Action Emulation

Jan van Eijck[1,2], Ji Ruan[3] and Tomasz Sadzik[4]
CWI, Amsterdam[1], Uil-OTS, Utrecht[2], University of Liverpool[3], Stanford University[4]

January 24, 2008

Abstract

The effects of public announcements, private communications, deceptive messages to groups, and so on, can all be captured by a general mechanism of updating multi-agent models with so-called update action models [3], now in widespread use (see [9] for a textbook treatment). There is a natural extension of the definition of a bisimulation to action models. Surely enough, updating with bisimilar action models gives the same result (modulo bisimulation). But the converse turns out to be false: there are examples of pairs of non-bisimilar update models with the same update effect. This paper is a quest for a notion of structural equivalence that is more appropriate for action models than bisimulation. We propose action emulation as a notion of structural equivalence more appropriate for action models, and generalizing standard bisimulation. It is proved that action emulation provides a full characterisation of update effect. The important case of updating with purely propositional information is treated first, as an easy stepping stone for the more general treatment.

1 Introduction

In everyday life, all our actions are conditioned by information. As soon as we get up, we have to make decisions on what to wear, on what to eat, on whether to take an umbrella. To take those decisions, we need information: check the refrigerator to see whether there is milk, check the weather forecast to see whether there is likely to be rain. Even if you wake up alone, there is multi-agent interaction, for it is possible (and sometimes natural) to consider the surrounding universe as an agent. Typically, agents interact without full information about the state of the whole system. If I send an email, then you may or may not know that it arrived, I may or may not know whether you checked your inbox, and so on. Knowledge and (lack of) knowledge about knowledge plays a key role in the interaction of agents. In systems that deal with public announcements (such as emails with lots of cc's), common knowledge is created: everyone on a cc list knowns that the content of the email is now common knowledge. To handle the effects of communicative actions that include public announcements, group announcement and private messages one needs a powerful logic that can (at least) express common knowledge.

In epistemic logic [12] knowledge is represented with multi-agent Kripke models (or possible world models) that contain for each agent an accessibility relation pointing at the situations that the agent considers possible. These models encode the state of the system. To talk about the system, a logical language is used that allows one to express things like 'agent a considers φ possible' (this would express that φ is consistent with what a knows or believes), or 'in all states that are linked to the current state via a and b accessibilities, φ is the case' (this would express common knowledge of a and b that φ). While standard epistemic logics do not deal with change, Dynamic Epistemic Logic (DEL) does: it introduces the representation of *actions*, and the method of updating a situation with the actions of agents. This area has seen a strong development in recent years, witness Gerbrandy [11], van Ditmarsch [8], van Benthem [5, 6], and Baltag, Moss and co-workers [3, 1, 2].

Our own version of DEL is based loosely on the Logic of Communication and Change (LCC) of [7], which is perhaps the most streamlined version of DEL so far. The basic insight is from [3]: a wide variety of information updates can be treated using a formal product construction with an action model, which is nothing but a multi-agent Kripke model with the valuations replaced by precondition formulas. The reason for this to work is that actions with epistemic effects are quite similar to situations with epistemic aspects. The uncertainty of agents about which action takes place is a lot like the uncertainty of agents about what is the case. If you receive a message φ and I am left in the dark, then this is modelled as an action that allows you to distinguish the φ situations from the rest, while I am not allowed to make that distinction. If the two of us get the φ message, and some outsider does not, then it makes a real difference whether the two of us know of each other that we get the same information, and this again is encoded in the update model. In this paper, we study equivalence of update actions: two update actions are equivalent if they always produce non-distinguishable results. Our main contribution is that we propose a concept called *action emulation* to characterize such equivalence, and prove that our characterization is correct.

The structure of the paper is as floows. In section 2, we present the version of Dynamic Epistemic Logic we work with, and motivate our choice. Also, we give a formal definition of the notion of 'same update effect' that we want to capture. Next, we make a distinction, and treat a number of cases in order of increasing complexity. The first case is the technically interesting case where the update actions are in fact ordinary epistemic models: this is achieved by taking as precondition formulas full factual descriptions of states of affairs (Section 3). The next case is the important one of updating with factual information, but using an arbitrarily complex communication pattern. Most of our everyday communications are like this. We exchange factual information, deciding whether to send cc's or not, we decide to keep some facts to ourselves, or only tell them to a few close friends. The epistemic pattern of *how* the information is conveyed may be incredibly complex, as when we decide to send private letters of invitation to a large group of acquaintances, but with a cc to our spouse (Section 4). The final threshold (that we cross in Section 5) is that we allow the conveyed information itself to contain epistemic elements. This turns out to be the most difficult case. It concerns examples like public admissions of ignorance "I can assure all of you that I did not know it at the time", or utterances like "the thing I am telling you now is something that nobody knowns yet". A famous example is the Moore sentence "φ is the case and you don't know it yet". Conveying this information makes it false, for after receiving it you *do* know that φ. Now it might seem that there is still a fourth, most general, case to consider: the case where the information that is conveyed refers to other information updates and their effects. As will be explained, this final case does not add new complexities: our epistemic logic was carefully designed in such manner that all descriptions of the effects of update actions can be translated to purely epistemic formulas. So this ends our treatment, and

Section 6 gives conclusions and questions for further research.

2 Dynamic Epistemic Logic and the Same Update Effect

In this section we formally introduce epistemic models (or multiagent Kripke models), followed by a tandem definition of action models and a suitable epistemic language. The definition of action models and epistemic language has to be in tandem (by mutual recursion), for the action models use formulas of the language as preconditions, and the action models themselves occur as operators in the language. The epistemic models capture a static description of what agents know about the world and about each other, the action models capture the ways of modifying these static systems by providing information (in a very general sense).

Definition 1 (Epistemic models) *Let a set of propositional variables P and a finite set of agents* Ag *be given. An epistemic model is a triple $M = (W, V, R)$ where W is a set of worlds, $V : W \to \mathcal{P}(P)$ assigns a valuation to each world $w \in W$, and $R : \mathrm{Ag} \to \mathcal{P}(W^2)$ assigns an accessibility relation $\xrightarrow{i}$ to each agent $i \in \mathrm{Ag}$.*

A pair (M, U) with $U \subseteq W$ is a multiple pointed epistemic model, indicating that the actual world is among U.

