



UvA-DARE (Digital Academic Repository)

Epistemic modelling and protocol dynamics

Wang, Y.

Publication date
2010

[Link to publication](#)

Citation for published version (APA):

Wang, Y. (2010). *Epistemic modelling and protocol dynamics*. [Thesis, fully internal, Universiteit van Amsterdam]. Institute for Logic, Language and Computation.

General rights

It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations

If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: <https://uba.uva.nl/en/contact>, or a letter to: Library of the University of Amsterdam, Secretariat, P.O. Box 19185, 1000 GD Amsterdam, The Netherlands. You will be contacted as soon as possible.

5.1 Introduction

In Part I, we proposed and studied several variants of PDL as logics for reasoning about knowledge, protocol and change. As we have argued, to verify epistemic protocols by model checking, it is important to have the *right* model. However, as argued in [vB09], to build a model is an *art*. For real life applications, the initial models and the action models can be huge (see, e.g., [DW07] for a case study in a protocol verification setting). Thus some (semi-) automatic method is inevitable in dynamic epistemic modelling. In this part, we make some modest contributions to dynamic epistemic modelling.

Our first inspiration is from the ETL approaches where the temporal epistemic models (interpreted systems) are generated in a distributed fashion with each agent acting as a component (cf. Definition 2.3.1). This distributed feature made ETL very handy in modelling various multi-agent scenarios [FHMV95], for example, adding one extra agent is done by adding one more component. On the other hand, DEL models are apparently not inherently distributed at the first glance: the static models and action models contain information about all the agents. For example, in DEMO [vE07], an implementation of DEL model checking, the initial models are generated by first considering an universal ignorance model, where the agents do not know any atomic proposition, and then restricting it by announcements. It is clear that this method cannot generate all the desired initial models nor is it easy to handle extra agents. To make the DEL approach more applicable for real-life applications, it is crucial to build a static model (or an event model) from local components according to each agent's perspective. This clearly gives rise to the need for a formal way for *composing* models.

In fact, a clue is hinted at in DEL itself: the product update is indeed a way of composing models, though between two different types of models (static models vs. event models). A straightforward idea is to extend this product operation to compositions between two static models or two event models. However, there is a difficulty in defining such a composition operation: the models we want to compose

may have different vocabularies. Note that in practice, one agent may be able to observe only part of the atomic facts (propositions) in the whole vocabulary, e.g., in the Muddy Children scenario each child can see whether other children are muddy but can make no observations about herself. Thus it is reasonable to divide the whole vocabulary into parts (sets of observables) and build “partial models” for agents according to their local vocabularies.

In this chapter, we define and demonstrate the use of *merging composition* between static models and between action models with arbitrary vocabularies. For example, we show how a 2^n worlds Muddy Children model can be viewed as a composition of n two-node models, each talking only about the muddiness of a single child. Next, we extend standard event model update to an update operation which works on static models and event models with different vocabularies, by incorporating vocabulary expansion in the update process. We also look at the models generated in the distributed fashion of ETL and claim that our merging composition with arbitrary vocabularies can achieve the same goal in the DEL setting.

Related work Our merging composition of event models may be viewed as a notion of parallel composition of events. The first concurrent operation in the framework of DEL has been introduced in [vDvdHK03c, vDvdHK03b], where the authors follow the treatment of concurrency as in concurrent PDL [Pel87]. The concurrent operator \cap as in [vDvdHK03c] essentially splits the system into copies with each copy executing a concurrent component (see also [vDvdHK07, Chapter 5] for details). In some sense, composing actions in concurrent DEL may be viewed as merging agents who are acting differently, while in this chapter we focus on merging propositional information which is distributed among agents in both static and event models. Compared to the large body of research about parallel compositions in various process algebra frameworks (e.g., [Mil82, BK85, BHR84, GP94]), the distinct feature of our operator is the merging of different vocabularies and preconditions. The restriction to epistemic models (S5 models) also gives specific results meaningful in the epistemic setting.

Structure of the chapter In Section 5.2 we introduce the operator of merging composition on static models, under which the Kripke models form a commutative monoid. We then structurally characterize the induced pre-order by this monoid. Based on the merging composition, we study a natural operation which expands a model with a larger vocabulary. Various logical preservation results between the components and the composed model are proved. Section 5.3 addresses the problem of decomposition by looking at a specific class of models which are useful in a multi-agent setting. We demonstrate that we can decompose a model either by *agents* or by *issues*. We introduce the composition of event models and the extended product update in Section 5.4. We show that under certain conditions the action update distributes over merging composition. We point out some future directions in the last section.

5.2 Composing Static Models

5.2.1 Merging Composition

Recall that an S5 Kripke model \mathcal{M} is a tuple $(S, \mathbf{P}, \mathbf{I}, \sim, V)$ where \mathbf{P} is the (finite) vocabulary, \mathbf{I} is the (finite) set of agents, \sim_i is an equivalence relation for each $i \in \mathbf{I}$, and the valuation function $V : S \mapsto 2^{\mathbf{P}}$ assigns a set of atomic propositions to each state. Given a Kripke model \mathcal{M} , we use $S_{\mathcal{M}}, \mathbf{P}_{\mathcal{M}}, \mathbf{I}_{\mathcal{M}}, \sim_{\mathcal{M}}$ and $V_{\mathcal{M}}$ to denote the corresponding elements in the definition of \mathcal{M} . In this chapter we consider the compositions of models with different vocabularies but the same set of agents. We define the *unit* model \mathcal{E} as the model $(\{s\}, \emptyset, \mathbf{I}, \sim, V)$ where $V(s) = \emptyset$ and $\sim_i = \{(s, s)\}$ for any i . In a picture:



Now we define the *merging composition* of two S5 models with arbitrary vocabularies.

5.2.1. DEFINITION. (Merging Composition of Kripke Models) Given two models with the same set of agents \mathbf{I} : $\mathcal{M} = (S, \mathbf{P}, \mathbf{I}, \sim, V)$ and $\mathcal{N} = (T, \mathbf{P}', \mathbf{I}, \sim', V')$, the *merging composition* $\mathcal{M} \oplus \mathcal{N}$ is given by $(S'', \mathbf{P} \cup \mathbf{P}', \mathbf{I}, \sim'', V'')$, where:

- $S'' = \{(s, t) \mid s \in S, t \in T, V(s) \cap \mathbf{P}' = V'(t) \cap \mathbf{P}\}$,
- $(s, s') \sim''_i (t, t')$ iff $s \sim_i t$ and $s' \sim'_i t'$,
- $V''(s, t) = V(s) \cup V'(t)$.

□

Intuitively, the accessibility relations in the composed model are defined by “synchronizing” the corresponding relations in the components, in the usual way as in product updates, restricted to the pairs of worlds where the old valuations agree on the common vocabulary $\mathbf{P} \cap \mathbf{P}'$. It is clear that $V(s, s')$ agrees with $V(s)$ on \mathbf{P} and with $V'(s')$ on \mathbf{P}' , thus *merging* the two component valuations. We say a state s in \mathcal{M} is *compatible* with a state t in \mathcal{N} if (s, t) is in the composition $\mathcal{M} \oplus \mathcal{N}$. It is not hard to verify that any merging composition of S5 models is again an S5 model.

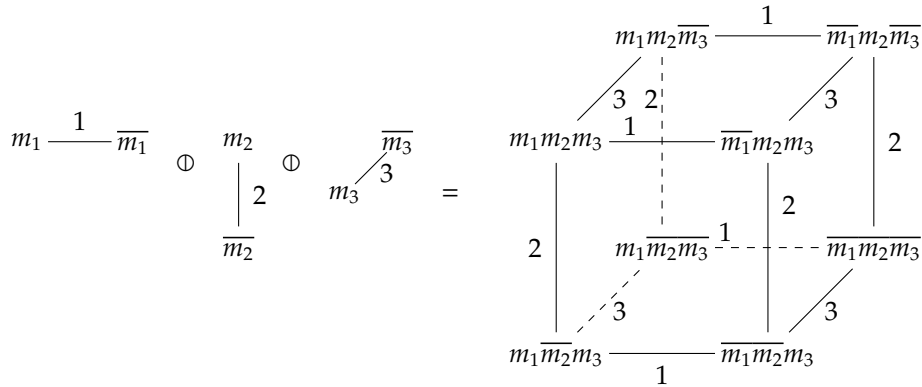
As a first example, here is a “compositional version” of the 2-Muddy Children scenario:

$$\begin{array}{ccc}
 m_1 \xrightarrow{1} \overline{m_1} & & m_1 m_2 \xrightarrow{1} \overline{m_1 m_2} \\
 & \oplus & \\
 & \left| \begin{array}{c} m_2 \\ 2 \\ \overline{m_2} \end{array} \right. & = & \left| \begin{array}{c} m_1 m_2 \\ 2 \\ m_1 \overline{m_2} \end{array} \right. \\
 & & & \left| \begin{array}{c} \overline{m_1 m_2} \\ 1 \\ \overline{m_1} m_2 \end{array} \right.
 \end{array}$$

where m_i expresses “child i is muddy”, the set of agents is $\{1, 2\}$, the vocabulary of each model $m_i \xleftrightarrow{i} \bar{m}_i$ is $\{m_i\}$, and as usual we leave out reflexive arrows which are present for all agents. Intuitively, each 2-world model represents the children’s observational power on whether child i is muddy, e.g., $m_i \xleftrightarrow{i} \bar{m}_i$ captures the situation that child i does not know whether she herself is muddy while all the others do know whether child i is muddy.

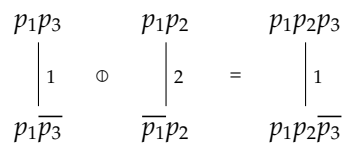
In the same fashion, composing the above models with a third model does give the 3-Muddy Children model¹:

5.2.2. EXAMPLE. Composing Muddy Children

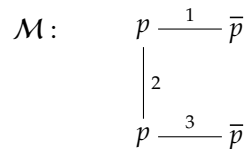


Multidimensional hypercubes with more and more children present can be composed in the same way. Ω

Here is an example of composing models with intersected vocabularies:



Note that according to our definition, self-composition $\mathcal{M} \oplus \mathcal{M}$ is not always bisimilar to \mathcal{M} . Consider the model \mathcal{M} :



¹Note that the set of agents needs to be extended from $\{1, 2\}$ to $\{1, 2, 3\}$.

