BMS revisited*

Guillaume Aucher
University of Luxembourg
6, rue Richard Coudenhove-Kalergi
L-1359 Luxembourg
guillaume.aucher@uni.lu

Abstract

The insight of the BMS logical framework (proposed by *B*altag, *M*oss and *S*olecki) is to represent how an event is perceived by several agents very similarly to the way one represents how a static situation is perceived by them: by means of a Kripke model. There are however some differences between the definitions of an epistemic model (representing the static situation) and an event model. In this paper we restore the symmetry. The resulting logical framework allows, unlike any other one, to express statements about *ongoing* events and to model the fact that our perception of events (and not only of the static situation) can also be updated due to other events. We axiomatize it and prove its decidability. Finally, we show that it embeds the BMS one if we add common belief operators.

Dynamic epistemic logic deals with the issue of representing from a logical point of view the beliefs of several agents (about a given situation) and how these beliefs change over time as new events occur [van Ditmarsch et al., 2007]. One of the most influential framework in this field has been proposed by Baltag, Moss and Solecki (to which we refer by the term BMS, [Baltag et al., 1998, Baltag and Moss, 2004]). Their insight is to represent the agents' beliefs about an event occurring completely similarly to the way the agents' beliefs about the static situation are represented: by means of a Kripke model. They then propose an update operation between these two Kripke models (one representing the initial situation and one representing the event) which yields a new Kripke model representing the agents' beliefs about the situation after the event has taken place. However, the events considered there are assumed to be instantaneous, at least from a formal point of view. This is a strong idealization because very often in everyday life, events take time: "a tub is being filled", "Ann is going to her office", "a computer program is running"... In that case we might talk of processes instead of (lasting) events, although we will use the general term event throughout the paper. Besides, the BMS language can only express statements about what is true before or after an event occurs and not *while* an event is occurring. Moreover, it can neither express that an event is currently occurring nor express some static properties about the world together with the fact that an event is occurring, such as: "the tub is not full *and* it is being filled". Actually these kinds of statement are widespread in natural languages, and it seems natural to expect from a logical framework to be able to express them if one wants for example to formally represent a given situation or talk in an abstract way about ongoing computation processes and programs.

Besides, this idealization precludes the logical study of important properties of the dynamics of beliefs. Indeed, it hides the fact that the agents' beliefs about *events/processes*, and not only about the static situation, can also change over time due to other events (in which they are temporally included). For example, assume that Ann and Bob do not know whether tub 1 or tub 2 is being filled. This (lasting) event can be described by a first event model. Now assume that one privately tells Bob that tub 1 is actually being filled. This new event triggers an update of the initial *event* so that Bob knows that tub 1 is being filled whereas Ann still does not know whether tub 1 or tub 2 is being filled. Formally, as we will see, this creates a kind of hierarchy among events.

The aim of this paper is to give a formal account of these phenomena by extending and refining the BMS framework, and to propose a unified language which can express statements of the kind above. The paper is organized as follows. In Section 1, we briefly recall and review the BMS framework. In Section 2, we propose a new definition of event models together with a simple and natural language for them. In Section 3, we propose a generic product update between event models which generalizes the BMS up-

*The proofs of this paper can be found at the following address: `http://publications.uni.lu/record/2457/files/TARK2009BIS.PDF`.

TARK '09, July 6-8, 2009, California
ISBN: 978-1-60558-560-4...$10.00

date product. In Section 4, we propose a general dynamic language that can express statements about the situation as well as the current events occurring in this situation. We then axiomatize it and show that the BMS system can be embedded in our framework if we add common belief operators. Finally, in Section 5 we compare our framework with related works and notably with process logics.

1 The BMS framework

Let Φ be a finite set of propositional letters also called atomic facts and let G be a finite set of agents.

Epistemic models are tuples of the form $M = (W, R, V)$, where W is a non-empty set of possible worlds, $V : \Phi \to 2^W$ a valuation and $R : G \to 2^{W\times W}$ assigns an accessibility relation to each agent. We write $R_j = R(j)$ and $R_j(w) = \{w' \in W \mid R_j(w, w')\}$. When we have $v \in R_j(w)$ then in world w agent j considers world v as being possible. The epistemic language for epistemic models is defined as follows:

$$\mathcal{L}^e : \varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid B_j\varphi \mid C_G\varphi$$

where p ranges over Φ and j over G. $B_j\varphi$ reads 'agents j believes φ' and $C_G\varphi$ reads 'it is common belief among the agents G that φ is true'. The degree of a formula without common belief $deg(\varphi)$ is defined inductively as usual.[1] The truth conditions for this language are defined inductively as follows. Let $w \in W$. $M, w \models p$ iff $w \in V(p)$; $M, w \models \neg\varphi$ iff not $M, w \models \varphi$; $M, w \models \varphi \wedge \varphi'$ iff $M, w \models \varphi$ and $M, w \models \varphi'$; $M, w \models B_j\varphi$ iff for all $v \in R_j(w)$ $M, v \models \varphi$; $M, w \models C_G\varphi$ iff for all $v \in \left(\underset{j\in G}{\cup} R_j\right)^+ (w)$ $M, v \models \varphi$.[2] See [Fagin et al., 1995] for details.

Example 1.1. ('tub' example) Assume there are two tubs and two agents Ann and Bob. They both know that at least one tub is *not* full but they do not know which one and this is even common belief. Tub 2 is actually full but tub 1 is not. This situation is depicted in the epistemic model (M^0, w^0_a) of Figure 1. The boxed world w^0_a represents the actual world. The accessibility relations are represented by arrows indexed by A (standing for Ann) or B (standing for Bob). The propositional letter p^0 (resp. q^0) stands for 'tub 2 (resp. tub 1) is full'. So we have $M^0, w^0_a \models C_G(\neg p^0 \vee \neg q^0)$: 'it is common belief among Ann and Bob that at least one tub is not full'. ◄

Event models are very similar to epistemic models and are of the form $A = (E, R, Pre, Post)$, where E is a finite and non-empty set, $Pre : E \to \mathcal{L}$, $Post : \Phi \times E \to \mathcal{L}$ and $R : G \to 2^{W\times W}$ are functions. When we have $b \in R_j(a)$ then the occurrence of a is perceived by agent j as being possibly the occurrence of b. Informally, $Pre(a)$ is the precondition that a world must fulfill so that possible event a can take place in this world. For example $Pre(a) = \top$ means that event a can take place in any world. $Post(p, a)$ specifies which conditions a possible world should fulfill so that propositional letter p is true in the resulting world after event a has occurred (this function was originally introduced in [van Benthem et al., 2006, van Ditmarsch et al., 2005]). However, note that unlike epistemic models, there is no valuation and also no (natural) language for event models to describe and talk about events.

[1] $deg(p) = 0, deg(\neg\varphi) = deg(\varphi), deg(\varphi \wedge \varphi') = max\{deg(\varphi), deg(\varphi')\}, deg(B_j\varphi) = deg(\varphi) + 1$.

[2] If R is a relation, we define $R^+(w) = \{v \mid$ there is $w = w_1, \ldots, w_n = v$ such that $w_i R w_{i+1}\}$.

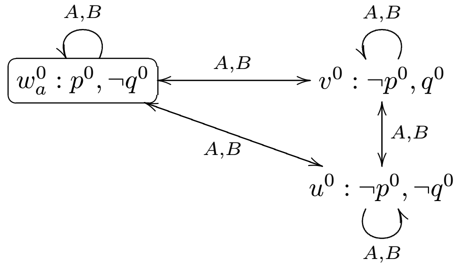

Figure 1: 'tub' example. (M^0, w^0_a)

Product update. Given $M = (W, R, V)$ and $A = (E, R, Pre, Post)$, their *product update* $M \otimes A = (W', R', V')$ is an epistemic model describing the new situation after the event described by A occurred in the situation described by M. The new set of possible worlds is $W' = \{(w, a) \mid M, w \models Pre(a)\}$, the new valuation is $V'(p) = \{(w, a) \mid M, w \models Post(p, w)\}$, and the new accessibility relation is defined by $(v, b) \in R_j(w, a)$ iff $v \in R_j(w)$ and $b \in R_j(a)$.