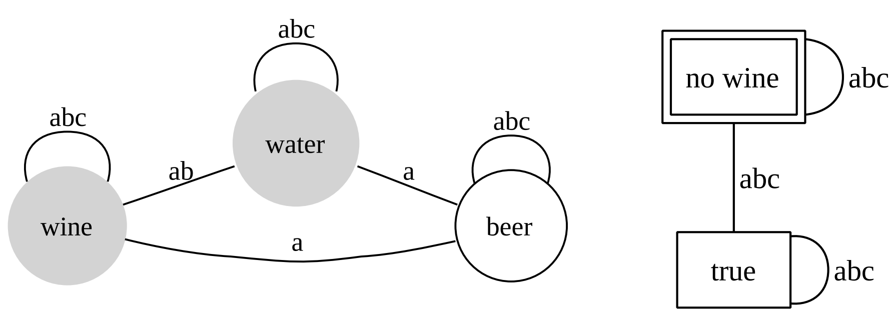

Figure 1: Epistemic model and update

Figure 1 gives an example of an epistemic model (on the left) that is either a wine or a water situation. The grey shaded situations are the distinguished points. In a distinguished point a knows nothing about the available drink, b knows it is either wine or water, and c knows that it is wine if it is wine, and that it is water if it is water. In other words, c knows what's for drinks. But note that the situation captures much more: for instance, it is common knowledge that c knows what's for drinks, and it is also common knowledge that a does not know. Baltag, Moss and Solecki [3] proposed to model updates on epistemic models as action models: epistemic models with valuations replaced by preconditions. An example is in Figure 1 on the right. The actual action (indicated by a double box) is that the information is conveyed that there is no wine. The agents a, b and c cannot distinguish this action from an action where no information is conveyed. The result of the action model update should be that the wine situation remains available as an epistemic option for the agents, but it removed from the set of actual worlds.

Here is the definition of action models. The language in which the information is conveyed will be given in the next definition.

Definition 2 (Action models for a given language $\mathcal{L}$) *Let a finite set of agents* Ag *and an epistemic language $\mathcal{L}$ be given. An action model for $\mathcal{L}$ is a triple $A = (W, pre, R)$ where W is a set of action states, $pre : W \to \mathcal{L}$ assigns a precondition to each action state, and $R : \mathrm{Ag} \to \mathcal{P}(W^2)$ assigns an accessibility relation $\xrightarrow{i}$ to each agent $i \in \mathrm{Ag}$.*

A pair (A, S) with $S \subseteq W$ is a multiple pointed action model, indicating that the actual action that takes place is a member of S.

The epistemic language $\mathcal{LANG}$ used in the action models is defined as follows.

Definition 3 ($\mathcal{LANG}$) *Assume p ranges over set of basic propositions P, i ranges over the set of agents* Ag*. The formulas of $\mathcal{LANG}$ are given by:*

$$\begin{aligned} \varphi &::= \top \mid p \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \Box_\alpha \varphi \mid [A, S]\varphi, \\ \alpha &::= i \mid ?\varphi \mid \alpha_1 \cup \alpha_2 \mid \alpha_1 ; \alpha_2 \mid \alpha^*, \end{aligned}$$

where (A, S) is a multiple pointed finite $\mathcal{LANG}$ (action) model.

We employ the usual abbreviations. In particular, $\varphi_1 \vee \varphi_2$ is shorthand for $\neg(\neg\varphi_1 \wedge \neg\varphi_2)$, $\varphi_1 \to \varphi_2$ for $\neg(\varphi_1 \wedge \neg\varphi_2)$, $\Diamond_\alpha\varphi$ for $\neg\Box_\alpha\neg\varphi$, $\langle A, S\rangle\varphi$ for $\neg[A, S]\neg\varphi$.

This language is more expressive than the usual epistemic languages, even apart from the presence of action models as modal operators. For instance, K_i, the knowledge operator for a, looks like $\Box_i$, while the common knowledge for a and b is given by by $\Box(a \cup b)^*$. Relativized common knowledge is expressed by $(?\varphi; a \cup b)^*$: this epistemic program talks about paths along a and b links, where everywhere on the path φ is true. The advantage of this expressiveness reveals itself when one wants to express the epistemic effects of (say) public announcements. How can we express epistemically that public announcement of φ created the common knowledge among a and b that ψ? For that we have to describe the situation in the model *before the announcement*, and common knowledge relativized to φ expresses just this.

The advantage of the availability of regular epistemic programs α turns out to be even more formidable: Van Benthem, Van Eijck and Kooi prove in [7] that the effects of every update action with an action model A, S can be described in purely epistemic terms. This means that from now on we can work with the sublanguage of $\mathcal{LANG}$ with expressions of the form $[A, S]\varphi$ removed. This is in fact the language of Epistemic Propositional Dynamic Logic (E-PDL) that will be used in section 5. This is the most general case, preconditions in action models that contain themselves action models can be replaced by equivalent purely epistemic preconditions.

There are a number of reasons for employing *multiple* pointed models for representing situations and for updating.

- In the first place, it allows us to generalize over a number of situations in a straightforward way. See the example above, where we could leave it undetermined whether the actual situation is a wine situation or a water situation.

- Similarly for the action models: it allows us to handle choice in a straighforward way. Consider an action of revealing the truth about φ as a choice between announcing φ if that is the case, and announcing its negation if $\neg\varphi$ is the case.
- It simplifies the definition of the update process. Updates are always defined, but they may result in epistemic models with an empty domain or with an empty set of distinguished worlds.
- Finally, we found that the action emulations that we are going to use relate sets of points to sets of points, so it matches well with allowing multiplepointedness. See Section 4 for further details.

With multiple pointed action models, the multiple points constrain the whereabouts of the actual action, with multiple pointed epistemic models they constrain the whereabouts of the actual world.

Let MOD be the class of multiple pointed epistemic models and ACT the class of multiple pointed finite $\mathcal{LANG}$ models. Then $\mathcal{LANG}$-update is an operation of the following type:

$$\otimes : \text{MOD} \times \text{ACT} \rightarrow \text{MOD}.$$

The operation $\otimes$ and the truth definition for $\mathcal{LANG}$ are defined by mutual recursion, as follows.

Definition 4 (Update, Truth) *Given a multiple pointed epistemic model (M, U) and an action model (A, S), we define*

$$(M, U) \otimes (A, S)$$

as

$$((W', V', R'), U'),$$

where

$$\begin{aligned}
W' &:= \{(w, s) \mid w \in W_M, s \in W_A, M \models_w pre_s\}, \\
V'(w, s) &:= V_M(w), \\
(w, s) \xrightarrow{i} (w', s') \in R' &:= w \xrightarrow{i} w' \in R_M, s \xrightarrow{i} s' \in R_A, \\
U' &:= \{(u, s) \mid u \in U, s \in S, (u, s) \in W'\},
\end{aligned}$$

and where the truth definition is given by:

$$\begin{aligned}
M \models_w \top & \quad && \textit{always} \\
M \models_w p &:\equiv && p \in V_M(w) \\
M \models_w \neg\varphi &:\equiv && \textit{not } M \models_w \varphi \\
M \models_w \varphi_1 \wedge \varphi_2 &:\equiv && M \models_w \varphi_1 \textit{ and } M \models_w \varphi_2 \\
M \models_w \Box_\alpha \varphi &:\equiv && \textit{for all } w' \textit{ with } w \xrightarrow{\alpha} w' \; M \models_{w'} \varphi \\
M \models_w [A, S]\varphi &:\equiv && (W', V', R') \models_{(w,s)} \varphi \textit{ for all } (w, s) \in U', \\
& && \textit{where } ((W', V', R'), U') = (M, \{w\}) \otimes (A, S).
\end{aligned}$$