Let us call the upper and lower p worlds s and t respectively. It is clear that (s, t) is in the composed model $\mathcal{M} \oplus \mathcal{M}$ and $V_{\mathcal{M} \oplus \mathcal{M}}((s, t)) = \{p\}$. However, according to the definition of relations in the composed model, (s, t) cannot reach a $\neg p$ world by just one step in the composed model. Therefore (s, t) is not bisimilar to any world in \mathcal{M} . Nevertheless, Kripke models with different vocabularies do form a commutative monoid:

5.2.3. THEOREM. *Kripke models with the same set of agents form a commutative monoid under the \oplus operation, with total bisimilarity (see Definition 2.2.2) as the appropriate equality notion. In particular, we have:*

$$\begin{aligned} \mathcal{E} \oplus \mathcal{M} &\Leftrightarrow \mathcal{M} \\ \mathcal{M} \oplus \mathcal{E} &\Leftrightarrow \mathcal{M} \\ \mathcal{M} \oplus (\mathcal{N} \oplus \mathcal{K}) &\Leftrightarrow (\mathcal{M} \oplus \mathcal{N}) \oplus \mathcal{K} \\ \mathcal{M} \oplus \mathcal{N} &\Leftrightarrow \mathcal{N} \oplus \mathcal{M} \end{aligned}$$

PROOF Commutativity and axioms about the unit are immediate. We only check associativity here. Let $A(l)_x^y$ be the abbreviation of $V_x(l) \cap \mathbf{P}_y$ for $l \in \{s, t, k\}$ and $x, y \in \{\mathcal{M}, \mathcal{N}, \mathcal{K}\}$, e.g., $A(s)_M^N$ represents $V_M(s) \cap \mathbf{P}_N$. Thus the condition

$$\text{PC} := (A(s)_M^N = A(t)_N^M \text{ and } A(s)_M^K = A(k)_K^M \text{ and } A(t)_N^K = A(k)_K^N)$$

expresses that s, t, k are pairwise compatible. A moment of reflection should assure that:

$$(s, (t, k)) \in S_{\mathcal{M} \oplus (\mathcal{N} \oplus \mathcal{K})} \iff \text{PC} \iff ((s, t), k) \in S_{(\mathcal{M} \oplus \mathcal{N}) \oplus \mathcal{K}}$$

Then it is straightforward to see that $\mathcal{M} \oplus (\mathcal{N} \oplus \mathcal{K}) \Leftrightarrow (\mathcal{M} \oplus \mathcal{N}) \oplus \mathcal{K}$. \spadesuit

Note that \Leftrightarrow is indeed a congruence of this monoid:

5.2.4. PROPOSITION. *If $\mathcal{M}_1 \Leftrightarrow \mathcal{M}_2$ and $\mathcal{N}_1 \Leftrightarrow \mathcal{N}_2$ then $\mathcal{M}_1 \oplus \mathcal{N}_1 \Leftrightarrow \mathcal{M}_2 \oplus \mathcal{N}_2$*

PROOF Let Z_1, Z_2 be the total bisimulations witnessing $\mathcal{M}_1 \Leftrightarrow \mathcal{M}_2$ and $\mathcal{N}_1 \Leftrightarrow \mathcal{N}_2$ respectively. Then the relation $Z \subseteq S_{\mathcal{M}_1 \oplus \mathcal{N}_1} \times S_{\mathcal{M}_2 \oplus \mathcal{N}_2}$ defined by:

$$(s_1, t_1)Z(s_2, t_2) \iff s_1Z_1s_2 \text{ and } t_1Z_2t_2$$

is clearly a total bisimulation between $\mathcal{M}_1 \oplus \mathcal{N}_1$ and $\mathcal{M}_2 \oplus \mathcal{N}_2$. \spadesuit

The commutative monoid yields the algebraic preordering \leq on the class of Kripke models with different vocabularies:

$$\mathcal{M} \leq \mathcal{N} \text{ iff there is a } \mathcal{K} \text{ with } \mathcal{M} \oplus \mathcal{K} \Leftrightarrow \mathcal{N}.$$

We proceed to give a structural characterization of this relation. For this, let a *left-simulation* between two restricted static models \mathcal{M} and \mathcal{N} be a bisimulation with the

invariance condition restricted to proposition letters in the vocabulary of \mathcal{M} , and without the Zig condition (see Definition 2.2.2). Formally, given two models \mathcal{M} and \mathcal{N} such that $\mathbf{P}_{\mathcal{M}} \subseteq \mathbf{P}_{\mathcal{N}}$, a left-simulation between \mathcal{M} and \mathcal{N} is a relation $R \subseteq S_{\mathcal{M}} \times S_{\mathcal{N}}$ such that sRt implies that the following hold:

Restricted Invariance $V_{\mathcal{M}}(s) = V_{\mathcal{N}}(t) \cap \mathbf{P}_{\mathcal{M}}$;

Zag If for some $i \in \mathbf{I}$ there is a $t' \in S_{\mathcal{N}}$ with $t \xrightarrow{i} t'$ then there is a $s' \in S_{\mathcal{M}}$ with $s \xrightarrow{i} s'$ and $s'Rt'$.

We will use $\mathcal{M}, s \Leftarrow \mathcal{N}, t$ to indicate that there is a left-simulation that connects s and t , and $\mathcal{M} \Leftarrow \mathcal{N}$ to indicate that there is a *total* left-simulation between \mathcal{M} and \mathcal{N} : there is a left-simulation R that links *every* world in \mathcal{N} to *some* world in \mathcal{M} .²

Ditmarsch and French [vDF09] prove that for finite static models \mathcal{M} and \mathcal{N} :

\mathcal{M} is a simulation of $\mathcal{N} \iff$ there exists an event model \mathcal{A} st. $\mathcal{N} \Leftrightarrow \mathcal{M} \otimes \mathcal{A}$

Here we prove a similar result in our setting:

5.2.5. THEOREM. *For any models \mathcal{M}, \mathcal{N} with arbitrary vocabularies:*

$$\mathcal{M} \leq \mathcal{N} \implies \mathcal{M} \Leftarrow \mathcal{N}$$

PROOF \implies : Assume $\mathcal{M} \leq \mathcal{N}$. Then there is a model \mathcal{K} with $\mathcal{M} \oplus \mathcal{K} \Leftrightarrow \mathcal{N}$. Let Z be a total bisimulation between $S_{\mathcal{M} \oplus \mathcal{K}}$ and $S_{\mathcal{N}}$. Define R as sRt iff there is some world $x \in S_{\mathcal{K}}$ with $(s, x)Zt$. R is easily seen to be a total left-simulation between \mathcal{M} and \mathcal{N} . The restricted invariance property follows from the definition of the valuation on $\mathcal{M} \oplus \mathcal{K}$. The zag property follows from the definition of the accessibility relations on $\mathcal{M} \oplus \mathcal{K}$. Thus, $\mathcal{M} \Leftarrow \mathcal{N}$. \star

Note that the converse does not hold without restrictions on the models. For example, let \mathcal{M} and \mathcal{N} be the following two S5 models:

$$\mathcal{M}: \quad \begin{array}{c} p \xrightarrow{1} p \\ | \\ \bar{p} \end{array} \quad \mathcal{N}: \quad \begin{array}{c} pq \xrightarrow{1} p\bar{q} \\ | \\ \overline{pq} \end{array}$$

It is clear that $\mathcal{M} \Leftarrow \mathcal{N}$. Now suppose towards a contradiction that there exists an \mathcal{M}' such that $\mathcal{M} \oplus \mathcal{M}' \Leftrightarrow \mathcal{N}$. Since there is a pq world in \mathcal{N} , there must be a world t in \mathcal{M}' such that $q \in V_{\mathcal{M}'}(t)$ and t is compatible with any p world in \mathcal{M} . Let us denote the upper-right world in \mathcal{M} as s . Then (s, t) must be in the composed model $\mathcal{M} \oplus \mathcal{M}'$ and $V((s, t)) = \{p, q\}$. However, according to the definition of \oplus , (s, t) cannot reach a

²Note that the *totality* here is different from the totality of bisimulation which requires that any world in any one of the two models is linked to some world in the other model.

$\neg p$ world in one step, thus (s, t) is not bisimilar to any pq world in \mathcal{N} . It follows that (s, t) is not bisimilar to any world in \mathcal{N} .

Now we look at a subclass of models. A model \mathcal{M} is called *propositionally differentiated* if for any $s, t \in \mathcal{M}$: $V_{\mathcal{M}}(s) \neq V_{\mathcal{M}}(t)$. For example, the models for Muddy Children and Russian Cards Problem as in Chapter 3 are indeed propositionally differentiated. Restricted to this simple but useful class of models we have the exact correspondence of \Leftarrow and \leq :

5.2.6. THEOREM. *Let \mathcal{M} be a propositionally differentiated model. Then*

$$\mathcal{M} \leq \mathcal{N} \iff \mathcal{M} \Leftarrow \mathcal{N}$$

PROOF \Rightarrow follows from Theorem 5.2.5.

For \Leftarrow : Assume $\mathcal{M} \Leftarrow \mathcal{N}$. Let R be a left-simulation between \mathcal{M} and \mathcal{N} . Since R is total, for each $t \in S_{\mathcal{N}}$ there is at least one world s in $S_{\mathcal{M}}$ such that sRt . Note that since \mathcal{M} is propositionally differentiated and $\mathbf{P}_{\mathcal{M}} \subseteq \mathbf{P}_{\mathcal{N}}$, for each $t \in S_{\mathcal{N}}$ there is at most one world $s \in S_{\mathcal{M}}$ such that $V_{\mathcal{M}}(s) = V_{\mathcal{N}}(t) \cap \mathbf{P}_{\mathcal{M}}$. Therefore for each $t \in \mathcal{N}$ there is one and only one world s such that sRt .

We will show that $\mathcal{M} \oplus \mathcal{N} \Leftarrow \mathcal{N}$. Let the relation Z between $\mathcal{M} \oplus \mathcal{N}$ and \mathcal{N} be defined as:

$$(s, t)Zt' \text{ iff } t = t'$$

We claim Z is a total bisimulation. Totality is straightforward. Suppose $(s, t)Zt'$. We now check the three conditions for bisimulation.

By the construction of $\mathcal{M} \oplus \mathcal{N}$, $V_{\mathcal{M} \oplus \mathcal{N}}((s, t)) = V_{\mathcal{M}}(s) \cup V_{\mathcal{N}}(t) = V_{\mathcal{N}}(t)$. This proves the invariance property.

Suppose $(s, t) \sim_i (s', t')$. By the construction of $\mathcal{M} \oplus \mathcal{N}$ this means $s \sim_i s'$ and $t \sim_i t'$. By the definition of Z , $(s', t')Zt'$. This proves the Zig property.