The BMS language $\mathcal{L}_{BMS}(A)$ is inspired from the one of Propositional Dynamic Logic (PDL) [Pratt, 1976, Harel et al., 2000] and takes as argument an event model A. It is just the epistemic one enriched with a new modality $[A, a]\varphi$ which reads 'after any execution of event a, φ is true'. Its truth condition is as follows:

$$M, w \models [A, a]\varphi \text{ iff}$$
$$M, w \models Pre(a) \text{ implies } M \otimes A, (w, a) \models \varphi$$

Note that the event model A, which a priori is a semantic object, is given in the very definition of the syntax of the language.

2 Languages for event models

In this section we are going to restore the symmetry between epistemic and event models.

2.1 Syntax

Let $\Phi^0, \ldots, \Phi^N$ be finite and disjoint sets of propositional letters.

Definition 2.1. Let $i \in \{0, \ldots, N\}$. The language $\mathcal{L}^i$ is defined inductively as follows

$$\mathcal{L}^i : \varphi^i ::= p^i \mid \neg\varphi^i \mid \varphi^i \wedge \varphi^i \mid B_j\varphi^i$$

where p^i ranges over Φ^i and j over G. $\langle B_j \rangle\varphi^i$ abbreviates $\neg B_j \neg\varphi^i$. $E\varphi^i$ abbreviates $\bigwedge_{j \in G} B_j\varphi$ and $E^n\varphi^i$ is defined inductively by $E^0\varphi^i = \varphi^i$ and $E^{n+1}\varphi^i = EE^n\varphi^i$. We also note $\mathcal{L}^i_n = \{\varphi^i \in \mathcal{L}^i \mid deg(\varphi^i) \leq n\}$ and by notation, $\varphi^i \in \mathcal{L}^i$ for all $i \in \{0, \ldots, N\}$.

The propositional letters $p^i \in \Phi^i$ for $i \geq 1$ are called *atomic events* (of type i) and the propositional letters $p^0 \in \Phi^0$ are called *atomic facts*. ◀

Language $\mathcal{L}^0$ corresponds to the classical epistemic language $\mathcal{L}^e$ of Section 1 (without common belief). The other languages $\mathcal{L}^i$ for $i \geq 1$ are used to describe (types of) events. Atomic events p^i for $i \geq 1$ describe events, just as atomic facts p^0 describe static properties of the world. For example p^1 = 'Ann shows her red card to Bob', p^2 = 'one truthfully announces that tub 2 is being filled', r^3 = 'Claire is observing Ann observing Bob opening the box'... Generally, atomic events are of the form 'something is happening', 'somebody *is doing* something' whereas atomic facts are of the form 'something has this static property'. Besides, the occurrence of these atomic events might change some properties of the world, unlike atomic facts. The negation $\neg p^i$ of an atomic event p^i should be interpreted as 'the atomic event p^i is not occurring'. However, this does not mean that another 'opposite' event is necessarily occurring.

Moreover, these atomic events might have preconditions. For example, the precondition that 'Ann shows her red card to Bob' (p^1) is that 'Ann has the red card' (r_A): $Pre(p^1) = r_A$. The precondition that 'one truthfully announces that tub 2 is being filled' (p^2) is that 'tub 2 is being filled' (p^1): $Pre(p^2) = p^1$. The precondition that 'Claire is observing Ann observing Bob opening the box' (r^3) is that 'Ann is observing Bob opening the box' (r^2) whose precondition is that 'Bob is opening the box' (r^1): $Pre(r^3) = r^2$ and $Pre(r^2) = r^1$. Note that in these last two examples the preconditions of (atomic) events are also events. This motivates our introduction of different types of events and this also leads us to introduce a precondition function which assigns to every atomic event p^i a formula of $\mathcal{L}^k$, for some $k \neq i$.

Definition 2.2. $Pre : \Phi^1 \cup \ldots \cup \Phi^N \to \mathcal{L}^0 \cup \ldots \cup \mathcal{L}^N$ is a function such that for all $i \geq 1$, there is a unique $k \neq i$ such that for all $p^i \in \Phi^i$, $Pre(p^i) \in \mathcal{L}^k$.

In that case, we (abusively) write $Pre(i) = k$ or $i \in Pre^{-1}(k)$. So $(\{0, \ldots, N\}, Pre^{-1})$ is a directed graph and we assume in this paper that it is a rooted tree with root 0. ◀

Note that because the atomic events of Φ^i are supposed to describe a particular type of event i, we assume that their preconditions should deal with the same type of event k (or with properties of the world) described by some $\mathcal{L}^k$. If this is not the case then the set Φ^i should be split up in subsets each dealing with a more specific type of event.

Moreover, the occurrence of atomic events might change the truth value of some atomic facts or of some other atomic events. For instance, the occurrence of the atomic event q^1='tub 1 is being filled' affects the atomic fact q^0='tub 1 is full': after the occurrence of q^1, the atomic fact q^0 is true. Likewise, pressing on a button b might trigger the filling of tub 2 (even if tub 1 is already being filled). So after the occurrence of the atomic event r^2='Ann presses button b' the atomic event p^1='tub 2 is being filled' is true. This leads us to introduce a postcondition function which specifies some *sufficient* conditions for a propositional letter to be true in case an atomic event occurs.

Definition 2.3. For all $i \in \{1, \ldots, N\}$ and $k \in \{0, \ldots, N\}$ such that $Pre(i) = k$, we define a function $Post(i, k) : \Phi^k \times \Phi^i \to \mathcal{L}^k$. $Post(i, k)$ is abusively written $Post$. ◀

$Post(p^k, p^i)$ is a sufficient condition *before* the occurrence of p^i for p^k to be true after the occurrence of p^i. So in the tub example, $Post(q^0, p^1) = \top$ and $Post(p^0, p^1) = p^0$, where we recall that p^0='tub 2 is full'.

2.2 Semantics

We are now ready to define a semantics for this hierarchy of languages.

Definition 2.4. Let $i \in \{0, \ldots, N\}$. A $\mathcal{L}^i$*-model* M^i is a triple $M^i = (W^i, R^i, V^i)$ such that

- W^i is a non-empty set of possible worlds;
- $R^i : G \to 2^{W^i \times W^i}$ assigns an accessibility relation to each agent;
- $V^i : \Phi^i \to 2^{W^i}$ assigns a set of possible worlds to each propositional letter.

We write $w^i \in M^i$ for $w^i \in W^i$ and (M^i, w^i) is called a *pointed* $\mathcal{L}^i$*-model*. ◀

So a $\mathcal{L}^i$-model is just an epistemic model where the set of propositional letters is Φ^i. The truth conditions are also identical to the ones of epistemic logic:

Definition 2.5. Let $i \in \{0, \ldots, N\}$. Let M^i be a $\mathcal{L}^i$-model, $w^i \in M^i$ and $\varphi^i \in \mathcal{L}^i$. $M^i, w^i \models \varphi^i$ is defined inductively as follows:

$$\begin{array}{lll} M^i, w^i \models p^i & \text{iff} & w^i \in V(p^i) \\ M^i, w^i \models \neg\varphi^i & \text{iff} & \text{not } M^i, w^i \models \varphi^i \\ M^i, w^i \models \varphi^i \wedge \psi^i & \text{iff} & M^i, w^i \models \varphi^i \text{ and } M^i, w^i \models \psi^i \\ M^i, w^i \models B_j\varphi^i & \text{iff} & \text{for all } v^i \in R_j(w^i), M^i, v^i \models \varphi^i \end{array}$$

We write $M^i \models \varphi^i$ when $M^i, w^i \models \varphi^i$ for all $w^i \in M^i$, and $\models^i \varphi^i$ when for all $\mathcal{L}^i$-model M^i, $M^i \models \varphi^i$. ◀

So the $\mathcal{L}^i$-models are free of the precondition and postcondition functions Pre and $Post$ that were present in the definition of event models. However, given a $\mathcal{L}^i$-model M^i and $w^i \in M^i$, we can get back the usual preconditions and postconditions $Pre(w^i)$ and $Post(p, w^i)$ of event models:

Definition 2.6. Let $i \in \{1, \dots, N\}$, $k = Pre(i)$ and $p^k \in \Phi^k$. Let M^i be a $\mathcal{L}^i$-model and $w^i \in M^i$. $Pre(w^i)$ and $Post(p^k, w^i)$ are defined as follows.

