Seeing is Believing: Formalising False-Belief Tasks in Dynamic Epistemic Logic

Thomas Bolander

Technical University of Denmark

Abstract. In this paper we show how to formalise false-belief tasks like the Sally-Anne task and the second-order chocolate task in *Dynamic Epistemic Logic* (DEL). False-belief tasks are used to test the strength of the *Theory of Mind* (ToM) of humans, that is, a human's ability to attribute mental states to other agents. Having a ToM is known to be essential to human social intelligence, and hence likely to be essential to social intelligence of artificial agents as well. It is therefore important to find ways of implementing a ToM in artificial agents, and to show that such agents can then solve false-belief tasks. In this paper, the approach is to use DEL as a formal framework for representing ToM, and use reasoning in DEL to solve false-belief tasks. In addition to formalising several false-belief tasks in DEL, the paper introduces some extensions of DEL itself: *edge-conditioned event models* and *observability propositions*. These extensions are introduced to provide better formalisations of the false-belief tasks, but expected to have independent future interest.

1 Introduction

Social intelligence is the ability to understand others and the social context effectively and thus to interact with other agents successfully. Research has suggested that *Theory of Mind* (ToM) may play an important role in explaining social intelligence. ToM is the ability to to attribute mental states—beliefs, intentions, etc.—to oneself and others and to understand that others might have mental states that are different from one's own [23]. The strength of a human child's ToM is often tested with a *false-belief task* such as the *Sally-Anne task* [29].

Example 1 (The Sally-Anne task). The Sally-Anne task is illustrated in Figure 7 in the appendix. It is based on a story with two agents, Sally and Anne, that has the following 5 steps, corresponding to the 5 pictures in Figure 7:

0. Sally and Anne are in a room. Sally is holding a marble. There is a basket and a box in the room.
1. Sally puts the marble into the basket.
2. Sally leaves the room.
3. Anne transfers the marble to the box.
4. Sally comes back.

When used as a cognitive test for children, the child is told or shown the story in the figure. At the end, the child is asked "where does Sally believe the marble to be?" Passing the test means answering "in the basket", since Sally didn't see Anne transfer the marble from the basket to the box, and hence Sally has the *false belief* that it is still in the basket. If the child answers "in the box", where in fact the marble is, the child has *failed* the test. Children under the age of 4, and autistic children in general, are generally unable to pass the Sally-Anne test [29, 9].

To create AI agents with social intelligence, it seems relevant to consider the possibility of equipping such agents with a ToM, and to test them using false-belief tasks. The idea here is that for an AI agent, e.g. a robot, to be considered truly 'socially intelligent', it should at least be able to pass these false-belief tasks. Hence it becomes important to find ways of *formalising* ToM and false-belief tasks in a way that will allow computers to do the required reasoning.

The goal of the present paper is to present one such possible formalisation, using the framework of *Dynamic Epistemic Logic* (DEL). We will now explain why DEL is a fairly natural choice here. First of all, we need a formalism that can represent the beliefs of other agents, e.g. the beliefs of Sally, Sally's beliefs about Anne, etc. This naturally leads one to consider an *epistemic logic* (or, more precisely, a *doxastic logic*, but we will here still refer to it as *epistemic*). Basic epistemic logic is however only sufficient to model static state of affairs, like "at this particular instant, Sally believes the marble to be in the basket." In the false-belief tasks we also need to be able to model the *dynamics*: how the beliefs of the involved agents change as actions occur, e.g. when Anne secretely transfers the marble. This is where DEL comes into the picture: it has a natural way to deal with static states of beliefs (the *epistemic models* of DEL), a natural way to describe actions with epistemic and/or world changing effects (the *event models* of DEL), and a simple way of calculating the result of executing an action in a state (the *product update* of DEL).

Below we will first, in Section 2, briefly present the qualities we aim for in our false-belief task formalisations. Next, in Section 3, we introduce the required parts of DEL, and then apply it to formalise the Sally-Anne task in Section 4. The formalisation turns out not to be entirely satisfactory, and hence we will, in Section 5, introduce an extension of DEL that gives more appropriate formalisations. The improved formalisations are in Section 6.

2 Robustness and faithfulness

Above we claim that DEL is a fairly natural choice for the formalisation of false-belief tasks. This of course doesn't imply that it is the *only* natural choice. Indeed, there are several existing formalisations of false-belief tasks in the literature, using different formal frameworks. Figure 8 of the appendix gives a brief overview of the full formalisations and implemented systems we know of. The Sally-Anne task is usually referred to as a *first-order* false-belief task since it only

involves *first-order belief attribution*: the child has to attribute beliefs to Sally, but not, say, to Sally's beliefs about Anne's beliefs (which would be second-order belief attribution). Most of the existing formalisations can only deal with first-order or at most second-order false-belief tasks. We wish to be more general, and at the same time have formalisations that stay as close as possible to the informal versions of the tasks, and so propose the following two criteria:

Robustness. *The formalism should not only be able to deal with one or two selected false-belief tasks, but with as many as possible, with no strict limit on the order of belief attribution.*

Faithfulness. *Each action of the false-belief story should correspond to an action in the formalism in a natural way, and it should be fairly straightforward, not requiring ingenuity, to find out what that action of the formalism is.*

One can distinguish approaches to formalising false-belief tasks that seek to: 1) provide formal models of human reasoning; 2) provide the basis for a reasoning engine of autonomous agents. These two are of course not always disjoint aims, as discussed by Rineke Verbrugge [27]. In this paper, however, we are exclusively concerned with the second aim. We will hence not be concerned with whether our formalisation has any correspondence with the cognitive processes of humans solving false-belief tasks.

3 Dynamic epistemic logic

In this section we will introduce the required basics of dynamic epistemic logic (DEL). The less technically inclined, or interested, reader can browse very quickly through the definitions and instead focus on the examples that illustrate the workings of the formalism in relation to the Sally-Anne task. Basic familiarity with epistemic logic, but not necessarily DEL, is expected. All definitions in this section are well-known and standard in DEL. The particular variant presented here is adopted from van Ditmarsch and Kooi [17].

Epistemic Models

Throughout this article, P is an infinite, countable set of atomic propositions (propositional symbols), and $\mathcal{A}$ is a non-empty finite set of agents. We will most often use lower case letters $p, q, r, \ldots$ for atomic propositions and capital letters $A, B, C, \ldots$ for agents. Variables ranging over agents will be denoted $i, j, k, \ldots$. The epistemic language $\mathcal{L}(P, \mathcal{A})$ is generated by the following BNF:

$$\phi \ ::= \ p \ \mid \ \neg\phi \ \mid \ \phi \wedge \phi \ \mid \ B_i\phi$$

where $p \in P$ and $i \in \mathcal{A}$. The intended interpretation of a formula $B_i\phi$ is "agent i believes ϕ". The formula $\phi \vee \psi$ is an abbreviations of $\neg(\neg\phi \wedge \neg\psi)$, and we define $\top$ as an abbreviation for $p \vee \neg p$ and $\bot$ as an abbreviation for $p \wedge \neg p$ for some arbitrarily chosen $p \in P$. The semantics of $\mathcal{L}(P, \mathcal{A})$ is defined through *epistemic models*.

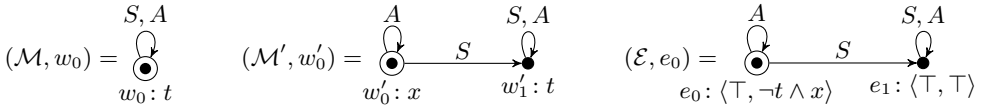


Fig. 1. Two states and an action.