Suppose $t \sim_i t'$. Recall that since $\mathcal{M} \Leftarrow \mathcal{N}$ and \mathcal{M} is propositionally differentiated, there must be a unique s in \mathcal{M} such that sRt . Then since R is left-simulation, there must be some s' such that $s \sim_i s'$ and $s'Rt'$. Since $s'Rt'$, $V_{\mathcal{M}}(s') = V_{\mathcal{N}}(t') \cap \mathbf{P}_{\mathcal{M}}$ so $(s', t') \in \mathcal{M} \oplus \mathcal{N}$. Since $s \sim_i s'$ and $t \sim_i t'$ then $(s, t) \sim_i (s', t')$. By the definition of Z , $(s', t')Zt'$. This proves the Zag property. \heartsuit

5.2.2 Expansion

Based on the merging composition operation, we can define the expansions of models with new vocabularies. Let $\mathcal{M}_{\mathbf{P}}^{\mathbf{I}}$ be the universal ignorance model for \mathbf{P} , i.e. $\mathcal{M}_{\mathbf{P}}^{\mathbf{I}} = (S, \mathbf{P}, \mathbf{I}, \sim, V)$ with $S = \mathcal{P}(\mathbf{P})$, $\sim_i = S \times S$, $V = \text{id}$. Given \mathcal{M} we define the *expansion* of \mathcal{M} w.r.t. vocabulary \mathbf{P}' as follows: $\mathcal{M} \triangleleft \mathbf{P}' = \mathcal{M} \oplus \mathcal{M}_{\mathbf{P}'}^{\mathbf{I}}$. Note that $\mathbf{P}_{\mathcal{M} \triangleleft \mathbf{P}'} = \mathbf{P}_{\mathcal{M}} \cup \mathbf{P}'$ and the states in the expansion are of the form (s, X) where $s \in S_{\mathcal{M}}$ and $X \subseteq \mathbf{P}$. In the sequel, we will use variables X, Y to denote (possibly empty) subsets of the whole vocabulary in the context of expansions.

Here is an example of expanding with a single new proposition letter m_2 . Note: Here and henceforth, worlds are *i*-linked if there is an *i*-path in the picture.

PROOF Let R be a left-simulation which witnesses $\mathcal{M}, s \Leftarrow \mathcal{N}, t$. Let

$$R' = \{(s, V_{\mathcal{N}}(t)), t \mid (s, t) \in R\}$$

Note that when $(s, t) \in R$, $(s, V_{\mathcal{N}}(t))$ is indeed in the model $\mathcal{M} \triangleleft \mathbf{P}_{\mathcal{N}}$ due to the restricted invariance condition of R . Thus R' is well-defined. Totality follows from the totality of R . We claim that R' is a left-simulation between $\mathcal{M} \triangleleft \mathbf{P}_{\mathcal{N}}, (s, V_{\mathcal{N}}(t))$ and \mathcal{N}, t . The condition of restricted invariance is obvious. For the Zag condition, suppose $t \xrightarrow{i} t' \in \mathcal{N}$ then there is an s' such that $s \xrightarrow{i} s'$ in \mathcal{M} and $s'Rt'$. Since $\mathcal{M} \triangleleft \mathbf{P}_{\mathcal{N}} = \mathcal{M} \oplus \mathcal{M}_{\mathbf{P}_{\mathcal{N}}}^I$, we have $(s, V_{\mathcal{N}}(t)) \xrightarrow{i} (s', V_{\mathcal{N}}(t'))$ in $\mathcal{M} \triangleleft \mathbf{P}_{\mathcal{N}}$ and $(s', V_{\mathcal{N}}(t'))R't'$. \times

5.2.3 Preservation

Now let us consider the PDL language over \mathbf{P}, \mathbf{I} (notation: $\text{PDL}_{\mathbf{P}, \mathbf{I}}$):

$$\begin{aligned} \phi & ::= \top \mid p \mid \neg\phi \mid \phi \vee \phi \mid \langle \pi \rangle \phi \\ \pi & ::= i \mid ?\phi \mid \pi \cdot \pi \mid \pi + \pi \mid \pi^* \end{aligned}$$

The semantics for $\text{PDL}_{\mathbf{P}, \mathbf{I}}$ is defined as usual (see Section 2.3.1). Note that the truth value of a $\text{PDL}_{\mathbf{P}, \mathbf{I}}$ formula may not be defined on a model with a different vocabulary other than \mathbf{P} . We will study a three valued semantics in chapter 7, while in this chapter, we stick to the 2-valued semantics and make sure the formulas are evaluated on the models where the semantics is defined.

Since PDL is bisimulation invariant, as a straightforward consequence of Proposition 5.2.8, we have:

5.2.10. PROPOSITION. *For any model \mathcal{M} , if $X \cap \mathbf{P}_{\mathcal{M}} = \emptyset$ and $Y \subseteq X$ then for any $\phi \in \text{PDL}_{\mathbf{P} \cup \mathbf{Y}, \mathbf{I}}$: $\mathcal{M} \triangleleft X, (s, X') \vDash \phi \iff \mathcal{M} \triangleleft Y, (s, X' \cap Y) \vDash \phi$.*

We will use this proposition to prove Theorem 5.4.5 in Section 5.4

The diamond fragment of $\text{PDL}_{\mathbf{P}, \mathbf{I}}$ is given by the following ϕ form of formulas:

$$\begin{aligned} \psi & ::= \top \mid p \mid \neg\psi \mid \psi \vee \psi \\ \pi & ::= i \mid ?\phi \mid \pi \cdot \pi \mid \pi + \pi \mid \pi^* \\ \phi & ::= \psi \mid \langle \pi \rangle \phi \mid \phi \vee \phi \mid \phi \wedge \phi. \end{aligned}$$

We can define the box fragment of $L_{\mathbf{P}, \mathbf{I}}$ i.e. the collection of formulas which are logically equivalent to $\neg\phi$ for some ϕ in the diamond fragment.

It is well-known that diamond formulas are preserved under simulation. The following theorem generalizes this to cases where the vocabularies of the two models may be different.

5.2.11. THEOREM. *If $\mathcal{M}, s \leq \mathcal{N}, t$ then all formulas ϕ in the diamond fragment of $\text{PDL}_{\mathbf{P}_{\mathcal{M}}, \mathbf{I}}$ are preserved from right to left under left simulation: if $\mathcal{N}, t \vDash \phi$ then $\mathcal{M}, s \vDash \phi$. Equivalently, the box fragment of $\text{PDL}_{\mathbf{P}_{\mathcal{M}}, \mathbf{I}}$ is preserved from left to right under left simulation.*

PROOF Suppose $\mathcal{M}, s \leq \mathcal{N}, t$. From Theorem 5.2.5, we know that $\mathcal{M}, s \Leftarrow \mathcal{N}, t$. Let R be a left simulation with sRt . We prove the property by induction on the construction of ϕ . The purely boolean cases are trivial according to the invariance property of R . We only need to show the case of $\langle \pi \rangle \phi$. Suppose $\mathcal{N}, t \models \langle \pi \rangle \phi$, then there is a path w starting from t in \mathcal{N} such that $w \in L(\pi)$ which leads to some ϕ world t' . By the Zag property of R , it is not hard to see that there is a matching w path in \mathcal{M} starting from s to some world s' such that $s'Rt'$. By the induction hypothesis, $\mathcal{M}, s' \models \phi$, and therefore $\mathcal{M}, s \models \langle \pi \rangle \phi$. \spadesuit

Based on Proposition 5.2.9 we can relax the restrictions of $\mathbf{P}_{\mathcal{M}}$ on the formulas:

5.2.12. COROLLARY. *If $\mathcal{M}, w \leq \mathcal{N}, t$ then all formulas ϕ in the diamond fragment of $\text{PDL}_{\mathbf{P}_N, \mathbf{I}}$ are preserved from right to left under left simulation: if $\mathcal{N}, t \models \phi$ then $\mathcal{M} \triangleleft \mathbf{P}_N, (w, V(t)) \models \phi$.*

Theorem 5.2.11 suggests a way of checking the properties (in terms of formulas in the box fragment) of a big model by looking at its components. The following theorem shows that the components can carry more information about the composed models, if we restrict ourselves to certain decomposition of the models. Formally we say model \mathcal{M} is *decomposable* into $\mathcal{M}_0, \dots, \mathcal{M}_n$ if $\mathcal{M} \Leftarrow \mathcal{M}_0 \oplus \dots \oplus \mathcal{M}_n$. A pointed model (\mathcal{M}, s) is decomposable into $(\mathcal{M}_0, s_0), \dots, (\mathcal{M}_n, s_n)$ if $\mathcal{M}, s \Leftarrow \mathcal{M}_1 \oplus \dots \oplus \mathcal{M}_n, (s_0, \dots, s_n)$.

5.2.13. THEOREM (PRESERVATION). *If a pointed model (\mathcal{M}, s) is decomposable into models $(\mathcal{M}_0, s_0), \dots, (\mathcal{M}_n, s_n)$ with disjoint vocabularies $\mathbf{P}_0, \mathbf{P}_1, \dots, \mathbf{P}_n$, then for any i : $\mathcal{M}_i, s_i \Leftarrow_{\mathbf{P}_i} \mathcal{M}, s$. Therefore for any ϕ in $\text{PDL}_{\mathbf{P}_i, \mathbf{I}}$: $\mathcal{M}_i, s_i \models \phi \iff \mathcal{M}, s \models \phi$.*

PROOF Suppose without loss of generality that $\mathcal{M}, s = \mathcal{M}_0 \oplus \dots \oplus \mathcal{M}_n, (s_0, \dots, s_n)$ where $s_i \in \mathcal{M}_i$. Given a tuple $\vec{t} = (t_0, t_1, \dots, t_n) \in \mathcal{M}$, we let $\vec{t}[i]$ be the i th element in the tuple t . Let Z_i be the relation on $S_{\mathcal{M}} \times S_{\mathcal{M}_i}$ given by $\vec{t}Z_it$ iff $\vec{t}[i] = t$. We show that Z_i is a \mathbf{P}_i -restricted bisimulation. It is clear that sZ_is_i . Assume $\vec{t}Z_it$ for some $\vec{t} \in \mathcal{M}$ and $t \in \mathcal{M}_i$. Then $V_{\mathcal{M}}(\vec{t}) \cap \mathbf{P}_i = V_{\mathcal{M}_i}(t)$, by the definition of the merging composition. Thus, \mathbf{P}_i -restricted invariance holds. Next suppose $\vec{t} \xrightarrow{k} \vec{t}'$. Then by the definition of the accessibility relations on \mathcal{M} , $\vec{t}[i] \xrightarrow{k} \vec{t}'[i]$, whence, by definition of Z_i , there is a t'' in the domain of \mathcal{M}_i with $\vec{t}'Z_it''$. It follows that the Zig condition holds. Finally, assume $t \xrightarrow{k} t'$ in \mathcal{M}_i and $\vec{t}Z_it$. Now consider the state \vec{t}' given by $\vec{t}'[i] = t'$ and $\vec{t}'[j] = \vec{t}[j]$ for $j \neq i$. Since \mathbf{P}_i is disjoint from any other vocabulary \mathbf{P}_j for $j \neq i$, \vec{t}' must be in \mathcal{M} and $\vec{t}'Z_it'$. Then by the reflexivity of the S5 component models and the fact that $t \xrightarrow{k} t'$, we have $\vec{t} \xrightarrow{k} \vec{t}'$. This proves the Zag condition. \spadesuit

As an example, consider the case of Muddy Children. From the above theorem, we know that any epistemic statement that talks about the muddiness of a single child in the big model can be checked in a two-world component, e.g., at the component model $m_1 \xleftarrow{1} \overline{m_1}$, we can verify that agent 2 knows that agent 1 does not know whether she herself is muddy.