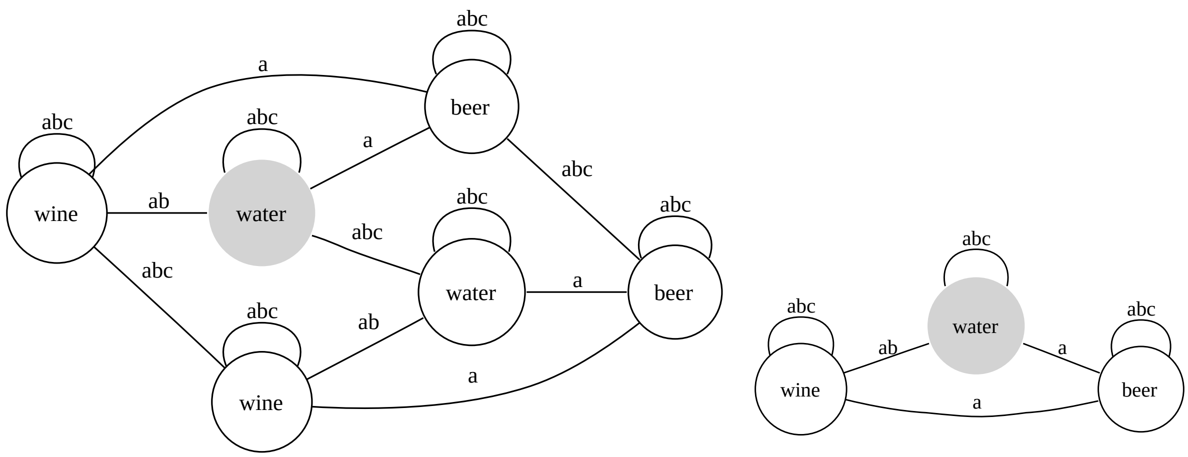

Figure 2: Two bisimilar epistemic models

In this definition, $\stackrel{?\varphi}{\to}$ is the relation $\{(x,x) \mid M \models_x \varphi\}$, $\stackrel{\alpha_1 \cup \alpha_2}{\to}$ is the relation $\stackrel{\alpha_1}{\to} \cup \stackrel{\alpha_2}{\to}$, $\stackrel{\alpha_1;\alpha_2}{\to}$ is $\{(x,y) | \exists z (x \stackrel{\alpha_1}{\to} z) \& (z \stackrel{\alpha_2}{\to} y)\}$ and $\stackrel{\alpha^*}{\to}$ is the transitive closure of $\stackrel{\alpha}{\to}$.

The standard notion of structural equivalence for epistemic models is bisimulation. The reader may care to check that the result of updating the situation described in Figure 1 with the update model that is also given there, is the situation given in Figure 2 on the left, and that the actual world in this result is bisimilar to the actual world in the figure on the right (those not familiar with the notion of bisimulation can use the definition below). So the effect is indeed that wine is eliminated as a possibility, while the epistemic relations remain unchanged: nobody is aware that something has happened. This is the act of running out of wine, without anybody noticing. It is very different from the private announcement at the Wedding at Cana "They have no more wine".

We give a general definition of bisimulation, that applies to epistemic models and action models alike. Let $\equiv_{\mathcal{L}}$ be the equivalence notion for a appropriate logical language $\mathcal{L}$.

Definition 5 (Bisimulation) *Let X, Y be epistemic (action) models. The relation $C \subseteq W_X \times W_Y$ is a bisimulation if whenever sCt the following hold:*

Invariance $V(s) = V(t)$ *($pre_s \equiv_{\mathcal{L}} pre_t$ and pre_s is consistent).*

Zig *for all $i \in \mathrm{Ag}$, all states s' with $s \stackrel{i}{\to} s'$ there is a state t' with $t \stackrel{i}{\to} t'$ and $s'Ct'$.*

Zag *same requirement vice versa.*

A pointed bisimulation between (X, S) and (Y, T) is a bisimulation that connects each $s \in S$ to some $t \in T$, and vice versa. If there is a pointed bisimulation between (X, S) and (Y, T), this is expressed as $(X, S) \underline{\leftrightarrow} (Y, T)$.

Note that pointed bisimulation is defined as a lift of a bisimulation from the state level to the state-set level. For the action bisimulation, we require that preconditions of actions should be consistent if they are bisimilar. This is natural, since actions with inconsistent preconditions will have no influence on epistemic update anyway: they simply drop out of the picture. The notion can be applied to action models with inconsistent actions, of course, but these inconsistent actions will simply not be in the bisimulation relation.

The following theorem shows that action update preserves ordinary bisimulation on epistemic models:

Theorem 6 (Preservation of epistemic bisimulation; Baltag, Moss, Solecki) *Given multiple pointed epistemic models* $(M, U), (N, K)$*, and an action model* (A, S)*,*

$$(M, U) \underline{\leftrightarrow} (N, K) \text{ implies } (M, U) \otimes (A, S) \underline{\leftrightarrow} (N, K) \otimes (A, S)$$

Thinking of the finite multiple pointed action models **A** as 'action programs', the basic semantic notion of equivalence between such programs is that of having the same update effect: if applied to the same epistemic model, they will yield bisimilar results.

Definition 7 (Same update effect) *Given multiple pointed action models* $(A, S), (B, T)$*,*

$$(A, S) \equiv_{ACT} (B, T) \text{ iff } \forall (M, U) : (M, U) \otimes (A, S) \underline{\leftrightarrow} (M, U) \otimes (B, T).$$

Bisimilar action models have the same update effect:

Proposition 8 (Preservation of action bisimulation) *Action update preserves action bisimulation: given two pointed action models (A,S) and (B,T),*

$$\text{if } (A, S) \underline{\leftrightarrow} (B, T) \text{ then } (A, S) \equiv_{ACT} (B, T).$$

Proof. We have to show that for any epistemic model (M, U) and every (u, s_i) among the pointed worlds of $(M, U) \otimes (A, S)$ there is a (v, t_j) among the actual worlds of $(M, U) \otimes (B, T)$ with $(u, s_i) \underline{\leftrightarrow} (v, t_j)$, and vice versa. This follows immediately from the existence of the bisimulation $\underline{\leftrightarrow}$ between (A, S) and (B, T), for the relation on $M \otimes A \times M \otimes B$ defined by means of

$$(u, s) C (v, t) \text{ iff } u = v \text{ and } s \underline{\leftrightarrow} t$$

is a bisimulation. □

In the next three sections, we characterize the notion of having the same update effect in terms of canonical updates. On the basis of that, we will define a relation on action models directly, called action emulation, and show that this notion exactly captures the update effects of action models.

