigma_2, ..., \sigma_n)$, 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:

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

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.

$$\mathcal{L}_{BMS}\ \phi := p, q, ... \mid \neg\phi \mid \phi \wedge \psi \mid K_i\phi \mid \langle A, a \rangle \phi$$

The semantics for $\langle A, a \rangle \phi$ are as follows:

$$M, w \models \langle A, a \rangle \phi \text{ iff } M, w \models \textsc{Pre}_a \text{ and } M \times A, (w, a) \models \phi.$$

The public announcement operator $\langle \phi! \rangle \psi$ is then definable as $\langle A, \phi! \rangle$, where A is the action model for $\phi!$ described just above.

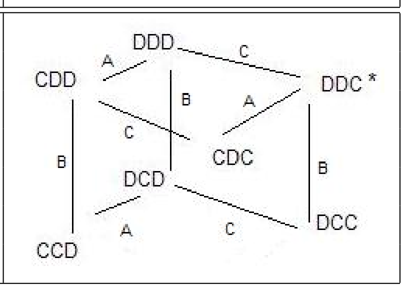
1.2 Examples and Problems

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.

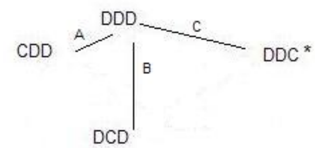
We can model the information in a table:

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 σ_2 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.

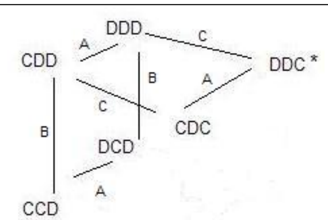
The update can be seen in the following two tables:

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

World	Cards	Player 3
w_1'	rbw	w_1', w_2'
w_2'	rwb	w_1', w_2'
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 with history works, when there are several different possible actions, some of which are differentiable from others. This example, however, does not make use of the history in any significant way. 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

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

R will be a set of relations, which can be empty. Its purpose will become clear below. 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 $(w_1, \sigma_1) \sim'_j (w_2, \sigma_2)$ iff $w_1 \sim_j w_2$ and $\sigma_1 \sim_j \sigma_2$. 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 $(w_1, \sigma_1) \in V'(p)$ 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. However, we now can extend the language once more.

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

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:

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

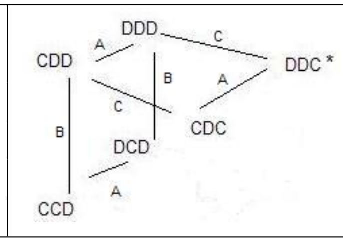
However, we will say more about choices of expressive power in the next section.

2.2 Expressive Power

We can now look at several instances of the increase in expressive power with this new modality.

- 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.

- Also, we have 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.

- 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 $\text{PRE}_\sigma = \neg P_\sigma \top$. In other words, σ can only be performed at states which are not of the form (s, σ).

2.3 About the Logic

We can prove some basic conservation results about the new language with respect to the old language, and preservation under taking products.

Claim. $M, w \models \phi$ iff $M \times A, w \models \phi$.
Proof Sketch. By induction on ϕ. For ϕ a proposition letter, it suffices to note that for $w \in W$, $w \in V(p) \Leftrightarrow w \in V'(p)$, since the two coincide on worlds in W. The cases for the propositional connectives are trivial, and follow straightforwardly from the definition of satisfaction.
Case K_i: Suppose $M, w \models K_i\phi$. Then for all v with $w \sim_i v$, $M, v \models \phi$. Because of the way we defined $M \times A$, we know that for all such v, $v \in M \times A$, and $w \sim_i v$ in $M \times A$. We also know that there are no worlds in $W \times \Sigma$ which are indiscernible from w to i, because of the way $\sim_i$ was constructed. So $M \times A, w \models K_i\phi$. The converse relies on the same observations.
Case P_σ: Suppose $M, w \models P_\sigma\phi$. Then $w = (v, \sigma)$, and $M, v \models \phi$. However, when we construct $M \times A$, worlds in W are carried over and the R_σ-relations between them are preserved. So $M \times A, w \models P_\sigma\phi$, since v is still a σ-successor of w in $M \times A$. The converse is similar, which completes the proof of the claim.

Another thing we might want to assure ourselves of is that modal formulas in our extended language are preserved under bisimulation. This result actually comes for free, because of the way we defined our models. P_σ is just a $\diamond$-operator, the way we have defined it, and given a standard multimodal notion of bisimulation, it is easy to see that all modal formulas in our extended language will be preserved under bisimulation.