5.3 Decomposition

At this stage a natural question to ask is: what kind of model can be decomposed into what kind of form? In this section we look at a particular class of models which is useful in multi-agent systems. In the interpreted systems literature, a basic proposition $p \in \mathbf{P}$ is *i-local* for $i \in \mathbf{I}$ in a model \mathcal{M} , if for any s, t in $S_{\mathcal{M}}$: $s \sim_i t$ implies that $(p \in V_{\mathcal{M}}(s) \iff p \in V_{\mathcal{M}}(t))$ (cf., e.g., [EvdMM98]). Intuitively, the *i-local* propositions are the *atomic observables* of agent i and thus agent i also knows whether they are true. Here we extend this idea by considering not only basic propositions but also their boolean combinations. We say \mathcal{M} is *locally generated* if, for every agent i , there is a non-empty set of boolean formulas Φ_i (the set of local observables) based on $\mathbf{P}_{\mathcal{M}}$ such that:

$$\text{for all } s, s' \in S_{\mathcal{M}}, s \sim_i s' \text{ iff for all } \varphi \in \Phi_i, \mathcal{M}, s \models \varphi \iff \mathcal{M}, s' \models \varphi$$

Intuitively, a model is locally generated if those local observables determine the epistemic relations in the model. The Muddy Children model is a typical example of a locally generated model (the set of local observables for i is $\{m_j \mid j \neq i, j \in \mathbf{I}\}$). As the following two propositions will show, locally generated models are essentially propositionally differentiated models, which we considered in Theorem 5.2.6.

5.3.1. PROPOSITION. *A locally generated model is bisimilar to a propositionally differentiated model. More precisely, its bisimulation contraction (see Definition 2.2.3) is propositionally differentiated.*

PROOF Given a locally generated model \mathcal{M} , suppose Φ_i is the set of local observables for i . Let $Z = \{(s, t) \mid V_{\mathcal{M}}(s) = V_{\mathcal{M}}(t)\}$. We show Z is a bisimulation. Assume sZt . The invariance condition is trivial. For Zig, suppose $s \stackrel{i}{\sim} s'$. Since \mathcal{M} is locally generated, for any $\phi \in \Phi_i$: $\mathcal{M}, s \models \phi \iff \mathcal{M}, s' \models \phi$. Since Φ_i contains only boolean formulas and $V_{\mathcal{M}}(s) = V_{\mathcal{M}}(t)$, we have for any $\phi \in \Phi_i$: $\mathcal{M}, t \models \phi \iff \mathcal{M}, s' \models \phi$. Again due to the definition of the relations in a locally generated model, we have $t \sim_i s'$. Obviously $s'Zs'$, thus it proves the Zig condition. The same argument works for the Zag condition. Therefore it is easy to see that the bisimulation contraction of \mathcal{M} is propositionally differentiated. \times

We also have:

5.3.2. PROPOSITION. *Propositionally differentiated models are locally generated.*

PROOF Suppose \mathcal{M} is propositionally differentiated. Let $|S_{\mathcal{M}}|_{\sim_i}$ be the partitioning of $S_{\mathcal{M}}$ according to the equivalence relation \sim_i . Since \mathcal{M} is propositionally differentiated, we can characterize each world by a conjunction of literals. Then we can characterize each equivalence class in $|S_{\mathcal{M}}|_{\sim_i}$ by a disjunction of these characterizations. Let Φ_i be the set of these disjunctions. Then clearly \mathcal{M} is locally generated from these Φ_i . \times

We can decompose a locally generated model into certain components in an intuitive way.

5.3.3. THEOREM (DECOMPOSITION BY AGENTS). *Given a set of agent $\mathbf{I} = \{1, 2, \dots, n\}$. If $\mathcal{M} = (S, \mathbf{P}, \mathbf{I}, \sim, V)$ is locally generated w.r.t. Φ_1, \dots, Φ_n , then there are models $\mathcal{M}_1, \dots, \mathcal{M}_n$ and \mathcal{M}_0 such that:*

- $\mathcal{M} \Leftrightarrow (\mathcal{M}_0 \oplus \mathcal{M}_1 \oplus \dots \oplus \mathcal{M}_n)$;
- $|S_{\mathcal{M}_i}| \leq |S|$ and \mathcal{M}_i is bisimulation contracted model;
- $\mathbf{P}_{\mathcal{M}_j} = \{p \in \mathbf{P}_{\mathcal{M}} \mid p \text{ appears in } \Phi_j\}$ for $j > 0$;

PROOF Let $\mathbf{P}_i = \{p \in \mathbf{P}_{\mathcal{M}} \mid p \text{ appears in } \Phi_i\}$ and $\mathbf{P}_0 = \mathbf{P}_{\mathcal{M}}$ then we define $\mathcal{N}_i = (S_{\mathcal{M}}, \mathbf{P}_i, \mathbf{I}, \sim_{\mathcal{N}_i}, V_i)$ where V_i is the restriction of $V_{\mathcal{M}}$ to \mathbf{P}_i and

$$s \overset{j}{\sim}_{\mathcal{N}_i} s' \iff \begin{cases} s \overset{j}{\sim}_{\mathcal{M}} s' & \text{if } j = i \\ \text{always} & \text{if } i \neq j \end{cases}$$

Intuitively, for each $i \in \mathbf{I}$: \mathcal{N}_i is a “local” model for agent i obtained by ignoring the non-local information: atomic propositions not mentioned in the i -observables and epistemic accessibility relations for agents other than i . Note that by ignoring the epistemic relations for j we mean setting \sim_j to be universal. For example:

$$\begin{array}{ccc} m_1 m_2 \overset{1}{\text{---}} \overline{m_1 m_2} & & m_2 \overset{1,2}{\text{---}} m_2 \quad m_2 \quad m_2 \\ 2 \left| \quad \quad \quad \right| 2 \Rightarrow \text{ignore } m_1 \text{ and } \sim_2 \Rightarrow & & 2 \left| \quad \quad \quad \right| 2 \Leftrightarrow \left| \quad \quad \quad \right| 2 \\ m_1 \overline{m_2} \overset{1}{\text{---}} \overline{m_1 m_2} & & \overline{m_2} \overset{1,2}{\text{---}} \overline{m_2} \quad \overline{m_2} \quad \overline{m_2} \end{array}$$

By our definition, the relations in \mathcal{N}_0 are universal. Intuitively, \mathcal{N}_i captures all the possible *states of affairs* in \mathcal{M} . Let a relation $Z \subseteq S_{\mathcal{M}} \times S_{\mathcal{N}_0 \oplus \dots \oplus \mathcal{N}_n}$ be given as follows:

$$sZ(s_0, s_1, \dots, s_n) \iff s = s_0$$

Now let us verify that Z is indeed a total bisimulation. Totality and invariance are trivial by definition of Z .

For Zig: Suppose $s \overset{i}{\sim}_{\mathcal{M}} s'$ and $sZ(s_0, s_1, \dots, s_n)$ then $s = s_0$. Since (s, s_1, \dots, s_n) exists, then $V_{\mathcal{M}}(s) \cap \mathbf{P}_i = V_{\mathcal{N}_i}(s_i)$. Therefore s and s_i satisfy the same set of boolean formulas based on \mathbf{P}_i . Since \mathcal{M} is locally generated then we know that s' and s_i agree on the formulas in Φ . Therefore s_i and s' must also agree on the truth values of the formulas in Φ , thus $s_i \overset{i}{\sim}_{\mathcal{M}_i} s'$. Since $\overset{j}{\sim}_{\mathcal{N}_i}$ is universal for $j \neq i$, it is clear that $(s, s_1, \dots, s_n) \overset{i}{\sim}_{\mathcal{N}_i} (s', s'_1, \dots, s'_n)$ in $\mathcal{N}_0 \oplus \dots \oplus \mathcal{N}_n$ and $s'Z(s', s'_1, \dots, s'_n)$.

For Zag: Suppose $(s_0, s_1, \dots, s_n) \overset{i}{\sim}_{\mathcal{N}_i} (s'_0, s'_1, \dots, s'_n)$ and $sZ(s_0, s_1, \dots, s_n)$, we then have $s = s_0$ and $s_i \overset{i}{\sim}_{\mathcal{N}_i} s'_i$. By the definition of \mathcal{N}_i , we have $s_i \overset{i}{\sim}_{\mathcal{M}} s'_i$. Thus s_i and s'_i agree on formulas in Φ_i . Since s_0 in \mathcal{M} is compatible with s_i in \mathcal{N}_i and s'_i in \mathcal{N}_i is

compatible with s'_0 in \mathcal{M} thus s'_0 also agrees with s_0 on formulas in Φ_i . Therefore $s \sim_{\mathcal{M}}^i s'_0$ and $s'_0 Z(s'_0, s'_1, \dots, s'_n)$. This proves the Zag condition.