3 Update Effect of Valuation Precondition Models

To explain in what sense action emulation generalizes standard bisimulation, we show in this Section that the two agree for the special class of action models with preconditions mirroring static valuations. Action models with valuation preconditions are in fact epistemic models masquerading as action models.

Let Q be a finite set of proposition letters. Then a valuation over Q is a subset of Q. For any $\mathbf{v} \in \mathcal{P}(Q)$, let $\overline{\mathbf{v}} :\equiv \bigwedge_{p\in\mathbf{v}} p \wedge \bigwedge_{p\notin\mathbf{v}} \neg p$. A *$Q$-valuation action model* has precondition formulas of the form $\overline{\mathbf{v}}$, for some $\mathbf{v} \in \mathcal{P}(Q)$.

The epistemic model VAL_Q is the model (W, V, R) where $W = \mathcal{P}(Q)$, V is the identity function, and R is the universal relation on W for every agent i. We use VAL_Q^* to denote the multiple pointed model (VAL_Q, W).

Proposition 9 *Given any two Q-valuation pointed action models (A, S) and (B, T),*

$$\textit{If } (A, S) \equiv_{ACT} (B, T) \textit{ then } (A, S) \underline{\leftrightarrow} (B, T)$$

Proof. Suppose $(A, S) \equiv_{\text{ACT}} (B, T)$. We have that $\text{VAL}_Q^* \otimes (A, S) \underline{\leftrightarrow} \text{VAL}_Q^* \otimes (B, T)$. It is immediate that if for any $\mathbf{v} \in \mathcal{P}(Q)$ we identify $\mathbf{v}$ with $\overline{\mathbf{v}}$, then $\text{VAL}_Q^* \otimes (A, S) \underline{\leftrightarrow} (A, S)$ and $\text{VAL}_Q^* \otimes (B, T) \underline{\leftrightarrow} (B, T)$. Due to the transitivity of bisimulation, this finishes the proof. □

Combining proposition 8 and 9 we have:

Theorem 10 *For any two Q-valuation pointed action models (A, S) and (B, T),*

$$(A, S) \equiv_{ACT} (B, T) \textit{ iff } (A, S) \underline{\leftrightarrow} (B, T).$$

The claim of Theorem 10 does not hold for action models in general. Figure 3 provides an example of two non-bisimilar action models with the same update effect.

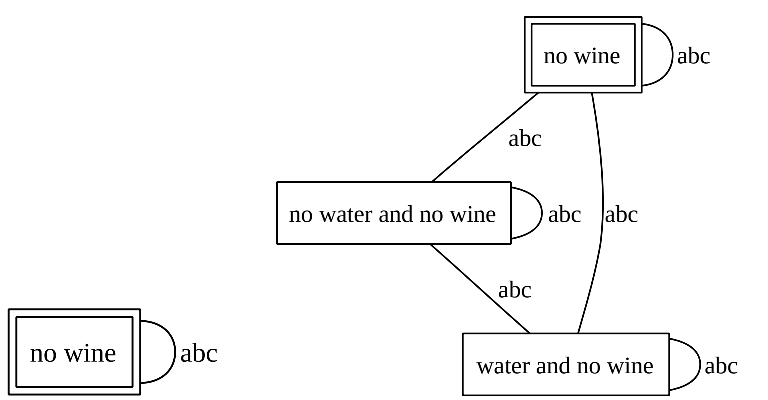

Figure 3: Non-bisimilar actions models with the same update effects

The leftmost action model in Figure 3 consists of a single "no wine" action accessible to all. This is the public announcement *"There is no (more) wine"*. The other action model has two extra states, both agreeing with the actual action on the non-availability of wine, but both making opposite statements about the availability of water.

Clearly, the only action of the model on the left is not bisimilar to the "no water and no wine" action on the right, for these actions have different preconditions. Also the only action of the leftmost model is not bisimilar to the "water and no wine" state on the right, for the same reason. Still the two action models have the same update effects: they both act as the public announcement that there is no wine.

For more examples, consider Figure 4. There are four action models here, all with a single agent (so the agent label was dropped): A_1 with state 0, A_2 with states $\{1, 2, 3\}$, A_3 with states $\{4, 5\}$, A_4 with states $\{6, 7, 8, 9\}$. Each of the action models has the effect of selecting the accessibility paths with $p \vee q$ holding at every node along the paths. It is not hard to verify that they have same update effects.

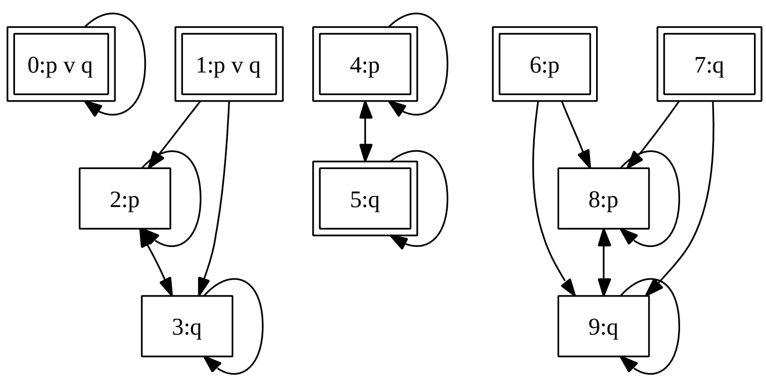

Figure 4: More non-bisimilar actions models with the same update effects

Examples like these suggest that the notion of bisimilarity in definition 5 is too strong to capture the 'essence' of our update actions, and they motivate our quest for a more appropriate relation.

4 Update Effects of Propositional Precondition Models

In this section, we focus on the important case of actions with propositional preconditions. We propose *propositional action emulation* to characterize the propositional precondition models with the same update effects, and we show that this solves the problems in Figures 3 and 4 in the previous section. These are all examples with purely propositional preconditions.

Compared to *Q-valuation* precondition actions, actions with propositional preconditions may be executable on *multiple* states with *different valuations*. For example, the "no wine" action in the single-action public announcement on the left in Figure 3 is executable on any state where there is no wine (e.g., it is executable in the "beer" and the "water" situations of the epistemic model on the left in Figure 1, but not in the "wine" situation).