- $Pre(w^i) = \bigwedge\{Pre(p^i) \mid M^i, w^i \models p^i\}$;
- $Post(p^k, w^i) = \begin{cases} \bigvee\{Post(p^k, p^i) \mid M^i, w^i \models p^i\} \\ \text{if } M^i, w^i \models p^i \text{ for some } p^i \in \Phi^i \\ p^k \text{ otherwise.} \end{cases}$

◀

For $Pre(w^i)$, we take the conjunction of the relevant $Pre(p^i)$s because these are *necessary* conditions for the possible event w^i to take place. On the other hand, for $Post(p^k, w^i)$ we take the disjunction of the relevant $Post(p^k, p^i)$s because these are *sufficient* conditions for p^k to be true after the occurrence of w^i. Besides, if w^i is the event where nothing happens, i.e. $M^i, w^i \models \neg p^i$ for all $p^i \in \Phi^i$, then the truth values of the p^ks should not change.

Finally we introduce a particular kind of $\mathcal{L}^i$-model which will be used in Section 4. For $i \in \{1, \dots, N\}$, we define $M^{i,\emptyset} = (\{w^{i,\emptyset}\}, R^{i,\emptyset}, V^{i,\emptyset})$ where $V^{i,\emptyset}(p^i) = \emptyset$ for all $p^i \in \Phi^i$, and $R_j^{i,\emptyset}(w^{i,\emptyset}) = \{w^{i,\emptyset}\}$ for all $j \in G$. So $M^{i,\emptyset}$ represents the event whereby nothing happens and this is common belief among the agents.

2.3 Axiomatization

The axiomatization for the class of $\mathcal{L}^i$-models is the same as the one for epistemic models.

Definition 2.7. Let $i \in \{0, \dots, N\}$. The logic L^i for the language $\mathcal{L}^i$ is defined by the following axiom schemes and inference rules. We write $\vdash^i \varphi^i$ for $\varphi^i \in \mathsf{L}^i$.

Taut	All propositional axiom schemes and inference rules
K^i	$\vdash^i B_j(\varphi^i \to \psi^i) \to (B_j\varphi^i \to B_j\psi^i)$ for all $j \in G$
Nec^i	If $\vdash^i \varphi^i$ then $\vdash^i B_j\varphi^i$ for all $j \in G$

◀

Theorem 2.8 ([Fagin et al., 1995])**.** *Let $i \in \{0, \dots, N\}$. For all $\varphi^i \in \mathcal{L}^i$, $\models^i \varphi^i$ iff $\vdash^i \varphi^i$. Besides, L^i is decidable.*

2.4 Examples

Example 2.9. ('card' example) This example shows that possible events of event models can be the combination of more elementary atomic events. Assume Ann, Bob and Claire play a card game with three cards: a red one, a green one and a yellow one. They have only one card and they only know the color of their cards. Ann has the red card, Bob the green card and Claire the yellow one. Then Ann and Bob show their card privately to each other in front of Claire who therefore does not know which card they show to each other. We model this example by introducing the atomic facts $\Phi^0 = \{AhR, AhG, AhY, BhR, BhG, BhY\}$ and the atomic events $\Phi^1 = \{AsR, AsG, BsG, BsG\}$. AhR stands for 'Ann has the Red card', AhG for 'Ann has the Green card',... and so on. AsR stands for 'Ann shows her Red card', BsG stands for 'Bob shows his Green card',... and so on. $Pre(1) = 0$ and $Pre(AsR) = AhR, Pre(AsG) = AhG, Pre(BsG) = BhG, Pre(BsG) = BhG$. Finally, $Post(p^0, p^1) = p^0$ for all $p^0 \in \Phi^0$ and $p^1 \in \Phi^1$ because these atomic events do not change atomic facts of the world (also called *epistemic events* in [Baltag and Moss, 2004]). The event of Ann and Bob showing privately their card to each other in front of Claire is depicted in Figure 2.

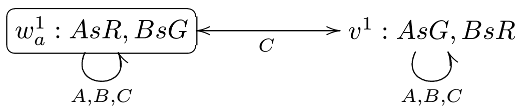

Figure 2: Ann and Bob show their cards to each other privately in front of Claire.

Applying Definition 2.6, we then obtain the usual preconditions and postconditions: $Pre(w_a^1) = Pre(AsR) \wedge Pre(BsG) = AhR \wedge BhG; Pre(v^1) = Pre(AsG) \wedge Pre(BsR) = AhG \wedge BhR; Post(p, w_a^1) = Post(p, v^1) = p$ for all $p \in \Phi^0$. ◀

Example 2.10. ('tub' example) Let $\Phi^0 = \{p^0, q^0\}$, $\Phi^1 = \{p^1, q^1\}$, $\Phi^2 = \{p^2\}$. p^0 stands for 'tub 2 is full' and q^0 for 'tub 1 is full'. p^1 stands for 'tub 2 is being filled' and q^1 for 'tub 1 is being filled'. p^2 stands for 'one truthfully announces that tub 1 is being filled'. $Pre(1) = 0$ and $Pre(2) = 1$. $Pre(p^1) = \neg p^0$, $Pre(q^1) = \neg q^0$. $Pre(p^2) = q^1$. We have $Post(p^0, p^1) = \top$, $Post(q^0, q^1) = \top$ and $Post(p^0, q^1) = p^0$, $Post(q^0, p^1) = q^0$. We also have $Post(p^1, p^2) = p^1$ and $Post(q^1, p^2) = q^1$. In Figure 3 (*up*) is depicted the $\mathcal{L}^1$-model (M^1, w_a^1) representing the event whereby tub 1 is being filled but the agents do not know wether it is tub 1 or tub 2 which is being filled: $M^1, w_a^1 \models q^1 \wedge (B_A(q^1 \leftrightarrow \neg p^1) \wedge \langle B_A\rangle p^1 \wedge \langle B_A\rangle q^1) \wedge (B_B(q^1 \leftrightarrow \neg p^1) \wedge \langle B_B\rangle p^1 \wedge \langle B_B\rangle\, q^1)$. In Fig-

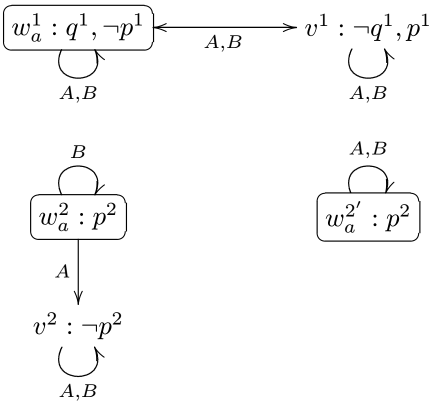

Figure 3: (*up*) One of the tubs is being filled (M^1, w^1_a); (*down left*) one privately informs Bob that tub 1 is being filled (M^2, w^2_a) and (*down right*) one publicly announces that tub 1 is being filled $(M^{2'}, w^{2'}_a)$.

ure 3 (*down left*) is depicted the $\mathcal{L}^2$-model (M^2, w^2_a) representing the event where one privately informs Bob that tub 1 is being filled, Ann suspecting nothing about it. So we have $M^2, w^2_a \models p^2 \wedge B_B p^2 \wedge B_A \neg p^2$ which somehow defines formally the notion of privacy: something happens and agent B knows it but agent A believes it does not happen. In Figure 3 (*down right*) is depicted the $\mathcal{L}^2$-model $(M^{2'}, w^{2'}_a)$ representing the event where one publicly informs Ann and Bob that tub 1 is being filled. So we have $M^{2'}, w^{2'}_a \models p^2 \wedge B_A p^2 \wedge B_B p^2$ which somehow defines formally the notion of publicness: something happens and everybody knows it happens. ◀

3 A generic product update

As we said in the introduction, because the events we consider might be processes, it is quite possible that an event represented by M^k be updated by another event represented by M^i. This gives rise to a generic product update between $\mathcal{L}^i$-models whose definition is very similar to the BMS one of Section 1.