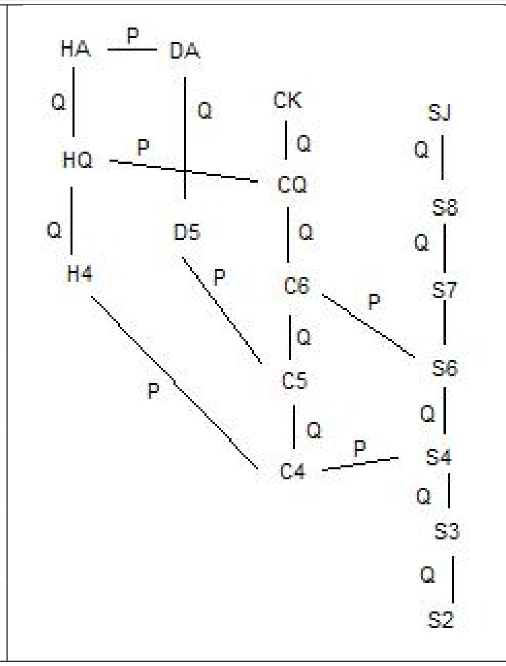
3 Applications and Problems

And as for applications of this past modality, we can actually use this with a public announcement logic to solve the following problem:

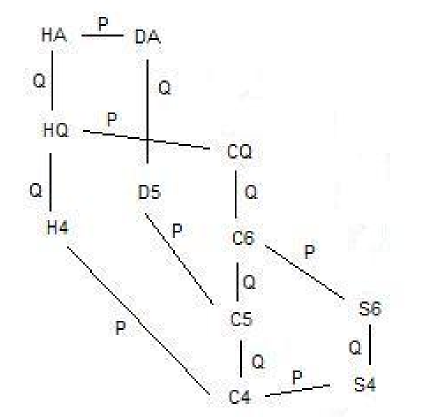
There are sixteen cards in a drawer.
Hearts: A, Q, 4
Spades: J, 8, 7, 4, 3, 2
Clubs: K, Q, 6, 5, 4
Diamonds: A, 5
One card is chosen. Mr. P is told the point value of the card, and Mr. Q is told the colour. This fact is common knowledge. Now we have a conversation:
P: I don't know what the card is.
Q: 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.

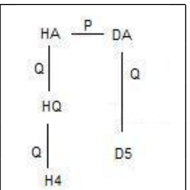
So our initial state model M has sixteen states, one corresponding to each possible card which could have been chosen. For the sake of readability, we will name each state by the card to which it corresponds, and since there is only one possible action (since we have public announcement) in each update,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

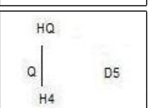
P: I don't know what the card is.

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.

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.

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

Q: I know it too.

World	Mr. P	Mr. Q
D5	D5	D5

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

4 Aucher Update

It is worth noting that the same insights which gave us our backward-looking modality can be applied to a system of quantitative update developed by Guillaume Aucher. Aucher's system is also a modification of BMS product update, but what it does is adds a quantitative element in the form of plausibilities for the worlds. In the sections which follow, we will outline Aucher update, and suggest a relatively minor modification to do it, before showing how it can also be extended to allow for looking backward.

A motivation behind Aucher update stems from a one shortcoming of product update, which is is that it does not allow us to model the unexpectedness of certain actions. It allows us to model agent's suspicions, but there are certain cases where we do not want so much symmetry between indistinguishable actions. For instance, suppose we have a situation with three agents. The third agent cannot distinguish between a case where the first two secretly share information while his back is turned, and a case where they do not. Product update allows us to model the third agent's suspicion that a communication did take place, simply by allowing there to be two indistinguishable actions in the first place, but it does not allow us to model how plausible the third agent takes it to be that a communication took place. With product update, there are only two possibilities for modeling this situation. Either there is a straightforward uncertainty relation between the actual world, where a communication took place, and a possible world where one did not, or the third agent does not think the actual world is possible. It seems natural to look for another option.

4.1 Language and Models

In Aucher update, we have our belief epistemic model

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

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. κ_j is an operator, ranging from 0 to some pre-defined threshold n, defined on all worlds. What κ_j is intended to model, is j's plausibility preferences in the worlds between which she cannot distinguish. Thus, one world in each equivalence class is always assigned plausibility 0, marking it as the most plausible world.
5. V is a valuation.
6. w_0 is the world corresponding to the actual world.