Now take two multiple pointed action models with the same update effect, and take some arbitrary epistemic model. Then the results of the updates are bisimilar. Now if we pick up two bisimilar states in the results, say (w, s) and (v, t), then we know that w, v agree on the valuation, and that pre(s) and pre(t) are both compatible with this valuation. This suggests the following definition of a structural relation that has to hold between s and t:

Definition 11 (Propositional Action Emulation) *Given action models A and B, a relation $E \subseteq W_A \times W_B$ is a propositional action emulation if whenever sEt the following hold:*

Invariance $pre_s \wedge pre_t$ *is consistent.*

Zig *If $s \xrightarrow{i} s'$ then there are $t_1, \ldots, t_n$ with*

$$t \xrightarrow{i} t_1, \ldots, t \xrightarrow{i} t_n, s'Et_1, \ldots, s'Et_n \text{ and } pre_{s'} \models pre_{t_1} \vee \cdots \vee pre_{t_n}.$$

Zag *If $t \xrightarrow{i} t'$ then there are $s_1, \ldots, s_n$ with*

$$s \xrightarrow{i} s_1, \ldots, s \xrightarrow{i} s_n, s_1Et', \ldots, s_nEt' \text{ and } pre_{t'} \models pre_{s_1} \vee \cdots \vee pre_{s_n}$$

For pointed action models (A,S), (B,T), we write $(A, S) \leftrightarrows_p (B, T)$ if additionally the following is satisfied: for every $s \in S(\subseteq W_A)$ with consistent pre_s there are $t_1, \ldots, t_n \in T(\subseteq W)$ such that $sEt_1, \ldots, sEt_n$ and $pre_s \models pre_{t_1} \vee \cdots \vee pre_{t_n}$, and for every $t \in T$ with consistent pre_t there are $s_1, \ldots, s_m \in S$ with $s_1Et, \ldots, s_mEt$ and $pre_t \models pre_{s_1} \vee \cdots \vee pre_{s_m}$.

Observe that the examples of propositional actions with the same update effect in the previous section all satisfy this structural requirement.

Recall that in Section 2, we gave a number of reasons for working with multiple pointed models. The definition following sheds extra light on our choice. The definition makes essential use of a relation linking sets of points to sets of points.

The following proposition states that propositional action emulation is a weakening of bisimulation, in a precise sense.

Proposition 12 *Given pointed action models (A, S) and (B, T) with propositional preconditions,*

$$\text{if } (A, S) \leftrightarrow (B, T) \text{ then } (A, S) \leftrightarrows_p (B, T).$$

Proof. The bisimulation Z witnessing $(A, S) \leftrightarrow (B, T)$, is also a propositional action emulation witnessing $(A, S) \leftrightarrows_p (B, T)$, since the three conditions of action emulation follow from three conditions of action bisimulation. □

In the following, we formally prove that propositional action emulation is a sufficient and necessary requirement for two propositional action models to have the same update effect.

Proposition 13 *Given pointed action models* (A, S) *and* (B, T) *with propositional preconditions,*

$$\text{if } (A, S) \leftrightarrows_p (B, T) \text{ then } (A, S) \equiv_{ACT} (B, T).$$

Proof. Let (M, X) be an arbitrary pointed epistemic model. Assume $(A, S) \leftrightarrows_p (B, T)$ and let E be an action emulation witnessing this.

Define $C \subseteq M \otimes A \times M \otimes B$ by means of: $(w, s)C(v, t) :\equiv w = v \wedge sEt$. We show that C is a bisimulation: suppose $(w, s)C(v, t)$,

Invariance From $(w, s)C(v, t)$ we get that $w = v$ and hence $V(w, s) = V(v, t)$.

Zig Let $(w, s) \xrightarrow{i} (w', s')$. Then $w \xrightarrow{i} w'$, $s \xrightarrow{i} s'$, and $M \models_{w'} \text{pre}_{s'}$. From $(w, s)C(v, t)$ we have that sEt. By sEt, there are $t_1, \ldots, t_n \in W_B$ with $t \xrightarrow{i} t_1, \ldots, t \xrightarrow{i} t_n$, $s'Et_1, \ldots, s'Et_n$, and $\text{pre}_{s'} \models \text{pre}_{t_1} \vee \cdots \vee \text{pre}_{t_n}$. Since $M \models_{w'} \text{pre}_{s'}$, it follows from $\text{pre}_{s'} \models \text{pre}_{t_1} \vee \cdots \vee \text{pre}_{t_n}$ that there is some t_i with $M \models_{w'} \text{pre}_{t_i}$. Thus $(w', s')C(w', t_i)$.

Zag Same reasoning vice versa.

Now we show that C connects the pointed models $(M, X) \otimes (A, S)$ and $(M, X) \otimes (B, T)$. Given $(w, s) \in M \otimes A$ with $w \in X$ and $s \in S$, we have $M \models_w \text{pre}_s$. Since E connects (A, S) and (B, T), there must be $t_1, \ldots, t_n$, such that $sEt_1, \ldots, sEt_n$ and $\text{pre}_s \models \text{pre}_{t_1} \vee \ldots \vee \text{pre}_{t_n}$; hence $M \models_w \text{pre}_{t_1} \vee \ldots \vee \text{pre}_{t_n}$. So there must be t_i such that $M \models_w \text{pre}_{t_i}$, therefore $(w, s)C(w, t_i)$. The other direction is similar. □

Proposition 14 *Given pointed action models* (A, S) *and* (B, T) *with propositional preconditions,*

$$\text{if } (A, S) \equiv_{ACT} (B, T) \text{ then } (A, S) \leftrightarrows_p (B, T).$$

Proof. Suppose $(A, S) \equiv_{\text{ACT}} (B, T)$, and let Q include all propositional letters occuring in preconditions of A and B. We have that $\text{VAL}_Q^* \otimes (A, S) \underline{\leftrightarrow} \text{VAL}_Q^* \otimes (B, T)$. Define a binary relation $E \subseteq W_A \times W_B$ by means of: $sEt :\equiv$ there is a $\mathbf{v} \in \text{VAL}_Q$ such that $(\mathbf{v}, s) \underline{\leftrightarrow} (\mathbf{v}, t)$. We show that E is a propositional action emulation. Suppose sEt:

Invariance By definition, there is $\mathbf{v}$ such that $(\mathbf{v}, s) \underline{\leftrightarrow} (\mathbf{v}, t)$, so $\mathbf{v} \models \text{pre}_s$ and $\mathbf{v} \models \text{pre}_t$. Therefore $\text{pre}_s \wedge \text{pre}_t$ is consistent.