Definition 1 (Epistemic models and states). *An* epistemic model of $\mathcal{L}(P, \mathcal{A})$ *is* $\mathcal{M} = (W, R, V)$*, where*

- W*, the* domain*, is a set of* worlds*;*
- $R : \mathcal{A} \to 2^{W \times W}$ *assigns an* accessibility relation $R(i)$ *to each agent* $i \in \mathcal{A}$*;*
- $V : P \to 2^W$ *assigns a set of worlds to each atomic proposition.*

The relation $R(i)$ *is usually abbreviated* R_i*, and we write* wR_iv *when* $(w, v) \in R_i$*. For* $w \in W$*, the pair* $(\mathcal{M}, w)$ *is called a* state *of* $\mathcal{L}(P, \mathcal{A})$*, and* w *is referred to as the* actual world*.*

The truth conditions (that is, the definition of $(\mathcal{M}, w) \models \phi$ for models $\mathcal{M}$, worlds w and formulas $\phi \in \mathcal{L}(P, \mathcal{A})$) are standard and provided in Figure 9 of the appendix.

Example 2. We will now illustrate the notion of a state relative to the Sally-Anne task of Example 1. The example states are $(\mathcal{M}, w_0)$ and $(\mathcal{M}', w_0')$ of Figure 1. Here we have two atomic propositions, x and t, where x is intended to mean "the marble is in the box$x$", and t means "the marble is in the basket$t$". We use the agent symbols S and A for Sally and Anne, respectively.

In $(\mathcal{M}, w_0)$ and $(\mathcal{M}', w_0')$, and states in general, each world is marked by its name followed by a list of the atomic propositions true at that world (which may be empty if none holds true). Sometimes we will drop names on worlds and just label them by the list of true propositions. Edges are labelled with the name of the relevant accessibility relations (agents). We use the symbol $\odot$ to mark the actual world.

Consider $(\mathcal{M}, w_0)$. The actual world is w_0, that is, the marble is in the basket (t holds). The loop at w_0 for S and A means that Sally and Anne consider the actual world w_0 possible, and the absence of other edges means that they *only* consider w_0 possible. Hence we have e.g. $(\mathcal{M}, w_0) \models B_S t \wedge B_A t \wedge B_S B_A t \wedge B_A B_S t$: both Sally and Anne believe the marble to be in the basket, and they both believe each other to have this belief. The state $(\mathcal{M}, w_0)$ corresponds to the situation before Anne has transferred the marble to the box.

Consider now $(\mathcal{M}', w_0')$. This corresponds to the situation after Anne has transferred the marble in Sally's absence. The actual world now satisfies x. In the actual world, w_0, Anne only considers w_0 possible (signified by the loop labelled A at w_0): she *knows* the marble to be in the box. However, Sally doesn't have such a loop at w_0, rather she has an edge going to w_1 where t holds. This means that in the actual world Sally only considers it possible that the actual world is

in fact w_1. Hence she has a *false belief* that the marble is in the basket (a false belief that t holds). Formally, $(\mathcal{M}', w_0') \models B_S t$.

We have now seen how we can use states to model the beliefs of Sally and Anne before and after the marble is moved. But we also need a way to model the act of moving the marble. This is done using DEL event models, presented next.

Event Models

DEL introduces the concept of *event model* (or *action model*) for modeling the changes to states brought about by the execution of actions [6, 5]. We here use a variant that includes postconditions [16, 10, 11], which means that actions can have both epistemic effects (changing the beliefs of agents) and ontic effects (changing the physical facts of the world).

Definition 2 (Event models and actions). *An* event model of $\mathcal{L}(P, \mathcal{A})$ *is* $\mathcal{E} = (E, Q, pre, post)$*, where*

- E*, the* domain*, is a finite non-empty set of* events*;*
- $Q : \mathcal{A} \to 2^{E \times E}$ *assigns an* accessibility relation $Q(i)$ *to each agent* $i \in \mathcal{A}$*;*
- $pre : E \to \mathcal{L}(P, \mathcal{A})$ *assigns to each event* a precondition*, which can be any formula in* $\mathcal{L}(P, \mathcal{A})$*.*
- $post : E \to \mathcal{L}(P, \mathcal{A})$ *assigns to each event* a postcondition*. Postconditions are conjunctions of propositional literals,* i.e.*, conjunctions of atomic propositions and their negations (including* $\top$ *and* $\bot$*).*

The relation $Q(i)$ *is generally abbreviated* Q_i*. For* $e \in E$*,* $(\mathcal{E}, e)$ *is called an* action *of* $\mathcal{L}(P, \mathcal{A})$*, and* e *is referred to as the* actual event*.*

Example 3. Consider the action $(\mathcal{E}, e_0)$ of Figure 1. Labeling events by the pair $\langle \phi_1, \phi_2 \rangle$ means that the event has precondition ϕ_1 and postcondition ϕ_2. Hence the actual event, e_0, corresponds to the action of making t false and x true, that is, it is the act of transferring the marble from the basket to the box. The event e_1 has trivial pre- and post-conditions meaning that it is a 'skip' action representing that nothing happens. Looking at the edges of the action, we see that Anne only considers it possible that the marble is transferred (the loop at e_0), whereas Sally only considers it possible that nothing happens (she only has en edge from the actual event to the 'skip' event e_1). Hence the model encodes an action where the marble is *actually* transferred from the basket to the box, Anne is aware of this, but Sally thinks that nothing happens. It hence encodes step 3 of the Sally-Anne task, cf. Example 1.

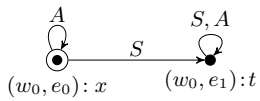
Product Update

Assume given a state $(\mathcal{M}, w_0)$ and an action $(\mathcal{E}, e_0)$. The product update yields a new state $(\mathcal{M}, w_0) \otimes (\mathcal{E}, e_0)$ representing the situation after the action $(\mathcal{E}, e_0)$ has been executed in the state $(\mathcal{M}, w_0)$.

Definition 3 (Product update). *Let $(\mathcal{M}, w_0)$ be a state and $(\mathcal{E}, e_0)$ an action, where $\mathcal{M} = (W, R, V)$ and $\mathcal{E} = (E, Q, pre)$, and where $\mathcal{M}, w_0 \models pre(e_0)$. The* product update *of $(\mathcal{M}, w_0)$ with $(\mathcal{E}, e_0)$ is defined as the state $(\mathcal{M}, w_0) \otimes (\mathcal{E}, e_0) = ((W', R', V'), (w_0, e_0))$, where*

- $W' = \{(w, e) \in W \times E \mid \mathcal{M}, w \models pre(e)\}$
- $R'_i = \{((w, e), (v, f)) \in W' \times W' \mid wR_iv \text{ and } eQ_if\}$
- $(w, e) \in V'(p)$ *iff* $post(e) \models p$ *or* $(\mathcal{M}, w \models p$ *and* $post(e) \not\models \neg p)$.

Example 4. Referring again to Figure 1, we can calculate the product update $(\mathcal{M}, w_0) \otimes (\mathcal{E}, e_0)$. Intuitively, the calculation works like this. For each event in $\mathcal{E}$, we first find the worlds in $\mathcal{M}$ that satisfies the precondition of the event. Each such matching world-event pair will become a world in the resulting model. Since both e_0 and e_1 have the trivial precondition $\top$, both have their precondition satisfied in the world w_0. This gives us two matching world-event pairs (w_0, e_0) and (w_0, e_1) that will become the worlds of the new model. Now we have to use the postconditions of the events in order to figure out what the labels of these new worlds will be. In (w_0, e_0) we have paired w_0 with e_0. This means that we should take the existing label of w_0 and then update it according to the postcondition of e_0. The label of w_0 is t and the postcondition of e_0 is $\neg t \wedge x$. The postcondition $\neg t \wedge x$ will force t to become false and x to become true, so the label of (w_0, e_0) will be x. The label of (w_0, e_1) is the same as of w_0, since e_0 has the trivial postcondition $\top$. So the updated model $(\mathcal{M}, w_0) \otimes (\mathcal{E}, e_0)$ will have the two worlds $(w_0, e_0) : x$ and $(w_0, e_1) : t$. Now we only need to find the edges connecting these two worlds. There will be an A-loop at (w_0, e_0), since there is both an A-loop at w_0 in $\mathcal{M}$ and an A-loop at e_0 in $\mathcal{E}$. Similarly there will be an $\{S, A\}$-loop at (w_0, e_1). Finally, we need to check the edges between (w_0, e_0) and (w_0, e_1). Since there is an S-loop at w_0 and an S-edge from e_0 to e_1, we get an S-edge from (w_0, e_0) to (w_0, e_1). In total, the product update becomes:

$(\mathcal{M}, w_0) \otimes (\mathcal{E}, e_0) =$ 

Note that the resulting model is isomorphic to $(\mathcal{M}', w'_0)$ of Figure 1. Since $(\mathcal{M}, w_0)$ represents the situation before Anne transfers the marble, and $(\mathcal{M}', w'_0)$ represents the situation afterwards (cf. Example 2), $(\mathcal{E}, e_0)$ correctly captures the action of transferring the marble in Sally's absence.