Now we have shown $\mathcal{M} \Leftrightarrow \mathcal{N}_0 \oplus \dots \oplus \mathcal{N}_n$. Let \mathcal{M}_i be the bisimulation contraction of \mathcal{N}_i . From Proposition 5.2.4, we know $\mathcal{M} \Leftrightarrow (\mathcal{M}_0 \oplus \mathcal{M}_1 \oplus \dots \oplus \mathcal{M}_n)$. \spadesuit

In the above proof, \mathcal{M}_0 is used to rule out unnecessary worlds created by the merging composition. It is not hard to see that if $\bigcup_{i \in \mathbf{I}} \mathbf{P}_i = \mathbf{P}_{\mathcal{M}}$ and for any $\mathbf{P} \subseteq \mathbf{P}_{\mathcal{M}}$ there is an $s \in S_{\mathcal{M}}$ such that $V_{\mathcal{M}}(s) = \mathbf{P}$, then we can drop the \mathcal{M}_0 in the decomposition.

The above theorem gives another way to decompose the Muddy Children models different from the one in Example 5.2.2. Recall that an n -Muddy Children model is locally generated by sets of observables Φ_1, \dots, Φ_n where $\Phi_i = \{m_j \mid j \neq i, j \in \mathbf{I}\}$. For example, if $n = 3$ then the set of observables for agent 1 is $\{m_2, m_3\}$. We can then decompose the 3-Muddy Children model \mathcal{M} by $\mathcal{M}_1, \dots, \mathcal{M}_3$, where e.g., \mathcal{M}_1 is as follows:

$$\begin{array}{ccc} m_2 m_3 & \xrightarrow{2,3} & m_2 \overline{m_3} \\ \left| \begin{array}{c} 2,3 \\ 2,3 \end{array} \right. & & \left| \begin{array}{c} 2,3 \\ 2,3 \end{array} \right. \\ \overline{m_2} m_3 & \xrightarrow{2,3} & \overline{m_2} \overline{m_3} \end{array}$$

Compared to the two-world model decomposition in Example 5.2.2, the above decomposition requires bigger size components (2^{n-1} worlds for the n children case). This is because we decompose the model in an *agent-based* fashion: each component represents one agent's observational power regardless of the others. Thus if the vocabulary of the set of observables Φ_i is big then so is the component model. In the Muddy Children example, if there are more children then the vocabulary of the observables for each child also increases (e.g., new m_j), therefore the component model for this agent also grows bigger. However, in other applications the vocabulary of observables may not increase even when the initial model grows bigger. For example, in the Russian Cards scenario, the agents can only observe their own cards, no matter how many other agents there are (cf. the locally generated initial model of $RCP_{n,n,k}$ in Section 3.4). Therefore, the size of each component can be constant and relatively small.

To decompose a Muddy Children model as in Example 5.2.2, we decompose the model in an *issue-based* fashion (every proposition is an issue), as the following theorem demonstrates:

5.3.4. THEOREM (DECOMPOSITION BY ISSUES). *Given a set of agent $\mathbf{I} = \{1, 2, \dots, n\}$ and a set of proposition letters $\mathbf{P} = \{p_1, \dots, p_k\}$, if $\mathcal{M} = (S, \mathbf{P}, \mathbf{I}, \sim, V)$ is locally generated by Φ_1, \dots, Φ_n such that Φ_i only contains atomic propositions (i.e., $\Phi_i \subseteq \mathbf{P}$), then there are models $\mathcal{M}_1, \dots, \mathcal{M}_k$ and \mathcal{M}_0 such that:*

- $\mathcal{M} \Leftrightarrow (\mathcal{M}_0 \oplus \mathcal{M}_1 \oplus \dots \oplus \mathcal{M}_k)$;
- $\mathbf{P}_{\mathcal{M}_j} = \{p_j\}$ for $j > 0$ and $\mathbf{P}_0 = \mathbf{P}$;

- $|S_{\mathcal{M}_j}| = 2$ for $j > 0$

PROOF Let \mathcal{M}_0 be the same as in the proof of Theorem 5.3.3. For $j > 0$, let $\mathcal{M}_j = (S_j, \mathbf{P}_j, \mathbf{I}, \sim_{\mathcal{M}_j}, V_j)$ where:

- $S_j = \{p_j, \bar{p}_j\}$;
- $\mathbf{P}_j = \{p_j\}$;
- $V_j(p_j) = \{p_j\}$ and $V_j(\bar{p}_j) = \emptyset$ with the obvious interpretation;
- for $j > 0$: $s \stackrel{i}{\sim}_{\mathcal{M}_j} t \iff s = t$ or $(s \neq t \text{ and } p_j \notin \Phi_i)$.

Let a relation $Z \subseteq S_{\mathcal{M}} \times S_{\mathcal{M}_0 \oplus \dots \oplus \mathcal{M}_k}$ be given as follows:

$$sZ(s_0, s_1, \dots, s_k) \iff s = s_0$$

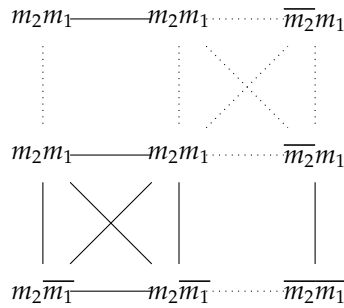
If $sZ(s_0, s_1, \dots, s_k)$ then for $0 < j \leq k$: s_j intuitively represents the truth value of p_j in s . Now let us verify that Z is indeed a total bisimulation. Totality and invariance are trivial.

For Zig: Suppose $s \stackrel{i}{\sim}_{\mathcal{M}} s'$ and $sZ(s_0, s_1, \dots, s_k)$. Since (s_0, s_1, \dots, s_k) exists, then $s_0 = s$ and $(s_j = p_j \iff p_j \in V_{\mathcal{M}}(s))$. It is easy to see there is some world (s', s'_1, \dots, s'_k) in $\mathcal{M}_0 \oplus \dots \oplus \mathcal{M}_k$ such that $s'Z(s', s'_1, \dots, s'_k)$. We need to show that for each $j > 0$: $s_j \stackrel{i}{\sim}_{\mathcal{M}_j} s'_j$. If $s_j = s'_j$ then $s_j \stackrel{i}{\sim}_{\mathcal{M}_j} s'_j$ by definition of $\stackrel{i}{\sim}_{\mathcal{M}_j}$. Now suppose $s_j \neq s'_j$. Since \mathcal{M} is locally generated by Φ_1, \dots, Φ_n , s and s' agree on the truth values of the atomic propositions in Φ_i . Therefore $p_j \notin \Phi_i$, thus $s_j \stackrel{i}{\sim}_{\mathcal{M}_j} s'_j$.

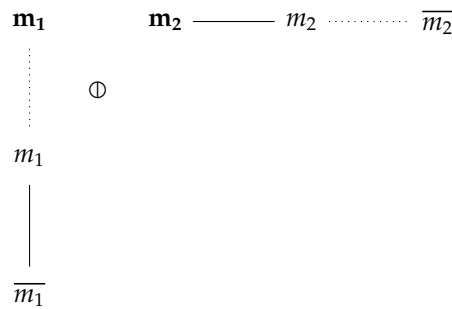
For Zag: Suppose $(s_0, s_1, \dots, s_k) \stackrel{i}{\sim}_{\mathcal{M}_i} (s'_0, s'_1, \dots, s'_k)$ and $sZ(s_0, s_1, \dots, s_k)$, we have $s_0 = s$ and for any $j \leq k$: $s_j \stackrel{i}{\sim}_{\mathcal{M}_j} s'_j$. By the definition of \mathcal{M}_i , we have $s_j = s'_j$ or $s_j \neq s'_j$ and $p_j \notin \Phi_i$. Namely, for $p_j \in \Phi_i$: $s_j = s'_j$. Therefore s and s' agree on the truth values of the propositions in Φ_i . Since \mathcal{M} is locally generated by Φ_1, \dots, Φ_n then $s \stackrel{i}{\sim}_{\mathcal{M}} s'$. This proves the Zag condition. \star

According to the above theorem, a locally generated model by sets of atomic propositions can be decomposed by components based on each atomic proposition. This gives us the desired decomposition of the n -Muddy Children models as in Example 5.2.2.

Theorems 5.3.3 and 5.3.4 show that we can decompose a locally generated model. On the other hand, there are models which are not locally generated but decomposable in a non-trivial way. For example, consider the following model (to ease the presentation, we use solid lines for agent 1 and dotted lines for agent 2):



This model is not bisimilar to any propositionally differentiated model. From Proposition 5.3.1 follows that it is not bisimilar to any locally-generated model. Nevertheless, \mathcal{M} can be decomposed into two models as follows:



If we take the boldface states as the real worlds in these two models respectively, then the two models capture the situations where agent 2 is not sure whether 1 knows m_1 and agent 1 is not sure whether 2 knows m_2 . If we interpret m_1 and m_2 as in Muddy Children, then the composed model, when taking the top-left corner state as the real world, captures the situation where the children can see each other's faces but are not sure whether the other has a mirror (actually they do have mirrors). Since the vocabularies of the above two models are disjoint, from Proposition 5.2.13, we know that any true claim about only m_2 or m_1 will be preserved at the components. For example, agent 1 knows agent 2 does not know whether agent 1 knows m_1 can be verified in the left hand component model.

5.4 Composing Updates

Recall Definition 2.3.2 that an (S5) event model $\mathcal{A} = (E, \mathbf{I}, \leftrightarrow, Pre)$ is like a static model, but with valuations replaced by precondition formulas taken from an appropriate language. Let $\mathbf{P}_{\mathcal{A}}$ be the set of proposition letters appearing in the preconditions of $e \in E$ according to Pre .

Note that the standard product update as in Definition 2.3.3 is defined on the pairs of a static model and an event model where $\mathbf{P}_{\mathcal{A}} \subseteq \mathbf{P}_{\mathcal{M}}$. We will now generalize the standard product update to an operation that works on Kripke models and event models with arbitrary vocabularies.

Model expansion is used in the following definition of product update to ensure that no matter what the vocabulary of the static model is, we can always check the preconditions of the events model on the static model. The vocabulary of the resulting updated model is the union of the vocabulary of the static model and the vocabulary of the event model.

5.4.1. DEFINITION. (Extended Product Update) Given a static model $\mathcal{M} = (S, \mathbf{P}, \mathbf{I}, \sim, V)$ and an event model $\mathcal{A} = (E, \mathbf{I}, \leftrightarrow, Pre)$ for the same set of agents \mathbf{I} . Let X be the differential vocabulary, i.e., $X = \mathbf{P}_{\mathcal{A}} - \mathbf{P}$. Then the extended product update $\mathcal{M} \otimes \mathcal{A}$ is the static model $(S', \mathbf{P} \cup \mathbf{P}_{\mathcal{A}}, \mathbf{I}, \sim', V')$ given by $(\mathcal{M} \triangleleft X) \otimes \mathcal{A}$, where \otimes denotes the usual update product. \mathfrak{M}

This definition boils down to the following:

1. $S' = \{(s, X', e) \mid s \in S, e \in E, X' \subseteq X, \mathcal{M} \triangleleft X, (s, X') \models Pre(e)\}$,
2. $(s, X', e) \sim'_i (t, X'', f)$ iff $s \sim_i t$ and $e \leftrightarrow_i f$,
3. $V'(s, X', e) = V(s) \cup X'$.