Zig Suppose $s \xrightarrow{i} s'$. There must be a set $\{\mathbf{v}_1, ..., \mathbf{v}_n\}$ such that $\models pre_{s'} \leftrightarrow \overline{\mathbf{v}}_1 \vee ... \vee \overline{\mathbf{v}}_n$. Since the accessibility relation in VAL_Q is universal, we have $\mathbf{v} \xrightarrow{i} \mathbf{v}_j, j = 1, ..., n$. So $(\mathbf{v}, s) \xrightarrow{i} (\mathbf{v}_j, s'), j = 1, ..., n$. According to $(\mathbf{v}, s) \underline{\leftrightarrow} (\mathbf{v}, t)$, there must be $t_1, ..., t_n$ such that $(\mathbf{v}, t) \xrightarrow{i} (\mathbf{v}_j, t_j)$ and $(\mathbf{v}_j, s') \underline{\leftrightarrow} (\mathbf{v}_j, t_j), j = 1, .., n$. It follows that $\text{pre}_{s'} \models \text{pre}_{t_1} \vee ... \vee \text{pre}_{t_n}$ since $\mathbf{v}_j \models \text{pre}_{t_j}, j = 1, .., n$.

Zag Same reasoning vice versa.

It is easy to check that E connects the states with consistent preconditions in S and T. $\square$

Combining proposition 13 and 14, we have:

Theorem 15 *Given pointed action models (A, S) and (B, T) with propositional preconditions,*

$$(A, S) \equiv_{ACT} (B, T) \text{ iff } (A, S) \leftrightarrows_p (B, T).$$

So we have achieved the characterization of having the same update effect in the case of propositional preconditions. However, the story does not stop here: we still have to deal with action models with modal preconditions. Now it turns out that the definition of propositional action emulation is still not adequate for such action models. In the action models of Figure 5, there is no propositional action emulation between 0 and 1, even though the two models have the same update effect. The rightmost action model is funny: the precondition in its actual action says that it is known that there is wine, and yet there is an arrow from that action to a "no wine" action. Clearly, this "no wine" action will never be executed.

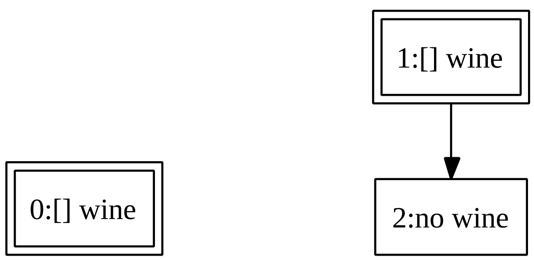

Figure 5: No propositional AE, yet same update effect

Or consider Figure 6. It is easy to check that $\{0, 1\}$ and $\{4\}$ are not connected by an action emulation, and yet they have the the same update effect. Here is a simple argument. If the updated epistemic world satisfies $\Box$wine then the state corresponding to 4 is 0, if the updated world satisfies $\Box\neg$wine, then it is 2. The example shows that the state might depend on the *type* of the updated epistemic world.

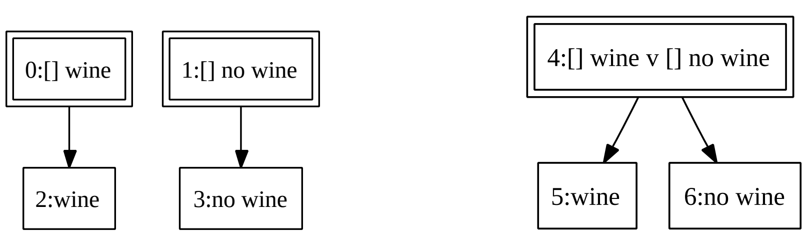

Figure 6: No propositinal AE, yet same update effect

In the next section we lift the restriction on precondition and define an appropriate, weaker version of action emulation that looks more complicated, but that has the merit that it works for all cases.

5 Update Effects of Modal Precondition Models

We now turn to the case of arbitrary modal preconditions in $\mathcal{LANG}$. As was mentioned already, it was shown in [7] that our language $\mathcal{LANG}$ is equally expressive as Epistemic PDL, i.e. the sub-language of $\mathcal{LANG}$, defined by:

$$\begin{aligned} \varphi &::= \top \mid p \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \Box_\alpha \varphi, \\ \alpha &::= a \mid ?\varphi \mid \alpha_1 \cup \alpha_2 \mid \alpha_1; \alpha_2 \mid \alpha^*. \end{aligned}$$

This was in fact the main reason for picking $\mathcal{LANG}$ as our language. An arbitrary modal precondition in $\mathcal{LANG}$ can be expressed equivalently in the Epistemic PDL sublanguage. So in this section, it is enough to consider action models with preconditions phrased in Epistemic PDL. This enables us to use existing tools developed in PDL, e.g., the method of canonical models.

To motivate the new (admittedly rather complicated) version of action emulation below, look again at Figure 5. As was already mentioned, the "no wine" action in the rightmost action model will never get executed. Let there be an epistemic model M with an actual world w. Then for $(w, 0)$ to end up in the update result of the left action model, it has to be the case that $\Box\neg$wine holds in w, and similarly for $(w, 1)$ in the update result of the right action model. But this means that in both cases, there will be no accessible v where "wine" holds, for that is what the formula $\Box\neg$wine expresses. The example shows that the modal precondition of an action can influence what happens elsewhere in the update result. It is clear, therefore, that propositional action emulation has to fail for such cases. The key to taking the non-local effects of preconditions into account turns out to be the construction of maximal consistent, yet finite, sets of formulas from the preconditions.

Let A and B be action models with modal preconditions. Let Π be the preconditions occurring in A, B, and Q be the set of all proposition letters occurring in Π. Let $FL(\Pi)$ be the *Fisher/Ladner closure* of Π (see e.g. [10, 16, 14]), i.e. the smallest set of formulas containing Π that is closed under subfomulas and such that

$$\begin{aligned} \text{if } \Box_{?\varphi_1}\varphi_2 &\in FL(\Pi) \text{ then } \neg(\varphi_1 \wedge \neg\varphi_2) \in FL(\Pi), \\ \text{if } \Box_{\alpha_1\cup\alpha_2}\varphi &\in FL(\Pi) \text{ then } \Box_{\alpha_1}\varphi \in FL(\Pi) \text{ and } \Box_{\alpha_2}\varphi \in FL(\Pi), \\ \text{if } \Box_{\alpha_1;\alpha_2}\varphi &\in FL(\Pi) \text{ then } \Box_{\alpha_1}\Box_{\alpha_2}\varphi \in FL(\Pi), \\ \text{if } \Box_{\alpha^*}\varphi &\in FL(\Pi) \text{ then } \Box_{\alpha}\varphi \in FL(\Pi) \text{ and } \Box_{\alpha}\Box_{\alpha^*}\varphi \in FL(\Pi). \end{aligned}$$

Let $\neg FL(\Pi)$ be the closure of $FL(\Pi)$ under single negation.