4 Formalising the Sally-Anne task in DEL

We now have all the necessary ingredients for our first formalisation of the Sally-Anne task. Consider again the 5 steps of the Sally-Anne story presented in Example 1. The first step, step 0, describes the initial state, whereas the rest, 1–4, describes a sequence of actions. We will now show how to represent step 0 as a state and steps 1–4 as actions. We use the same symbols as in the previous

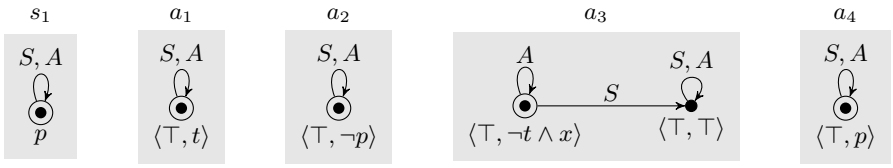


Fig. 2. The states and actions in the DEL formalisation of Sally-Anne.

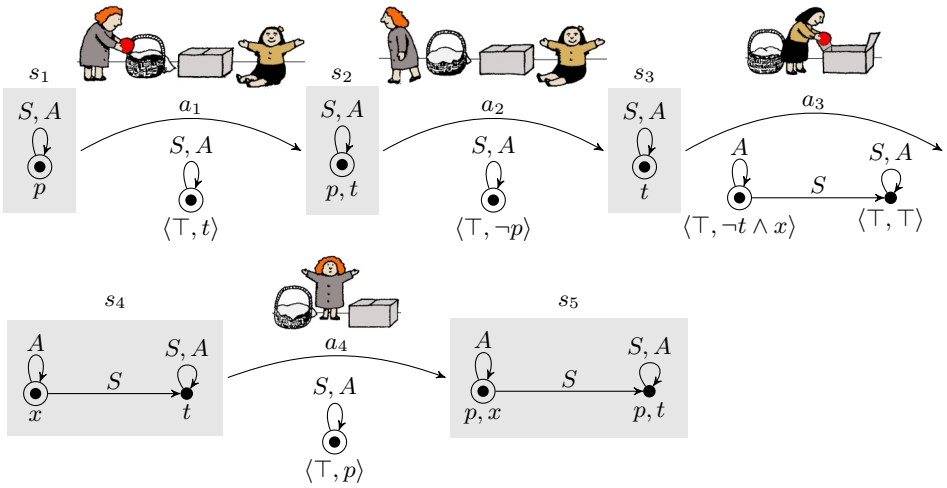


Fig. 3. The DEL-formalisation of the Sally-Anne task

examples, except we add a new atomic proposition p meaning "Sally is *p*resent in the room with Anne". The following 5 step list, corresponding to the list of Example 1, shows the relevant states and actions:

0. Sally is in the room, holding the marble: state s_1 of Figure 2.
1. Sally puts the marble into the basket: action a_1 of Figure 2.
2. Sally leaves the room: action a_2 of Figure 2.
3. Anne transfers the marble to the box: action a_3 of Figure 2.
4. Sally re-enters: action a_4 of Figure 2.

Figure 3 calculates the result of executing the action sequence $a_1, \ldots, a_4$ in s_1, that is, $s_{i+1} = s_i \otimes a_i$ for all $i = 1, \ldots, 4$, and hence $s_5 = s_1 \otimes a_1 \otimes \cdots \otimes a_4$. The first two actions, a_1 and a_2, are very simple. As seen from Figure 3, executing a_1 in the initial state s_1 simply adds the proposition t to the actual world (in s_2), signifying that now the marble is in the basket. Executing a_2 in the resulting state s_2 amounts to deleting p from the actual world: in s_3 Sally is no longer present

in the room. The action a_3, the most complex one, has already been discussed in Example 3, and in Example 4 we carefully checked that $s_4 = s_3 \otimes a_3$. The final action, a_4, simply adds p to every world of the model, corresponding to the fact the Sally returns to the room, and this is observed by both agents.

What is important is now of course to check what holds in s_5, the model resulting from executing $a_1, \ldots, a_4$ in s_1. From Figure 3 we can see that $s_5 \models \neg t \wedge B_S t$, that is, Sally mistakenly believes the marble to be in the basket. Assume an agent presented with steps 0–4 of the original informal story is able to formalise the steps as $s_1, a_1, \ldots, a_4$, and is afterwards asked "where does Sally believe the marble to be". Then that agent can first calculate the final state $s_5 = s_1 \otimes a_1 \otimes \cdots \otimes a_4$ and conclude that $s_5 \models B_S t$ holds. From this the agent can answer "in the basket", hence passing the Sally-Anne test!

5 Extending the DEL formalism

So far so good, or at least it seems that way. But a closer look shows that there are two problems with the DEL-formalisation that need to be addressed. The first is: where do the event models come from? How is an agent supposed to get from the informal steps of the story to the formalisations $s_1, a_1, \ldots, a_4$? It seems to require ingenuity to come up with the right event models to formalise the informal action descriptions, in particular action a_3. Hence the proposed solution doesn't yet really satisfy the *faithfulness* criterion presented in Section 2.

The second problem with the formalisation can be illustrated by considering a shortened version of the Sally-Anne task where Sally doesn't leave the room, that is, it only includes the steps 0, 1 and 3 of Example 1. These steps ought to have the same formalisations as before, that is, s_1, a_1 and a_3, respectively. Hence the situation after the shortened Sally-Anne story should correspond to $s_1 \otimes a_1 \otimes a_3$. However, consulting Figure 3 it can be checked that $s_1 \otimes a_1 \otimes a_3 = s_5$ (since a_2 only makes p false, and a_4 makes it true again). Hence, an agent presented with the shortened Sally-Anne story would conclude that $s_1 \otimes a_1 \otimes a_3 \models B_S t$, implying that Sally ends up believing the marble to be in the basket. This is clearly not correct, since in this version she never left the room!

In the following we will propose an improved formalisation that solves both of these problems. We start out by analysing the source of the second problem, which is in the formalisation of a_3 (see Figure 2). As explained in Example 3, a_3 "encodes an action where the marble is *actually* transferred from the basket to the box, Anne is aware of this, but Sally thinks that nothing happens". All this is clearly not part of step 3 of the story, which simply states "Sally transfers the marble to the box". The problem with a_3 is that it is hardcoded into the event model who observes the action taking place. In most real-life cases, who observes an action depends on the state in which the action is applied. This is also the case in the Sally-Anne story: whether Sally observes the marble being moved depends on whether she is in the room. So the edges of the event model for action a_3 ought to depend on whether Sally is present, that is, whether p holds or not. This leads us to a more general type of event model like a_3^{edge} of Figure 4.