Definition 3.1. Let $i \in \{1, \dots, N\}$ and $k = Pre(i)$. Let $M^i = (W^i, R^i, V^i, w^i_a)$ be a pointed $\mathcal{L}^i$-model and $M^k = (W^k, R^k, V^k, w^k_a)$ be a pointed $\mathcal{L}^k$-model such that $M^k, w^k_a \models Pre(w^i_a)$. We define the pointed $\mathcal{L}^k$-model $(M^k, w^k_a) \otimes (M^i, w^i_a) = (W', R', V', w'_a)$ as follows.

1. $W' = \{(w^k, w^i) \mid M^k, w^k \models Pre(w^i)\}$;
2. $(v^k, v^i) \in R'_j(w^k, w^i)$ iff $v^k \in R^k_j(w^k)$ and $v^i \in R^i_j(w^i)$;
3. $V'(p^k) = \{(w^k, w^i) \mid M^k, w^k \models Post(p^k, w^i)\}$;
4. $w'_a = (w^k_a, w^i_a)$.

◀

Example 3.2. ('tub' example) In Figure 4 is depicted the product update of the models (M^1, w^1_a) and (M^2, w^2_a) (*up*) and (M^1, w^1_a) and $(M^{2'}, w^{2'}_a)$ (*down*) of Figure 3. So we have $(M^1, w^1_a) \otimes (M^2, w^2_a) \models (q^1 \wedge B_B q^1) \wedge (B_A(q^1 \leftrightarrow \neg p^1) \wedge \langle B_A \rangle p^1 \wedge \langle B_A \rangle q^1) \wedge B_A \left(B_B(q^1 \leftrightarrow \neg p^1) \wedge \langle B_B \rangle p^1 \wedge \langle B_B \rangle q^1 \right)$: Bob knows that tub 1 is being filled whereas Ann does not know whether tub 1 or tub 2 is being filled and believes that Bob does not know neither. We also have $(M^1, w^1_a) \otimes (M^{2'}, w^{2'}_a) \models q^1 \wedge B_A q^1 \wedge B_B q^1$: both Ann and Bob know that tub 1 is being filled. ◀

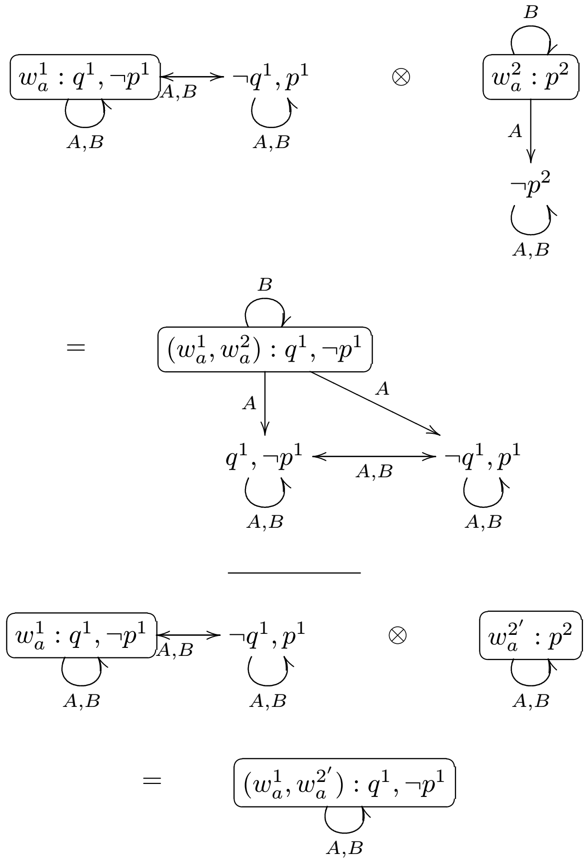

Figure 4: (*up*) Product update for the private announcement to Bob that tub 1 is being filled. (*down*) Product update for the public announcement that tub 1 is being filled.

4 A general language

Definition 4.1. The language $\mathcal{L}$ is defined inductively as follows.

$$\mathcal{L} : \varphi ::= \top^k \mid \varphi^k \mid \neg\varphi \mid \varphi \wedge \varphi \mid [i\ ends]\varphi \mid [i\ starts]\varphi$$

where k ranges over $\{0, \dots, N\}$, φ^k over $\mathcal{L}^k$ and i over $\{1, \dots, N\}$. As usual, $\langle i\ ends \rangle \varphi$ abbreviates $\neg[i\ ends]\neg\varphi$ and $\langle i\ starts \rangle \varphi$ abbreviates $\neg[i\ starts]\neg\varphi$.

The language $\mathcal{L}^{St}$ is the language $\mathcal{L}$ without the operators $[i\ ends]$ and $[i\ starts]$. ◀

$\top^k$ reads 'an event of type k is occurring', $[i\ ends]\varphi$ reads 'φ holds after an event of type i ends', and $[i\ starts]\varphi$ reads 'φ holds when a new event of type i starts'.

We extend the function Pre to $\mathcal{T} = \{\top^k \mid k \in \{0, \ldots, N\}\}$ by stating $Pre(\top^i) = \top^k$ when $Pre(i) = k$.

4.1 The 'static' part: $\mathcal{L}^{St}$

4.1.1 Semantics

Definition 4.2. A $\mathcal{L}^{St}$*-model* $\mathcal{M} = \{(M^0, w^0), \ldots, (M^n, w^n)\}$ is a non-empty set of pointed $\mathcal{L}^i$-models (M^i, w^i) such that for all pointed $\mathcal{L}^i$-model $(M^i, w^i) \in \mathcal{M}$ (with $i \geq 1$),

1. there exists a unique pointed $\mathcal{L}^k$-model $(M^k, w^k) \in \mathcal{M}$ with $k = Pre(i)$ such that $M^k, w^k \models Pre(w^i)$,
2. there is at most one pointed $\mathcal{L}^l$-model $(M^l, w^l) \in \mathcal{M}$ with $i = Pre(l)$.

By notation, $(M^i, w^i) \in \mathcal{M}$ is supposed to be a pointed $\mathcal{L}^i$-model. ◀

A $\mathcal{L}^{St}$-model models the state of the world at a given time t: each $\mathcal{L}^i$-model (M^i, w^i) of the $\mathcal{L}^{St}$-model (for $i \geq 1$) models an actual event occurring at time t in the actual world and the static properties of this world are modeled by (M^0, w^0).