Note that the definition of accessibility relations does not require $X = X'$ since all the different values for the novel atoms are the same for i . From Proposition 5.2.8, we can equivalently (modulo bisimulation) define the update as $(\mathcal{M} \triangleleft \mathbf{P}_{\mathcal{A}}) \otimes \mathcal{A}$. However, for the ease in proofs we will stick to the above definition where \mathcal{M} is expanded with $\mathbf{P}_{\mathcal{A}} - \mathbf{P}_{\mathcal{M}}$. Here is an example of an update with a public announcement “at least one of you is muddy” (i.e., an event model with only one world whose precondition is $m_1 \vee m_2$). As usual, we denote this event model as $!(m_1 \vee m_2)$. Note that the update involves model expansion:

$$\begin{array}{ccc}
 m_1 \xrightarrow{1} \overline{m_1} & & m_1 m_2 \xrightarrow{1} \overline{m_1} m_2 \\
 & \triangleleft \{m_2\} & \left| \begin{array}{c} 1,2 \\ 1,2 \end{array} \right. \\
 & & m_1 \overline{m_2} \xrightarrow{1} \overline{m_1} \overline{m_2} \\
 & & \left| \begin{array}{c} 1,2 \\ 1,2 \end{array} \right. \\
 & & m_1 m_2 \xrightarrow{1} \overline{m_1} m_2 \\
 & & \left| \begin{array}{c} 1,2 \\ 1,2 \end{array} \right. \\
 & & m_1 \overline{m_2}
 \end{array}$$

Here is an update of the other component of the 2-Muddy Children model:

$$\begin{array}{ccc}
 m_2 & & m_1 m_2 \xrightarrow{1,2} \overline{m_1} m_2 \\
 \left| \begin{array}{c} 2 \\ 2 \end{array} \right. & \triangleleft \{m_1\} & \left| \begin{array}{c} 2 \\ 2 \end{array} \right. \\
 \overline{m_2} & & m_1 \overline{m_2} \xrightarrow{1,2} \overline{m_1} \overline{m_2} \\
 & & \left| \begin{array}{c} 1,2 \\ 1,2 \end{array} \right. \\
 & & m_1 m_2 \xrightarrow{1,2} \overline{m_1} m_2 \\
 & & \left| \begin{array}{c} 2 \\ 2 \end{array} \right. \\
 & & m_1 \overline{m_2}
 \end{array}$$

And here is the outcome of composing the two update results:

$$\begin{array}{ccc}
m_1 m_2 \xrightarrow{1} \overline{m_1 m_2} & m_1 m_2 \xrightarrow{1,2} \overline{m_1 m_2} & m_1 m_2 \xrightarrow{1} \overline{m_1 m_2} \\
1,2 \Big| & \oplus \quad 2 \Big| & \Leftrightarrow 2 \Big| \\
m_1 \overline{m_2} & m_1 \overline{m_2} & m_1 \overline{m_2}
\end{array}$$

This is the same as the result of public announcement of $m_1 \vee m_2$ on the composition of $m_1 \xrightarrow{1} \overline{m_1}$ and $m_2 \xrightarrow{2} \overline{m_2}$. In the following, we will show that this outcome is not accidental: updating a composed model yields the same result (modulo bisimulation) as composing the updates of its components, provided the event model has certain property. We call an event model \mathcal{A} *propositionally differentiated* if the preconditions are purely boolean formulas and any two states in \mathcal{A} have mutually exclusive preconditions. For a boolean precondition $Pre(e)$ of e in \mathcal{A} , a vocabulary \mathbf{P} such that $\mathbf{P}_{\mathcal{A}} \subseteq \mathbf{P}$, and a set $X \subseteq \mathbf{P}$, we write $X \vDash_{\mathbf{P}} Pre(e)$ if X (viewed as a valuation for \mathbf{P}) makes $Pre(e)$ true. It is clear that $X \cap \mathbf{P}_{\mathcal{A}} \vDash_{\mathbf{P}} Pre(e)$ iff $X \vDash_{\mathbf{P}} Pre(e)$. In case $\mathbf{P} = \bigcup_{j \in J} \mathbf{P}_{\mathcal{M}_j}$ we write $X \vDash_{\{\mathcal{M}_j | j \in J\}} Pre(e)$.

5.4.2. THEOREM. *If \mathcal{A} is a propositionally differentiated event model then:*

$$(\mathcal{M} \oplus \mathcal{N}) \odot \mathcal{A} \Leftrightarrow (\mathcal{M} \odot \mathcal{A}) \oplus (\mathcal{N} \odot \mathcal{A}).$$

PROOF Let $\mathcal{M}_1 = (\mathcal{M} \oplus \mathcal{N}) \odot \mathcal{A}$ and $\mathcal{M}_2 = (\mathcal{M} \odot \mathcal{A}) \oplus (\mathcal{N} \odot \mathcal{A})$. Let relation $Z \subseteq S_{\mathcal{M}_1} \times S_{\mathcal{M}_2}$ be given by:

$$((s, t, X), e) Z ((s', X_1, e'), (t', X_2, e'')) \text{ iff } s = s', t = t', e = e' = e'' \text{ and } X = X_1 \cap X_2$$

We need to show that Z is a total bisimulation between \mathcal{M}_1 and \mathcal{M}_2 . The totality of Z is proved in Lemma 5.4.3. Here we focus on the three conditions of bisimulation. Suppose $((s, t, X), e)$ and $((s, X_1, e'), (t, X_2, e''))$ exist in \mathcal{M}_1 and \mathcal{M}_2 respectively and $((s, t, X), e) Z ((s, X_1, e'), (t, X_2, e''))$.

For invariance we need to show

$$V_{\mathcal{M}}(s) \cup V_{\mathcal{N}}(t) \cup (X_1 \cap X_2) = V_{\mathcal{M}}(s) \cup X_1 \cup V_{\mathcal{N}}(t) \cup X_2$$

Since the only difference between the left hand side and right hand side is about $X, X_1, X_2 \subseteq \mathbf{P}_{\mathcal{A}}$, then showing the following suffices:

$$(V_{\mathcal{M}}(s) \cup V_{\mathcal{N}}(t) \cup (X_2 \cap X_2)) \cap \mathbf{P}_{\mathcal{A}} = (V_{\mathcal{M}}(s) \cup X_1 \cup V_{\mathcal{N}}(t) \cup X_2) \cap \mathbf{P}_{\mathcal{A}} \quad (\star)$$

Since (s, X_1, e) in $\mathcal{M} \odot \mathcal{A}$ and (t, X_2, e) in $\mathcal{N} \odot \mathcal{A}$ are compatible, we have:

$$(V_{\mathcal{M}}(s) \cup X_1) \cap \mathbf{P}_{\mathcal{A}} = (V_{\mathcal{N}}(t) \cup X_2) \cap \mathbf{P}_{\mathcal{A}} \quad (\star\star)$$

Now let us massage the left hand side of (\star) :

$$\begin{aligned}
& (V_{\mathcal{M}}(s) \cup V_{\mathcal{N}}(t) \cup (X_1 \cap X_2)) \cap \mathbf{P}_{\mathcal{A}} \\
&= ((V_{\mathcal{M}}(s) \cup V_{\mathcal{N}}(t) \cup X_1) \cap (V_{\mathcal{M}}(s) \cup V_{\mathcal{N}}(t) \cup X_2)) \cap \mathbf{P}_{\mathcal{A}} \\
&= ((V_{\mathcal{M}}(s) \cup V_{\mathcal{N}}(t) \cup X_1) \cap \mathbf{P}_{\mathcal{A}}) \cap (V_{\mathcal{M}}(s) \cup V_{\mathcal{N}}(t) \cup X_2) \cap \mathbf{P}_{\mathcal{A}} \\
&= (((V_{\mathcal{M}}(s) \cup X_1) \cap \mathbf{P}_{\mathcal{A}}) \cup (V_{\mathcal{N}}(t) \cap \mathbf{P}_{\mathcal{A}})) \cap (((V_{\mathcal{N}}(t) \cup X_2) \cap \mathbf{P}_{\mathcal{A}}) \cup (V_{\mathcal{M}}(s) \cap \mathbf{P}_{\mathcal{A}})) \\
&= (((V_{\mathcal{M}}(s) \cup X_1) \cap \mathbf{P}_{\mathcal{A}}) \cup (V_{\mathcal{N}}(t) \cap \mathbf{P}_{\mathcal{A}})) \cap ((V_{\mathcal{M}}(s) \cup X_1) \cap \mathbf{P}_{\mathcal{A}}) \cup (V_{\mathcal{M}}(s) \cap \mathbf{P}_{\mathcal{A}})) \text{ (by } (\star\star)) \\
&= ((V_{\mathcal{M}}(s) \cup X_1) \cap \mathbf{P}_{\mathcal{A}}) \cup (V_{\mathcal{N}}(t) \cap V_{\mathcal{M}}(s) \cap \mathbf{P}_{\mathcal{A}}) \\
&= ((V_{\mathcal{M}}(s) \cup X_1) \cap \mathbf{P}_{\mathcal{A}}) \quad (\text{since } V_{\mathcal{N}}(t) \cap V_{\mathcal{M}}(s) \subseteq V_{\mathcal{M}}(s)) \\
&= ((V_{\mathcal{M}}(s) \cup X_1) \cap \mathbf{P}_{\mathcal{A}}) \cup ((V_{\mathcal{N}}(t) \cup X_2) \cap \mathbf{P}_{\mathcal{A}}) \quad (\text{from } (\star\star)) \\
&= (V_{\mathcal{M}}(s) \cup X_1 \cup V_{\mathcal{N}}(t) \cup X_2) \cap \mathbf{P}_{\mathcal{A}}
\end{aligned}$$

This proves the invariance requirement.