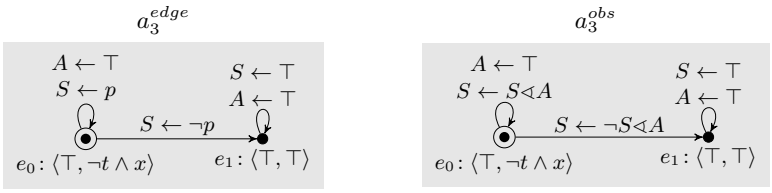


Fig. 4. Two generalised variants of the action a_3 in Sally-Anne

Here $A \leftarrow \top$ (A gets $\top$) at the loop of e_0 means that A unconditionally has an edge here: Anne unconditionally observes the event e_0. The other label $S \leftarrow p$ at the loop of e_0 means that there is an edge here for agent S if p is true: Sally observes the event e_0 if she is present in the room. Similarly, the label $S \leftarrow \neg p$ on the edge from e_0 to e_1 means that if Sally is not in the room ($\neg p$) then she thinks that nothing (e_1) happens. This is a new type of event model, called an *edge-conditioned event model*, to be defined formally in the next subsection.

With edge-conditioned event models we can solve the second problem mentioned above. We now have an event model that will behave correctly both if applied in a state where Sally is present (p holds) and in a state where Sally is not present (p doesn't hold). If a_3^{edge} is applied in a state where p holds, from e_0 Sally will only consider e_0 possible (have a loop at e_0), but if p doesn't hold, from e_0 she will only consider e_1 possible (have an edge from e_0 to e_1). Hence, if p holds she observes the event e_0, otherwise she doesn't. Using edge-conditioned event models also brings us a step closer to satisfying the first criterion of *faithfulness*. In almost all existing false-belief tasks, all ontic actions have the same structure as a_3^{edge}, and we can hence define a *generic event model* for all such ontic actions (which we will do in Section 5). However, it is still not quite satisfactory to use ad hoc symbols like p to state that a certain agent is present. This leads us to our next new idea.

In addition to the propositional symbols P, we add to the language a new set of propositional symbols $i \lhd j$ (i sees j) for each pair of agents $i \neq j$. The intended meaning of $i \lhd j$ is that agent i observes the actions of agent j. Using such symbols we can replace the event model a_3^{edge} by a_3^{obs}, see Figure 4. The meaning of the label $S \leftarrow S \lhd A$ at the loop of e_0 is that agent S observes the event e_0 if S currently sees A ($S \lhd A$ is the case). We will now define these new technical constructs formally, and afterwards apply them to give an improved formalisation of the Sally-Anne task.

Edge-conditioned event models

Definition 4 (Edge-conditioned event models). *An* edge-conditioned event model *of $\mathcal{L}(P, \mathcal{A})$ is $\mathcal{E} = (E, Q, pre, post)$, where E, pre and $post$ are defined as for standard event models (Definition 2), and $Q : \mathcal{A} \to (E \times E \to \mathcal{L}(P, \mathcal{A}))$ assigns to each agent i a mapping $Q(i)$ from pairs of events into formulas of*

$\mathcal{L}(P, \mathcal{A})$. *The mapping* $Q(i)$ *is generally abbreviated* Q_i. *For* $e \in \mathcal{E}$, $(\mathcal{E}, e)$ *is called an* edge-conditioned action *of* $\mathcal{L}(P, \mathcal{A})$.

For standard event models (Definition 2), eQ_if means that event f is accessible from event e by agent i, and we draw an edge from e to f labelled i in the graph of the event model. In edge-conditioned event models, accessibility has become conditioned by a formula: $Q_i(e, f) = \phi$ means that f is accessible from e by i *under condition* ϕ. When $Q_i(e, f) = \phi$, we draw an edge from e to f labelled $i \leftarrow \phi$ in the graph of the event model (except when $Q_i(e, f) = \bot$). We already saw an example of such an edge-conditioned event model: a_3^{edge} of Figure 4. Note that edge-conditioned event models naturally generalise standard event models: Any standard event model $\mathcal{E} = (E, Q, pre, post)$ can be equivalently represented as an edge-conditioned event model $\mathcal{E}' = (E, Q', pre, post)$ by simply letting $Q'_i(e, f) = \top$ for all $(e, f) \in Q_i$ and $Q'_i(e, f) = \bot$ otherwise. We also have to generalise the notion of product update:

Definition 5 (Edge-conditioned product update). *Let a state* $(\mathcal{M}, w_0)$ *and an edge-conditioned action* $(\mathcal{E}, e_0)$ *be given, where* $\mathcal{M} = (W, R, V)$ *and* $\mathcal{E} = (E, Q, pre)$, *and where* $\mathcal{M}, w_0 \models pre(e_0)$. *The* product update *of* $(\mathcal{M}, w_0)$ *with* $(\mathcal{E}, e_0)$ *is defined as the state* $(\mathcal{M}, w_0) \otimes (\mathcal{E}, e_0) = ((W', R', V'), (w_0, e_0))$, *where* W' *and* V' *are defined as in the standard product update (Definition 3) and* $R'_i = \{((w, e), (v, f)) \in W' \times W' \mid wR_iv$ *and* $\mathcal{M}, w \models Q_i(e, f)\}$.

The only difference to the standard product update is that the R'_i relations have become parametrised by the $Q_i(e, f)$ formulas. There is an i-edge from a world-event pair (w, e) to a world-event pair (v, f) iff there is an i-edge from w to v in the epistemic model, and the condition $Q_i(e, f)$ for having an edge from e to f in the event model is true in w.

It can be shown that any edge-conditioned event model induces a standard event model in a canonical way, but the induced standard event model might be exponentially bigger. In technical terms, it can be shown that edge-conditioned event models are exponentially more succinct than standard event models (we will prove this and other interesting properties of edge-conditioned event models in a future paper). In particular, our generic event models for ontic actions and observability change (to be presented in Section 5) are going to consist of 2 events each, whereas the same actions using only standard event models would contain $2^{n-1} + 1$ events, where n is the number of agents!

Observability propositions

We now define a new language $\mathcal{L}^{obs}(P, \mathcal{A})$ extending $\mathcal{L}(P, \mathcal{A})$ by the addition of *observability propositions* on the form $i \lhd j$: $\phi ::= p \mid i \lhd j \mid \neg\phi \mid \phi \wedge \phi \mid B_i\phi$, where $p \in P$, $i, j \in \mathcal{A}$ and $i \neq j$. As noted above, the intended meaning of $i \lhd j$ is that "agent i observes all actions performed by agent j". The reason we do not include the reflexive propositions $i \lhd i$ is that we will assume that all agents always observe their own actions, so $i \lhd i$ is implicitly always true. This assumption can of course be relaxed, but we will not consider that here. In

the expression $i \lhd j$ we call i the *observer* and j the *observed*. Given a formula ϕ, we use $\pi_1(\phi)$ to denote the set of agents occurring as observers in ϕ, that is, $\pi_1(\phi) = \{i \mid i \lhd j \text{ is a subformula of } \phi \text{ for some } j\}$. For instance we have $\pi_1(i \lhd j \wedge \neg k \lhd l) = \{i, k\}$ (note that k is in the set even though the formula $k \lhd l$ occurs negated).

The idea of introducing observability propositions in the context of DEL was first introduced in [15]. They, however, only use a simpler type of proposition h_i with the intended meaning "agent i observes *all* actions" (agent i is in a state of paying attention to everything that happens). Here we need something more fine-grained, in particular for our later formalisation of the chocolate task (Section 6) where we need to be able to represent that an agent i is observing the actions of an agent j without j observing the actions of i.

Ontic actions and observability change

The previous definitions of edge-conditioned event models and product update extend to the language $\mathcal{L}^{obs}(P, \mathcal{A})$ in the obvious way (after all, we only added some additional atomic propositions). We can now finally define two generic types of edge-conditioned actions that are sufficient to formalise a number of different false-belief tasks of varying order. The first action type is an ontic action $do(i, \phi)$: agent i makes ϕ true. Step 1 of the Sally-Anne task is for instance going to be formalised by $do(S, t)$: Sally makes t true. The second is an *o*bservability *c*hanging action $oc(\phi)$ for changing who observes who. For instance step 2 of the Sally-Anne task where Sally leaves the room is going to be formalised by $oc(\neg S \lhd A \wedge \neg A \lhd S)$: Sally stops observing Anne ($\neg S \lhd A$), and Anne stops observing Sally ($\neg A \lhd S$).