Definition 4.3. Let $\mathcal{M} = \{(M^0, w^0), \ldots, (M^n, w^n)\}$ be a $\mathcal{L}^{St}$-model and $\varphi^{St} \in \mathcal{L}^{St}$. $\mathcal{M} \models \varphi^{St}$ is defined inductively as follows.

$$\begin{array}{lll}
\mathcal{M} \models \top^i & \text{iff} & \text{there is } (M^i, w^i) \in \mathcal{M} \\
\mathcal{M} \models \varphi^i & \text{iff} & \begin{cases} M^i, w^i \models \varphi^i \\ \text{if there is } (M^i, w^i) \in \mathcal{M} \\ M^{i,\emptyset}, w^{i,\emptyset} \models \varphi^i \\ \text{otherwise} \end{cases} \\
\mathcal{M} \models \neg\varphi & \text{iff} & \text{not } \mathcal{M} \models \varphi \\
\mathcal{M} \models \varphi \wedge \varphi' & \text{iff} & \mathcal{M} \models \varphi \text{ and } \mathcal{M} \models \varphi'.
\end{array}$$

◀

If there is no $\mathcal{L}^i$-model in $\mathcal{M}$ this means that no event of type i is occurring and the agents all know that, i.e. that the event modeled by the $\mathcal{L}^i$-model $(M^{i,\emptyset}, w^{i,\emptyset})$ is occurring (defined in Section 2.2). That is why in that case the truth value of a formula $\varphi^i \in \mathcal{L}^i$ is determined by $(M^{i,\emptyset}, w^{i,\emptyset})$. Note that it is quite possible that a $\mathcal{L}^i$-model in $\mathcal{M}$ is bisimilar to $(M^{i,\emptyset}, w^{i,\emptyset})$ (i.e. contains the same information as $(M^{i,\emptyset}, w^{i,\emptyset})$). In that case we still have that $\mathcal{M} \models \top^i$ although no genuine event of type i is occurring. But because this is a very marginal case, we prefer to keep the intuitive reading of $\top^i$ as 'an event of type i is occurring'.

Example 4.4. ('tub' example) In Figure 5 is depicted the $\mathcal{L}^{St}$-model $\mathcal{M} = \{(M^0, w_a^0), (M^1, w_a^1), (M^2, w_a^2)\}$. So we have $\mathcal{M} \models [\neg q^0 \wedge \neg B_A q^0 \wedge \neg B_B q^0] \wedge [q^1 \wedge B_A(q^1 \leftrightarrow \neg p^1) \wedge \langle B_A \rangle p^1 \wedge \langle B_A \rangle q^1 \wedge B_B(q^1 \leftrightarrow \neg p^1) \wedge \langle B_B \rangle p^1 \wedge \langle B_B \rangle q^1)] \wedge (p^2 \wedge B_A \neg p^2)$: tub 1 is not full but Ann and Bob do not know it, and tub 1 is being filled but Ann and Bob do not know wether tub 1 or tub 2 is being filled, and one informs Bob that tub 1 is being filled but Ann believes that nothing happens. So our language allows us to express at the same time statements about static properties of the world and about events occurring in this world. ◀

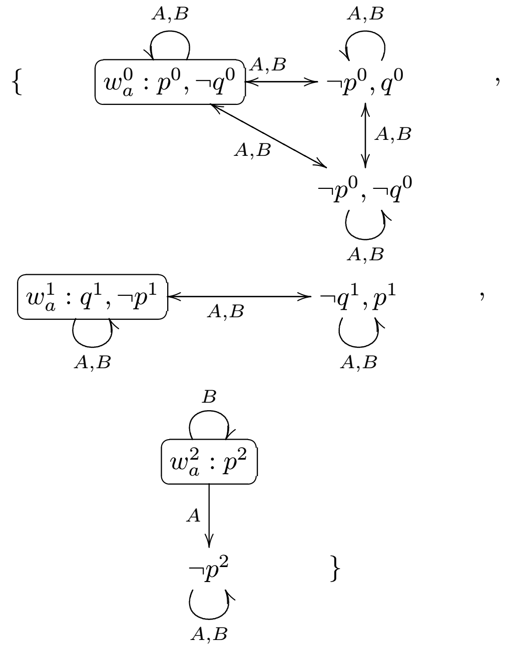

Figure 5: A $\mathcal{L}$-model: tub 2 is full, tub 1 is being filled and one privately informs Bob that this happens.

Some notations. Let $\mathcal{M} = \{(M^0, w^0), \ldots, (M^n, w^n)\}$ be a $\mathcal{L}^{St}$-model and let $(M^i, w^i) \in \mathcal{M}$ (with $i \geq 1$). $Pre_{\mathcal{M}}(M^i, w^i)$ is the unique $\mathcal{L}^k$-model $(M^k, w^k) \in \mathcal{M}$ such that $k = Pre(i)$. Finally for $i \in \{0, \ldots, N\}$, we define $last(i) = \top^i \wedge \bigwedge_{l \in Pre^{-1}(i)} \neg\top^l$. So we have $\mathcal{M} \models last(i)$ iff there is $(M^i, w^i) \in \mathcal{M}$ and there is no $(M^l, w^l) \in \mathcal{M}$ such that $Pre_{\mathcal{M}}(M^l, w^l) = (M^i, w^i)$. $last(i)$ for $i \geq 1$ reads 'the *last* event which occurred and which is still occurring is of type i'. $last(0)$ reads 'no event is occurring'.

Definition 4.5. Let $\mathcal{M} = \{(M^0, w^0), \ldots, (M^n, w^n)\}$ be a $\mathcal{L}^{St}$-model such that $\mathcal{M} \models last(n)$. We define $\otimes(\mathcal{M})$ by $\otimes(\mathcal{M}) = \mathcal{M}$ if $n = 0$ and $\otimes(\mathcal{M}) = \{(M^0, w^0), \ldots, Pre_{\mathcal{M}}(M^n, w^n) \otimes (M^n, w^n)\}$ otherwise. ◀

So $\otimes(\mathcal{M})$ is just $\mathcal{M}$ updated by the most recent event when this one ends.

Example 4.6. ('tub' example) If we take up the $\mathcal{L}$-model $\mathcal{M}$ of Example 4.4 then $\otimes(\mathcal{M}) = \{(M^0, w_a^0), (M^1, w_a^1) \otimes (M^2, w_a^2)\}$ where $(M^1, w_a^1) \otimes (M^2, w_a^2)$ is depicted in Figure 4. ◀

However, because the product update might change truth values of atomic events, the preconditions of the possible events might change during an update. So even if $\mathcal{M}$ is a $\mathcal{L}^{St}$ model, $\otimes(\mathcal{M})$ is not necessarily a $\mathcal{L}^{St}$-model. This leads us to define the notion of $\mathcal{L}$-model.

Definition 4.7. A $\mathcal{L}$-*model* is a $\mathcal{L}^{St}$-model which is stable under $\otimes$, i.e. a $\mathcal{L}^{St}$-model $\mathcal{M}$ such that $\otimes(\mathcal{M})$ is a $\mathcal{L}$-model. ◀

We are now going to determine under which conditions a $\mathcal{L}^{St}$ model is a $\mathcal{L}$-model.

Definition 4.8. Let $p^i \in \Phi^0 \cup \ldots \cup \Phi^N$. $Post(p^i)$ is defined inductively as follows.

- $Post(p^0) = \top$;
- $Post(p^i) = \bigwedge_{p^k \in \Phi^k} (Post(p^k, p^i) \to (Pre(p^k) \wedge Post(p^k)))$ if $i \geq 1$ and $k = Pre(i)$.

Then $Post^i$ is defined inductively as follows.

- $Post^0 = \top$;
- $Post^i = \bigwedge_{p^i \in \Phi^i} \left(p^i \to Post(p^i)\right) \wedge (\bigwedge_{p^i \in \Phi^i} \neg p^i \to Post^k)$ if $i \geq 1$ and $k = Pre(i)$.

Finally we define $Post$ and Pre.

- $Post = \bigwedge_{i \in \{0,\ldots,N\}} (last(i) \to Post^i)$;
- $Pre = \bigwedge_{p \in \Phi^0 \cup \ldots \cup \Phi^N \cup \mathcal{T}} (p \to Pre(p))$.