Now assume $((s, t, X), e)Z((s, X_1, e_1), (t, X_2, e_2))$ and $((s, t, X), e) \stackrel{i}{\sim} ((s', t', X'), e')$ in \mathcal{M}_1 , then $e = e_1 = e_2$, $s \stackrel{i}{\sim} s'$ in \mathcal{M} , $t \stackrel{i}{\sim} t'$ in \mathcal{N} and $e \stackrel{i}{\sim} e'$ in \mathcal{A} . From totality (Lemma 5.4.3), in \mathcal{M}_2 there exists $((s', X'_1, e'), (t', X'_2, e'))$ for some X'_1 and X'_2 such that

$$((s', t', X'), e')Z((s', X'_1, e'), (t', X'_2, e')).$$

According to the definition of relations in \mathcal{M}_2 , it is not hard to see that

$$((s, X_1, e), (t, X_2, e)) \stackrel{i}{\sim} ((s', X'_1, e'), (t', X'_2, e'))$$

This proves the Zig requirement.

Suppose $((s, t, X), e)Z((s, X_1, e_1), (t, X_2, e_2))$ and

$$((s, X_1, e), (t, X_2, e)) \stackrel{i}{\sim} ((s', X'_1, e'), (t', X'_2, e'))$$

Therefore $s \stackrel{i}{\sim}_{\mathcal{M}_1} s'$, $t \stackrel{i}{\sim}_{\mathcal{M}_2} t'$, and $e \stackrel{i}{\sim}_{\mathcal{A}} e'$. From Lemma 5.4.3, in \mathcal{M}_1 there exists $((s', X', e'))$ for some X' such that:

$$((s', t', X'), e')Z((s', X'_1, e'), (t', X'_2, e'))$$

It follows that $((s, t, X), e) \stackrel{i}{\sim} ((s', t', X'), e')$. This proves the Zag condition. \star

5.4.3. LEMMA. *The relation Z defined above is total.*

PROOF We need to show for any state u that exists in \mathcal{M}_1 there is a state v exists in \mathcal{M}_2 such that uZv , and for any v exists in \mathcal{M}_2 there is an u in \mathcal{M}_1 such that uZv .

Suppose $((s, t, X), e)$ exists in \mathcal{M}_1 then the following hold:

Fact 1 $X \subseteq \mathbf{P}_{\mathcal{A}} - (\mathbf{P}_{\mathcal{M}} \cup \mathbf{P}_{\mathcal{N}})$;

Fact 2 $V_{\mathcal{M}}(s) \cap \mathbf{P}_{\mathcal{N}} = V_{\mathcal{N}}(t) \cap \mathbf{P}_{\mathcal{M}}$;

Fact 3 $V_M(s) \cup V_N(t) \cup X \vDash_{M,N,\mathcal{A}} \text{Pre}(e)$.

Now we let:

$$X_1 = X \cup ((V_N(t) - V_M(s)) \cap \mathbf{P}_{\mathcal{A}}) \text{ and } X_2 = X \cup ((V_M(s) - V_N(t)) \cap \mathbf{P}_{\mathcal{A}}).$$

Clearly $X = X_1 \cap X_2$. To show $((s, X_1, e), (t, X_2, e))$ exists in \mathcal{M}_2 , we need to show:

1. X_1 and X_2 are well-defined: $X_1 \subseteq \mathbf{P}_{\mathcal{A}} - \mathbf{P}_M$ and $X_2 \subseteq \mathbf{P}_{\mathcal{A}} - \mathbf{P}_N$.
2. e can be executed on both (s, X_1) and (t, X_2) : $V_M(s) \cup X_1 \vDash_{M,\mathcal{A}} \text{Pre}(e)$ and $V_N(t) \cup X_2 \vDash_{N,\mathcal{A}} \text{Pre}(e)$.
3. (s, X_1, e) and (t, X_2, e) can be composed: $(V_M(s) \cup X_1) \cap (\mathbf{P}_N \cup \mathbf{P}_{\mathcal{A}}) = (V_N(t) \cup X_2) \cap (\mathbf{P}_M \cup \mathbf{P}_{\mathcal{A}})$.

For (1): Recall that $V_M(s) \cap \mathbf{P}_N = V_N(t) \cap \mathbf{P}_M$, thus we have $V_N(t) \cap \mathbf{P}_M \subseteq V_M(s)$ and $V_M(s) \cap \mathbf{P}_N \subseteq V_N(t)$. Therefore, $(V_N(t) - V_M(s)) \cap \mathbf{P}_M = \emptyset$ and $(V_M(s) - V_N(t)) \cap \mathbf{P}_N = \emptyset$. It means $((V_N(t) - V_M(s)) \cap \mathbf{P}_{\mathcal{A}}) \subseteq \mathbf{P}_{\mathcal{A}} - \mathbf{P}_M$ and $((V_M(s) - V_N(t)) \cap \mathbf{P}_{\mathcal{A}}) \subseteq \mathbf{P}_{\mathcal{A}} - \mathbf{P}_N$. Also note that $X \subseteq \mathbf{P}_{\mathcal{A}} - (\mathbf{P}_M \cup \mathbf{P}_N) \subseteq \mathbf{P}_{\mathcal{A}} - \mathbf{P}_M$ and similarly $X \subseteq \mathbf{P}_{\mathcal{A}} - \mathbf{P}_N$. Therefore by the definitions of X_1 and X_2 we have $X_1 \subseteq \mathbf{P}_{\mathcal{A}} - \mathbf{P}_M$ and $X_2 \subseteq \mathbf{P}_{\mathcal{A}} - \mathbf{P}_N$.

For (2): By the definition of X_1 :

$$V_M(s) \cup X_1 = V_M(s) \cup X \cup ((V_N(t) - V_M(s)) \cap \mathbf{P}_{\mathcal{A}})$$

Then we have:

$$\begin{aligned} & (V_M(s) \cup X_1) \cap \mathbf{P}_{\mathcal{A}} \quad (\#) \\ &= (V_M(s) \cup X \cup ((V_N(t) - V_M(s)) \cap \mathbf{P}_{\mathcal{A}})) \cap \mathbf{P}_{\mathcal{A}} \\ &= ((V_M(s) \cup X) \cap \mathbf{P}_{\mathcal{A}}) \cup ((V_N(t) - V_M(s)) \cap \mathbf{P}_{\mathcal{A}}) \\ &= (V_M(s) \cup V_N(t) \cup X) \cap \mathbf{P}_{\mathcal{A}} \end{aligned}$$

Since $V_M(s) \cup V_N(t) \cup X \vDash_{M,N,\mathcal{A}} \text{Pre}(e)$ we have

$$(V_M(s) \cup V_N(t) \cup X) \cap \mathbf{P}_{\mathcal{A}} \vDash_{M,N,\mathcal{A}} \text{Pre}(e)$$

Therefore from the derivation (#), $(V_M(s) \cup X_1) \cap \mathbf{P}_{\mathcal{A}} \vDash_{M,N,\mathcal{A}} \text{Pre}(e)$ and then $V_M(s) \cup X_1 \vDash_{M,\mathcal{A}} \text{Pre}(e)$. Similarly we can prove $V_N(t) \cup X_2 \vDash_{N,\mathcal{A}} \text{Pre}(e)$.

For (3): By the definition of X_1 :

$$(V_M(s) \cup X_1) \cap (\mathbf{P}_N \cup \mathbf{P}_{\mathcal{A}}) = ((V_M(s) \cup X_1) \cap \mathbf{P}_{\mathcal{A}}) \cup ((V_M(s) \cup X_1) \cap \mathbf{P}_N)$$

From (#), we know that: $(V_M(s) \cup X_1) \cap \mathbf{P}_{\mathcal{A}} = (V_M(s) \cup V_N(t) \cup X) \cap \mathbf{P}_{\mathcal{A}}$, thus

$$(V_M(s) \cup X_1) \cap (\mathbf{P}_N \cup \mathbf{P}_{\mathcal{A}}) = ((V_M(s) \cup V_N(t) \cup X) \cap \mathbf{P}_{\mathcal{A}}) \cup (V_M(s) \cap \mathbf{P}_N) \cup (X_1 \cap \mathbf{P}_N) \quad (+)$$

Note that

$$\begin{aligned} & X_1 \cap \mathbf{P}_N \\ &= (X \cup ((V_N(t) - V_M(s)) \cap \mathbf{P}_{\mathcal{A}})) \cap \mathbf{P}_N \\ &= ((V_N(t) - V_M(s)) \cap \mathbf{P}_{\mathcal{A}}) \cap \mathbf{P}_N \quad (\text{since } X \cap \mathbf{P}_N = \emptyset) \\ &= (V_N(t) - V_M(s)) \cap \mathbf{P}_{\mathcal{A}} \quad (\text{since } V_N(t) \subseteq \mathbf{P}_N) \end{aligned}$$

Therefore go back to (†) we have:

$$\begin{aligned} & (V_M(s) \cup X_1) \cap (\mathbf{P}_N \cup \mathbf{P}_A) \\ &= ((V_M(s) \cup V_N(t) \cup X) \cap \mathbf{P}_A) \cup (V_M(s) \cap \mathbf{P}_N) \cup ((V_N(t) - V_M(s)) \cap \mathbf{P}_A) \\ &= ((V_M(s) \cup V_N(t) \cup X) \cap \mathbf{P}_A) \cup (V_M(s) \cap \mathbf{P}_N) \quad (\ddagger) \end{aligned}$$

Similarly we can show

$$(V_N(t) \cup X_2) \cap (\mathbf{P}_M \cup \mathbf{P}_A) = ((V_M(s) \cup V_N(t) \cup X) \cap \mathbf{P}_A) \cup (V_N(t) \cap \mathbf{P}_M) \quad (\S)$$

From the **Fact 1** ($V_N(t) \cap \mathbf{P}_M = V_M(s) \cap \mathbf{P}_N$), (†) and (§) we have:

$$(V_M(s) \cup X_1) \cap (\mathbf{P}_N \cup \mathbf{P}_A) = (V_N(t) \cup X_2) \cap (\mathbf{P}_M \cup \mathbf{P}_A)$$

This proves (3).

Till now we have proved that for any state u that exists in \mathcal{M}_1 there is a state v exists in \mathcal{M}_2 such that uZv . Now suppose $((s, X_1, e), (t, X_2, e'))$ exists in \mathcal{M}_2 we need to show that there is an u in \mathcal{M}_1 such that uZv . Since \mathcal{A} is propositionally differentiated, no two actions can be executed under the same valuation over \mathbf{P}_A , thus events $e = e'$. We now only need to show that $((s, t, X_1 \cap X_2), e)$ exists in \mathcal{M}_1 . Formally we need to verify the following claims:

1. s and t are compatible.
2. X is well-defined: $X_1 \cap X_2 \subseteq \mathbf{P}_A - (\mathbf{P}_M \cup \mathbf{P}_N)$.
3. e can be executed on $(s, t, X_1 \cap X_2)$: $V_M(s) \cup V_N(t) \cup (X_1 \cap X_2) \vDash_{\mathcal{M}, N, \mathcal{A}} \text{Pre}(e)$.

