Dynamic Epistemic Logic with ASP Updates: Application to Conditional Planning

Pedro Cabalar[a], Jorge Fandinno[b] and Luis Fariñas del Cerro[c]

[a]*University of Corunna, SPAIN*

[b]*University of Potsdam, GERMANY*

[c]*Institut de Recherche en Informatique de Toulouse, Universty of Toulouse, CNRS, FRANCE*

Abstract

Dynamic Epistemic Logic (DEL) is a family of multimodal logics that has proved to be very successful for epistemic reasoning in planning tasks. In this logic, the agent's knowledge is captured by modal epistemic operators whereas the system evolution is described in terms of (some subset of) dynamic logic modalities in which actions are usually represented as semantic objects called *event models*. In this paper, we study a variant of DEL, that we call DEL[ASP], where actions are syntactically described by using an Answer Set Programming (ASP) representation instead of event models. This representation directly inherits high level expressive features like indirect effects, qualifications, state constraints, defaults, or recursive fluents that are common in ASP descriptions of action domains. Besides, we illustrate how this approach can be applied for obtaining conditional plans in single-agent, partially observable domains where knowledge acquisition may be represented as indirect effects of actions.

Keywords

Answer Set Programming, Dynamic Epistemic Logic, Epistemic Logic Programs, Epistemic Specifications, Conditional Planning, Equilibrium Logic, Non-Monotonic Reasoning

Introduction

Automated planning is the field of Artificial Intelligence concerned with the generation of strategies to achieve a goal in a given dynamic domain. A planner usually starts from a formal representation of the domain, a particular instance of the problem and the goal to achieve. The planner output is some strategy, expressed in terms of actions that cause the state transitions to reach the goal. The most common situation is that such a strategy is just a sequence of actions called a *plan*. In Classical Planning [1] some simplifying restrictions are assumed: the system has a *finite* number of states, the world is *fully observable* and the transition relation is *deterministic* and *static* (i.e. transitions are only caused by the execution of actions). However, a rational agent may easily face planning problems that require relaxing these assumptions. For instance, a robot may not possess all the information about the environment, either because its sensors have a limited scope, or because its actions may have non-deterministic effects that require observation for finding out the real outcome. Removing the assumptions of determinism and fully observable world naturally leads to two important questions [2]: (i) *how does a plan*

look like in this new context? and (ii) *how to represent the changes in the agent's knowledge along the plan execution?*

Regarding (i), two new categories of plans have been defined in this context: *conformant plans* and *conditional plans*. A *conformant plan* is a sequence of actions that guarantees achieving the goal regardless unknown values of the fluents in the initial situation or the precise effect of the non-deterministic actions. If we further allow sensing actions (acquiring knowledge from the environment) then the structure of a sequential plan is not reasonable any more: a *conditional plan* may contain "if-then-else" constructs that allow the agent to follow different strategies depending on the knowledge she acquired when executing the plan. Approaches to both conformant and conditional planning have been broadly studied in the literature [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27].

With respect to question (ii), several approaches studied the effects of sensing actions [4, 6, 18, 19, 21, 22, 28, 29, 30, 31]. One prominent line of research is based on *Dynamic Epistemic Logic* (DEL) [32, 33], a multi-modal approach where the agent's knowledge is captured by modal epistemic operators whereas the system evolution is described in terms of (some subset of) dynamic logic [34] modalities. For instance, the DEL expression $[\texttt{watch}](\mathbf{K}\,rain \vee \mathbf{K} \sim rain)$ represents that, after any possible outcome of sensing action $\texttt{watch}$, the agent knows whether $rain$ holds or not. Different variants of DEL have been successfully applied to the problem of planning with non-deterministic, partially observable multi-agent domains [18, 19, 21, 22, 35]. Although DEL has proved to be very convenient for epistemic reasoning in planning, it shows some important drawbacks when analysed from a Knowledge Representation (KR) viewpoint. This is because, for representing actions and their effects, DEL uses the so-called *event models* [36, 37], that inherit some of the expressive limitations of the STRIPS planning language [38]. In particular, event models do not allow some important KR features, like the treatment of indirect effects, action qualifications, state constraints or recursive fluents, that are quite common in modern representation of action theories.

One popular KR formalism that naturally covers these expressive features is *Answer Set Programming* (ASP) [39, 40, 41], a well-established paradigm for problem solving and non-monotonic reasoning based on the *stable models* semantics [42, 43]. The use of ASP for classical planning was introduced in [44, 45], leading to a methodology adopted by different high-level action languages (see [46] and references there) and, more recently, to a temporal extension of ASP [47]. Besides, there exists a growing list of ASP applications [48], many of them dealing with classical planning problems. When moving to conformant planning, though, the application of ASP is still under an exploratory stage. Most of the attempts in this direction relied on a extension called *epistemic specifications* [49] that incorporate modal constructs (called *subjective literals*) for representing the agent's knowledge. However, the semantic interpretation of this formalism is still under debate and only some preliminary implementations are still available – see [50] for a recent survey. On the other hand, the use of ASP to obtain conditional plans was still an unexplored territory.

In this paper, we study the case of single-agent planning and combine both approaches, DEL and (epistemic) ASP, to exploit the advantages of both formalisms in a single language. Our proposal, called DEL[ASP], relies on replacing event models by epistemic logic programs. In that way, the basic event to describe the transition between two epistemic models becomes an ASP epistemic specification, while we keep the same dynamic logic operators for temporal reasoning

among transitions. On the one hand, with respect to DEL, the new approach provides all the expressive power of ASP for action domains: indirect effects, qualifications, state constraints, defaults, or recursive fluents are directly inherited from ASP. Moreover, when a classical planning scenario (represented in ASP) becomes partially observable, the new approach allows *keeping the scenario representation untouched*, possibly adding new epistemic rules to describe the effects of sensing actions. On the other hand, with respect to (non-temporal) epistemic ASP, dynamic operators provide a comfortable way for explicitly representing, and formally reasoning about conformant and conditional plans.

The rest of the paper is organised as follows. In the next section, we provide some background on the formalisms that conform our proposal. After these preliminaries, we introduce the formalism of DEL[ASP] and explain its behaviour using some examples. Then, we study the representation of conditional plans in this formalism. Finally, we discuss some related work and conclude the paper.

Preliminaries

In this section, we provide some background on planning in DEL, planning in ASP, and the ASP extension of epistemic specifications, since these three components will be present in DEL[ASP] up to some degree. In the case of DEL, we will present a slight generalisation of [19, 21] that admits abstract *updating objects*. These objects correspond to event models for standard DEL, which we denote here as DEL[$\mathcal{E}$], and will become epistemic specifications for DEL[ASP]. For the case of epistemic logic programs, we will use a recent logical formalisation [51] that avoids the problem of self-supported conclusions present in the original semantics [49]. This logic, called *Founded Autoepistemic Equilibrium Logic* (FAEEL) is a combination of Pearce's *Equilibrium Logic* [52], a well-known logical characterisation of stable models, with Moore's *Autoepistemic Logic* (AEL) [53], one of the most representative approaches among modal non-monotonic logics.

Dynamic Epistemic Logic with Abstract Updating Objects

Given a set of propositional symbols $\mathbf{P}$ and a set of updating objects $\mathbf{O}$, a *(dynamic epistemic) formula* φ is defined according to the following grammar:

$$\varphi ::= \bot \mid p \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \varphi \rightarrow \varphi \mid \sim\varphi \mid \mathbf{K}\,\varphi \mid [o]\varphi$$

where $p \in \mathbf{P}$ is a proposition and $o \in \mathbf{O}$ an updating object. The modal epistemic operator $\mathbf{K}$ represents the (planning) agent's knowledge: formula $\mathbf{K}\,\varphi$ means that "the agent knows φ." The symbol "$\sim$" stands here for classical negation (we reserve the symbol "$\neg$" for intuitionistic negation later on). A formula φ is called *objective* if the operator $\mathbf{K}$ does not occur in it. It is called *subjective* if it has at least some proposition and every proposition is in the scope of $\mathbf{K}$. As usual, we define the following abbreviations: $\varphi \leftrightarrow \psi \stackrel{\text{def}}{=} (\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi)$, $(\varphi \leftarrow \psi) \stackrel{\text{def}}{=} (\psi \rightarrow \varphi)$, and $\top \stackrel{\text{def}}{=} \sim\bot$. We also define the dual of $\mathbf{K}$ as follows: $\hat{\mathbf{K}}\,\varphi \stackrel{\text{def}}{=} \sim\mathbf{K}\sim\varphi$. We keep the Boolean operators $\vee, \wedge, \rightarrow, \bot$ and avoid defining ones in terms of the others, since this will not be valid when we use an intuitionistic reading later on. By $\mathcal{L}_{\text{E}}(\mathbf{P})$ we denote the language containing all dynamic epistemic formulas over $\mathbf{P}$.

We provide next an abstract semantics that just relies on two basic concepts: *epistemic models* that represent the agent's knowledge; and the *updating evaluation*, a pair of generic functions that describe how updating objects cause transitions among those models.

Definition 1 (Epistemic Model). *Given a (possibly infinite) set of propositional symbols* **P** *and a (possibly infinite) set of possible worlds* **W**, *a* model *is a triple* $\mathcal{M} = \langle W, \mathcal{K}, V \rangle$ *where*

- $W \subseteq \mathbf{W}$ *is a finite set of worlds,*

- $\mathcal{K} \subseteq W \times W$ *is an accessibility relation on* W, *and*

- $V : W \longrightarrow 2^{\mathbf{P}}$ *is a valuation function.*

$D(\mathcal{M}) = W$ *denotes the domain of* $\mathcal{M}$. *An* epistemic model *is a model where* $\mathcal{K}$ *is an equivalence relation: it is further called* information cell *if* $\mathcal{K}$ *is an universal relation. A* belief model *is a model where* $\mathcal{K} = W \times W'$ *with* $W' = W \setminus \{w_0\}$ *for some* $w_0 \in W$. *By* **M** *we denote the set of all possible epistemic models over* **P** *and* **W**. $\square$

We assume that the modeller coincides with the planning agent (the one whose knowledge is captured by the epistemic models). This is usually called an *internal* point of view, as opposed to the *external* one where the modeller is a different agent, an omniscient and external observer who can differentiate the actual world and knows its configuration [54, 55]. Adopting the internal orientation translates in the lack of a designated world (all worlds in the model are equally possible). A second consequence is that, even for single-agent epistemic models, we cannot replace the equivalence relation by a universal one.

Before going into further technical details, let us introduce the following scenario from [21], which will be our running example throughout the paper.

Example 1. *After following carefully laid plans, a thief has almost made it to her target: the vault containing the invaluable Pink Panther diamond. Standing outside the vault, she now deliberates on how to get her hands on the diamond. She* knows *the light inside the vault is off, and that the* Pink Panther is on either the right or the left pedestal inside. *Obviously, the diamond cannot be on both the right and left pedestal, but nonetheless the agent may be uncertain about its location. Note that the thief is perfectly capable of moving in the darkness and take whatever is on top any of the pedestals, but she is not able to know whether the diamond has been taken or not. It is assumed that there are four possible actions:* move, flick, take_left *and* take_right. *The action* move *changes the location of the thief from outside the vault* ($\sim v$) *to inside the vault* (v) *and vice-versa. The action* flick *turns on the light* (l). *Furthermore, if the thief is in the vault* (v) *and the light is on* (l), *the thief can see* (s) *where the Pink Panther is. Finally, actions* take_left *and* take_right *respectively take the diamond* (d) *from the left* ($\sim r$) *or right* (r) *pedestal if the diamond is in the intended pedestal.* $\square$

The set of propositions for the example is $\mathbf{P} = \{v, l, r, s, d\}$. Figure 1 depicts three consecutive epistemic models, $\mathcal{M}_0$, $\mathcal{M}_1$ and $\mathcal{M}_2$, respectively corresponding to the initial state of Example 1 and the resulting states after performing the sequence of actions move and then flick.

For improving readability, we represent the world valuations as strings of propositions and underline those that are false. For instance, the valuation for w_1 in Figure 1a is depicted
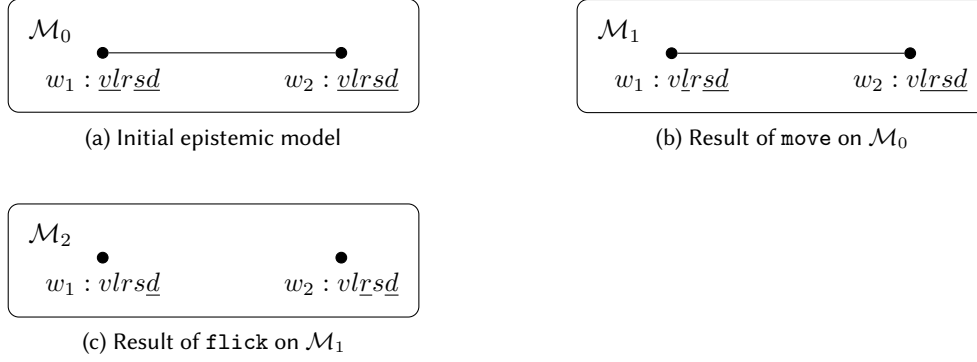
Figure 1: Sequence of epistemic models $\mathcal{M}_0, \mathcal{M}_1, \mathcal{M}_2$ that result from actions `move` and then `flick` starting in the initial state $\mathcal{M}_0$ of Example 1.

as $\underline{vlrsd}$ and corresponds to the set of true atoms $\{r\}$, making false all the rest. The initial model, $\mathcal{M}_0$ represents the triple $\langle W_0, \mathcal{K}_0, V_0 \rangle$ where we have two worlds $W_0 = \{w_1, w_2\}$ universally connected, that is, $\mathcal{K}_0 = W_0 \times W_0 = \{(w_1, w_2), (w_2, w_1), (w_1, w_1), (w_2, w_2)\}$ with valuations $V_0(w_1) = \{r\}$ and $V_0(w_2) = \emptyset$. This means that the agent knows that the thief is outside the vault ($\underline{v}$), the light is off ($\underline{l}$), she cannot see where the Pink Panther is ($\underline{s}$) and she does not have the diamond ($\underline{d}$). The two connected worlds reveal that she does not know whether the Pink Panther is on the right pedestal (world w_1) or on the left one (world w_2). The epistemic model $\mathcal{M}_1$ in Figure 1b reflects the same configuration, with the only exception that now the thief is in the vault (v), as a result of moving inside. That is, $\mathcal{M}_1 = \langle W_0, \mathcal{K}_0, V_1 \rangle$ with V_1 satisfying $V_1(w_1) = \{v, r\}$ and $V_0(w_2) = \{v\}$. Finally, $\mathcal{M}_2$ shows the epistemic model that results from performing action `flick` on $\mathcal{M}_1$. In this third model, two relevant changes occur: first, l and s became true in both worlds, since flicking turns on the light, and then, the thief can see the interior of the vault. Second, more importantly, the two worlds became completely disconnected, something that reveals that the agent has now complete knowledge of the world configuration in the two possible cases, w_1 and w_2. Formally, we have that $\mathcal{M}_2 = \langle W_0, \mathcal{K}_2, V_2 \rangle$ with $\mathcal{K}_2 = \{(w_1, w_1), (w_2, w_2)\}$ and with V_2 satisfying $V_2(w_1) = \{v, l, r, s\}$ and $V_2(w_2) = \{v, l, s\}$.