◀

Pre characterizes condition 1 of Definition 4.2. $Post(p^i)$ is a necessary condition for a $\mathcal{L}^{St}$-model $\mathcal{M}$ to be a $\mathcal{L}$-model in case $\mathcal{M} \models p^i \wedge last(i)$.

Proposition 4.9. *Let $\mathcal{M}$ be a $\mathcal{L}^{St}$-model. $\mathcal{M}$ is a $\mathcal{L}$-model iff $\mathcal{M} \models Post$.*

4.1.2 Axiomatization

Let $\varphi \in \mathcal{L}^{St}$. We write $\models \varphi$ when for all $\mathcal{L}$-model $\mathcal{M}$, $\mathcal{M} \models \varphi$.

Definition 4.10. The logic L^{St} for the language $\mathcal{L}^{St}$ is defined by the following axiom schemes and inference rules. We write $\vdash^{St} \varphi$ for $\varphi \in \mathsf{L}^{St}$.

L^i	All axiom schemas and inference rules of L^i for all $i \in \{0, \ldots, N\}$	
A_1	$\vdash^{St} \neg last(0) \to \bigvee_{i \in \{1,\ldots,N\}} last(i)$	
A_2	$\vdash^{St} last(i) \to \neg last(i')$	for all $i \neq i'$
A_3	$\vdash^{St} \neg \top^i \to E^n \left(\neg p^i \wedge \langle B_j \rangle \neg p^i\right)$	for all $n \in \mathbb{N}$
A_4	$\vdash^{St} Pre \wedge Post$	

◀

Axiom A_1 expresses that if at least one event is occurring then one of these events is the most recent. Axiom schema A_2 characterizes condition 2 of Definition 4.2 and expresses that there is a unique most recent event. Axiom scheme A_3 characterizes the special event of type i $(M^{i,\emptyset}, w^{i,\emptyset})$ where nothing happens and this is common knowledge.

Theorem 4.11. *For all $\varphi^{St} \in \mathcal{L}^{St}$, $\models \varphi^{St}$ iff $\vdash^{St} \varphi^{St}$.*

Theorem 4.12. *L^{St} is decidable.*

4.2 Adding dynamics: $\mathcal{L}$

4.2.1 Semantics

Definition 4.13. Let $i \in \{1, \ldots, N\}$. The relations R^i_{ends} and R^i_{starts} on $\mathcal{L}$-models are defined as follows. Let $\mathcal{M}$ and $\mathcal{M}'$ be two $\mathcal{L}$-models.

- $\mathcal{M}' \in R^i_{ends}(\mathcal{M})$ iff there is $(M^i, w^i) \in \mathcal{M}$ such that $\begin{cases} \mathcal{M}' = \otimes(\mathcal{M}) \\ \text{if } \mathcal{M} \models last(i); \\ \mathcal{M}' \in R^l_{ends} \circ R^i_{ends}(\mathcal{M}) \\ \text{where } Pre_{\mathcal{M}}(M^l, w^l) = (M^i, w^i)\text{, otherwise.} \end{cases}$
- $\mathcal{M}' \in R^i_{starts}(\mathcal{M})$ iff there is a pointed $\mathcal{L}^i$-model (M^i, w^i) such that $\mathcal{M}' = \mathcal{M} \cup \{(M^i, w^i)\}$.

Let $\varphi \in \mathcal{L}$. $\mathcal{M} \models \varphi$ is defined inductively as follows. The boolean cases are as in Definition 4.3.

$$\mathcal{M} \models [i\ ends]\varphi \quad \text{iff} \quad \text{for all } \mathcal{M}' \in R^i_{ends}(\mathcal{M}),\ \mathcal{M}' \models \varphi$$
$$\mathcal{M} \models [i\ starts]\varphi \quad \text{iff} \quad \text{for all } \mathcal{M}' \in R^i_{starts}(\mathcal{M}),\ \mathcal{M}' \models \varphi$$

We write $\models \varphi$ when for all $\mathcal{L}$-model $\mathcal{M}$, $\mathcal{M} \models \varphi$. ◀

If an event of type l presupposes an event of type i, i.e. if $Pre(l) = i$, then if the event of type i ends then the event of type l also ends. For example, if 'Bob is opening a box to look at a coin' (p^1) and 'Ann is observing Bob opening the box' (p^2) then $Pre(p^2) = p^1$. So if Bob stops opening the box to look at the coin ($\neg p^1$), Ann stops observing Bob opening the box ($\neg p^2$). This explains the inductive definition of R^i_{ends}.

$$\{(M^0,w^0),\ldots,(M^k,w^k),(M^i,w^i),\ldots,(M^n,w^n)\}$$

$$\downarrow R^i_{ends}$$

$$\{(M^0,w^0),\ldots,\underbrace{(M^k,w^k)\otimes\ldots\otimes((M^{n-1},w^{n-1})\otimes(M^n,w^n))}_{\text{pointed } L^k\text{-model}}\}$$

$$\{(M^0,w^0),\ldots,(M^n,w^n)\}$$

$$\downarrow R^{n+1}_{starts}$$

$$\{(M^0,w^0),\ldots,(M^n,w^n),(M^{n+1},w^{n+1})\}$$

Note that the above figures (where $k = Pre(i)$) also explain our reading of $last(i)$ introduced in Section 4.1.1.

Example 4.14. ('tub' example) If we take up Example 4.4 then $\mathcal{M} \models [2\ ends](q^1 \wedge B_B q^1 \wedge B_A(q^1 \leftrightarrow \neg p^1) \wedge \langle B_A\rangle p^1 \wedge \langle B_A\rangle q^1)$: after the event of type 2 ends (i.e. after the private announcement to Bob that tub 1 is being filled) Bob knows that tub 1 is being filled while Ann still does not know whether tub 1 or 2 is being filled. We also have $\models [2\ starts](p^2 \wedge B_A p^2 \wedge B_B p^2 \rightarrow [2\ ends](q^1 \wedge B_A q^1 \wedge B_B q^1))$: after any event where one publicly announces that tub 1 is being filled everybody knows that tub 1 is being filled. ◀

4.2.2 Axiomatization

In the BMS axiomatization one needs to refer to the modal structure of the event model, introducing it henceforth directly into the language. In our axiomatization we will also need to refer to it. However, we will do so thanks to our languages $\mathcal{L}^i$ and more particularly thanks to formulas δ_n, originally introduced in [Balbiani and Herzig, 2007]. These formulas can completely characterize the modal structure of a $\mathcal{L}^i$-model up to modal depth n [Balbiani and Herzig, 2007].

Definition 4.15. [Balbiani and Herzig, 2007] Let $i \in \{0,\ldots,N\}$. We define inductively the sets E^i_n as follows.

- $E^i_0 = \{\bigwedge_{p^i\in S_0} p^i \wedge \bigwedge_{p^i\notin S_0} \neg p^i \mid S_0 \subseteq \Phi^i\}$;
- $E^i_{n+1} = \{\delta_0 \wedge \bigwedge_{j\in G}\left(\bigwedge_{\delta_n\in S^j_n}\langle B_j\rangle\delta_n \wedge B_j \bigvee_{\delta_n\in S^j_n}\delta_n\right) \mid \delta_0 \in E^i_0, S^j_n \subseteq E^i_n\}$.

Let $\delta_{n+1} \in E^i_{n+1}$. δ_{n+1} can be written under the form $\delta_{n+1} = \delta_0 \wedge \bigwedge_{j\in G}\left(\bigwedge_{\delta_n\in S^j_n}\langle B_j\rangle\delta_n \wedge B_j \bigvee_{\delta_n\in S^j_n}\delta_n\right)$.