Definition 6. *We define the following edge-conditioned actions on* $\mathcal{L}^{obs}(P, \mathcal{A})$.

- $do(i, \phi)$*: for each agent* i *and each conjunction of propositional literals* ϕ*, this is the* ontic action *shown at the top of Figure 5.*
- $oc(\phi)$*: for each conjunction of observability literals (observability propositions and their negations), this is the* observability changing action *shown at the bottom of Figure 5.*

These new actions need a little explanation. Consider first $do(i, \phi)$. As mentioned, this is an action where agent i makes ϕ true (since the actual event e_0 has postcondition ϕ). From the label at the loop of e_0 we can see that the agents who observe the action taking place, and hence come to believe ϕ, are: 1) agent i itself (since we have $i \leftarrow \top$); 2) any other agent who is observing agent i (since we have $j \in \mathcal{A} \setminus \{i\} : j \leftarrow j \lhd i$). The agents who are not observing i will think that nothing happens (the label $j \in \mathcal{A} \setminus \{i\} : j \leftarrow \neg j \lhd i$ on the edge to e_1). This also explains the title of the paper, "Seeing is believing": If agent j *sees* agent i, $j \lhd i$, then j comes to *believe* any formula ϕ that i brings about. The action $oc(\phi)$ is a bit more complicated, but follows the same principle (note that the two event models only differ in their edge labels). Looking at the labels of the loop at e_0, we can see that the agents observing the observability change are: 1) any agent

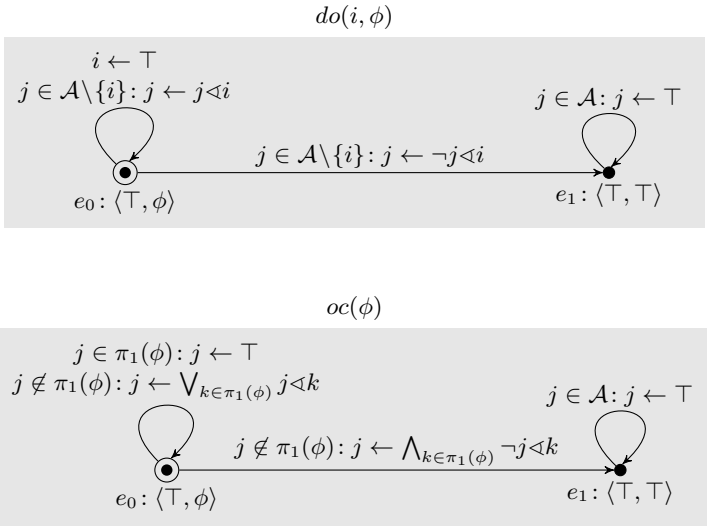


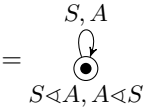
Fig. 5. The edge-conditioned actions $do(i, \phi)$ and $oc(\phi)$.

whose observer status is affected by the action (since we have $j \in \pi_1(\phi) : j \leftarrow \top$); 2) any other agent who is observing at least one of the aforementioned agents (since we have $j \notin \pi_1(\phi) : j \leftarrow \bigvee_{k \in \pi_1(\phi)} j \lhd k$). This means that if i is currently observing j, and j either starts or stops to observe k, then i will also observe this change. One could imagine intricate situations where this wouldn't hold, but for our purposes it is sufficient. The event model $oc(\phi)$ is a generalisation of the event models $+S$ and $-S$ introduced in [15].

A few final remarks before turning to present our improved formalisations of false-belief tasks. In standard DEL there is no explicit notion of agency, that is, an action simply happens without any need to say who did it. But in our *do* action we need to include the agent performing it as a parameter, since what will be observed by the other agents depends on it. To save space, we have chosen not to include an *announcement action* in our framework, even though this would be very simple: simply take the event model for $do(i, \phi)$ and put ϕ in the precondition instead of the postcondition of e_0.

6 New formalisations of false-belief tasks

Example 5 (Formalising the Sally-Anne task). Given the generic actions from the previous section, it is now quite straightforward to provide a new formalisation of the Sally-Anne task using these actions:

0. Sally is in the room with Anne, holding the marble: state $s_1 =$ 

1. Sally puts the marble into the basket: $a_1 = do(S, t)$.
2. Sally leaves the room: $a_2 = oc(\neg S \lhd A \wedge \neg A \lhd S)$.
3. Anne transfers the marble to the box: $a_3 = do(A, \neg t \wedge x)$.
4. Sally re-enters: $a_4 = oc(S \lhd A \wedge A \lhd S)$.

Note that we no longer use the atomic proposition p, as we now have a more generic way to deal with observability through our observability propositions. Similar to the previous formalisation in Section 4, it can now be checked that $s_1 \otimes a_1 \otimes \cdots \otimes a_4 \models B_S t$, hence again the formalisation gives the right answer to the Sally-Anne test. We should also note that now we have $s_1 \otimes a_1 \otimes a_3 \models B_S x$, so if Sally doesn't leave the room, she will not get a false belief. Thus we have successfully solved the problem of the shortened Sally-Anne task that was discussed in the beginning of Section 5. We will not show the detailed calculations, as we will save that for the next example, which formalises a more complex false-belief task.

Example 6 (Formalising the second-order chocolate task). We now consider a compact version of the second-order chocolate task presented in [18, 4]. It is illustrated in Figure 10 in the appendix. It has the following steps:

0. John and Mary are in a room. There is a chocolate bar in the room.
1. John puts the chocolate into the drawer.
2. John leaves the room.
3. John starts peeking into the room through the window, without Mary seeing.
4. Mary transfers the chocolate to the box.

The child taking the test is now asked "where does Mary believe that John believes the chocolate to be?" It is a second-order task since this question concerns second-order belief attribution (Mary's beliefs about John's beliefs). The correct answer is "in the drawer", since Mary is not aware that John was peeking while she moved the chocolate. It is immediate that step 1 and 4 above are ontic actions, and steps 2 and 3 are observability changing actions. Let us use atomic propositions d for the "the chocolate is in the $d$rawer" and x for "the chocolate is in the bo$x$." We use agent symbols J for John and M for Mary. Step 1, "John puts the chocolate into the drawer", must then be the ontic action $do(J, d)$. Step 2, "John leaves the room", must be the observability change $oc(\neg J \lhd M \wedge \neg M \lhd J)$ (John stops observing Mary and Mary stops observing John). Step 3 is again an observability change, but this time it is simply $oc(J \lhd M)$: John starts observing Mary. Finally, step 4 is the ontic action $do(M, \neg d \wedge x)$. Figure 6 calculates the result of executing the action sequence of steps 1–4 in the initial state described by step 0. The actions in the figure show the applied instances of $do(i, \phi)$ and $oc(\phi)$ calculated from Figure 5. Some of the states and actions contain grey nodes. These are nodes that are not accessible from the initial world/event, and can hence be ignored (by bisimulation contraction, to be technically precise).