As we can see, the accessibility relation needs not be universal: for instance, we had $(w_1, w_2) \notin \mathcal{K}_2$ in $\mathcal{M}_2$ above. In general, when $(w, w') \notin \mathcal{K}$ we say that the two worlds are *indistinguishable at plan-time*, given epistemic model $\mathcal{M} = \langle W, \mathcal{K}, V \rangle$ with $\{w_1, w_2\} \subseteq W$. In the example, model $\mathcal{M}_2$ tells us that, before executing the plan, the agent cannot tell which of the two possibilities represented by worlds w_1 (diamond on the right) and w_2 (diamond on the left) will correspond to the actual world. However, once she executes the plan, she will acquire that knowledge in both cases: w_1 and w_2 are disconnected, so uncertainty in the agent's knowledge is completely removed. As a result, at that point, she will be able to make a decision whether she should perform the `take_right` or `take_left` action.

If, on the contrary, two worlds w, w' are connected $(w, w') \in \mathcal{K}$ in some epistemic model, we say that they are *indistinguishable at run-time*. This expresses a higher degree of uncertainty, since the agent has no way to tell which world corresponds to "reality" either before or during

the plan execution. For instance, at model $\mathcal{M}_1$ we have $(w_1, w_2) \in \mathcal{K}_1$ meaning that, during the plan execution, the agent will not be able to decide (at that point) whether actions `take_right` or `take_left` will work.

Until now, we have only described the information captured by epistemic models and presented an example with transitions among them, but did not specify how those transitions were obtained. For that purpose, we will only assume, by now, the existence of a pair of generic functions called *updating evaluation* defined below.

Definition 2 (Updating evaluation). *Given a set of models $\mathbf{M}$ over set of worlds $\mathbf{W}$ and a set of updating objects $\mathbf{O}$, an* updating evaluation *is a pair $\langle \otimes, \mathcal{R} \rangle$ of partial functions $\otimes : \mathbf{M} \times \mathbf{O} \longrightarrow \mathbf{M}$ and $\mathcal{R} : \mathbf{M} \times \mathbf{O} \longrightarrow 2^{\mathbf{W} \times \mathbf{W}}$ satisfying $\mathcal{R}(\mathcal{M}, o) \subseteq D(W) \times D(\otimes(\mathcal{M}, o))$.* $\square$

Function $\otimes$ takes an initial model $\mathcal{M}$ and some updating object o and provides a successor model $\mathcal{M}' = \otimes(\mathcal{M}, o)$. We will usually write $\otimes$ in infix notation and assume that it is left associative so that $\mathcal{M} \otimes o$ stands for $\otimes(\mathcal{M}, o)$ and $\mathcal{M} \otimes o_1 \otimes o_2$ stands for $(\mathcal{M} \otimes o_1) \otimes o_2$. Relation $\mathcal{R}(\mathcal{M}, o)$ matches worlds from the initial model $\mathcal{M}$ and its successor model $\mathcal{M}' = \mathcal{M} \otimes o$. We will also write $\mathcal{R}_{\mathcal{M}, o}$ instead of $\mathcal{R}(\mathcal{M}, o)$.

At this point, we have all the elements for defining the satisfaction of dynamic epistemic formulas.

Definition 3 (Satisfaction). *Let $\langle \otimes, \mathcal{R} \rangle$ be an updating evaluation. Then, given an epistemic model $\mathcal{M} = \langle W, \mathcal{K}, V \rangle$ and world $w \in W$, satisfaction of formulas is given by the following recursive definition:*

- $\mathcal{M}, w \not\models \bot$,

- $\mathcal{M}, w \models p$ *iff* $p \in V(w)$,

- $\mathcal{M}, w \models \varphi_1 \wedge \varphi_2$ *iff* $\mathcal{M}, w \models \varphi_1$ *and* $\mathcal{M}, w \models \varphi_2$,

- $\mathcal{M}, w \models \varphi_1 \vee \varphi_2$ *iff* $\mathcal{M}, w \models \varphi_1$ *or* $\mathcal{M}, w \models \varphi_2$,

- $\mathcal{M}, w \models \varphi_1 \rightarrow \varphi_2$ *iff* $\mathcal{M}, w \not\models \varphi_1$ *or* $\mathcal{M}, w \models \varphi_2$,

- $\mathcal{M}, w \models \sim\varphi$ *iff* $\mathcal{M}, w \not\models \varphi$,

- $\mathcal{M}, w \models \mathbf{K}\varphi$ *iff* $\mathcal{M}, w' \models \varphi$ *for all* w' *with* $(w, w') \in \mathcal{K}$, *and*

- $\mathcal{M}, w \models [o]\varphi$ *iff* $\mathcal{M} \otimes o$ *and* $\mathcal{R}_{\mathcal{M}, o}$ *are defined and* $\mathcal{M} \otimes o, w' \models \varphi$
  holds for all w' *with* $(w, w') \in \mathcal{R}_{\mathcal{M}, o}$. $\square$

As usual, we write $\mathcal{M} \models \varphi$ iff $\mathcal{M}, w \models \varphi$ for every world $w \in W$. Furthermore, given a theory Γ, we write $\mathcal{M} \models \Gamma$ iff $\mathcal{M} \models \varphi$ for every formula $\varphi \in \Gamma$. We say that theory Γ *entails* formula ψ, also written $\Gamma \models \psi$, iff $\mathcal{M} \models \Gamma$ implies $\mathcal{M} \models \psi$ for any epistemic model $\mathcal{M} \in \mathbf{M}$.

It is easy to see that the semantics for the dynamic-free fragment of the language (i.e., without $[\cdot]$ operator) corresponds to modal logic S5 (see [56] for instance).

Dynamic Epistemic Logic with Event Model Updates: $\mathrm{DEL}[\mathcal{E}]$

Let us now see how these definitions apply to the case in which updating objects are event models [36]. The following is an adaptation of the definition from [21]. A first peculiarity of event models is that, when making an update $\mathcal{M} \otimes o = \mathcal{M}'$, the resulting epistemic model $\mathcal{M}'$ uses world names of the form (w, e) where w is a world from the updated epistemic model $\mathcal{M}$ and e is a world (or event) from the event model o. For this reason, along this section, we assume that the global set of available world names $\mathbf{W}$ is closed under formation of pairs. In other words, $\mathbf{W}$ satisfies $(w, w') \in \mathbf{W}$ for all $w, w' \in \mathbf{W}$. For instance, given a unique "atomic" world name $w_0 \in \mathbf{W}$, the set $\mathbf{W}$ would contain infinitely many pairs $(w_0, w_0), ((w_0, w_0), w_0), (w_0, (w_0, w_0)), \ldots$ and so on.

Definition 4 (Event Model). *An* event model *over* $\mathbf{P}$ *and* $\mathbf{W}$ *is a quadruple* $\mathcal{E} = \langle E, \mathcal{K}, pre, post \rangle$ *where*

- $E \subseteq \mathbf{W}$ *is a finite set of worlds called* events,

- $pre : E \longrightarrow \mathcal{L}_E(\mathbf{P})$ *assigns to each event a precondition, and*

- $post : E \longrightarrow (\mathbf{P} \longrightarrow \mathcal{L}_E(\mathbf{P}))$ *assigns to each event a postcondition, for some propositions in* $\mathbf{P}$.

- $\mathcal{K} \subseteq E \times E$

By $D(\mathcal{E}) = E$ *we denote the domain of* $\mathcal{E}$. *A pair* $\langle \mathcal{E}, e \rangle$ *with* $e \in E$ *is called a* pointed event model. □

Definition 5 (Event updating evaluation). *Let* $\mathcal{M} = \langle W, \mathcal{K}, V \rangle$ *be an epistemic model and* $\mathcal{E} = \langle E, \hat{\mathcal{K}}, pre, post \rangle$ *an event model, both over* $\mathbf{P}$ *and* $\mathbf{W}$. *The* product update $\mathcal{M} \otimes \mathcal{E} \overset{\mathrm{def}}{=} \langle W', \mathcal{K}', V' \rangle$ *is another epistemic model where*

- $W' = \{ (w, e) \in W \times E \mid \mathcal{M}, w \models pre(e) \} \subseteq \mathbf{W}$ *is a set of worlds,*

- $\mathcal{K}' = \{ ((w_1, e_1), (w_2, e_2)) \in W' \times W' \mid (w_1, w_2) \in \mathcal{K} \text{ and } (e_1, e_2) \in \hat{\mathcal{K}} \}$,

- $V'((w, e)) = \{ p \in \mathbf{P} \mid \mathcal{M}, w \models post(e)(p) \}$ *for every* $(w, e) \in W'$,

Given a pointed event model $\langle \mathcal{E}, e \rangle$, *the* event updating evaluation *is a pair* $\langle \otimes, \mathcal{R} \rangle$ *with*

- $\mathcal{M} \otimes \langle \mathcal{E}, e \rangle \overset{\mathrm{def}}{=} \mathcal{M} \otimes \mathcal{E}$

- $\mathcal{R}(\mathcal{M}, \langle \mathcal{E}, e \rangle) \overset{\mathrm{def}}{=} \{ (w, w') \in W \times W' \mid w' = (w, e) \}$. □

For simplicity, we will usually write $[\mathcal{E}, e]\varphi$ instead of $[\langle \mathcal{E}, e \rangle]\varphi$. We will also use the following shorthands

$$[\mathcal{E}]\varphi \overset{\mathrm{def}}{=} \bigwedge_{e \in D(\mathcal{E})} [\mathcal{E}, e]\varphi \qquad\qquad \langle \mathcal{E} \rangle \varphi \overset{\mathrm{def}}{=} {\sim}[\mathcal{E}]{\sim}\varphi$$

The following result shows that, indeed, the semantics described above coincides with the semantics from [21] for the case of event models.

Proposition 1. *Let $\mathcal{M}$ be an epistemic model, $w \in D(\mathcal{M})$ be a world in $\mathcal{M}$, $\langle \mathcal{E}, e \rangle$ be pointed event model and $\varphi \in \mathcal{L}_E(\mathbf{P})$ be a formula. Then,*

– *$\mathcal{M}, w \models [\mathcal{E}, e]\varphi$ iff $\mathcal{M}, w \models pre(e)$ implies $\mathcal{M} \otimes \mathcal{E}, (w, e) \models \varphi$,*

Proof. By definition, we have $\mathcal{M}, w \models [\mathcal{E}, e]$
iff $\mathcal{M} \otimes \langle \mathcal{E}, e \rangle, w' \models \varphi$ for all w' with $(w, w') \in \mathcal{R}_{\mathcal{M}, \langle \mathcal{E}, e \rangle}$
iff $\mathcal{M} \otimes \mathcal{E}, w' \models \varphi$ for all w' with $(w, w') \in \mathcal{R}_{\mathcal{M}, \langle \mathcal{E}, e \rangle}$.
Note now that, by definition, we have either $\mathcal{R}_{\mathcal{M}, \langle \mathcal{E}, e \rangle} \cap (\{w\} \times D(\mathcal{M} \otimes \mathcal{E})) = \{(w, (w, e))\}$
or $\mathcal{R}_{\mathcal{M}, \langle \mathcal{E}, e \rangle} \cap (\{w\} \times D(\mathcal{M} \otimes \mathcal{E})) = \emptyset$. Hence, the above holds
iff $\mathcal{M} \otimes \mathcal{E}, (w, e) \models \varphi$ or $\mathcal{R}_{\mathcal{M}, \langle \mathcal{E}, e \rangle} \cap (\{w\} \times D(\mathcal{M} \otimes \mathcal{E})) = \emptyset$,
iff $\mathcal{M} \otimes \mathcal{E}, (w, e) \models \varphi$ or $\mathcal{M}, w \not\models pre(e)$
iff $\mathcal{M}, w \models pre(e)$ implies $\mathcal{M} \otimes \mathcal{E}, (w, e) \models \varphi$. $\qquad\qquad\qquad\qquad\square$

The following result helps understanding the semantics of (non-pointed) event models.