For Aucher update, our static language is similar to that for BMS, but we can now include a belief operator.

$$\mathcal{L}_{St}\ \phi := p, q, ... \mid \neg\phi \mid \phi \wedge \psi \mid K_i\phi \mid B_i^k\phi \text{ where } k \in \mathbb{Z}$$

The semantics for $B_i^k\phi$ are as follows:

$$M, w \models B_i^k\phi \text{ iff for all } v \text{ with } w \sim_i v, \text{ and } \kappa_i(v) \leq k,\ M, v \models \phi.$$

Now when we move to the dynamic language, we have our action model

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

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 $a \sim_j b$ whenever j cannot differentiate between actions a and b.

3. κ^*_j is an operator, ranging from 0 to our pre-defined threshold n, defined on all actions. What κ^*_j is intended to model, is j's plausibility preferences among the actions between which she is uncertain.

4. 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.

5. σ_0 is the actual action in our update.

Finally, we can define $M \times A$ as the epistemic action model

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

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 $(w_1, \sigma_1) \sim'_j (w_2, \sigma_2)$ iff $w_1 \sim_j w_2$ and $\sigma_1 \sim_j \sigma_2$. 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. $\kappa'_j((w, \sigma)) = \mathrm{Cut}_n(\kappa_j(w) + \kappa^*_j(\sigma) - \min\{\kappa_j(v) : M, v \models \textsc{Pre}_\sigma \text{ and } v \sim_j w\})$. The Cut operator is a correction factor such that
$$Cut_n(x) = \begin{cases} x, & \text{if } 0 \leq x \leq n; \\ n, & \text{if } x > n. \end{cases}$$

4. V' is essentially the old valuation on worlds.

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

Correspondingly, we can add the update operator to our language.

$$\mathcal{L}_A \ \phi := p, q, ... \mid \neg\phi \mid \phi \wedge \psi \mid K_i\phi \mid B^k_i\phi \mid \langle A, a\rangle\phi$$

4.2 World-Relative Plausibility

However, Aucher's approach does not deal as well with certain examples, of instance, cases where an agent lies. After all, in the Aucher system, an action has particular preconditions. So in general, an agent cannot report that she has the red card if she does not in fact have the red card. But if we countenance lying as a possibility in a game, then an agent can report that she has the red card even if actually has the blue card.

In Aucher's updates, this just means that we need to remove the preconditions for all such actions, where agents report information. But then we fail to capture the sense in which an agent is *not supposed to* report that she has the red card if she actually has the blue one. After all, the plausibilities an agent assigns to actions in the action model are absolute. But it seems that to capture lying (and an agent's suspicions about how likely her fellow players are to lie), we need some sort of world-relative plausibility. So we can replace some of our preconditions with probabilities.

So most of the formal machinery, such as the language, is exactly the same. Our belief epistemic model M does not change. However, our action model changes slightly, in that we have

$$A = (\Sigma, \{\sim_j : j \in G\}, \{\kappa^*_j(\sigma|[w]_j) : j \in G, w \in W\}, \{\text{Pre}_\sigma : \sigma \in \Sigma\}, \sigma_0).$$

1. $[w]_j = \{v : w \sim_j v\}$. It is the equivalence class of that world, relative to that agent's plausibility.

2. $\kappa^*_j(\sigma|[w]_j)$ is is an operator, ranging from 0 to our pre-defined threshold n, defined at an action and an equivalence class, whenever there is a world in that equivalence class which satisfies the preconditions for that action. What κ^*_j is intended to model, is j's plausibility preferences among the actions between which she is uncertain, at the worlds between which she is uncertain.

The conditional κ^*-function is the only change in the action model. As we might expect, it changes the update mechanism slightly. However, it the update model is almost exactly the same - all that changes is what we assign to κ'.

When product update is performed,

$$\kappa'_j((w,\sigma)) = \text{Cut}_n(\kappa_j(w) + \kappa^*_j(\sigma|[w]_j) - \min\{\kappa(v) : M, v \models \text{Pre}_\sigma\}).$$

So instead of updating with respect to an absolute plausibility for an action, we update with respect to our conditional plausibility.