Before going into the detailed calculations of Figure 6, let us have a look at the resulting model s_5. This is the model in which it should be checked where Mary believes John believes the chocolate to be. Clearly we have $s_5 \models B_M B_J d$, so the

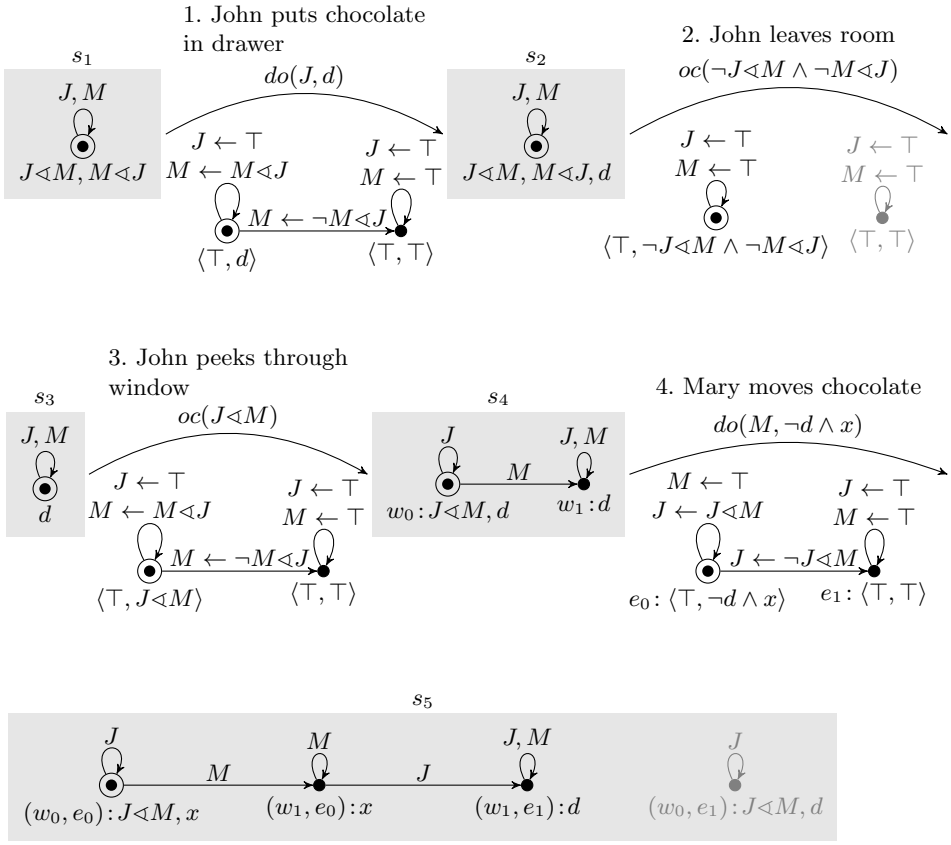


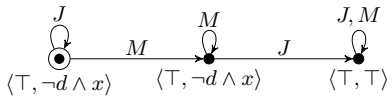
Fig. 6. The DEL-formalisation of the second-order chocolate task

agent's answer will be "in the drawer", hence passing the false-belief test. But s_5 can do more than just answer this question, in fact it is a full description of the final situation, including all beliefs to arbitrary order. Concerning observability, we can for instance see that $s_5 \models J \lhd M \wedge B_M \neg J \lhd M \wedge B_J B_M \neg J \lhd M$: John sees Mary, Mary believes he doesn't, and John knows this. We can also imagine a third-order version of the task, where the question is "Where does John believe that Mary believes that John believes the chocolate to be", and by consulting s_5 we immediately get the answer "in the drawer": $s_5 \models B_J B_M B_J d$.

The most interesting part of the calculation in Figure 6 is the last step, $s_5 = s_4 \otimes do(M, \neg d \wedge x)$, so we will explain this in more detail. Calculating the product $s_4 \otimes do(M, \neg d \wedge x)$ follows the same strategy as in Example 4. First we find the matching world-event pairs which, in this case, is all four world-event combinations (w_0, e_0), (w_0, e_1), (w_1, e_0) and (w_1, e_1), since both e_0 and e_1 have trivial preconditions (see Figure 6 where $do(M, \neg d \wedge x)$ is the event model

of step 4). In the world-event pairs containing e_0, the postcondition of e_0 is enforced, that is, d is made false and x true. The other world-event pairs simply inherit their label from the first element of the pair. Hence the four worlds of the resulting model s_5 are $(w_0, e_0) : J \lhd M, x$; $(w_0, e_1) : J \lhd M; d$, $(w_1, e_0) : x$; $(w_1, e_1) : d$. Now for the interesting part, the edges. At (w_0, e_0) we get a J-loop, since there is J-loop at w_0 and *the condition for having a J-loop at e_0 is $J \lhd M$, which is satisfied in w_0*. This should be contrasted with the situation at (w_1, e_0): Here we also have a J-loop at the world of the pair, w_1, but now *the condition $J \lhd M$ for having a J-loop at the event of the pair is* not *satisfied in the world of the pair*. At (w_1, e_0) we hence only get an M-loop (since both w_1 and e_0 unconditionally have such a loop). We leave the calculation of the rest of the edges to the (enthusiastic) reader.

Note that to get from s_4 to s_5 we only have to apply an instance of a generic edge-conditioned action with 2 events. This situation is much better than what can be achieved with standard event models. In Proposition 1 in the appendix we prove that there is no standard event model a with 2 events such that $s_5 = s_4 \otimes a$. This implies that the smallest standard event model that can produce s_5 from s_4 is this:



The problem with this event model is that it is already a 'second-order model' that fully encodes the structure of the model s_5 we wish to obtain. Hence if we had to formalise the second-order chocolate task using standard event models, we would have to formalise the step "Mary moves the chocolate" as this event model that already fully encodes the final structure achieved at the end of the story. This would certainly be very far from achieving the *faithfulness* criterion introduced in Section 2. So indeed the edge-conditioned event models make a real difference to the formalisation of false-belief tasks.

7 Conclusion, related work and future work

In this paper we have shown how to formalise two false-belief tasks—a first- and a second-order one—in an extension of dynamic epistemic logic. In the end, we were able to express the formalisations rather compactly:

- **Sally-Anne task**: $do(S, t), oc(\neg S \lhd A \wedge \neg A \lhd S), do(A, \neg t \wedge x), oc(S \lhd A \wedge A \lhd S)$.
- **Chocolate task**: $do(J, d), oc(\neg J \lhd M \wedge \neg M \lhd J), oc(J \lhd M), do(M, \neg d \wedge x)$.

We started out expressing two overall criteria for our formalisations of false-belief tasks: robustness and faithfulness. To be robust, the formalism should be able to formalise false-belief tasks of arbitrary order. We claim to have such robustness in our current formalism, put proving it formally is future work. Nevertheless, we *have* been able to show that we could go from a formalisation of a first-order false-belief task to a second-order one at no extra cost, which as discussed above

is *not* the case in standard DEL (and not in most other frameworks either). To have faithfulness, we required that it should be relatively straightforward to get from the informal action descriptions of the false-belief task to the corresponding formalised actions. We believe we have taken a big step closer towards achieving this. If the (semi-)informal description says "agent i makes ϕ true" it is our action $do(i, \phi)$. If the informal description says, e.g., "now agent i starts observing j without agent j noticing" it is $oc(i \lhd j)$. The formalisation step can of course still not be fully automated, but we are much closer than if we just had to build all the relevant event models from scratch, which was where this paper started.

There is of course also a limit to the types of false-belief tasks that can be dealt with using only *do* and *oc*. In particular, a lot of the existing false-belief tasks involve untruthful announcements such as the 'ice-cream task' [22], the 'birthday puppy task' [26] and the 'clown-in-the-park task' [28]. These can not be dealt with in the current framework. To be able to deal with untruthful announcements and the revision of false beliefs, we need another type of model called plausibility models [7]. We plan to show how these models can be used to formalise the aforementioned false-belief tasks in a future paper.

In our approach, observability amounts to 'who sees who', that is, it is a relation between agents. Other approaches to modelling observability can be found in e.g. [14, 20, 8, 15]. In these approaches, observability is instead connected either to propositions [14, 20], particular actions [8] or *all* actions [15]. The paper [24] uses a similar approach to observability as we do, but in a more complex 2-dimensional dynamic epistemic logic. In the papers [14, 8], observability is encoded using axioms instead of being encoded into the states as we do. For us, it is very important to encode observability directly into the states to be able to deal with higher-order observability ('you don't see me seeing you').

Even though edge-conditioned event models is an original idea of this paper, they are close in spirit to the *generalized arrow updates* of [21]. However, arrow updates are rather an *alternative* to event models, whereas our edge-conditioned event models is a straightforward *generalisation* of event models. Furthermore, arrow updates are purely epistemic (without postconditions), and would hence not be able to represent the ontic actions of the false-belief tasks.