Proposition 2. *Let $\mathcal{M}$ be an epistemic model, $w \in D(\mathcal{M})$ be a world in $\mathcal{M}$, $\mathcal{E}$ be an event model and $\varphi \in \mathcal{L}_E(\mathbf{P})$ be a formula. Then,*

– *$\mathcal{M}, w \models [\mathcal{E}]\varphi$ iff $\mathcal{M} \otimes \mathcal{E}, (w, e) \models \varphi$ for every $e \in D(\mathcal{E})$ such that $\mathcal{M}, w \models pre(e)$,*

– *$\mathcal{M}, w \models \langle \mathcal{E} \rangle \varphi$ iff $\mathcal{M}, w \models pre(e)$ and $\mathcal{M} \otimes \mathcal{E}, (w, e) \models \varphi$ for some $e \in D(\mathcal{E})$.*

Proof. The second statement follows directly from its definitions. For the third, we have $\mathcal{M}, w \models \langle \mathcal{E} \rangle \varphi$ iff $\mathcal{M}, w \models \sim[\mathcal{E}] \sim \varphi$ iff $\mathcal{M}, w \not\models [\mathcal{E}] \sim \varphi$
iff $\mathcal{M}, w \models pre(e)$ does not imply $\mathcal{M} \otimes \mathcal{E}, (w, e) \models \sim \varphi$ for some $e \in D(\mathcal{E})$
iff $\mathcal{M}, w \models pre(e)$ and $\mathcal{M} \otimes \mathcal{E}, (w, e) \not\models \sim \varphi$ for some $e \in D(\mathcal{E})$
iff $\mathcal{M}, w \models pre(e)$ and $\mathcal{M} \otimes \mathcal{E}, (w, e) \models \varphi$ for some $e \in D(\mathcal{E})$. $\qquad\qquad\square$

Going back to our running example, Figure 2 depicts the event models corresponding to the actions of Example 1. For instance, Figure 2a depicts the event model of the action `take_left` with a single event e_1 whose precondition is $v \wedge \sim d$ and whose postcondition $\{d \mapsto \sim r\}$ states that, in the next state, d is assigned the truth value that formula $\sim r$ had in the previous state. More interestingly, Figure 2c depicts the event model of the action `flick` which has two events e_1 and e_2 with the same postcondition but different preconditions. The precondition of e_1 makes it applicable when the thief is in the vault and the diamond is on the right pedestal while the precondition of e_2 is analogous but for the left pedestal.[1] In this sense, when the action `flick` is executed in the epistemic model $\mathcal{M}_1$ (Figure 1b), it follows that only w_1 satisfies the preconditions of e_1 and only w_2 satisfies the preconditions of e_2. As a result, we can see that $\mathcal{M}_1 \otimes \texttt{flick}$ has two worlds, that is, $D(\mathcal{M}_1 \otimes \texttt{flick}) = \{(w_1, e_1), (w_2, e_2)\}$. Furthermore, since events e_1 and e_2 are disconnected, we also get that worlds (w_1, e_1) and (w_2, e_2) are disconnected. In fact, the epistemic model $\mathcal{M}_1 \otimes \texttt{flick}$ is isomorphic to the model $\mathcal{M}_2$ depicted in Figure 1c and can be obtained just by renaming each world w_i in $\mathcal{M}_2$ as (w_i, e_i).

---

[1]Note how we must specify the diamond's location in both preconditions, although the only real physical requirement for `flick` is being inside the vault. This need for specifying unrelated preconditions may obviously become a representational problem.
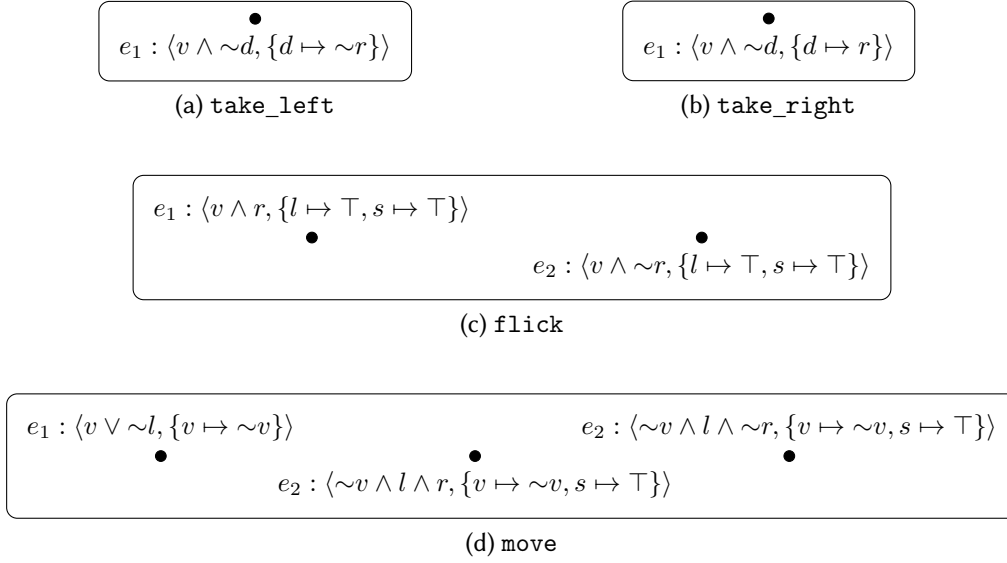
(a) `take_left`

(b) `take_right`

(c) `flick`

(d) `move`

Figure 2: Event models corresponding to the actions of Example 1. The first element of the pair assigned to each world corresponds to its preconditions while the second one corresponds to its postconditions. For instance, in (a) the precondition is $v \wedge \sim d$ and the postcondition $\{d \mapsto \sim r\}$. This postcondition means that, in the next state, d takes the value that formula $\sim r$ had in the previous state.
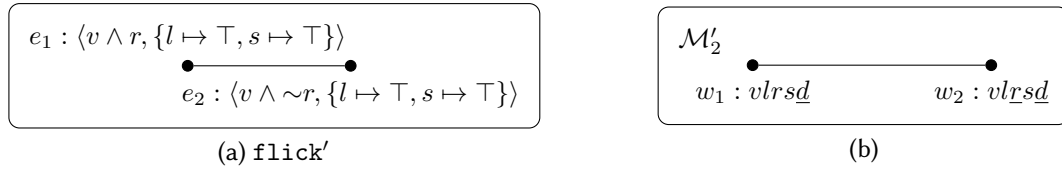

(a) `flick'`

(b)

Figure 3: (a) Event model corresponding to a variation of the action `flick` of Example 1 without observing the position of the diamond. (b) Epistemic model obtained after executing the action `flick'` in the model $\mathcal{M}_1$.

Note that the existence of two disconnected events in the action `flick` encodes the observation that happens when the light is turned on, that is, the agent obtains the knowledge about the actual place of the diamond. For instance, if we consider the action `flick'` depicted in Figure 3a, and obtained from the action `flick` by connecting events e_1 and e_2, we can see that $\mathcal{M}_1 \otimes \texttt{flick}'$ is isomorphic to the epistemic model $\mathcal{M}_2'$ depicted in Figure 3b. Model $\mathcal{M}_2'$ only differs from $\mathcal{M}_2$ in that worlds w_1 and w_2 are now connected, revealing that the agent cannot tell where is the diamond. In other words, `flick'` encodes the same ontic changes in the world than `flick` but does not capture the agent's observation about the position of the diamond.

Finally, Figure 4a (model $\mathcal{M}_3$) corresponds to an state where the thief is inside the vault with the diamond in her possession. Intuitively, this model represent the result of executing action `take_left` or `take_right` according to the agent's knowledge about the position of the diamond, whereas Figure 4b represents model $\mathcal{M}_4 = \mathcal{M}_3 \otimes \texttt{move}$, that is, the result of moving (outside the vault) afterwards.

$$\mathcal{M}_3 \quad \bullet \quad\quad\quad \bullet$$
$$w_1 : vlrsd \quad\quad\quad w_2 : vl\underline{r}sd$$

$$\mathcal{M}_4 \quad \bullet \quad\quad\quad \bullet$$
$$w_1 : \underline{v}lrsd \quad\quad\quad w_2 : \underline{v}l\underline{r}sd$$

(a) (b)

Figure 4: Epistemic models representing (a) the state corresponding to execution of action `take_left` or `take_right` according to the agent's knowledge about the position of the diamond and (b) the result of executing `move` on $\mathcal{M}_3$ in (a).

Planning in Answer Set Programming

In this subsection, we informally describe the ASP methodology for representing problems of classical planning: for a more formal approach we refer to [45, 57].

Our purpose is, by now, merely introductory, trying to illustrate the main representational features of ASP planning that are relevant for the current discussion. For this reason, we delay the introduction of a formal semantics for later on, when epistemic ASP is introduced.

ASP specifications or *logic programs* are sets of rules of the form:

$$a \leftarrow b_1, \ldots, b_n, not\ c_1, \ldots, not\ c_m$$

where $\leftarrow$ is a reversed implication, so its left hand side is called the rule *head* and its right hand side receives the name of rule *body*. The rule head a is a proposition or the symbol $\bot$: when the latter happens, the rule is a *constraint* forbidding that the body holds. The elements in the body (b_i and $not\ c_j$) are called *literals*, where b_i and c_j are propositions. The ordering among literals is irrelevant: in fact, commas just stand for conjunctions. Operator not represents *default negation*: we read $not\ c_j$ as "there is no evidence on c_j" or "there is no way to prove c_j". We will also use non-deterministic rules of the form:

$$m\ \{a_1; \ldots; a_k\}\ n \leftarrow Body$$

where $m, n \geq 0$ meaning that, when $Body$ holds, we can arbitrarily add a subset of atoms from $\{a_1, \ldots, a_k\}$ of cardinality c with $n \leq c \leq m$. ASP allows a second kind of negation called *strong* or *explicit* that we will represent $\sim p$. From a practical point of view, we can assume that "$\sim p$" is a new kind of atom and that models cannot make p and $\sim p$ true simultaneously.

For a simple representation of rules describing transitions we partly adopt the syntax of [57] and assume that, for each proposition p, we handle a second atom "$\bullet p$" that stands for p at the immediately *previous* situation. In temporal ASP, actions are represented as regular propositions in **P**: the rest of non-action propositions in **P** are called *fluents*.

Taking all these considerations, the behaviour of action `take_left` can be encoded in ASP as the following three rules:

$$d \leftarrow \texttt{take\_left} \tag{1}$$

$$\bot \leftarrow \texttt{take\_left}, \sim\bullet v \tag{2}$$

$$\bot \leftarrow \texttt{take\_left}, \bullet r \tag{3}$$

where (1) describes its direct effect (grasping the diamond) whereas the other two rules describe the preconditions: (2) forbids executing `take_left` when the thief was not in the vault and (3) forbids its execution when the diamond is in the right pedestal. Analogously, the following three rules encode the action `take_right`:

$$d \leftarrow \texttt{take\_right} \tag{4}$$

$$\bot \leftarrow \texttt{take\_right}, \sim \bullet v \tag{5}$$

$$\bot \leftarrow \texttt{take\_right}, \sim \bullet r \tag{6}$$

Similarly, actions `flick` and `move` are respectively represented by the rules:

$$l \leftarrow \texttt{flick} \tag{7}$$

$$\bot \leftarrow \texttt{flick}, \sim \bullet v \tag{8}$$

$$v \leftarrow \texttt{move}, \sim \bullet v \tag{9}$$

$$\sim v \leftarrow \texttt{move}, \bullet v \tag{10}$$

Rule (7) states the postcondition of `flick`, that is, the light is turned on, while rule (8) states its precondition, that is, we forbid its execution when being outside vault. Rules (9) and (10) together state the postconditions of `move`: its execution just flips the truth value of v.

To illustrate the use of indirect effects, we can just assert that seeing the diamond (s) just depends on being in the vault (v) with the light on (l), regardless of the actions that have been performed to reach that situation. This is naturally captured by the single ASP rule:

$$s \leftarrow v, l \tag{11}$$

Default negation allows a natural representation of the inertia law, that is, a fluent normally remains unchanged, unless there is evidence on the contrary. We divide the set of fluents into *inertial* $\mathcal{F}^{\mathcal{I}} = \{v, l, r, d\}$ and *non-inertial* fluents $\mathcal{F}^{\mathcal{N}} = \{s\}$. Inertia is then guaranteed by the pair of rules:

$$f \leftarrow \bullet f, not \sim f \tag{12}$$

$$\sim f \leftarrow \sim \bullet f, not\ f \tag{13}$$

for each inertial fluent $f \in \mathcal{F}^{\mathcal{I}}$. In our running example, the fluent (s) is considered false by default, that is, the following rule:

$$\sim s \leftarrow not\ s \tag{14}$$

sating that, unless (s) is proved, we should consider that its explicit negation ($\sim s$) is added to the set of conclusions.

If we consider now the following simplification of Example 1 where the value of all fluents in the initial situation are known, we can use ASP to obtain a plan to achieve the thief's goal.

Example 2 (Example 1 continued). *Consider now the case where the thief is outside the vault ($\sim v$) and already* knows the Pink Panther is inside the vault on the right (r) pedestal. □

```
1   #program dynamic.
    d :- take_left.
    :- take_left, not 'v.
    :- take_left, not -'r.
    d :- take_rigth.
6   :- take_rigth, not 'v.
    :- take_rigth, not 'r.
    l :- flick.
    :- flick, not 'v.
    v :- move, -'v.
11  -v :- move, 'v.
    s :- v,l.
     v :- 'v, not -v.
    -v :- -'v, not v.
     d :- 'd, not -d.
16  -d :- -'d, not d.
     l :- 'l, not -l.
    -l :- -'l, not l.
     r :- 'r, not -r.
    -r :- -'r, not r.
21  {take_left; take_rigth; flick; move}1.

    #program always.
    -s :- not s.

26  #program initial.
    -v. r. -d. -l.

    #program final.
    :- not -v.
31  :- not d.

    #show take_left/0.
    #show take_rigth/0.
    #show flick/0.
36  #show move/0.
```

Listing 1: Program corresponding to Example 2 in the syntax of the solver telingo.