For all $j \in G$, we note $R_j(\delta_{n+1}) = S^j_n$ and $R_0(\delta_{n+1}) = \{p^i \in \Phi^i \mid\ \vdash^i \delta_0 \rightarrow p^i\}$ ◀

Thanks to these formulas δ_n, we can now express what is true in $M^k \otimes M^i, (w^k, w^i)$ on the basis of what is true in (M^k, w^k) and (M^i, w^i). Intuitively, $Pre^{\delta_n}(\varphi)$ in the next definition is the formula that (M^k, w^k) must satisfy so that φ be true in $(M^k, w^k) \otimes (M^i, w^i)$, in case $M^i, w^i \models \delta_n$.

Definition 4.16. For all $i, k \in \{0,\ldots,N\}$ such that $k = Pre(i)$ we define for all $n \in \mathbb{N}$ the function $Pre : E^i_n \times \mathcal{L}^k_n \rightarrow \mathcal{L}^k_n$ inductively as follows:

- $Pre^{\delta_n}(p^k) = \begin{cases} \bigvee\{Post(p^k,p^i) \mid p^i \in R_0(\delta_n)\} \\ \text{if } R_0(\delta_n) \neq \emptyset \\ p^k \text{ otherwise;}\end{cases}$
- $Pre^{\delta_n}(\varphi \wedge \varphi') = Pre^{\delta_n}(\varphi) \wedge Pre^{\delta_n}(\varphi')$;
- $Pre^{\delta_n}(\neg\varphi) = \neg Pre^{\delta_n}(\varphi)$;
- $Pre^{\delta_n}(B_j\varphi) = \bigwedge_{\delta_{n-1}\in R_j(\delta_n)} B_j((\bigwedge_{p^i\in R_0(\delta_{n-1})} Pre(p^i)) \rightarrow Pre^{\delta_{n-1}}(\varphi))$

◀

Proposition 4.17. *Let $\varphi^k \in \mathcal{L}^k_n$. Let (M^k, w^k) be a pointed $\mathcal{L}^k$-model and (M^i, w^i) be a pointed $\mathcal{L}^i$-model such that $M^k, w^k \models Pre(w^i)$. Let $\delta_n \in E^i_n$.*

If $M^i, w^i \models \delta_n$ then

$M^k, w^k \models Pre^{\delta_n}(\varphi^k)$ *iff* $(M^k, w^k) \otimes (M^i, w^i) \models \varphi^k$.

We are now ready to axiomatize the full language $\mathcal{L}$.

Definition 4.18. The logic L for the language $\mathcal{L}$ is defined by the following axiom schemes and inference rules. We write $\vdash \varphi$ for $\varphi \in \mathsf{L}$. For all $i, k \in \{0,\ldots,N\}$ such that $Pre(i) = k$:

L^{St}	All axiom schemas and inference rules of L^{St}
A_5	$\vdash [i\ ends](last(k))$
A_6	$\vdash [i\ ends]\varphi \leftrightarrow \bigwedge\{last(i_n) \rightarrow [i_n\ ends]\ldots [i_1\ ends][i\ ends]\varphi \mid i = i_0,\ldots,i_n$ and $Pre(i_{l+1}) = i_l\}$
A_7	$\vdash last(i) \rightarrow (\langle i\ ends\rangle\varphi \leftrightarrow [i\ ends]\varphi)$
A_8	$\vdash last(i) \rightarrow ([i\ ends]\varphi^n \leftrightarrow \varphi^n)$ for all $n \neq i, k$
A_9	$\vdash last(i) \rightarrow \left([i\ ends]\varphi^k \leftrightarrow \bigwedge_{\delta_n\in E^i_n}\left(\delta_n \rightarrow Pre^{\delta_n}(\varphi^k)\right)\right)$ for all $\varphi^k \in \mathcal{L}^k_n$ and $n \in \mathbb{N}$
A_{10}	$\vdash [i\ starts]last(i)$
A_{11}	$\vdash \neg last(k) \leftrightarrow [i\ starts]\bot$
A_{12}	$\vdash [i\ starts](t \vee \varphi^0 \vee \ldots \vee \varphi^N) \leftrightarrow (([i\ starts]t) \vee\varphi^0 \vee \ldots \vee ([i\ starts]\varphi^i) \vee \ldots \vee \varphi^N)$ where t is a boolean combination of elements of $\mathcal{T}$
A_{13}	$\vdash last(k) \rightarrow (\langle i\ starts\rangle\varphi^i \leftrightarrow \bigwedge_{\{p^i\in\Phi^i \mid \vdash^{St}\varphi^i\rightarrow p^i\}} Post(p^i) \wedge Pre(p^i))$ for all $\varphi^i \in \mathcal{L}^i$ such that $\neg\varphi^i \notin \mathsf{L}^{St}$
A_{14}	$\vdash [i\ ends](\varphi \rightarrow \psi) \rightarrow ([i\ ends]\varphi \rightarrow [i\ ends]\psi)$

A_{15} $\vdash [i\ starts](\varphi \to \psi) \to ([i\ starts]\varphi \to [i\ starts]\psi)$
R_1 If $\vdash \varphi$ then $\vdash [i\ starts]\varphi$ and $\vdash [i\ ends]\varphi$

◀

Axiom A_{10} expresses that when a new event of type i starts to occur then this event is the most recent one, and similarly for Axiom A_5. Axiom A_{11} expresses that an event of type i can occur if and only if an event of type k is already occurring (and this event is the most recent one). Axiom A_6 captures the fact that when an event ends then this implies that all the other events that depended on this event also end (see Definition 4.13). Axiom A_8 captures the fact that only what is true about an event of type i and about its preconditions are affected when this one ends; and similarly for axiom A_{12}. Axiom A_9 captures Proposition 4.17. Axiom A_{13} expresses that an event satisfying φ^i can occur if and only if the necessary preconditions and postconditions associated to φ^i are fulfilled.

Proposition 4.19. *Let $\varphi \in \mathcal{L}$. Then there is $\varphi^{St} \in \mathcal{L}^{St}$ such that $\vdash \varphi \leftrightarrow \varphi^{St}$.*

Theorem 4.20. *For all $\varphi \in \mathcal{L}$, $\models \varphi$ iff $\vdash \varphi$.*

Theorem 4.21. *L is decidable.*

4.3 Embedding of the BMS framework

We add a common belief operator to our languages $\mathcal{L}^i$ and we assume as in BMS that the $\mathcal{L}^i$-models are finite (for $i \geq 1$). Let $A = (E, R, Pre, Post)$ be an event model with $E = \{a_1, \ldots, a_n\}$. We define the set of atomic events $\Phi^1 = \{p^1_1, \ldots, p^1_n\}$, where $Pre(p^1_i) = Pre(a_i)$ and $Post(p^0, p^1_i) = Post(p^0, a^1_i)$. We define the pointed $\mathcal{L}^1$-model $t(A, a) = (W^1, R^1, V^1, a)$ by $W^1 = E, R^1 = R$ and $V^1(p^1_i) = \{a_i\}$ for all $i \in \{1, \ldots, n\}$. $t(A, a)$ can be characterized[3] by a single formula $\chi(t(A, a))$ (thanks to the common belief operator). We also define the operator t from $\mathcal{L}_{BMS}(A)$ to $\mathcal{L}$ by $t(p^0) = p^0, t(\neg\varphi) = \neg t(\varphi), t(\varphi \wedge \varphi') = t(\varphi) \wedge t(\varphi'), t(B_j\varphi) = B_j t(\varphi), t(C_G\varphi) = C_G t(\varphi)$ and $t([A, a]\varphi) = [1\ starts](\chi(t(A, a)) \to [1\ ends]t(\varphi))$.

Theorem 4.22. *Let A be an event model and $\varphi \in \mathcal{L}_{BMS}(A)$. For all pointed epistemic model (M^0, w^0),*

$$M^0, w^0 \models_{BMS} \varphi \text{ iff } \{(M^0, w^0)\} \models t(\varphi).$$

However, note that the $*$ operator of the BMS language cannot be expressed in our framework.