5 Aucher Update + History

Now, in this section of the paper, I will look at how we can combine world-relative Aucher update with our temporal modality. Everything in the models carries over; we simply need to redefine the manner in which the plausibilities are assigned. This will allow us to model agents' beliefs about past actions, based on current states of affairs. To that end, I will sketch below a system for back-propagating plausibilities. This is a multi-step process, and requires first calculating the plausibilities for worlds in $W \times \Sigma$, and then revising past plausibilities. So the preservation results from the previous sections will not go through; but that is what we want. Although what is true at a world over time will not have changed, we want to be able to revise an agent's beliefs about which worlds are most likely.

5.1 Language and Models

Now we need to describe a way to combine our modified Aucher systems with our Product Update + History models. First, we define belief epistemic models such that a model M is now a tuple

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

- Here, κ_j is only defined on worlds which are leaves. We have a different function expressing an agent's plausibilities for past worlds.
- κ_j^P is a two-place function, which is defined on w_1, w_2 such that at w_1, j holds it possible that w_2 was an ancestor of w_1. It expresses j's degree of belief at w_1 that w_2 was a past state of affairs. So it is defined at w_1, w_2 whenever there is a v such that $w_1 \sim_j v$ and w_2 is an ancestor of v.

An action model is still a tuple

$$A = (\Sigma, \{\sim_j : j \in G\}, \{\kappa_j^*(\sigma | [w]_j) : j \in G, w \in W\}, \{\textsc{Pre}_\sigma : \sigma \in \Sigma\}, \sigma_0).$$

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

$$(W \cup (W \times \Sigma), \{\sim_j' : j \in G\}, \{\kappa_j' : j \in G\}, \{\kappa_j^P \cup \kappa_j'^P : j \in G\}, R \cup \{R_\sigma : \sigma \in \Sigma\}, V', w_0').$$

So all that is left is to explain how $\kappa_j'^P$ is calculated. Now, our mechanism for assigning probabilities to previous worlds will be a multi-step process. It applies only to the newly updated model, since we still want to be able to talk about our beliefs at previous stages of the game. Here is the rough idea.

1. Ignore uncertainties for parent nodes whose children we can now differentiate. This models the idea that we can retrospectively distinguish between certain past states of affairs.
2. Normalise each level of the tree, starting at the leaves and moving toward the root. This is to account for the changes we have made between uncertainty links, since there must always be one world in each equivalence class whose plausibility is 0.

The algorithm at each stage, simultaneously builds up a set X of worlds which j, at w, thinks are possible past states of affairs.

Stage 0: Let $X_0 = [w]_j$. So X is the set of worlds v such that j cannot distinguish between w and v. For $v \in X_0$, let $\kappa_j^P(w,v) = \kappa_j'(v)$.
Stage $n+1$: Let $X_{n+1} = \{v : \exists u \in X_n, \sigma \text{ such that } R_\sigma(u,v)\}$. For $v \in X_{n+1}$, let $\kappa_j^P(w,v) = \kappa_j(v) - \min(X_{n+1})$. So this step just normalizes the probabilities.

The only case in which this back-propagation changes plausibilities is in cases where the update results in the erasure of uncertainty links. If an update does not result in the erasure of uncertainty links, there will be nothing to do, since the κ-value for each parent will be the minimum value of all its nodes. We can see this by considering the update formula:

$$\kappa_j'((w,\sigma)) = \mathrm{Cut}_n(\kappa_j(w) + \kappa_j^*(\sigma|[w]_j) - \min\{\kappa(v) : M, v \models \textsc{Pre}_\sigma\}).$$

First, we note that since some action which can be performed at that world must have plausibility 0, there will be some child node whose κ-value is equal to that of the parent. Second, for $\kappa((w,\sigma)) > \kappa(w)$, it must be the case that $\min\{\kappa(v) : M, v \models \textsc{Pre}_\sigma\} > 0$. But this is only possible if σ cannot be performed at the most plausible world in the uncertainty equivalence class; and that means that after removing uncertainty links, we can differentiate between that world and w.

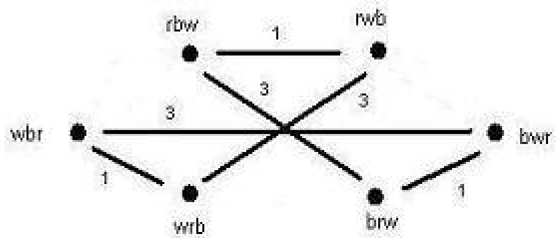
So the language is simply Aucher's original language with the addition of a past modality operator whose semantics are the same.