Listing 1 shows the full encoding representing Example 2 in the syntax of the ASP solver telingo.[2] In this syntax, ← is represented as :-, •p as 'p and ~p as -p. By copying that encoding into a file called pink.lp and executing "telingo pink.lp" we can obtain a plan for this example.

As we said before, an important difference between ASP and event models is the treatment of indirect effects. In the example, note how s was captured by the ASP rule (11), which only depends on other fluents (v and l) but does not refer to the original actions that caused their

values. There is no flexible way to express this feature when using event models: the value of s must be expressed as a direct effect of actions `flick` and `move`, that respectively determine the values of l and v. If new actions could alter the values of those fluents, directly or indirectly, then their effect on s should also be included in their post-conditions. This is, in fact, an instance of the well-known *ramification problem* [58].

The ramification problem may also occur for epistemic effects, if we are not careful enough for their treatment. For instance, the encoding of Example 1 in [21] did not use our fluent s (where the ramification is evident), but transferred the problem to the epistemic effect of *knowing* the position of the diamond (r). Again, this epistemic effect must be specified as a direct consequence of some action, something that does not always seem reasonable.

In the rest of the paper, we develop an extension of DEL and ASP, that we denote DEL[ASP] where the ontic and epistemic effects of actions can be described both in a direct or indirect way, as needed. In particular, in DEL[ASP], the observation of the diamond position when the thief is in the illuminated vault can be expressed by the following rule analogous to (11):

$$\mathbf{O}\, r \leftarrow v, l \tag{15}$$

Here, $\mathbf{O}\, r$ is a new construct whose intuitive meaning is that "the actual value of the fluent r is observed (by the agent)." Note that we just replace the fluent s in (11), whose intuitive meaning is that the agent sees the position of the diamond, by this new construct $\mathbf{O}\, r$, which makes this observation affect the agent's beliefs.

Epistemic Logic Programs

As explained before, we will use FAEEL for the interpretation of epistemic specifications, the epistemic extension of ASP. FAEEL inherits both the ASP capabilities for knowledge representation and the AEL capabilities for introspective reasoning. For the sake of coherence, we adapt the definitions of [51] to the use of Kripke structures. We also add strong negation [52, 59] to FAEEL, which for simplicity, is restricted to be used only in front of atoms, something that suffices for the goals of this paper and is usual in the ASP literature [43, 60].

Autoepistemic formulas are defined according to the following grammar:

$$\varphi ::= \bot \mid p \mid {\sim}p \mid \varphi_1 \wedge \varphi_2 \mid \varphi_1 \vee \varphi_2 \mid \varphi_1 \rightarrow \varphi_2 \mid \mathbf{L}\, \varphi$$

where $p \in \mathbf{P}$ is a proposition.

We assume that ASP notation is transformed into its logical counterpart: $not\ F$ is represented as $\neg F$, commas in the body are replaced by conjunctions and rules $G \leftarrow F$ are written from left to right $F \rightarrow G$.

Intuitively, the new construct, $\mathbf{L}\, \varphi$, is read as "it is reasonable (for the planning agent) to believe φ." Weak or intuitionistic negation is defined as usual: $\neg\varphi \overset{\text{def}}{=} \varphi \rightarrow \bot$. The knowledge modality is defined as true belief: $\mathbf{K}\, \varphi \overset{\text{def}}{=} \varphi \wedge \mathbf{L}\, \varphi$. We also introduce the following abbreviations:

$$
\begin{aligned}
\mathbf{U}\, p &\overset{\text{def}}{=} \neg p \wedge \neg{\sim}p \\
\mathbf{O}\, p &\overset{\text{def}}{=} (p \rightarrow \mathbf{L}\,\neg\neg p) \wedge ({\sim}p \rightarrow \mathbf{L}\,\neg\neg{\sim}p) \wedge (\mathbf{U}\, p \rightarrow \mathbf{L}\,\neg\neg\mathbf{U}\, p)
\end{aligned}
$$

whose respective intuitive meanings are that the value of proposition $p \in \mathbf{P}$ is *undefined* and that the actual value of proposition $p \in \mathbf{P}$ is *observed*. Note that when an atom p is observed, the agent's beliefs have to agree with the actual value of the atom p. The use of double negation here implies that only the agent's beliefs will be modified, without providing any justification for believing p. Besides, we assume all previous abbreviations too, that is, $(\varphi \leftarrow \psi) \stackrel{\text{def}}{=} (\psi \rightarrow \varphi)$, $\varphi \leftrightarrow \psi \stackrel{\text{def}}{=} (\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi)$, and $\top \stackrel{\text{def}}{=} \bot \rightarrow \bot$. An *autoepistemic theory* Γ is a set of autoepistemic formulas as defined above. When a theory is a singleton, we will usually write just φ instead of $\{\varphi\}$.

A *literal* L is either a proposition $p \in \mathbf{P}$ or a proposition preceded by strong negation $\sim p$ and by $Lit \stackrel{\text{def}}{=} \mathbf{P} \cup \{\sim p \mid p \in \mathbf{P}\}$ we denote the set of all literals over the signature $\mathbf{P}$.

We define next an (autoepistemic) HT-model, as a combination of modal epistemic logic with the logic of here-and-there (HT) [61], an intermediate logic with two intuitionistic worlds, h (standing for "here") and t (read as "there") satisfying $h \leq t$.

Definition 6 (HT-Model). *Given a set of propositional symbols $\mathbf{P}$, an* HT-model *is a quadruple $\mathcal{M} = \langle W, \mathcal{K}, V^h, V^t \rangle$ where*

- *W is a set of worlds,*

- *$\mathcal{K} \subseteq W \times W$ is an accessibility relation on W, and*

- *$V^x : W \longrightarrow 2^{Lit}$ is a valuation with $x \in \{h, t\}$ such that $V^h(w) \subseteq V^t(w)$ for all $w \in W$.*

$D(\mathcal{M}) = W$ denotes the domain of $\mathcal{M}$. A belief HT-model is an HT-model where $\mathcal{K} = \mathcal{K} \times \mathcal{K}'$ with $\mathcal{K}' = \mathcal{K} \setminus \{w_0\}$ for some distinguish world $w_0 \in \mathcal{K}$. □

A HT-model $\mathcal{M} = \langle W, \mathcal{K}, V^h, V^t \rangle$ is called *total* iff $V^h = V^t$. Furthermore, by $\mathcal{M}^t \stackrel{\text{def}}{=} \langle W, \mathcal{K}, V^t, V^t \rangle$ we denote the total model corresponding to $\mathcal{M}$. Satisfaction of autoepistemic formulas is then given by the following recursive definition:

- $\mathcal{M}, w \not\models \bot$,

- $\mathcal{M}, w \models L$ iff $L \in V^h(w)$ for any $L \in Lit$,

- $\mathcal{M}, w \models \varphi_1 \wedge \varphi_2$ iff $\mathcal{M}, w \models \varphi_1$ and $\mathcal{M}, w \models \varphi_2$,

- $\mathcal{M}, w \models \varphi_1 \vee \varphi_2$ iff $\mathcal{M}, w \models \varphi_1$ or $\mathcal{M}, w \models \varphi_2$,

- $\mathcal{M}, w \models \varphi_1 \rightarrow \varphi_2$ iff both $\mathcal{M}, w \not\models \varphi_1$ or $\mathcal{M}, w \models \varphi_2$ and
  $\mathcal{M}^t, w \not\models \varphi_1$ or $\mathcal{M}^t, w \models \varphi_2$,

- $\mathcal{M}, w \models \mathbf{L}\, \varphi$ iff $\mathcal{M}, w' \models \varphi$ for all w' with $(w, w') \in \mathcal{K}$

As usual, we say that $\mathcal{M}$ is an *HT-model* of some theory Γ, in symbols $\mathcal{M} \models \Gamma$, iff $\mathcal{M}, w \models \varphi$ for every world $w \in D(\mathcal{M})$ and every formula $\varphi \in \Gamma$. As mentioned before, when $\Gamma = \{\varphi\}$ is a singleton we will omit the brackets, so that $\mathcal{M} \models \varphi$ stands for $\mathcal{M} \models \{\varphi\}$ and holds iff $\mathcal{M}, w \models \varphi$ for every world $w \in D(\mathcal{M})$.

Definition 7 (Bisimulation between HT-models). *Let $\mathcal{M}_1 = \langle W_1, \mathcal{K}_1, V_1^h, V_1^t \rangle$ and $\mathcal{M}_2 = \langle W_2, \mathcal{K}_2, V_2^h, V_2^t \rangle$ be two HT-models. Given some binary relation $Z \subseteq W_1 \times W_2$, we write $\mathcal{M}_1 \preceq^Z \mathcal{M}_2$ iff*

- *every $(w_1, w_2) \in Z$ satisfies $V^t(w_1) = V^t(w_2)$ and $V^h(w_1) \subseteq V^h(w_2)$,*

- *for every $(w_1, w_1') \in \mathcal{K}_1$, there is $(w_2, w_2') \in \mathcal{K}_2$ such that $(w_1', w_2') \in Z$,*

- *for every $(w_2, w_2') \in \mathcal{K}_2$, there is $(w_1, w_1') \in \mathcal{K}_1$ such that $(w_1', w_2') \in Z$.*

We write $\mathcal{M}_1 \preceq \mathcal{M}_2$ iff there is a total relation Z s.t. $\mathcal{M}_1 \preceq^Z \mathcal{M}_2$. We also say that $\mathcal{M}_1$ and $\mathcal{M}_2$ are bisimilar, *in symbols $\mathcal{M}_1 \approx \mathcal{M}_2$, iff there is a total relation Z s.t $\mathcal{M}_1 \preceq^Z \mathcal{M}_2$ and $\mathcal{M}_2 \preceq^Z \mathcal{M}_1$. As usual, we write $\mathcal{M}_1 \prec \mathcal{M}_2$ iff $\mathcal{M}_1 \preceq \mathcal{M}_2$ and $\mathcal{M}_1 \not\approx \mathcal{M}_2$.* □

Definition 8 (Equilibrium model). *A total belief HT-model $\mathcal{M}$ of some theory Γ is said to be an* equilibrium model *of Γ iff there is no other belief HT-model $\mathcal{M}'$ of Γ such that $\mathcal{M}' \prec \mathcal{M}$.* □

Given some information cell $\mathcal{M} = \langle W, \mathcal{K}, V \rangle$ and some set of literals $I \subseteq Lit$, by $\mathcal{M} + I$ we denote the total belief HT-model $\mathcal{M}' = \langle W', \mathcal{K}', V', V' \rangle$ where $W' = \{w_0\} \cup W$ with $w_0 \notin W$, $\mathcal{K}' = W' \times W$ and $V' : W' \longrightarrow 2^{Lit}$ satisfies $V'(w) = V(w)$ for all $w \in W$ and $V'(w_0) = I$.

Definition 9 (World view). *Given a set of propositions $\mathbf{P}$, an information cell $\mathcal{M} = \langle W, \mathcal{K}, V \rangle$ over Lit is called a* world view *of some theory Γ iff the following two conditions hold:*

- *$\mathcal{M} + V(w)$ is an equilibrium model of Γ, for every world $w \in W$,*

- *$\mathcal{M} + I$ is not an equilibrium model of Γ for every set of literals $I \subseteq Lit$ satisfying $I \neq V(w)$ for all $w \in W$, and*

- *either $p \notin V(w)$ or $\sim p \notin V(w)$ for all $p \in \mathbf{P}$ and $w \in W$.*

We say that a theory Γ is consistent *iff it has some world view and by $\mathrm{WV}[\Gamma]$ we denote the set of all world views of Γ.* □

Example 3 (Example 1 continued). *For instance, the formula*

$$\varphi_0 = \sim v \wedge \sim l \wedge (r \vee \sim r) \wedge \sim s \wedge \sim d \tag{16}$$

has a unique world view that is depicted in Figure 5. Note that every propositional theory has a unique world view [51] that corresponds to the set of all answer sets of the theory. Furthermore, since φ_0 contains no negation, its answer sets coincide with its minimal classical models when we treat each strong negated literal $\sim p$ as a new atom. □

Dynamic Epistemic Logic with ASP Updates: DEL[ASP]

In this section, we present the major contribution of this paper, DEL[ASP], an instance of the abstract DEL framework where updating objects correspond to logic programs. Our motivation is twofold: on the one hand, to allow unrestricted use of indirect effects (both ontic and epistemic); on the other hand, to preserve the ASP representation of non-epistemic planning problems without need of any adjustment or modification. We illustrate these two objectives through our running example.
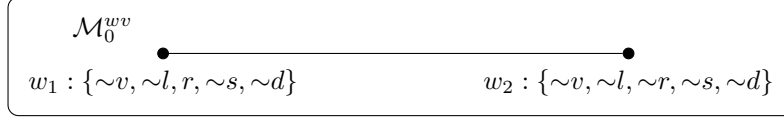
$$\mathcal{M}_0^{wv}$$

$w_1 : \{\sim v, \sim l, r, \sim s, \sim d\}$ \qquad\qquad $w_2 : \{\sim v, \sim l, \sim r, \sim s, \sim d\}$

Figure 5: Unique world view of the formula $\varphi_0 = \sim v \wedge \sim l \wedge (r \vee \sim r) \wedge \sim s \wedge \sim d$.

Characterising information cells in FAEEL

Let us start by showing how any information cell can be represented by some autoepistemic formula in FAEEL. Note that world views are an information cell over Lit, so they represent a kind of three valued epistemic models where each proposition p can be true $p \in V(w)$, false $\sim p \in V(w)$ or undefined $p, \sim p \notin V(w)$. We will show here how (two-valued) information cells over **P** can be simply represented as propositional formulas in FAEEL, allowing to map these three valued epistemic models into standard two valued ones.