From the existence of $((s, X_1, e), (t, X_2, e))$, clearly s and t can be composed. Since $X_1 \subseteq \mathbf{P}_A - \mathbf{P}_M$ and $X_2 \subseteq \mathbf{P}_A - \mathbf{P}_N$, (2) is also straightforward. Now we prove (3). Since $((s, X_1, e)$ and (t, X_2, e')) can be composed

$$V_M(s) \cup X_1 \cup V_N(t) = V_M(s) \cup X_2 \cup V_N(t)$$

Now we have:

$$\begin{aligned} & V_M(s) \cup X_1 \cup V_N(t) \\ &= (V_M(s) \cup X_1 \cup V_N(t)) \cap (V_M(s) \cup X_2 \cup V_N(t)) = (V_M(s) \cup V_N(t) \cup (X_1 \cap X_2)) \end{aligned}$$

Since $((s, X_1, e), (t, X_2, e))$ exists, it is not hard to see that $(V(s) \cup V(t) \cup X_1) \cap \mathbf{P}_A \vDash_{\mathcal{M}, N, \mathcal{A}} \text{Pre}(e)$. Thus $V_M(s) \cup V_N(t) \cup (X_1 \cap X_2) \vDash_{\mathcal{M}, N, \mathcal{A}} \text{Pre}(e)$. \spadesuit

Event models are very similar to static models, and it turns out that composition on event models can be defined in a natural way.

5.4.4. DEFINITION. (Merging Composition of Event Models) The composition $\mathcal{A} \oplus \mathcal{B}$ of two event models \mathcal{A} and \mathcal{B} with the same set of agents \mathbf{I} is given by $(E, \mathbf{I}, \leftrightarrow, \text{Pre})$, where:

- $E = \{(e, f) \mid e \in E_{\mathcal{A}}, f \in E_{\mathcal{B}}\}$
- $(e, f) \leftrightarrow_i (e', f')$ iff $e \leftrightarrow_i e'$ in \mathcal{A} and $f \leftrightarrow_i f'$ in \mathcal{B}
- $Pre((e, f)) = Pre_{\mathcal{A}}(e) \wedge Pre_{\mathcal{B}}(f)$.

□

Note that in the composed event model, some e may have an unsatisfiable precondition. We do not delete such non-executable actions in the composed model. The simplest example is composing two announcements $!\phi$ and $!\psi$, which results in an announcement of $\phi \wedge \psi$. The composition operator presented here can be viewed as a kind of parallel compositions of events. Consider the following example (where $\mathbf{I} = \{1, 2\}$ and the propositions are preconditions):

$$\begin{array}{ccc}
 \begin{array}{c} p \\ \left| \begin{array}{c} 2 \\ \oplus \end{array} \right. \\ q \end{array} & \begin{array}{c} \bar{q} \xrightarrow{1} \bar{p} \end{array} & = & \begin{array}{c} p\bar{q} \xrightarrow{1} p\bar{p} \\ \left| \begin{array}{c} 2 \\ \text{pruned} \end{array} \right. \\ q\bar{q} \xrightarrow{1} \bar{p}q \end{array} & & \begin{array}{c} p\bar{q} \\ \bar{p}q \end{array}
 \end{array}$$

The first model captures the event that agent 1 is being told that either p or q is true, while agent 2 can only see it without hearing the exact message. Similarly, the second model reflects the event that 2 is being told either p or q is false without 1 hearing the message. The composition of the two captures that both events are happening at the same time. As we can see, the effect of updating this composed event is the same as updating an announcement $p \wedge \neg q$ or $\neg p \wedge q$. Intuitively, if agent 1 is told p and he knows that 2 is (truthfully) told either $\neg q$ or $\neg p$ then he actually knows that $\neg p$.

Updating with a composite event model should yield the same outcome as first updating with its components and then composing the results. The following theorem says that it does, notably, without any restriction to certain class of models.

5.4.5. THEOREM. $\mathcal{M} \otimes (\mathcal{A} \oplus \mathcal{B}) \Leftrightarrow (\mathcal{M} \otimes \mathcal{A}) \oplus (\mathcal{M} \otimes \mathcal{B})$.

PROOF Let $\mathcal{M}_1 = \mathcal{M} \otimes (\mathcal{A} \oplus \mathcal{B})$ and $\mathcal{M}_2 = (\mathcal{M} \otimes \mathcal{A}) \oplus (\mathcal{M} \otimes \mathcal{B})$. Let the relation $Z \subseteq S_{\mathcal{M}_1} \times S_{\mathcal{M}_2}$ be given by:

$$(s, X, (e, f)) Z ((s', X_1, e'), (s'', X_2, f')) \text{ iff } s = s' = s'', e = e', f = f' \text{ and } X = X_1 \cup X_2$$

We first show Z is total.

\Rightarrow : Suppose that $(s, X, (e, f))$ is in $S_{\mathcal{M}_1}$, we need to show that there are X_1 and X_2 with $X = X_1 \cup X_2$ such that $((s, X_1, e), (s, X_2, f))$ exists in $S_{\mathcal{M}_2}$. Notice that:

$$\begin{aligned}
 & (s, X, (e, f)) \in S_{\mathcal{M} \otimes (\mathcal{A} \oplus \mathcal{B})} \\
 \Leftrightarrow & \mathcal{M} \triangleleft ((\mathbf{P}_{\mathcal{A}} \cup \mathbf{P}_{\mathcal{B}}) - \mathbf{P}_{\mathcal{M}}), (s, X) \models Pre_{\mathcal{A}}(e) \wedge Pre_{\mathcal{B}}(f) \\
 \Leftrightarrow & \mathcal{M} \triangleleft (\mathbf{P}_{\mathcal{A}} - \mathbf{P}_{\mathcal{M}}), (s, X \cap (\mathbf{P}_{\mathcal{A}} - \mathbf{P}_{\mathcal{M}})) \models Pre_{\mathcal{A}}(e) \\
 \text{and } & \mathcal{M} \triangleleft (\mathbf{P}_{\mathcal{B}} - \mathbf{P}_{\mathcal{M}}), (s, X \cap (\mathbf{P}_{\mathcal{B}} - \mathbf{P}_{\mathcal{M}})) \models Pre_{\mathcal{B}}(f) \quad (\text{From Proposition 5.2.10})
 \end{aligned}$$

Now let $X_1 = X \cap (\mathbf{P}_{\mathcal{A}} - \mathbf{P}_{\mathcal{M}})$ and $X_2 = X \cap (\mathbf{P}_{\mathcal{B}} - \mathbf{P}_{\mathcal{M}})$. Since $X \subseteq (\mathbf{P}_{\mathcal{A}} \cup \mathbf{P}_{\mathcal{B}}) - \mathbf{P}_{\mathcal{M}}$, $X_1 \cup X_2 = X \cap ((\mathbf{P}_{\mathcal{A}} - \mathbf{P}_{\mathcal{M}}) \cup (\mathbf{P}_{\mathcal{B}} - \mathbf{P}_{\mathcal{M}})) = X$. From the above derivation, $(s, X_1, e) \in S_{M \otimes \mathcal{A}}$ and $(s, X_2, f) \in S_{M \otimes \mathcal{B}}$ exist and they are compatible. Therefore $((s, X_1, e), (s, X_2, f)) \in S_{M_2}$.

\Leftarrow : Suppose $((s, X_1, e), (s, X_2, f))$ exists in S_{M_2} . Let $X = X_1 \cup X_2$, clearly $X \subseteq (\mathbf{P}_{\mathcal{A}} \cup \mathbf{P}_{\mathcal{B}}) - \mathbf{P}_{\mathcal{M}}$. It is then easy to show that $(s, X, (e, f)) \in S_{M_1}$.

The invariance is straightforward since $X = X_1 \cup X_2$. Based on totality, Zig and Zag properties of Z are immediate. \spadesuit

5.5 Discussion and Future Work

In this chapter, we presented a preliminary report on composing static models and event models with different vocabularies. We studied the algebraic properties of the newly introduced composition operator with the presence of the product update. We gave some results on the decomposition of locally generated models³. Definitely more questions about decomposability of models should be asked. For example, in [CLDQ09] a symmetry reduction technique is proposed in the setting of multi-agent systems, while it is not very clear whether every symmetric model can be decomposed in a non-trivial way by merging compositions. Note that non-symmetric models can also be decomposed: there are locally-generated models which do not have non-trivial symmetric structures⁴. More ambitious agenda is to logically characterize some decomposable classes of models. To systematically answer such decomposition questions, we may need to find help in both graph theory and modal logic.

Ditmarsch and French [vDF09] studied \Leftarrow between models with the same vocabulary in the context of product updates. They prove that when restricted to finite models:

$M, e \Leftarrow N, t$ iff there exists an event model (\mathcal{A}, e) such that $N, t \Leftrightarrow M \otimes \mathcal{A}, (s, e)$. Compared to this result, our Theorem 5.2.6 requires a much stronger condition. The reason is that essentially we only have very weak propositional “preconditions” when composing the model: matching the values of the basic propositions in the common vocabulary. This motivates the future extension with more complicated *matching* conditions when composing the models.

In Theorem 5.3.3 we decompose a locally-generated model w.r.t. the merging composition only. However, we can also decompose the model by composing M_1, \dots, M_n and then, instead of composing M_0 , restrict the resulting model by a public announcement of the possible states of affairs. Here a more general question arises naturally: what are the natural basic operations to construct a model besides

³We gave a characterization of models that can be decomposed into two-world building blocks in [vEWS10]. However, the criteria are not very intuitive, thus we have chosen to omit it here.

⁴When the sets of observables are not symmetric, the locally-generated model may be asymmetric.

composition? Is there a *normal form* of any model \mathcal{M} by composition, relativization (public announcement) and perhaps general product updates? Again some clues were hinted at in [vDF09]: the author showed that a simulation can be seen as a bisimulation transformation followed by a model restriction. We leave further studies on this topic to future work.

The combination of communicative actions and vocabulary expansion deserves further study. There is an obvious connection to the dynamics of awareness, as studied in [vBVQ09, dJ09], while our expansion operation can be seen as an action for (publicly) extending the awareness set. In this chapter we fix the set of agent in our discussion, while in other applications (e.g., about awareness) expansions with extra agents may be also relevant.

Finally our investigation is encouraging for epistemic model checking with dynamic epistemic logic, for it suggests ways to check relevant epistemic properties on small components of large models. We will pursue this line of research further in Chapter 7 of this thesis.