The axiom system we use for consistency is taken from [16]. Let MCONS_Π be the set of all maximal consistent subsets taken from $\neg FL(\Pi)$. Since $\neg FL(\Pi)$ is finite, these maximal consistent subsets are finite as well. Let EXP_Π be the triple (W, V, R) where $W = \text{MCONS}_\Pi$, V is the function that assigns to every maximal consistent subset $\Gamma \in \text{MCONS}_\Pi$ a set of propositional variables X such that $\Gamma \cap Q \subseteq X$ and if $\Gamma_1 \neq \Gamma_2$ then $V(\Gamma_1) \neq V(\Gamma_2)$, and R assigns to each agent i the relation $\xrightarrow{i}$ given by:

$$\Gamma \xrightarrow{i} \Gamma' \text{ iff } \forall\varphi \in \Gamma', \Box_i\neg\varphi \notin \Gamma.$$

Thus, in comparison to VAL_Q, the accessibilities now take the modal constraints imposed by the preconditions into acccount. Let EXP^*_Π be (EXP_Π, W). Let Π^M_w be the set

$$\{\varphi \in \neg FL(\Pi) \mid M \models_w \varphi\}.$$

Note that $\Pi^M_w \in \mathrm{MCONS}_\Pi$.

The following lemma will play a role in our proof:

Lemma 16 (Truth Lemma) *Given EXP_Π and $\Gamma \in MCONS_\Pi$. For any $\varphi \in \neg FL(\Pi)$,*

$$EXP_\Pi \models_\Gamma \varphi \textit{ iff } \varphi \in \Gamma$$

Proof. The proof follows from a standard induction on the structure of formulas. A detailed proof could be found in [16]. □

For $x \in W_A \cup W_B$, let $G(x)$ be the set

$$\{\Gamma \mid \Gamma \in \mathrm{MCONS}_\Pi, \mathrm{pre}_x \in \Gamma\}.$$

Thus, $G(x)$ is the set of those maximal consistent subsets taken from $\neg FL(\Pi)$ that contain the precondition of x.

Definition 17 (Action Emulation: full version) *Let A and B be action models, and Π be the preconditions occurring in A, B. Action emulation is a set of relations $\{E_\Gamma\}_{\Gamma\in MCONS_\Pi}$ on $W_A \times W_B$ such that whenever $sE_\Gamma t$ the following hold:*

Invariance *$pre_s \in \Gamma$ and $pre_t \in \Gamma$.*

Zig *if $s \xrightarrow{i} s'$ and $\Gamma' \in G(s')$ such that $\Gamma \xrightarrow{i} \Gamma'$, then there is a $t' \in W_B$ with $t \xrightarrow{i} t'$ and $s'E_{\Gamma'}t'$.*

Zag *if $t \xrightarrow{i} t'$ and $\Gamma' \in G(t')$ such that $\Gamma \xrightarrow{i} \Gamma'$, then there is a $s' \in W_A$ with $s \xrightarrow{i} s'$ and $s'E_{\Gamma'}t'$.*

Use $(A, S) \leftrightarrows (B, T)$ if there is an action emulation satisfying the following conditions:

- *for any $s \in S$ and $\Gamma \in G(s)$ there is a $t \in T$ with $sE_\Gamma t$.*
- *for any $t \in T$ and $\Gamma \in G(t)$ there is a $s \in T$ with $sE_\Gamma t$.*

It is not hard to see that propositional action emulation is a special case of the definition above. In the propositional case, each maximal consistent set Γ corresponds to a $\mathbf{v}$ such that $\Gamma \cap Q = \mathbf{v}$. We always have $\Gamma \xrightarrow{i} \Gamma'$ for any $i \in Ag$ because there is no modal operator in propositions. For each consistent precondition pre_s, there is a non-empty set $\{\mathbf{v}_1, ..., \mathbf{v}_n\}$ such that $\models pre_s \leftrightarrow \overline{\mathbf{v}}_1 \vee ... \vee \overline{\mathbf{v}}_n$. The sets of points in the zigzag conditions of the propositional case correspond to the set of maximal consistent sets, as required by the zigzag conditions in the full version.

Similar to Theorem 12, we have:

Theorem 18 *Given multiple pointed action models* (A,S) *and* (B,T)*,*

$$\textit{If } (A,S) \underline{\leftrightarrow} (B,T) \textit{ then } (A,S) \leftrightarrows (B,T).$$

We will now prove that emulating action models have the same update effect:

Proposition 19 *Given multiple pointed action models* (A,S) *and* (B,T)*,*

$$\textit{If } (A,S) \leftrightarrows (B,T) \textit{ then } (A,S) \equiv_{ACT} (B,T).$$

Proof. Let $\{E_\Gamma\}_{\Gamma \in \text{MCONS}_\Pi}$ be a set of relations witnessing $(A,S) \leftrightarrows (B,T)$. We wish to show for arbitrary (M,U) that $(M,U) \otimes (A,S) \underline{\leftrightarrow} (M,U) \otimes (B,T)$.

Define a relation C on $W_{M\otimes A} \times W_{M\otimes B}$ by means of:

$$(w,s)C(v,t) :\equiv w = v \text{ and } sE_{\Pi^M_w}t.$$

Suppose $(w,s)C(v,t)$. We show that C is a bisimulation.

Invariance (w,s) and (v,t) have the same valuation.

Zig Let $(w,s) \xrightarrow{i} (w',s')$. Then $w \xrightarrow{i} w'$, $s \xrightarrow{i} s'$, and $M \models_{w'} \text{pre}_{s'}$. It follows from $M \models_{w'} \text{pre}_{s'}$ that $\text{pre}_{s'} \in \Pi^M_{w'}$.

To show $\Pi^M_w \xrightarrow{i} \Pi^M_{w'}$, let $\varphi \in \Pi^M_{w'}$ and assume, for a contradiction, that $\Box_i\neg\varphi \in \Pi^M_w$. From $\varphi \in \Pi^M_{w'}$, we have $M \models_{w'} \varphi$, thus $M \models_w \Diamond_i\varphi$ by $w \xrightarrow{i} w'$. Because Π^M_w is maximally consistent, it follows from the assumption that $\neg\Box_i\neg\varphi \notin \Pi^M_w$, contradicting the fact that $M \models_w \Diamond_i\varphi$. Therefore $\Box_i\neg\varphi \notin \Pi^M_w$. Thus, $\Pi^M_w \xrightarrow{i} \Pi^M_{w'}$

Since $sE_{\Pi^M_w}t$, $s \xrightarrow{i} s'$, and $\Pi^M_w \xrightarrow{i} \Pi^M_{w'}$, there must be t', such that $t \xrightarrow{i} t'$ and $s'E_{\Pi^M_{w'}}t'$. It follows that $\text{pre}_{t'} \in \Pi^M_{w'}$, i.e. $M \models_{w'} \text{pre}_{t'}$. So $(w',s')C(w',t')$.