Example 4 (Example 3 continued). *Continuing with our running example, we can see now that this model satisfies either $p \in V(w)$ or $\sim p \in V(w)$ for every proposition $p \in \{v, l, r, s, d\}$ and world $w \in \{w_1, w_2\}$. Hence, we can map this model into a (two-valued) information cell by considering as true every proposition $p \in V(w)$ and as false every proposition satisfying $\sim p \in V(w)$. It is easy to see that the obtained information cell is precisely the model $\mathcal{M}_0$ depicted in Figure 1a, that is, the epistemic model corresponding to the initial situation of Example 1. In this sense, we can use the formula φ_0 to represent the initial state of this example.* □

Definition 10. *Given some information cell $\mathcal{M} = \langle W, \mathcal{K}, V \rangle$, its* characteristic *(autoepistemic) formula is $\varphi_{\mathcal{M}} \overset{\text{def}}{=} \bigvee_{w \in W} \varphi_{\mathcal{M}}^w$ where $\varphi_{\mathcal{M}}^w$ is defined as follows:*

$$\varphi_{\mathcal{M}}^w \quad \overset{\text{def}}{=} \quad \Big(\bigwedge_{p \in V(w)} p \Big) \wedge \Big(\bigwedge_{p \in \mathbf{P} \setminus V(w)} \sim p \Big) \qquad\qquad □$$

Definition 11 (Bisimulation). *Given two models $\mathcal{M}_1 = \langle W_1, \mathcal{K}_1, V_1 \rangle$ and $\mathcal{M}_2 = \langle W_2, \mathcal{K}_2, V_2 \rangle$, we say that they are* bisimilar, *in symbols $\mathcal{M}_1 \approx \mathcal{M}_2$, if and only if $\langle W_1, \mathcal{K}_1, V_1, V_1 \rangle \approx \langle W_2, \mathcal{K}_2, V_2, V_2 \rangle$.* □

Definition 12. *Given a set of propositions $P \subseteq \mathbf{P}$, we say that an HT-model $\mathcal{M} = \langle W, \mathcal{K}, V^h, V^t \rangle$ over Lit is P-*classical *iff every world $w \in W$ and proposition $p \in P$ satisfy that either $p \in V^h(w)$ or $\sim p \in V^h(w)$ holds. A theory Γ is P-*classical *iff it is consistent and, in addition, every world view is P-classical.* □

Definition 13. *Given a set of propositions $P \subseteq \mathbf{P}$ and any P-classical total HT-model $\mathcal{M} = \langle W, \mathcal{K}, V^t, V^t \rangle$ over Lit, by $\mathcal{M} \downarrow P = \langle W, \mathcal{K}, V \rangle$ we denote the model over P where $V : W \longrightarrow 2^P$ is a valuation satisfying $V(w) = V^t(w) \cap P$ for every world $w \in W$.* □

Proposition 3. *Let $\mathcal{M}$ be an information cell over $\mathbf{P}$. Then, $\varphi_{\mathcal{M}}$ has a unique world view $\mathcal{M}'$ and we have that $\mathcal{M}$ and $\mathcal{M}' \downarrow \mathbf{P}$ are bisimilar.*
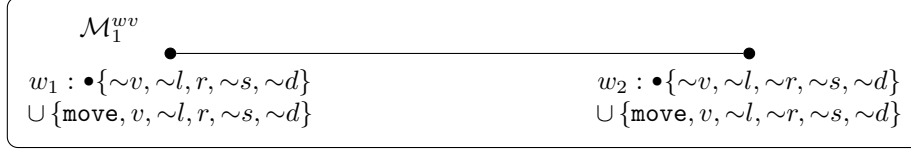
Figure 6: Unique world view of the program Γ_1.

Proof. First note that, since $\varphi_{\mathcal{M}}$ is a propositional formula, it has a unique world view $\mathcal{M}'$ [51, Proposition 3]. Let $\mathcal{M} = \langle W, \mathcal{K}, V \rangle$ and $\mathcal{M}' = \langle W', \mathcal{K}', V' \rangle$. Then, we have that $w' \in W'$ iff $V'(w')$ is a stable model of $\varphi_{\mathcal{M}}$. Note also that the stable models of $\varphi_{\mathcal{M}}$ are exactly its classical models understood as sets of literals. Hence, for every $w' \in W'$, there is some $w \in W$ such that $V'(w') = V(w) \cup \sim(\mathbf{P} \setminus V(w))$ and vice-versa. Consequently, $\mathcal{M}$ and $\mathcal{M}' \downarrow \mathbf{P}$ are bisimilar. $\quad\square$

Example 5 (Example 4 continued). *Continuing with our running example, we have $\varphi_{\mathcal{M}_0} = \varphi_{\mathcal{M}_0}^{w_1} \vee \varphi_{\mathcal{M}_0}^{w_2}$ with*

$$\varphi_{\mathcal{M}_0}^{w_1} = \sim v \wedge \sim l \wedge \ r \wedge \sim s \wedge \sim d$$
$$\varphi_{\mathcal{M}_0}^{w_2} = \sim v \wedge \sim l \wedge \sim r \wedge \sim s \wedge \sim d$$

By applying distributivity of conjunctions over disjunctions, it is easy to see that $\varphi_{\mathcal{M}_0}$ is classically (and intuitionistically) equivalent to (16). *As a result, $\mathcal{M}_0^{wv}$ is the unique world view of $\varphi_{\mathcal{M}_0}$ and, as expected from Proposition 3, it can be checked that it satisfies $\mathcal{M}_0^{wv} \downarrow \mathbf{P} = \mathcal{M}_0$.* $\quad\square$

Epistemic Model Updates with FAEEL

In this section, we show how autoepistemic equilibrium logic can be used to define epistemic model updates just by using an extended signature. Given a set of propositions $S \subseteq \mathbf{P}$, we define $\bullet S \overset{\text{def}}{=} \{\ \bullet p \mid p \in S \cap \mathbf{P}\ \}$ and $\mathbf{P}^{bi} = \mathbf{P} \cup \bullet\mathbf{P}$ where $\bullet p$ intuitively means that p is true in the previous state. It will also be convenient to use $\bullet$ in front of any propositional formula φ such that $\bullet\varphi$ is as an abbreviation for the formula obtained by writing $\bullet$ in front of every proposition occurring in φ.

Example 6 (Example 3 continued). *Let Γ_{pink} be a theory containing formulas* (1)-(15) *and let $\Gamma_1 = \Gamma_{pink} \cup \{\text{move}, \bullet\varphi_{\mathcal{M}_0}\}$. This program has a unique world view shown in Figure 6. Note that, if we disregard all the information corresponding to the previous situation (that is all literals preceded by $\bullet$) and the action* move, *then we have the same information as the epistemic model $\mathcal{M}_1$ in Figure 1a. In other words, Γ_1 encodes the transition that occurs between the epistemic models $\mathcal{M}_0$ and $\mathcal{M}_1$ when executing action* move. $\quad\square$

As shown in the example above, we can represent the transition between two epistemic models as an autoepistemic theory. Let us now formalise this intuition. We begin introducing some auxiliary definitions.

Given a set of epistemic models $\mathcal{S} = \{\mathcal{M}_1, \mathcal{M}_2, \dots\}$ where each $\mathcal{M}_i$ is a model over a set of atoms $\mathbf{P}$ of the form $\mathcal{M}_i = \langle W_i, \mathcal{K}_i, V_i \rangle$ and satisfying $W_i \cap W_j = \emptyset$ for all $\mathcal{M}_i, \mathcal{M}_j \in \mathcal{S}$ with $i \neq j$, by $\bigsqcup \mathcal{S} \overset{\text{def}}{=} \langle W, \mathcal{K}, V \rangle$, we denote an epistemic model where

- $W' = \bigcup\{ W_i \mid \mathcal{M}_i \in \mathcal{S} \}$,

- $\mathcal{K}' = \bigcup\{ \mathcal{K}_i \mid \mathcal{M}_i \in \mathcal{S} \}$, and

- $V' : W' \longrightarrow 2^{\mathbf{P}}$ with $V'(w) = V_i(w)$ for all $w \in W_i$ and all $\mathcal{M}_i \in \mathcal{S}$.

As usual, if $\mathcal{S} = \{\mathcal{M}_1, \mathcal{M}_2\}$, we just write $\mathcal{M}_1 \sqcup \mathcal{M}_2$ instead of $\bigsqcup\{\mathcal{M}_1, \mathcal{M}_2\}$.

Definition 14. *Let P be a set of atoms and Γ be some P-classical autoepistemic theory. Then, by* $\mathrm{mwv}(\Gamma, P) \stackrel{\mathrm{def}}{=} \bigsqcup\{ \mathcal{M} \downarrow P \mid \mathcal{M} \in \mathrm{WV}[\Gamma] \}$ *we denote the epistemic model capturing all the world views of Γ projected into the vocabulary P. If Γ is not P-classical, we assume that $\mathrm{mwv}(\Gamma, P)$ is not defined.* $\square$

In other words, for every P-classical autoepistemic theory Γ, $\mathrm{mwv}(\Gamma, P)$ is the two-valued epistemic model that has an information cell for every world view of Γ such that the valuation of every proposition $p \in P$ in every world in $\mathrm{mwv}(\Gamma, P)$ corresponds to the valuation of that proposition in that same world in the corresponding world view. Recall that, in a P-classical theory, all its worlds views satisfy either p or $\sim p$ for every proposition $p \in P$. This is necessary so it is possible to map three-valued world views into two-valued epistemic models. We could remove this restriction by allowing three-valued epistemic models, but we have decided to stay as close as possible to the original DEL semantics, which is only two-valued.

Example 7 (Example 6 continued). *Note that if $P \subseteq \mathbf{P}$, then the epistemic model $\mathrm{mwv}(\Gamma, P)$ always corresponds to the current situation, discarding all information about the previous one. In this sense, if we consider the program Γ_1 of Example 6 and the epistemic model $\mathcal{M}_1$ of Figure 1b, we have that $\mathrm{mwv}(\Gamma_1, P) = \mathcal{M}_1$ where $P = \{v, d, r, l, s\}$ is the set of fluent of Example 1.*

As a further example, consider now the theory $\Gamma_2 = \Gamma_{pink} \cup \{\mathtt{flick}, \bullet\varphi_{\mathcal{M}_1}\}$. Then, Γ_2 has two world views which correspond to the two cell informations in the epistemic model $\mathcal{M}_2$ depicted in Figure 1c. This explains why we need to join together all the world views of the theory in a single epistemic model. Every world view, which becomes an information cell, represents the knowledge the agent will have after executing the action $\mathtt{flick}$, while the set of all world views represent the knowledge the agent had before executing it. $\square$

Let us now define the transition between states borrowing the notion of product update from DEL.

Definition 15. *Given an information cell $\mathcal{M}$ over $P \subseteq \mathbf{P}$ and a theory Γ over $\mathbf{P}^{bi}$ such that $\Gamma \cup \{\bullet\varphi_{\mathcal{M}}\}$ is $\mathbf{P}^{bi}$-classical, we define:*

- *the product update of $\mathcal{M}$ with respect to Γ as the epistemic model $\mathcal{M} \otimes \Gamma \stackrel{\mathrm{def}}{=} \mathrm{mwv}(\Gamma \cup \{\bullet\varphi_{\mathcal{M}}\}, P)$.*

- *the binary relation $\mathcal{R}_{\mathcal{M},\Gamma} \subseteq D(\mathcal{M}) \times D(\mathcal{M} \otimes \Gamma)$ s.t. $(w, w') \in \mathcal{R}_{\mathcal{M},\Gamma}$ iff $\mathcal{M}', w' \models \bullet\varphi_{\mathcal{M}}^w$ with $\mathcal{M}' = \mathrm{mwv}(\Gamma \cup \{\bullet\varphi_{\mathcal{M}}\}, \bullet\mathbf{P})$.* $\square$

Example 8 (Example 7 continued). *It is now easy to see that*

$$\mathcal{M}_0 \otimes \Gamma_3 = \mathrm{mwv}(\Gamma_3 \cup \{\bullet\varphi_{\mathcal{M}_0}\}, P) = \mathrm{mwv}(\Gamma_1, P) = \mathcal{M}_1^{wv} \downarrow P = \mathcal{M}_1$$
$$\mathcal{M}_1 \otimes \Gamma_4 = \mathrm{mwv}(\Gamma_4 \cup \{\bullet\varphi_{\mathcal{M}_0}\}, P) = \mathrm{mwv}(\Gamma_2, P) = \mathcal{M}_2$$

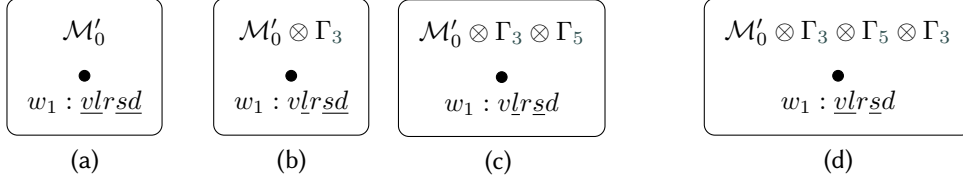| $\mathcal{M}'_0$	$\mathcal{M}'_0 \otimes \Gamma_3$	$\mathcal{M}'_0 \otimes \Gamma_3 \otimes \Gamma_5$	$\mathcal{M}'_0 \otimes \Gamma_3 \otimes \Gamma_5 \otimes \Gamma_3$
$\bullet$	$\bullet$	$\bullet$	$\bullet$
$w_1 : v\underline{lr}s\underline{d}$	$w_1 : v\underline{lr}s\underline{d}$	$w_1 : v\underline{lr}s\underline{d}$	$w_1 : v\underline{lr}s\underline{d}$
(a)	(b)	(c)	(d)