Solving false-belief tasks using DEL as we do in this paper is part of a larger research effort in *epistemic planning*: combining automated planning with DEL to integrate higher-order social cognition into intelligent planning agents [11, 1]. Combining the ideas of [11, 1] with the ideas of this paper will allow us to devise algorithms not only for *analysing* false beliefs (as is done in the false-belief tasks), but also for *synthesising* them. It could e.g. be that Anne plans to deceive Sally by asking her to go outside and then she moves the marble meanwhile. This is a case of epistemic planning where the goal is to achieve a state where Sally does not know the location of the marble.

References

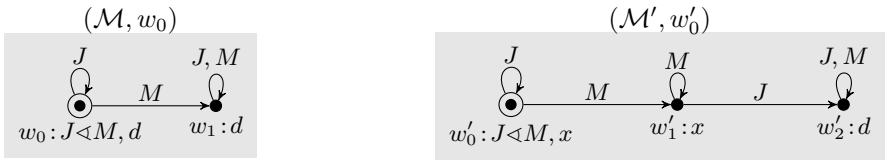
1. Andersen, M.B., Bolander, T., Jensen, M.H.: Conditional epistemic planning. Lecture Notes in Artificial Intelligence 7519, 94–106 (2012), proceedings of JELIA

2012
2. Arkoudas, K., Bringsjord, S.: Toward formalizing common-sense psychology: An analysis of the false-belief task. In: Ho, T.B., Zhou, Z.H. (eds.) PRICAI. Lecture Notes in Computer Science, vol. 5351, pp. 17–29. Springer (2008)
3. Arslan, B., Verbrugge, R., Taatgen, N., Hollebrandse, B.: Teaching children to attribute second-order false beliefs: A training study with feedback. Submitted.
4. Arslan, B., Taatgen, N., Verbrugge, R.: Modeling developmental transitions in reasoning about false beliefs of others. In: Proc. of the 12th International Conference on Cognitive Modelling (2013)
5. Baltag, A., Moss, L.: Logic for epistemic programs. Synthese 139(2), 165–224 (2004)
6. Baltag, A., Moss, L.S., Solecki, S.: The logic of public announcements and common knowledge and private suspicions. In: TARK. pp. 43–56 (1998)
7. Baltag, A., Smets, S.: A qualitative theory of dynamic interactive belief revision. In: Bonanno, G., van der Hoek, W., Wooldridge, M. (eds.) Logic and the Foundations of Game and Decision Theory (LOFT7). Texts in Logic and Games, vol. 3, pp. 13–60. Amsterdam University Press (2008)
8. Baral, C., Gelfond, G., Son, T.C., Pontelli, E.: An action language for reasoning about beliefs in multi-agent domains. In: Proceedings of the 14th International Workshop on Non-Monotonic Reasoning (2012)
9. Baron-Cohen, S., Leslie, A.M., Frith, U.: Does the autistic child have a theory of mind? Cognition 21(1), 37–46 (1985)
10. van Benthem, J., van Eijck, J., Kooi, B.: Logics of communication and change. Information and Computation 204(11), 1620–1662 (2006)
11. Bolander, T., Andersen, M.B.: Epistemic planning for single- and multi-agent systems. Journal of Applied Non-Classical Logics 21, 9–34 (2011)
12. Braüner, T.: Hybrid-logical reasoning in false-belief tasks. In: Schipper, B. (ed.) Proceedings of Fourteenth Conference on Theoretical Aspects of Rationality and Knowledge (TARK). pp. 186–195 (2013)
13. Breazeal, C., Gray, J., Berin, M.: Mindreading as a foundational skill for socially intelligent robots. In: Robotics Research, pp. 383–394. Springer (2011)
14. Brenner, M., Nebel, B.: Continual planning and acting in dynamic multiagent environments. Autonomous Agents and Multi-Agent Systems 19(3), 297–331 (2009)
15. van Ditmarsch, H., Herzig, A., Lorini, E., Schwarzentruber, F.: Listen to me! public announcements to agents that pay attention—or not. In: Logic, Rationality, and Interaction, pp. 96–109. Springer (2013)
16. van Ditmarsch, H., van der Hoek, W., Kooi, B.: Dynamic epistemic logic with assignment. In: Dignum, F., Dignum, V., Koenig, S., Kraus, S., Singh, M.P., Wooldridge, M. (eds.) Autonomous Agents and Multi-agent Systems (AAMAS 2005). pp. 141–148. ACM (2005)
17. van Ditmarsch, H., Kooi, B.: Semantic results for ontic and epistemic change. In: Bonanno, G., van der Hoek, W., Wooldridge, M. (eds.) Logic and the Foundation of Game and Decision Theory (LOFT 7). pp. 87–117. Texts in Logic and Games 3, Amsterdam University Press (2008)
18. Flobbe, L., Verbrugge, R., Hendriks, P., Krämer, I.: Childrens application of theory of mind in reasoning and language. Journal of Logic, Language and Information 17(4), 417–442 (2008), special issue on formal models for real people, edited by M. Counihan
19. Frith, U.: Mind blindness and the brain in autism. Neuron 32(6), 969–979 (2001)
20. van der Hoek, W., Troquard, N., Wooldridge, M.: Knowledge and control. In: The 10th International Conference on Autonomous Agents and Multiagent Systems-

Volume 2. pp. 719–726. International Foundation for Autonomous Agents and Multiagent Systems (2011)
21. Kooi, B., Renne, B.: Generalized arrow update logic. In: Proceedings of the 13th Conference on Theoretical Aspects of Rationality and Knowledge. pp. 205–211. ACM (2011)
22. Perner, J., Wimmer, H.: john thinks that mary thinks that attribution of second-order beliefs by 5-to 10-year-old children. Journal of experimental child psychology 39(3), 437–471 (1985)
23. Premack, D., Woodruff, G.: Does the chimpanzee have a theory of mind? Behavioral and Brain Sciences 1(4), 515–526 (1978)
24. Seligman, J., Liu, F., Girard, P.: Facebook and the epistemic logic of friendship. In: Schipper, B. (ed.) Proceedings of Fourteenth Conference on Theoretical Aspects of Rationality and Knowledge (TARK). pp. 229–238 (2013)
25. Sindlar, M.P.: In the Eye of the Beholder: Explaining Behavior through Mental State Attribution. Ph.D. thesis, Universiteit Utrecht (2011)
26. Sullivan, K., Zaitchik, D., Tager-Flusberg, H.: Preschoolers can attribute second-order beliefs. Developmental Psychology 30(3), 395 (1994)
27. Verbrugge, R.: Logic and social cognition. Journal of Philosophical Logic 38(6), 649–680 (2009)
28. Wahl, S., Spada, H.: Childrens reasoning about intentions, beliefs and behaviour. Cognitive Science Quarterly 1(1), 3–32 (2000)
29. Wimmer, H., Perner, J.: Beliefs about beliefs: Representation and constraining function of wrong beliefs in young children's understanding of deception. Cognition 13(1), 103–128 (1983)

Appendix

Proposition 1. *Let* $(\mathcal{M}, w_0)$ *and* $(\mathcal{M}', w_0)$ *be the following two models:*



There exists no standard event model $\mathcal{E}$ *with only 2 events* e_0 *and* e_1 *such that* $(\mathcal{M}, w_0) \otimes (\mathcal{E}, e_0)$ *contains* $(\mathcal{M}', w_0')$ *as a submodel.*

Proof. Assume the opposite. Then since $(\mathcal{M}', w_0')$ is a submodel of $(\mathcal{M}, w_0) \otimes (\mathcal{E}, e_0)$, all of the worlds w_0', w_1' and w_2' of $(\mathcal{M}', w_0')$ must be among the pairs (w_0, e_0), (w_0, e_1), (w_1, e_0) and (w_1, e_1) of $(\mathcal{M}, w_0) \otimes (\mathcal{E}, e_0)$. First we can conclude that $w_0' = (w_0, e_0)$, since the actual worlds and events have to match. Since w_0 has the label $J \lhd M, d$ whereas w_0' has the label $J \lhd M, x$, e_0 must be an event with a postcondition including $\neg d$ and x as conjuncts. Since w_2' doesn't satisfy x, w_2' can then not be of the form $(\cdot, e_0)$ (that is, it is not (w, e_0) for any w). Therefore w_2' must be of the form $(\cdot, e_1)$. Since w_2' doesn't satisfy x, e_1 must then be an event that *doesn't* have x as a (positively occurring) conjunct in its postcondition. Since w_1' satisfies x but none of w_0 or w_1 satisfies it, we can conclude that w_1' can not be on the form $(\cdot, e_1)$. Hence, we must have $w_1' = (w_0, e_0)$ or $w_1' = (w_1, e_0)$, but since $w_0' = (w_0, e_0)$ and $w_0' \neq w_1'$ we can conclude $w_1' = (w_1, e_0)$.