5 Related work

Other languages for event models have been proposed but none of them allows to express statements describing events as such. In [Baltag et al., 1999], the event language is the same as the epistemic language $\mathcal{L}^e$ and one sets $A, a \models p$ when $Pre(a) = p$. In [Rodenhäuser, 2001], labels are introduced that refer to the possible events of the event model, as in hybrid logic. New operators are also introduced: $A, a \models_{\downarrow 1} \varphi$ means 'any state reachable with a makes φ true' and $A, a \models_{\downarrow 2} \varphi$ means 'any state that makes φ true can be reached with a'.

At the outset of PDL [Pratt, 1976], a number of logical frameworks called process logics were proposed to express what happens *during* the computation of programs. As in PDL, the semantics of these frameworks all consider a set of states (possible worlds) as given, and the primitive programs at stake are represented by accessibility relations (transitions) between states. All these logics are propositional based and do not consider a set of agents. In [Pratt, 1979], the language of PDL is augmented with two additional operators $\perp$ and $[$. If a is a path (i.e. a sequence of primitive programs) and φ a propositional formula then $a \perp \varphi$ is true in w if at least one of the states of any computation of a starting from w satisfies φ. $a[\varphi$ is true in w if in any computation starting from w, if φ is true in some state then it remains true until the end of the computation. One can show that our logic is more expressive than Pratt's process logic (yet without the $*$ operator). In [Harel et al., 1982] the language of PDL is augmented with two additional operators $f\varphi$ and $\varphi suf \psi$: $f\varphi$ is true on a path if φ is true at the initial state of this path, and the operator suf corresponds to the until operator of temporal logic [Pnuelli, 1977]. Their process logic is more expressive than Pratt's process logic [Pratt, 1979], Parikh's SOAPL [Parikh, 1978], Nishimura's process logic [Nishimura, 1980] and Pnueli's Temporal Logic [Pnuelli, 1977]. This logic is refined in [Harel and Peleg, 1985] where f and suf are replaced by $chop$ and $slice$ yielding a strictly more expressive logic yet still decidable. Another process logic is defined in [Harel and Singerman, 1999] in the spirit of [Harel and Peleg, 1985] which also models concurrency and infinite computations. All these process logics have in common to evaluate truth of formulas on paths (a state being a path of length 0). This makes it difficult to compare them formally with our framework since our $\mathcal{L}$-models model what is true at a certain time and not throughout a history of programs (a path). In that respect they cannot express as we can that a primitive program is currently running but only express what is true at each step of a sequence of primitive programs.

6 Conclusion

We have proposed a logical framework that really exploits the power of the BMS notions of event model and product update. We showed that our framework embeds the BMS one and is still decidable (yet without common be-

[3] A formula χ characterizes a finite and pointed $\mathcal{L}^i$-model (M^i, w^i) iff $M^i, w^i \models \chi$ and for all finite and pointed $\mathcal{L}^i$-model $(M^{i'}, w^{i'})$, if $M^{i'}, w^{i'} \models \chi$ then (M^i, w^i) is bisimilar to $(M^{i'}, w^{i'})$.

lief). Unlike any other logical framework it can express statements about ongoing events (together with some static properties about the world). From a conceptual point of view, its formal structure reveals new aspects on the notion of event and belief dynamics. Firstly, as we saw, our beliefs about an event occurring can also be updated due to other events. Secondly, the set of all events has an internal logical structure and the classical Manichaean distinction between event and fact is not fine enough to account for the dynamics of beliefs.

A final remark on future work. In Definition 4.2, for simplicity and technical reasons we assumed that there is *at most* one pointed $\mathcal{L}^l$-model with $i = Pre(l)$ (condition 2). We can perfectly remove this assumption but then other kinds of product update should also be introduced. Indeed, assume that while tub 1 is being filled one publicly informs the agents that tub 2 is actually full. The preconditions of both events (the tub 1 being filled and the public announcement) are of type 0. However, after this public announcement, the agents know that tub 2 is full so they should update their beliefs and infer that tub 1 is currently being filled. Formally, this calls for the introduction of a 'reverse' update product which takes as argument a $\mathcal{L}^k$-model and a $\mathcal{L}^i$-model with $Pre(i) = k$ and yields a new $\mathcal{L}^i$-model. We leave the investigation of this new kind of update product for future work.

Acknowledgements

This paper originates from a discussion with Johan van Benthem during my master's thesis where he suggested that event languages might be defined similarly to epistemic languages. I thank him for that and also indirectly for the long journey it sparked. I also thank Emil Weydert for comments on this paper.

References

[Balbiani and Herzig, 2007] Balbiani, P. and Herzig, A. (2007). Talkin'bout Kripke models. In Braüner, T. and Villadsen, J., editors, *International Workshop on Hybrid Logic 2007 (Hylo 2007)*, Dublin.

[Baltag and Moss, 2004] Baltag, A. and Moss, L. (2004). Logic for epistemic programs. *Synthese*, 139(2):165–224.

[Baltag et al., 1998] Baltag, A., Moss, L., and Solecki, S. (1998). The logic of common knowledge, public announcement, and private suspicions. In Gilboa, I., editor, *Proceedings of the 7th conference on theoretical aspects of rationality and knowledge (TARK98)*, pages 43–56.

[Baltag et al., 1999] Baltag, A., Moss, L., and Solecki, S. (1999). The logic of public announcements, common knowledge and private suspicions. Technical report, Indiana University.

[Fagin et al., 1995] Fagin, R., Halpern, J., Moses, Y., and Vardi, M. (1995). *Reasoning about knowledge*. MIT Press.

[Harel et al., 1982] Harel, D., Kozen, D., and Parikh, R. (1982). Process logic: Expressiveness, decidability and completeness. *Journal of Computer and System Sciences*, 25(2).

[Harel et al., 2000] Harel, D., Kozen, D., and Tiuryn, J. (2000). *Dynamic Logic*. MIT Press.

[Harel and Peleg, 1985] Harel, D. and Peleg, D. (1985). Process logic with regular formulas. *Theoretical Computer Science*, 38:307–322.

[Harel and Singerman, 1999] Harel, D. and Singerman, E. (1999). Computation paths logic: An expressive, yet elementary, process logic. *Annals of pure and applied logic*, 96:167–186.

[Nishimura, 1980] Nishimura, H. (1980). Descriptively complete process logic. *Acta Informatica*, 14(4):359–369.

[Parikh, 1978] Parikh, R. (1978). A decidability result for second order process logic. In *Proceedings of 19th FOCS*, pages 177–183.

[Pnuelli, 1977] Pnuelli, A. (1977). The temporal logic of programs. In *Proceedings of 18th FOCS*, pages 46–57.

[Pratt, 1976] Pratt, V. (1976). Semantical considerations on floyd-hoare logic. In *Proceedings of the 17th IEEE Symposium on the Foundations of Computer Science*, pages 109–121.

[Pratt, 1979] Pratt, V. R. (1979). Process logic. In *Proceedings of the 6th ACM symposium on Principles of Programming Languages*, San Antonio.

[Rodenhäuser, 2001] Rodenhäuser, B. (2001). Updating epistemic uncertainty: an essay in the logic of information. Master's thesis, ILLC, University of Amsterdam.

[van Benthem et al., 2006] van Benthem, J., van Eijck, J., and Kooi, B. (2006). Logics of communication and change. *Information and Computation*, 204(11):1620–1662.

[van Ditmarsch et al., 2005] van Ditmarsch, H., van der Hoek, W., and Kooi, B. (2005). Dynamic epistemic logic with assignment. In Dignum, F., Dignum, V., Koenig, S., Kraus, S., Singh, M., and Wooldridge, M., editors, *Autonomous Agents and Multi-agent Systems (AAMAS 2005)*, pages 141–148. ACM.

[van Ditmarsch et al., 2007] van Ditmarsch, H., van der Hoek, W., and Kooi, B. (2007). *Dynamic Epistemic Logic*, volume 337 of *Synthese library*. Springer.