Zag Same reasoning vice versa.

□

Proposition 19 shows that action emulation is a sufficient condition for having the same update effect. We show that it is also necessary.

Proposition 20 *Given multiple pointed action models* (A,S) *and* (B,T)*,*

$$\textit{if } (A,S) \equiv_{ACT} (B,T) \textit{ then } (A,S) \leftrightarrows B,T).$$

Proof. Assume $(A,S) \equiv_{\text{ACT}} (B,T)$. It follows that $(\text{EXP}_\Pi, W) \otimes (A,S) \underline{\leftrightarrow} (\text{EXP}_\Pi, W) \otimes (B,T)$. Let C be the above bisimulation. We define a set of binary relations $\{E_\Gamma\}_{\Gamma \in \text{MCONS}_\Pi}$ as follows $sE_\Gamma t :\equiv (\Gamma,s)C(\Gamma,t)$, $\Gamma \in \text{MCONS}_\Pi$. We show that $\{E_\Gamma\}_{\Gamma\in\text{MCONS}_\Pi}$ is an emulation. Suppose $sE_\Gamma t$,

Invariance We have $(\Gamma, s)C(\Gamma, t)$. Therefore $\text{EXP}_\Pi \models_\Gamma \text{pre}_s$, and $\text{EXP}_\Pi \models_\Gamma \text{pre}_t$. According to Truth Lemma 16, we have $\text{pre}_s \in \Gamma$ and $\text{pre}_t \in \Gamma$.

Zig Suppose $s \xrightarrow{i} s'$ and $\Gamma' \in G(s')$ such that $\Gamma \xrightarrow{i} \Gamma'$. It follows that $\text{pre}_{s'} \in \Gamma'$. Again by Truth Lemma 16, we have $\text{EXP}_\Pi \models_{\Gamma'} \text{pre}_{s'}$, so $(\Gamma', s') \in W_{\text{EXP}_\Pi \otimes A}$. Therefore, we have $(\Gamma, s) \xrightarrow{i} (\Gamma', s')$. By C, there must be (Γ'', t') such that $(\Gamma, t) \xrightarrow{i} (\Gamma'', t')$ and $(\Gamma', s')C(\Gamma'', t')$. Since in our construction of EXP_Π, the valuation of each world is different, here $\Gamma' = \Gamma''$. Therefore $t \xrightarrow{i} t'$ and $s' E_{\Gamma'} t'$.

Zag Same reasoning vice versa.

It is easy to check that the last two conditions in action emulation are also fulfilled. □

Combining the above, we have:

Theorem 21 *Given multiple pointed action models (A, S) and (B, T),*

$$(A, S) \equiv_{ACT} (B, T) \text{ iff } (A, S) \leftrightarrows (B, T).$$

6 Conclusion and Further Issues

Action emulation is defined at the state set level, in terms of a set-theoretic lift of indexed bisimulation. This family tie with standard bisimulation generates a number of behavioural similarities to bisimulation. E.g., as with bisimulations, there always is a largest action emulation. Therefore, the union of all action emulations connecting (A, S) and (B, T) is an action emulation, i.e., there always is a largest action emulation connecting (A, S) and (B, T). This largest action emulation might be used to construct the minimal model contraction for any multiple pointed action model (A, S) , i.e. the minimal multiple pointed action model emulating with (A, S)., just as in the case of bisimulation (see, e.g., [15], where it is shown that such bisimulation minimal model can be found in an efficient way). This suggest the following question. What is the complexity of determining whether two action models emulate? Is this more complex than bisimulation, or is it also polynomial, like the decision problem for bisimilarity? In particular, can something like a partition refinement algorithm in the style of [15] be made to work for this? In the case of action models with propositional preconditions it might also be feasible to construct action emulation minimal models. This is future work.

Here is a more theoretical question. For bisimulation there are elegant characterizations: modal logic is the bisimulation invariant fragment of first order logic [4]. What this means is that a first order logic is equivalent to a modal formula precisely when it cannot see the difference between bisimulation structures). A similar bisimulation characterisation exists for monadic second order and modal mu calculus [13]. Can something along the same lines be found for action emulation?

Acknowledgement Thanks to Johan van Benthem, Hans van Ditmarsch and Albert Visser for inspiring discussions, and thanks to two anonymous reviewers for helpful comments. We are grateful to our editor Wiebe van der Hoek for encouragement and patience. The first author is grateful to the Netherlands Institute for Advanced Studies in Wassenaar (NIAS) for providing the opportunity to complete this paper as Fellow-in-Residence.

References

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

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

[3] Baltag, A., Moss, L., and Solecki, S. The logic of public announcements, common knowledge, and private suspicions. In *Proceedings of TARK'98* (1998), I. Bilboa, Ed., pp. 43–56.

[4] Benthem, J. v. *Modal Correspondence Theory*. PhD thesis, University of Amsterdam, 1976.

[5] Benthem, J. v. Language, logic, and communication. In *Logic in Action*, J. van Benthem, P. Dekker, J. van Eijck, M. de Rijke, and Y. Venema, Eds. ILLC, 2001, pp. 7–25.

[6] Benthem, J. v. One is a lonely number: on the logic of communication. Tech. Rep. PP-2002-27, ILLC, Amsterdam, 2002.

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

[8] Ditmarsch, H. v. *Knowledge Games*. PhD thesis, ILLC, Amsterdam, 2000.

[9] Ditmarsch, H. v., van der Hoek, W., and Kooi, B. *Dynamic Epistemic Logic*, vol. 337 of *Synthese Library*. Springer, 2006.

[10] Fischer, M. J., and Ladner, R. E. Propositional dynamic logic of regular programs. *Journal of Computer and System Sciences 18*, 2 (1979), 194–211.

[11] Gerbrandy, J. *Bisimulations on planet Kripke*. PhD thesis, ILLC, 1999.

[12] Hintikka, J. *Knowledge and Belief: An Introduction to the Logic of the Two Notions*. Cornell University Press, Ithaca N.Y., 1962.

[13] Janin, D., and Walukiewicz, I. On the expressive completeness of the propositional mu-calculus with respect to monadic second order logic. In *CONCUR '96: Proceedings of the 7th International Conference on Concurrency Theory* (1996), Springer-Verlag, pp. 263–277.

[14] Kozen, D., and Parikh, R. An elementary proof of the completeness of PDL. *Theoretical Computer Science 14* (1981), 113–118.

[15] Paige, R., and Tarjan, R. E. Three partition refinement algorithms. *SIAM J. Comput. 16*, 6 (1987), 973–989.

[16] SEGERBERG, K. A completeness theorem in the modal logic of programs. In *Universal Algebra and Applications*, T. Traczyck, Ed. Polish Science Publications, 1982, pp. 36–46.