Figure 7: Epistemic models corresponding execution of the sequence of actions $\langle \texttt{move}, \texttt{take\_right}, \texttt{move} \rangle$ in initial state of Example 2, represented here by the model $\mathcal{M}'_0$ in (a).

with $\Gamma_3 = \Gamma_{pink} \cup \{\texttt{move}\}$ and $\Gamma_4 = \Gamma_{pink} \cup \{\texttt{flick}\}$ *respectively being the theories representing the execution of the action* $\texttt{move}$ *and* $\texttt{flick}$ *according to program* Γ_{pink}, $\mathcal{M}_1^{wv}$ *the epistemic model depicted in Figure 6 and* $\mathcal{M}_1$ *and* $\mathcal{M}_2$ *the epistemic models depicted in Figure 1b and Figure 1c. In other words,* $\mathcal{M}_1$ *is the result of executing action* $\texttt{move}$ *in epistemic model* $\mathcal{M}_0$ *according to the description provided by* Γ_{pink}. *Furthermore, for each world* $w \in \{w_1, w_2\}$, *we have* $\mathcal{M}_1^{wv}, w \models \bullet\varphi_{\mathcal{M}}^w$. *In its turn, this implies* $\mathcal{M}_1^{wv} \downarrow \bullet\mathbf{P}, w \models \bullet\varphi_{\mathcal{M}}^w$ *and, thus, that* $\mathcal{R}_{\mathcal{M}_0, \Gamma_3} = \{(w_1, w_1), (w_2, w_2)\}$ *is the identity[3], that is, it maps each world in* $\mathcal{M}_0$ *to a world in* $\mathcal{M}_1$ *with the same name. Similarly,* $\mathcal{M}_2$ *is the result of executing the action* $\texttt{flick}$ *in the epistemic model* $\mathcal{M}_1$ *according to the description provided by* Γ_{pink} *and we can check that* $\mathcal{R}_{\mathcal{M}_1, \Gamma_4}$ *is also the identity.* $\square$

We define now the updating evaluation for ASP epistemic specifications for epistemic models. In a nutshell, this evaluation is the result of combining the evaluation for each individual information cell in the model.

Definition 16 (ASP updating evaluation). *Given any epistemic model* $\mathcal{M}$ *and theory* Γ, *the* ASP updating evaluation *is a pair* $\langle \otimes, \mathcal{R} \rangle$ *satisfying*

$$\mathcal{M} \otimes \Gamma \quad \overset{\text{def}}{=} \quad \bigsqcup \{ \mathcal{M}' \otimes \Gamma \mid \mathcal{M}' \in \text{cell}(\mathcal{M}) \}$$
$$\mathcal{R}_{\mathcal{M}, \Gamma} \quad \overset{\text{def}}{=} \quad \bigcup \{ \mathcal{M}' \otimes \Gamma \mid \mathcal{M}' \in \text{cell}(\mathcal{M}) \} \qquad \square$$

We can now directly apply Definition 3 to obtain the satisfaction of DEL[ASP] formulas, that is, DEL formulas in which the updating objects are autoepistemic theories.

Example 9 (Example 2 continued). *Let us now resume the simplified version of our running example introduced in Example 2. In this case, the initial situation can be represented by an epistemic model* $\mathcal{M}'_0$ *depicted in Figure 7a. Then, it can be checked that*

$$\mathcal{M}'_0 \models \mathbf{K}[\Gamma_3][\Gamma_5][\Gamma_3](\sim v \wedge d)$$

holds with $\Gamma_5 = \Gamma_{pink} \cup \{\texttt{take\_right}\}$. *In other words, the thief knows that after executing the sequence of actions* $\langle \texttt{move}, \texttt{take\_right}, \texttt{move} \rangle$ *she will be out of the vault with the diamond.*

---

[3]Note that, for the sake of clarity, the names of worlds have been chosen so that $\mathcal{R}_{\mathcal{M}_0, \Gamma_3}$ is the identity, but this is not necessarily the case. In fact, worlds from $\mathcal{M}_0$ and $\mathcal{M}_1$ could be disjoint or even be switched so that w_1 could be called w_2 and vice-versa.
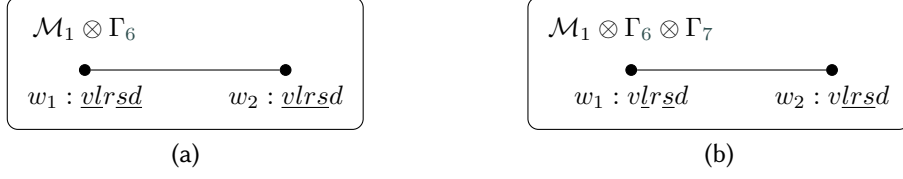
Figure 8: Epistemic models corresponding to (a) the execution of `try_take_left` in the model $\mathcal{M}_1 = \mathcal{M}_0 \otimes \Gamma_6$ and (b) the execution of `try_take_right` in the resulting state.

That is, this sequence of actions is a valid plan that achieves the goal of getting out of the vault with the diamond regardless of the actual initial situation. This means that this is a conformant plan.

For the sake of completeness, Figures 7b, c and d respectively depict the epistemic models $\mathcal{M}_0' \otimes \Gamma_3$, $\mathcal{M}_0' \otimes \Gamma_3 \otimes \Gamma_5$ and $\mathcal{M}_0' \otimes \Gamma_3 \otimes \Gamma_5 \otimes \Gamma_3$. $\square$

Example 10 (Example 1 continued). *As a further example, consider another variation of Example 1 where we have actions* `try_take_left` *and* `try_take_right` *that are similar to* `take_left` *and* `try_take_left`*, but that can be executed even when the diamond is not in the right location, having no effect in such case. This can be represented by a theory Γ'_{pink} obtained from Γ_{pink} by replacing rules (1-6) by the following rules:*

$$d \leftarrow \texttt{try\_take\_left} \wedge {}^\bullet\!\sim\! r \tag{17}$$

$$\bot \leftarrow \texttt{try\_take\_left} \wedge \sim\!{}^\bullet v \tag{18}$$

$$d \leftarrow \texttt{try\_take\_right} \wedge {}^\bullet r \tag{19}$$

$$\bot \leftarrow \texttt{try\_take\_right} \wedge \sim\!{}^\bullet v \tag{20}$$

Now, we can check that

$$\mathcal{M}_0 \models \mathbf{K}[\Gamma_3][\Gamma_6][\Gamma_7][\Gamma_3](\sim\! v \wedge d) \tag{21}$$

holds with $\Gamma_6 = \Gamma'_{pink} \cup \{\texttt{try\_take\_left}\}$ and $\Gamma_7 = \Gamma'_{pink} \cup \{\texttt{try\_take\_right}\}$. Recall that $\mathcal{M}_0 \otimes \Gamma_3 = \mathcal{M}_1$ is the epistemic model depicted in Figure 1a. We also can check that $\mathcal{M}_0 \otimes \Gamma_3 \otimes \Gamma_6 \otimes \Gamma_7 \otimes \Gamma_3 = \mathcal{M}_4$ is the epistemic model depicted in Figure 4b. Figure 8 depicts the epistemic models corresponding to intermediate states. With these models, we can see that (21) holds. That is, the thief knows that after executing the sequence of actions actions $\langle \texttt{move}, \texttt{try\_take\_left}, \texttt{try\_take\_right}, \texttt{move} \rangle$ she will be outside the vault with the diamond. Therefore, this sequence of actions constitutes a conformant plan for this problem. Note that the thief achieves her goal without ever getting to know where the diamond actually was. $\square$

Conditional Planning in DEL[ASP]

In this section, we show how to use DEL[ASP] to represent conditional plans. Let start by defining what a plan is by introducing the following plan language from [21].

Definition 17 (Plan Language). *Given disjunct sets of actions $\mathcal{A}$ and fluents $\mathcal{F}$, a* plan *is an expression π built with the following grammar:*

$$\pi \quad ::= \quad a \mid \text{skip} \mid \text{if } \mathbf{K}\,\varphi \text{ then } \pi \text{ else } \pi \mid \pi;\pi$$

where $a \in \mathcal{A}$ and φ is a formula over $\mathcal{F}$. We write (if $\mathbf{K}\,\varphi$ then π) as a shorthand for the plan (if $\mathbf{K}\,\varphi$ then π else skip). □

As mentioned in the introduction, conditional plans contain "if-then-else" structures that allow the agent to apply different strategics depending on the knowledge she has obtained along the execution of the plan. For instance,

$$\texttt{move ; flick ; if } \mathbf{K}\,r \text{ then } \texttt{take\_right} \text{ else } \texttt{take\_left ; move} \tag{22}$$

is a conditional plan for the problem laid out in Example 1. It is a plan since, as we will prove next, the thief eventually takes the diamond out in all possible outcomes, and it is conditional because the third step contains an alternative decision. If the thief acts according to her knowledge about the diamond position at that point, the plan is guaranteed to succeed. We will show that in fact, after executing the actions move and flick, the thief knows that she will know where the diamond is.

Let us now formalise these intuitive ideas by providing a translation from plans into DEL[ASP] as follows:

Definition 18 (Translation). *Let $\mathcal{A} \subseteq \mathbf{P}$ and $\mathcal{F} \subseteq \mathbf{P}$ be a pair of disjoint sets of propositions, respectively corresponding to actions and fluents. The translation of a plan π over $\mathcal{A}$ applied to a formula ψ over $\mathcal{F}$, with respect to a theory Γ over $\mathbf{P}^{bi}$ is denoted as $[\![\,\pi\,]\!]^{\Gamma}\psi$ and is recursively defined as:*

$$
\begin{aligned}
[\![\,a\,]\!]^{\Gamma}\psi &\overset{\text{def}}{=} \langle \Gamma \cup \{a\} \rangle \top \wedge [\Gamma \cup \{a\}]\psi \\
[\![\,\text{skip}\,]\!]^{\Gamma}\psi &\overset{\text{def}}{=} \psi \\
[\![\,\pi;\pi'\,]\!]^{\Gamma}\psi &\overset{\text{def}}{=} [\![\,\pi\,]\!]^{\Gamma}\,([\![\,\pi'\,]\!]^{\Gamma}\psi) \\
[\![\,\text{if } \mathbf{K}\,\varphi \text{ then } \pi \text{ else } \pi'\,]\!]^{\Gamma}\psi &\overset{\text{def}}{=} (\mathbf{K}\,\varphi \to [\![\,\pi\,]\!]^{\Gamma}\psi) \wedge (\sim\!\mathbf{K}\,\varphi \to [\![\,\pi'\,]\!]^{\Gamma}\psi)
\end{aligned}
$$

where φ is any formula over $\mathcal{F}$. □

As a first remark, note that the translation $[\![\,a\,]\!]^{\Gamma}$ of an action a is always made by adding a constant theory Γ that defines the behaviour of the action domain (fixing the transition relation). As a result, each elementary action in the plan becomes a complete autoepistemic theory $\Gamma \cup \{a\}$ in the translation. When Γ is clear from the context, we will simply write $[\![\,\pi\,]\!]$ instead of $[\![\,\pi\,]\!]^{\Gamma}$. Conjunct $[\Gamma \cup \{a\}]\,\psi$ requires that ψ becomes true in any resulting state whereas $\langle \Gamma \cup \{a\} \rangle \top$ ensures that action a is executable indeed.

We check next that the evaluation of plan (22) corresponds to what we have already seen in Example 8. For the sake of clarity, we gather together all rules of the theory Γ_{pink} in Figure 9.

Example 11 (Example 8 continued). Going on with our running example, let us consider plans $[\![\,\pi\,]\!]^{\Gamma_{pink}}$ simply denoted as $[\![\,\pi\,]\!]$. We have seen that $\mathcal{M}_0 \otimes \Gamma_3 = \mathcal{M}_1$ where $\Gamma_3 =$

$$
\begin{array}{ll}
d \;\leftarrow\; \texttt{take\_left} & (1) \\
\bot \;\leftarrow\; \texttt{take\_left}, \sim\!\bullet v & (2) \\
\bot \;\leftarrow\; \texttt{take\_left}, \bullet r & (3) \\
d \;\leftarrow\; \texttt{take\_right} & (4) \\
\bot \;\leftarrow\; \texttt{take\_right}, \sim\!\bullet v & (5) \\
\bot \;\leftarrow\; \texttt{take\_right}, \sim\!\bullet r & (6) \\
l \;\leftarrow\; \texttt{flick} & (7) \\
\bot \;\leftarrow\; \texttt{flick}, \sim\!\bullet v & (8) \\
v \;\leftarrow\; \texttt{move}, \sim\!\bullet v & (9) \\
\sim\!v \;\leftarrow\; \texttt{move}, \bullet v & (10)
\end{array}
\qquad
\begin{array}{ll}
s \;\leftarrow\; v, l & (11) \\
\mathbf{O}\,r \;\leftarrow\; v, l & (15) \\
\sim\!s \;\leftarrow\; not\ s & (14) \\
v \;\leftarrow\; \bullet v, not \sim\!v & \\
\sim\!v \;\leftarrow\; \sim\!\bullet v, not\ v & \\
l \;\leftarrow\; \bullet l, not \sim\!l & \\
\sim\!l \;\leftarrow\; \sim\!\bullet l, not\ l & \\
r \;\leftarrow\; \bullet r, not \sim\!r & \\
\sim\!r \;\leftarrow\; \sim\!\bullet r, not\ r & \\
d \;\leftarrow\; \bullet d, not \sim\!d & \\
\sim\!d \;\leftarrow\; \sim\!\bullet d, not\ d &
\end{array}
$$

Figure 9: Theory Γ_{pink}: the left column contains the direct effects and preconditions of actions while the right one contains the indirect effects and the inertia axioms.