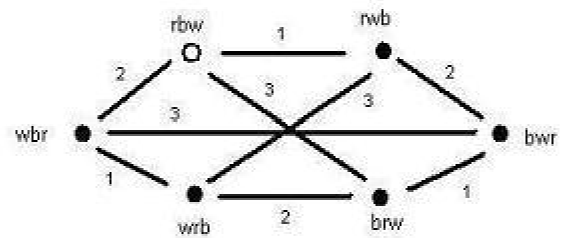
$$\mathcal{L}_{A+H}\ \phi := p, q, ... \mid \neg\phi \mid \phi \wedge \psi \mid K_i\phi \mid B_i^k\phi \mid \langle A, a\rangle\phi \mid P_\sigma\phi$$

5.2 Expressive Power

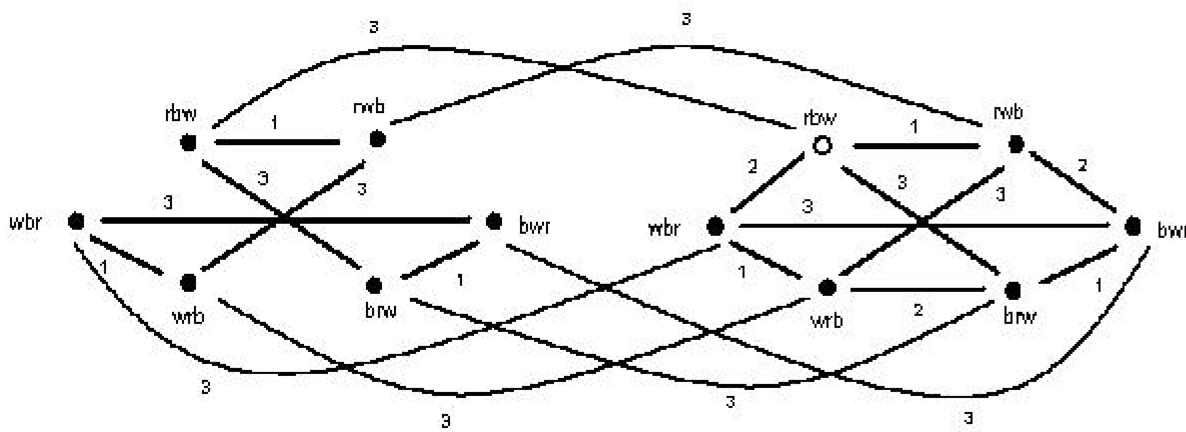
- First, we can illustrate the increase in expressive power that Aucher update provides to product update. Consider our familiar three cards example with the standard uncertainties (everyone looks at their cards). Now suppose that 3 leaves the room for a minute, and while he is gone, 1 tells 2 which card she has. Then 2 knows the complete configuration of cards, and 1 knows that 2 knows the configuration. But what about 3?

  There are two situations for which BMS accounts. In one situation, the clandestine information exchange happens with 3 being entirely oblivious to it. In this case, we are in a strange situation, in which an update has occurred, but 3 does not think so. So 3 does not in fact think that the actual world is possible. In the diagram below, the worlds on the left half are the worlds in which information exchange occurred, and the worlds on the right are those in which the null action occurred. Thus, the world rbw on the left is the actual world, but 3 does not consider the left hand worlds to be possible.





But consider another situation in which, as 3 is coming back into the room, he thinks he hears 1 and 2 talking, but isn't sure what he heard. So in this case, he suspects that some information exchange might have taken place, but isn't certain. In this situation, 3 does think the actual world is possible, but is uncertain between the actual world and a world in which no action (or the null action) has taken place.

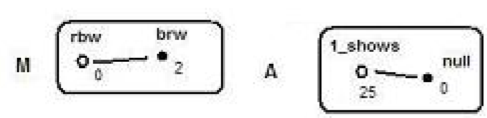


(Some lines in these figures will be omitted for the sake of readability.)

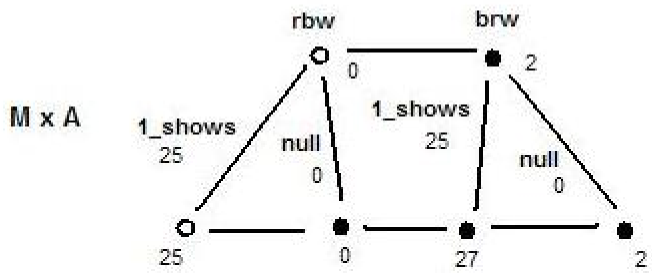
Neither of these options seem like a particularly satisfactory way to model the fact that 1 and 2 secretly exchanged information. The issue here is one of degree; we want to be able to represent cases in which 3 holds it possible that information has been exchanged, but does not really think that it has. In other words, we want to be able to make a distinction between what an agent acknowledges is possible, and what she thinks is likely.