We have now concluded $w_0' = (w_0, e_0)$ and $w_1' = (w_1, e_0)$. Since there is a J-loop at w_0' and $w_0' = (w_0, e_0)$, e_0 must then also have a J-loop (cf. the definition of product update). Similarly, since there is an M-loop at w_1' and $w_1' = (w_1, e_0)$, e_0 must also have an M-loop. We can conclude that e_0 contains a J, M-loop. Now since w_1 contains a J, M-loop, we get that $w_1' = (w_1, e_0)$ must also contain a J, M-loop. But looking at the world w_1' of $(\mathcal{M}', w_0')$ we see it only has an M-loop, and hence we have a contradiction, completing the proof.

The proof given is slightly intricate, but the intuition is rather clear: w_0' and w_1' must necessarily be updates with the same event (e_0), but there is no way a standard such event can produce a J-loop at w_0' and an M-loop at w_1' without having both a J-loop and an M-loop itself. But if it has, then when updating w_1 with this event we get a J, M-loop either at w_0' or w_1'. The reason that it works with edge-conditioned event models is of course that they allow us to let the edges of events depend on the world they are applied in, so in this case it is not a problem to have a single event e_0 which produces one type of loop in one world and another type of loop in another.

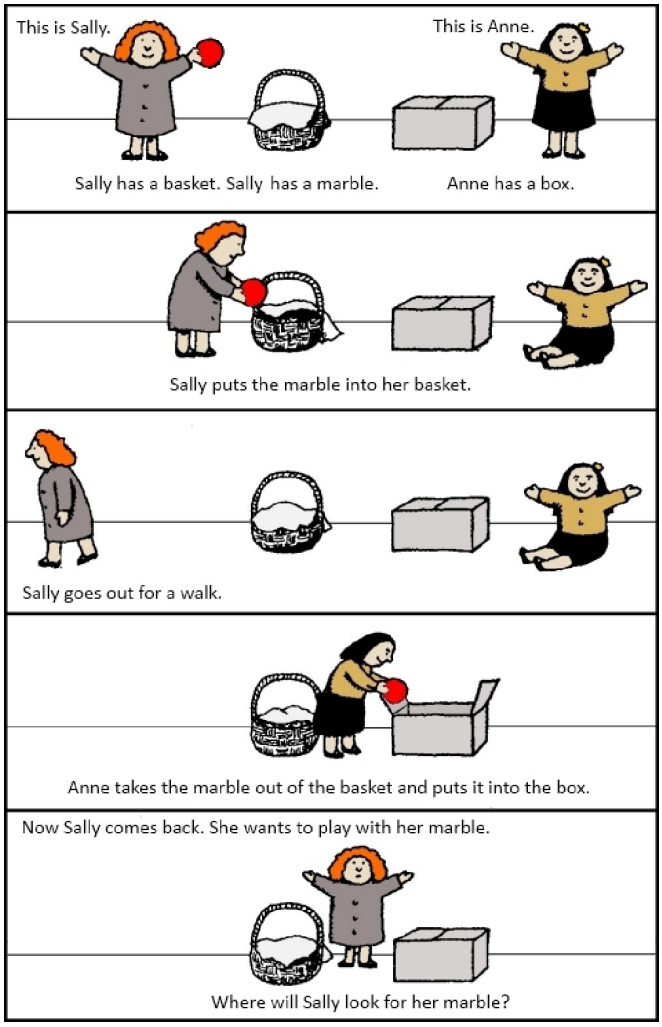


Fig. 7. An illustration of the Sally-Anne false belief task adapted from [19].

system/reference	year	formalism/platform	h-o reas.	other features
CRIBB [28]	2000	Prolog	≤ 2	goal recognition, plan recognition
Edd Hifeng [2]	2008	event calculus	≤ 1	Second Life avatar
Leonardo [13]	2011	C5 agent architecture	≤ 1	goal recognition, learning
[25]	2011	extension of PDL, implementation in 2APL	≤ 1	goal recognition
ACT-R agent [4]	2013	ACT-R cognitive architecture	∞	learning
[12]	2013	hybrid logic	∞	temporal reasoning

Fig. 8. Existing full formalisations/implementations of false-belief tasks, ordered chronologically. The numbers in the 'h-o reas.' column refer to the highest level of belief attribution the formalism/system allows (∞ if there is no upper bound).

$$
\begin{array}{lll}
(\mathcal{M}, w) \models p & \text{iff} & w \in V(p) \\
(\mathcal{M}, w) \models \neg\phi & \text{iff} & \mathcal{M}, w \not\models \phi \\
(\mathcal{M}, w) \models \phi \wedge \psi & \text{iff} & \mathcal{M}, w \models \phi \text{ and } \mathcal{M}, w \models \psi \\
(\mathcal{M}, w) \models B_i\phi & \text{iff} & \text{for all } v \in W, \text{ if } wR_iv \text{ then } \mathcal{M}, v \models \phi
\end{array}
$$

Fig. 9. Truth conditions for the epistemic language where $\mathcal{M} = (W, R, V)$ is an epistemic model, $i \in \mathcal{A}$, $w \in W$ and $\phi, \psi \in \mathcal{L}(P, \mathcal{A})$.

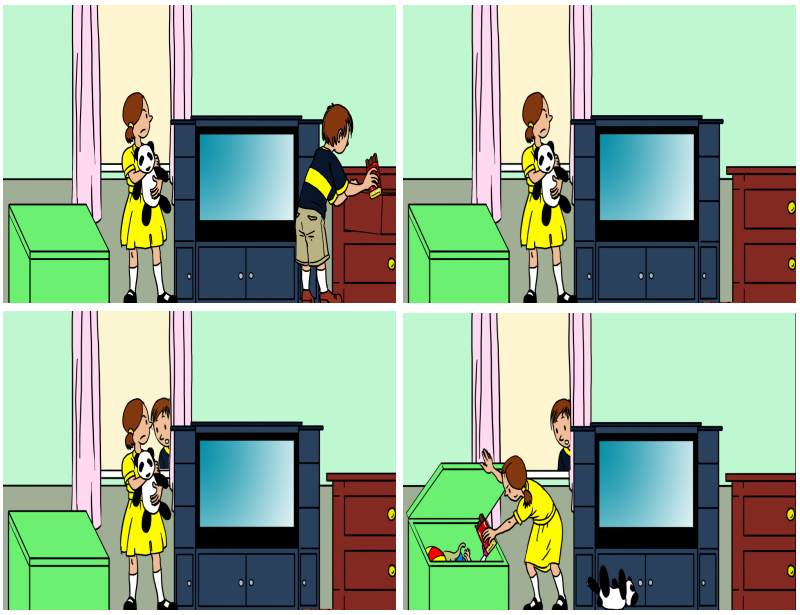


Fig. 10. Illustrations © Avik Kumar Maitra, with kind permission of the authors of [4, 3].