$\Gamma_{pink} \cup \{\texttt{move}\}$ was the theory representing the execution of action $\texttt{move}$ according to Γ_{pink}. We have also seen that $\mathcal{R}_{\mathcal{M}_0,\Gamma_3} = \{(w_1, w_1), (w_2, w_2)\}$ is the identity. Similarly, given theory $\Gamma_8 = \Gamma_{pink} \cup \{\texttt{flick}\}$ representing the execution of action $\texttt{flick}$, we have $\mathcal{M}_1 \otimes \Gamma_8 = \mathcal{M}_2$ and $\mathcal{R}_{\mathcal{M}_1,\Gamma_8} = \{(w_1, w_1), (w_2, w_2)\}$. Figure 10 shows these three models together with the corresponding relations $\mathcal{R}_{\mathcal{M}_0,\Gamma_3}$ and $\mathcal{R}_{\mathcal{M}_1,\Gamma_8}$. Looking at this figure, we observe that $\mathcal{M}_2, w_1 \models v \wedge \mathbf{K}\,r$ and, thus, also $\mathcal{M}_2, w_1 \models v \wedge (\mathbf{K}\,r \vee \mathbf{K} \sim\!r)$. From this we can conclude that $\mathcal{M}_1, w_1 \models [\Gamma_8](v \wedge (\mathbf{K}\,r \vee \mathbf{K} \sim\!r))$. Note that $\mathcal{M}_1 \models \langle\Gamma_8\rangle\top$ holds and, thus, it follows

$$\mathcal{M}_1, w_1 \models [\![\,\texttt{flick}\,]\!](v \wedge (\mathbf{K}\,r \vee \mathbf{K} \sim\!r))$$

Now we can check that $\mathcal{M}_0, w_1 \models [\Gamma_3][\![\,\texttt{flick}\,]\!](v \wedge (\mathbf{K}\,r \vee \mathbf{K} \sim\!r))$ and $\mathcal{M}_0 \models \langle\Gamma_3\rangle\top$ hold and, thus, we can conclude

$$\mathcal{M}_0, w_1 \models [\![\,\texttt{move}\,]\!]\big([\![\,\texttt{flick}\,]\!](v \wedge (\mathbf{K}\,r \vee \mathbf{K} \sim\!r))\big)$$

An analogous reasoning, allow us to see that the same holds for $\mathcal{M}_0, w_2$ and, thus, we obtain

$$\mathcal{M}_0 \models \mathbf{K}[\![\,\texttt{move}\,]\!]\big([\![\,\texttt{flick}\,]\!](v \wedge (\mathbf{K}\,r \vee \mathbf{K} \sim\!r))\big)$$

By definition, these two facts imply

$$\mathcal{M}_0 \models \mathbf{K}[\![\,\texttt{move};\texttt{flick}\,]\!](v \wedge (\mathbf{K}\,r \vee \mathbf{K} \sim\!r)) \qquad (23)$$

In other words, the thief knows that, after executing actions $\texttt{move}$ and $\texttt{flick}$, she will be inside the vault and that she will know where the diamond is. So she will be ready for the next step: using her knowledge to decide what is the suitable action to continue the plan. $\qquad\square$

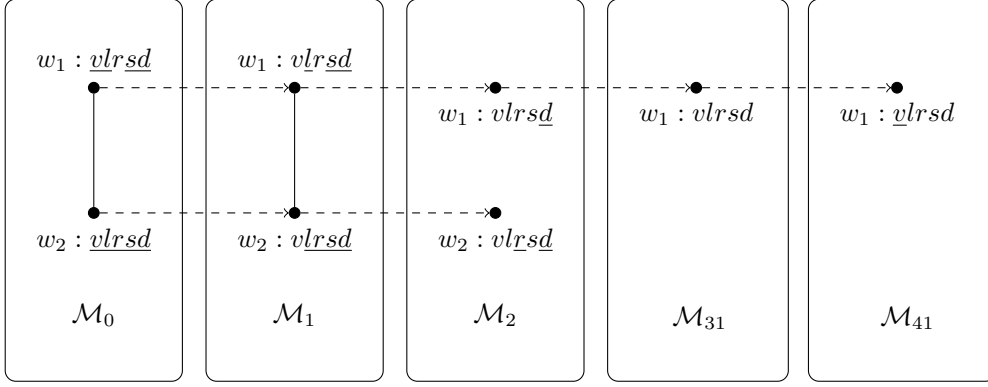Let us now continue with the thief's reasoning process after the execution of the first two actions.

Figure 10: Execution of the sequence of actions $\langle \texttt{move}, \texttt{flick}, \texttt{take\_right}, \texttt{move} \rangle$ starting at $\mathcal{M}_0, w_1$ of Example 1. We have $\mathcal{M}_{i+1} = \mathcal{M}_i \otimes (\Gamma_{pink} \cup \{a_i\})$ with a_i the corresponding action in the sequence. The dotted arrows depict the $\mathcal{R}$ relation associated with the update of $\mathcal{M}_i$ with respect to $\Gamma_{pink} \cup \{a_i\}$. Note that action $\texttt{take\_right}$ is not executable in $\mathcal{M}_2, w_2$ and, as a result, w_2 has no associated world in $\mathcal{M}_3$.

Example 12 (Example 11 continued). We will show now that

$$\mathcal{M}_2, w_1 \models [\![\text{ if } \mathbf{K}\, r \text{ then } \texttt{take\_right} \text{ else } \texttt{take\_left}; \texttt{move}]\!](v \wedge d) \tag{24}$$

is satisfied. First note that

$$\begin{aligned} \mathcal{M}_{31} \otimes \Gamma_3 &= \mathcal{M}_{41} \\ \mathcal{R}_{\mathcal{M}_2, \Gamma_3} &= \{(w_1, w_1)\} \end{aligned}$$

and that $\mathcal{M}_{41}, w_1 \models \sim v \wedge d$ and, thus, we get $\mathcal{M}_{41}, w_1 \models [\![\texttt{move}]\!](\sim v \wedge d)$. Let now $\Gamma_9 \stackrel{\text{def}}{=} \Gamma_{pink} \cup \{\texttt{take\_right}\}$. Then, we have

$$\begin{aligned} \mathcal{M}_2 \otimes \Gamma_9 &= \mathcal{M}_{31} \\ \mathcal{R}_{\mathcal{M}_2, \Gamma_9} &= \{(w_1, w_1)\} \end{aligned}$$

from where we get $\mathcal{M}_2, w_1 \models [\Gamma_9][\![\texttt{move}]\!](\sim v \wedge d)$. Furthermore, this implies $\mathcal{M}_2, w_1 \models [\![\texttt{take\_right}]\!][\![\texttt{move}]\!](\sim v \wedge d)$, which in its turn implies

$$\mathcal{M}_2, w_1 \models \mathbf{K}\, r \to [\![\texttt{take\_right}]\!][\![\texttt{move}]\!](\sim v \wedge d)$$

Note that $\mathcal{M}_2, w_1 \models \mathbf{K}\, r$ and, thus,

$$\mathcal{M}_2, w_1 \models \sim\mathbf{K}\, r \to [\![\texttt{take\_right}]\!][\![\texttt{move}]\!](\sim v \wedge d)$$

also follows. As a result, we can see that (24) holds. Now we follow the reasoning from Example 11 to show that $\mathcal{M}_0, w_1 \models [\![(22)]\!](\sim v \wedge d)$. That is, (22) is a plan that achieves the goal of Example 1 in the case that the diamond is in the right pedestal. Analogously, Figure 11 shows the models needed to prove $\mathcal{M}_0, w_2 \models [\![(22)]\!](\sim v \wedge d)$, that is, when the diamond was on the left. As a result, we obtain $\mathcal{M}_0 \models \mathbf{K}[\![(22)]\!](\sim v \wedge d)$. In other words, the thief knows that after executing (22), she will succeed in her goal: being outside of the vault with the diamond. $\quad\square$
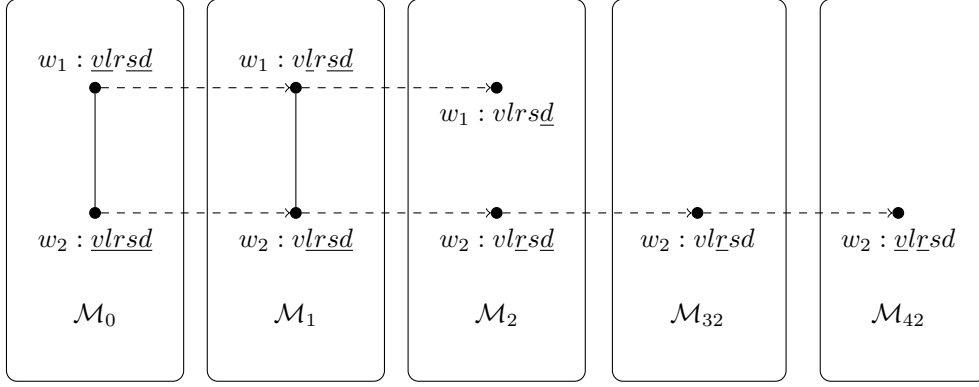
Figure 11: Execution of the sequence of actions $\langle \texttt{move}, \texttt{flick}, \texttt{take\_left}, \texttt{move} \rangle$ starting at $\mathcal{M}_0, w_2$ of Example 1. This figure is analogous to Figure 10 but replacing action $\texttt{take\_right}$ by $\texttt{take\_left}$ and models $\mathcal{M}_{31}$ and $\mathcal{M}_{41}$ by $\mathcal{M}_{32}$ and $\mathcal{M}_{42}$, respectively.

To conclude this section, we formalise the concepts of planing task and planning solution.

Definition 19 (Planning task). *Given the disjoint sets of actions $\mathcal{A} \subseteq \mathbf{P}$ and fluents $\mathcal{F} \subseteq \mathbf{P}$, a planning task is a triple $\Pi = \langle \Gamma_0, \Gamma, \varphi_g \rangle$ where Γ_0 is a theory over $\mathbf{P} \setminus \mathcal{A}$ defining the initial state, Γ is a theory over $\mathbf{P}^{bi}$ defining the interpretation of actions and φ_g is the* goal formula *over $\mathcal{F}$.* $\quad\square$

Definition 20 (Planning solution). *A plan π is a* conditional solution *for the planning task $\Pi = \langle \Gamma_0, \Gamma, \varphi_g \rangle$ iff $\texttt{mwv}(\Gamma_0, \mathcal{F}) \models [\![\pi]\!]^\Gamma \varphi_g$. A conditional solution without occurrences of the "if-then-else" construct is called a* conformant solution. $\quad\square$

In particular, Example 1 can be now formalised as the planning task $\Pi = \langle \Gamma_0, \Gamma_{pink}, \varphi_g \rangle$ where Γ_0 is a singleton containing (16), describing the initial situation, and $\varphi_g = {\sim}v \wedge d$. Then, we can see that (22) is a conditional solution for the planning task Π. We can also formalise Example 2 as the planning task $\Pi = \langle \Gamma'_0, \Gamma_{pink}, \varphi_g \rangle$ where Γ'_0 contains the single formula:

$$\varphi'_0 = {\sim}v \wedge {\sim}l \wedge r \wedge {\sim}s \wedge {\sim}d \tag{25}$$

that describes the corresponding initial situation. It can be checked that (22) is also a conditional solution for Π', though this example also has the simpler (conformant) solution:

$$\texttt{move} \ ; \ \texttt{take\_right} \ ; \ \texttt{move} \tag{26}$$

Finally, Example 10 becomes the task $\Pi = \langle \Gamma_0, \Gamma'_{pink}, \varphi_g \rangle$ for which

$$\texttt{move} \ ; \ \texttt{try\_take\_right} \ ; \ \texttt{try\_take\_left} \ ; \ \texttt{move} \tag{27}$$

is a conformant solution.

Conclusions and Future Work

As discussed in [62], the traditional DEL[$\mathcal{E}$] approach with event model updates is a *semantic approach*, where states and actions are represented as semantic objects, epistemic and event models respectively. On the other hand, DEL[ASP] is a *syntactic approach*, where states and actions are represented as knowledge-bases, that is, sets of formulas known to be true. Semantic and syntactic approaches are mutually dual, with the semantic approach modelling *ignorance* (the more ignorance, the bigger the state) and the syntactic approach modelling *knowledge* (the more knowledge, the bigger the knowledge-base). The generalisation of DEL for abstract updating objects can easily accommodate both approaches: it suffices with allowing both event models and epistemic programs to occur in the dynamic operator, and selecting the corresponding updating evaluation.

Another interesting observation is that both DEL[$\mathcal{E}$] and ASP can be considered as generalisations of the STRIPS planning language in orthogonal directions. On the one hand, DEL[$\mathcal{E}$] allows planning in domains where the world is not fully observable, the effects of actions are not necessarily deterministic and where sensing actions may allow to gain knowledge about the actual state of the world. On the other hand, ASP introduces high level KR features like the treatment of indirect effects, action qualifications, state constraints or recursive fluents (for motivation about the need of such features we refer to [63]). The approach presented here, DEL[ASP], combines the strengths of both generalisations so that it is possible to use high level KR features in non-fully observable or non-deterministic domains where observing the world may be needed to achieve a valid plan.

Similar to our approach, the action language $m\mathcal{AL}$ [64] also combined the treatment of indirect effects and action qualifications with the possibility of defining sensing actions. The main handicap of $m\mathcal{AL}$ with respect to DEL[ASP] is that the former only allows ramifications on the ontic effects, but not on the epistemic ones, as we did for instance with rule (15). In $m\mathcal{AL}$, as in DEL[$\mathcal{E}$], this indirect observation needs to be encoded as a direct effect of all actions that may affect those fluents. On the other hand, an advantage of both DEL[$\mathcal{E}$] and $m\mathcal{AL}$ is that they can be applied on domains that involve several agents and in which those agents may even hold false beliefs [22], while, so far, DEL[ASP] is only able to deal with domains involving a single agent. Extending DEL[ASP] to cover these domains is a matter of future work. It will be also interesting to study the relation between DEL[ASP] and Temporal ASP [57] and the possibility of extending the latter with an epistemic modality to deal with non-fully observable or non-deterministic domains.