However, this is simple to do using Aucher update. What we do is attend to 3's plausibility assignment to the action in which 1 and 2 exchange information. However, the update models are structurally the same, and this is what we want. The only thing that differs is the plausibility 3 assigns to an information exchange having taken place. We can take a simplified version of the action model, only taking into account 3's uncertainties in the actual world rbw. For instance, suppose we have the following valuation in the initial model:

$\kappa(\text{rbw}) = 0$ $\qquad$ $\kappa^*(\text{1 shows}) = 25$
$\kappa(\text{brw}) = 2$ $\qquad$ $\kappa^*(\text{null}) = 0$

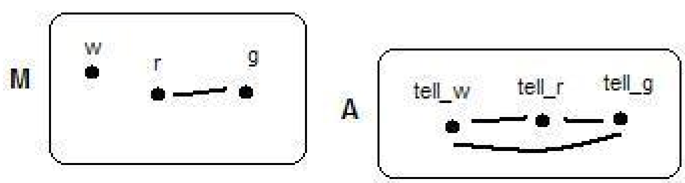


This leads to the following update, which is easy to calculate:



Thus we can model 3's varying degrees of suspicion with respect to an information exchange between 1 and 2, simply by varying κ^*(1 shows), instead of by a difference in the structure of the update model.

- Now we can illustrate the increase in expressive power accorded by adding in world-relative plausibility. Suppose we have an agent who is red/green colour blind, and is looking at a card in front of him which he knows might be either red, green, or white. In this case, he can distinguish between two equivalence classes of worlds - the class where the card is red or green, and the one where it is white. Now, suppose a trustworthy source passes him a note, face down, upon which the colour of his card is written. But until he looks at it, he cannot tell the difference between the three possible actions.



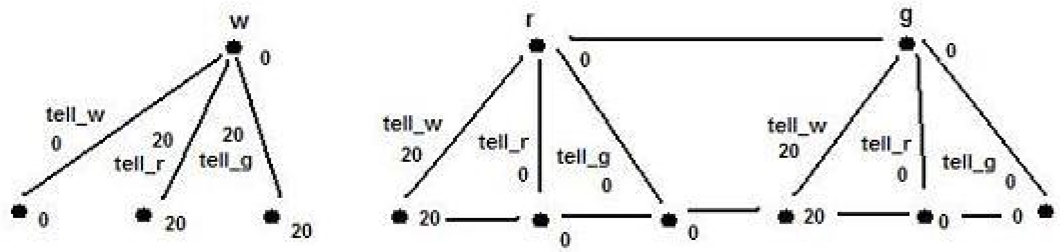
Yet, the agent, believing his source to be trustworthy, thinks the information reported to him will be accurate. If he has the red or green card, he thinks it likely that he will be told so, though he does not know which of the two statements is more likely. However, in that case, he would hold it quite unlikely (since the source is trustworthy) that he would be told he has the white card. In this sort of situation, however, Aucher update does not allow us to model this information. All we have are absolute plausibilities for the actions between which the agent cannot distinguish. So with Aucher update, we might simply assign all actions and all worlds plausibility 0, since we have no reason to think that any one action is likelier, without any further information. But we might think that our agent should not have the same plausibility for being told he has a white card, in the world where he has a white card, as in the world where he has a red one.

Thus, in this example, we might assign the following world-relative plausibilities:

$\kappa(r) = 0$
$\kappa(g) = 0$
$\kappa(w) = 0$

$\kappa^*(\text{tell r}|[r]) = 0$
$\kappa^*(\text{tell g}|[r]) = 0$
$\kappa^*(\text{tell w}|[r]) = 20$

$\kappa^*(\text{tell r}|[w]) = 20$
$\kappa^*(\text{tell g}|[w]) = 20$
$\kappa^*(\text{tell w}|[w]) = 0$



Thus, I can reflect the fact that if an agent knows he does not have a white card, he thinks it very unlikely that a trustworthy source would tell him he does have one. Similarly, if he knows he does have a white card, he thinks it very unlikely that he would be told he does not have one.

5.3 About the Logic

In this section, we will show parallel results as those shown with BMS + H, with respect to the extended language being a conservative extension of the old.