Regarding the computation of planning solutions in DEL[ASP], it is worth to mention that the algorithm based on planning trees described in [21] for DEL[$\mathcal{E}$] is general enough and does not really depend of the kind of updating object used. In this sense, we can apply that same algorithm with the only variation of using the ASP updating evaluation when we expand the tree. Then, solutions can be retrieved from the planning tree in exactly the same way as described there.

Acknowledgments

References

[1] M. Ghallab, D. S. Nau, P. Traverso, Automated planning - theory and practice, Elsevier, 2004.

[2] P. H. Tu, T. C. Son, C. Baral, Reasoning and planning with sensing actions, incomplete information, and static causal laws using answer set programming, Theory and Practice of Logic Programming 7 (2007) 377–450.

[3] M. A. Peot, D. E. Smith, Conditional nonlinear planning, in: Artificial Intelligence Planning Systems, Elsevier, 1992, pp. 189–197.

[4] K. Golden, D. S. Weld, Representing sensing actions: The middle ground revisited, in: KR, Morgan Kaufmann, 1996, pp. 174–185.

[5] L. Pryor, G. Collins, Planning for contingencies: A decision-based approach, J. Artif. Intell. Res. 4 (1996) 287–339.

[6] J. Lobo, G. Mendez, S. R. Taylor, Adding knowledge to the action description language A, in: B. Kuipers, B. L. Webber (Eds.), Proceedings of the Fourteenth National Conference on Artificial Intelligence and Ninth Innovative Applications of Artificial Intelligence Conference, AAAI 97, IAAI 97, July 27-31, 1997, Providence, Rhode Island, USA., AAAI Press / The MIT Press, 1997, pp. 454–459.

[7] A. L. Blum, M. L. Furst, Fast planning through planning graph analysis, Artificial intelligence 90 (1997) 281–300.

[8] D. E. Smith, D. S. Weld, Conformant graphplan, in: AAAI/IAAI, 1998, pp. 889–896.

[9] K. Golden, Leap before you look: Information gathering in the PUCCINI planner, in: AIPS, AAAI, 1998, pp. 70–77.

[10] D. S. Weld, C. R. Anderson, D. E. Smith, Extending graphplan to handle uncertainty & sensing actions, in: Aaai/iaai, 1998, pp. 897–904.

[11] J. Rintanen, Constructing conditional plans by a theorem-prover, Journal of Artificial Intelligence Research 10 (1999) 323–352.

[12] B. Bonet, H. Geffner, Planning with incomplete information as heuristic search in belief space, in: Proceedings of the Fifth International Conference on Artificial Intelligence Planning Systems, AAAI Press, 2000, pp. 52–61.

[13] C. Castellini, E. Giunchiglia, A. Tacchella, Sat-based planning in complex domains: Concurrency, constraints and nondeterminism, Artificial Intelligence 147 (2003) 85–117.

[14] T. Eiter, W. Faber, N. Leone, G. Pfeifer, A. Polleres, A logic programming approach to knowledge-state planning, ii: The dlvk system, Artificial Intelligence 144 (2003) 157–211.

[15] A. Cimatti, M. Roveri, P. Bertoli, Conformant planning via symbolic model checking and heuristic search, Artificial Intelligence 159 (2004) 127–206.

[16] D. Bryce, S. Kambhampati, D. E. Smith, Planning graph heuristics for belief space search, Journal of Artificial Intelligence Research 26 (2006) 35–99.

[17] J. Hoffmann, R. I. Brafman, Conformant planning via heuristic forward search: A new approach, Artificial Intelligence 170 (2006) 507–541.

[18] B. Löwe, E. Pacuit, A. Witzel, Planning based on dynamic epistemic logic, Technical Report, Citeseer, 2010.

[19] T. Bolander, M. B. Andersen, Epistemic planning for single and multi-agent systems, Journal of Applied Non-Classical Logics 21 (2011) 9–34.

[20] E. Pontelli, T. C. Son, C. Baral, G. Gelfond, Answer set programming and planning with knowledge and world-altering actions in multiple agent domains, in: Correct Reasoning, Springer, 2012, pp. 509–526.

[21] M. B. Andersen, T. Bolander, M. H. Jensen, Conditional epistemic planning, in: JELIA, volume 7519 of *Lecture Notes in Computer Science*, Springer, 2012, pp. 94–106.

[22] T. Bolander, Seeing is believing: Formalising false-belief tasks in dynamic epistemic logic, in: ECSI, volume 1283 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2014, pp. 87–107.

[23] F. Kominis, H. Geffner, Beliefs in multiagent planning: From one agent to many, in: R. I. Brafman, C. Domshlak, P. Haslum, S. Zilberstein (Eds.), Proceedings of the Twenty-Fifth International Conference on Automated Planning and Scheduling, ICAPS 2015, Jerusalem, Israel, June 7-11, 2015., AAAI Press, 2015, pp. 147–155.

[24] F. Kominis, H. Geffner, Multiagent online planning with nested beliefs and dialogue, in: L. Barbulescu, J. Frank, Mausam, S. F. Smith (Eds.), Proceedings of the Twenty-Seventh International Conference on Automated Planning and Scheduling, ICAPS 2017, Pittsburgh, Pennsylvania, USA, June 18-23, 2017., AAAI Press, 2017, pp. 186–194.

[25] M. C. Cooper, A. Herzig, F. Maffre, F. Maris, P. Régnier, Simple epistemic planning: Generalised gossiping, in: G. A. Kaminka, M. Fox, P. Bouquet, E. Hüllermeier, V. Dignum, F. Dignum, F. van Harmelen (Eds.), ECAI 2016 - 22nd European Conference on Artificial Intelligence, 29 August-2 September 2016, The Hague, The Netherlands, volume 285 of *Frontiers in Artificial Intelligence and Applications*, IOS Press, 2016, pp. 1563–1564.

[26] M. C. Cooper, A. Herzig, F. Maffre, F. Maris, P. Régnier, A simple account of multi-agent epistemic planning, in: G. A. Kaminka, M. Fox, P. Bouquet, E. Hüllermeier, V. Dignum, F. Dignum, F. van Harmelen (Eds.), ECAI 2016 - 22nd European Conference on Artificial Intelligence, 29 August-2 September 2016, The Hague, The Netherlands, volume 285 of *Frontiers in Artificial Intelligence and Applications*, IOS Press, 2016, pp. 193–201.

[27] T. Bolander, T. Engesser, R. Mattmüller, B. Nebel, Better eager than lazy? how agent types impact the successfulness of implicit coordination, in: M. Thielscher, F. Toni, F. Wolter (Eds.), Principles of Knowledge Representation and Reasoning: Proceedings of the Sixteenth International Conference, KR 2018, Tempe, Arizona, 30 October - 2 November 2018., AAAI Press, 2018, pp. 445–453.

[28] R. C. Moore, A formal theory of knowledge and action, Technical Report, SRI International, Menlo Park, CA, Artificial Intelligence Center, 1984.

[29] M. Thielscher, Representing the knowledge of a robot, in: KR, 2000, pp. 109–120.

[30] T. C. Son, C. Baral, Formalizing sensing actions—a transition function based approach, Artificial Intelligence 125 (2001) 19–91.

[31] R. B. Scherl, H. J. Levesque, Knowledge, action, and the frame problem, Artificial Intelligence 144 (2003) 1–39.

[32] H. Van Ditmarsch, W. van Der Hoek, B. Kooi, Dynamic epistemic logic, volume 337,

Springer Science & Business Media, 2007.

[33] H. van Ditmarsch, B. Kooi, Semantic results for ontic and epistemic change, Logic and the foundations of game and decision theory (LOFT 7) 3 (2008) 87–117.

[34] V. Pratt, Semantical consideration on floyd-hoare logic, in: Proceedings of the Seventeenth Annual Symposium on Foundations of Computer Science (SFCS'76), IEEE Computer Society Press, 1976, pp. 109–121.

[35] H. P. van Ditmarsch, W. van der Hoek, B. P. Kooi, Dynamic epistemic logic with assignment, in: Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems, ACM, 2005, pp. 141–148.

[36] A. Baltag, L. S. Moss, S. Solecki, The logic of public announcements and common knowledge and private suspicions, in: TARK, Morgan Kaufmann, 1998, pp. 43–56.

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

[38] R. E. Fikes, N. J. Nilsson, STRIPS: A new approach to the application of theorem proving to problem solving, Artificial Intelligence 2 (1971) 189 – 208.

[39] V. W. Marek, M. Truszczyński, Stable models and an alternative logic programming paradigm, in: K. R. Apt, V. W. Marek, M. Truszczyński, D. Warren (Eds.), The Logic Programming Paradigm, Artificial Intelligence, Springer Berlin Heidelberg, 1999, pp. 375–398.

[40] I. Niemelä, Logic programs with stable model semantics as a constraint programming paradigm, Annals of Mathematics and Artificial Intelligence 25 (1999) 241–273.

[41] C. Baral, Knowledge Representation, Reasoning and Declarative Problem Solving, Cambridge University Press, 2010.

[42] M. Gelfond, V. Lifschitz, The stable model semantics for logic programming, in: Proc. of the 5th Intl. Conference on Logic Programming (ICLP'88), 1988, pp. 1070–1080.

[43] M. Gelfond, V. Lifschitz, Classical negation in logic programs and disjunctive databases, New Generation Comput. 9 (1991) 365–386.

[44] V. Lifschitz, Answer set planning, in: D. de Schreye (Ed.), Proceedings of the International Conference on Logic Programming (ICLP'99), MIT Press, 1999, pp. 23–37.

[45] V. Lifschitz, Answer set programming and plan generation, Artificial Intelligence 138 (2002) 39 – 54. Knowledge Representation and Logic Programming.

[46] J. Lee, V. Lifschitz, F. Yang, Action language $\mathcal{BC}$ preliminary report, in: F. Rossi (Ed.), IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013, IJCAI/AAAI, 2013, pp. 983–989.

[47] P. Cabalar, R. Kaminski, T. Schaub, A. Schuhmann, Temporal answer set programming on finite traces, Theory and Practice of Logic Programming 18 (2018) 406–420.

[48] E. Erdem, M. Gelfond, N. Leone, Applications of ASP, AI Magazine 37 (2016) 53–68.

[49] M. Gelfond, Strong introspection, in: T. L. Dean, K. McKeown (Eds.), Proceedings of the AAAI Conference, volume 1, AAAI Press/The MIT Press, 1991, pp. 386–391.

[50] A. P. Leclerc, P. T. Kahl, A survey of advances in epistemic logic program solvers, CoRR abs/1809.07141 (2018). URL: http://arxiv.org/abs/1809.07141. arXiv:1809.07141.

[51] P. Cabalar, J. Fandinno, L. Fariñas del Cerro, Founded world views with autoepistemic equilibrium logic, Under consideration for 15th International Conference on Logic Programming and Nonmonotonic Reasoning (LPNMR 2019), Corr abs/1902.07741 (2019). URL: https://arxiv.org/abs/1902.07741.

[52] D. Pearce, A new logical characterisation of stable models and answer sets, in: J. Dix, L. M. Pereira, T. C. Przymusinski (Eds.), Non-Monotonic Extensions of Logic Programming, NMELP 1996, Bad Honnef, Germany, September 5-6, 1996, Selected Papers, volume 1216 of *Lecture Notes in Computer Science*, Springer, 1996, pp. 57–70.

[53] R. C. Moore, Semantical considerations on nonmonotonic logic, Artificial Intelligence 25 (1985) 75–94.

[54] G. Aucher, An internal version of epistemic logic, Studia Logica 94 (2010) 1–22.

[55] R. Fagin, Reasoning about knowledge, MIT Press, 1995.

[56] J. Y. Halpern, Reasoning about knowledge: an overview, in: Theoretical aspects of reasoning about knowledge, Elsevier, 1986, pp. 1–17.

[57] P. Cabalar, R. Kaminski, T. Schaub, A. Schuhmann, Temporal answer set programming on finite traces, Theory and Practice of Logic Programming 18 (2018) 406–420.

[58] H. Kautz, The logic of persistence, in: Proceedings of the 5th National Conference of Artificial Intelligence, 1986, pp. 401–405.

[59] D. Nelson, Constructible falsity, J. Symbolic Logic 14 (1949) 16–26.

[60] V. Lifschitz, L. R. Tang, H. Turner, Nested expressions in logic programs, Ann. Math. Artif. Intell. 25 (1999) 369–389.

[61] A. Heyting, Die formalen Regeln der intuitionistischen Logik, in: Sitzungsberichte der Preussischen Akademie der Wissenschaften, Deutsche Akademie der Wissenschaften zu Berlin, 1930, p. 42–56. Reprint in Logik-Texte: Kommentierte Auswahl zur Geschichte der Modernen Logik, Akademie-Verlag, 1986.

[62] T. Bolander, A gentle introduction to epistemic planning: The DEL approach, in: M4M@ICLA, volume 243 of *EPTCS*, 2017, pp. 1–22.

[63] V. Lifschitz, Answer set programming and plan generation, Artificial Intelligence 138 (2002) 39–54.

[64] C. Baral, G. Gelfond, E. Pontelli, T. C. Son, Reasoning about the beliefs of agents in multi-agent domains in the presence of state constraints: The action language mal, in: J. Leite, T. C. Son, P. Torroni, L. van der Torre, S. Woltran (Eds.), Computational Logic in Multi-Agent Systems, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 290–306.