Conditional Epistemic Planning

Mikkel Birkegaard Andersen, Thomas Bolander, and Martin Holm Jensen

Technical University of Denmark

Abstract. Recent work has shown that Dynamic Epistemic Logic (DEL) offers a solid foundation for automated planning under partial observability and non-determinism. Under such circumstances, a plan must branch if it is to guarantee achieving the goal under all contingencies (strong planning). Without branching, plans can offer only the possibility of achieving the goal (weak planning). We show how to formulate planning in uncertain domains using DEL and give a language of conditional plans. Translating this language to standard DEL gives verification of both strong and weak plans via model checking. In addition to plan verification, we provide a tableau-inspired algorithm for synthesising plans, and show this algorithm to be terminating, sound and complete.

1 Introduction

Whenever an agent deliberates about the future with the purpose of achieving a goal, she is engaging in the act of planning. When planning, the agent has a view of the environment and knowledge of how her actions affect the environment. Automated Planning is a widely studied area of AI, in which problems are expressed along these lines. Many different variants of planning, with different assumptions and restrictions, have been studied. In this paper we consider planning under uncertainty (nondeterminism and partial observability), where exact states of affairs and outcomes of actions need not be known by the agent. We formulate such scenarios in an epistemic setting, where states, actions and goals are infused with the notions of knowledge from Dynamic Epistemic Logic (DEL). Throughout this exposition, our running example, starting with Example 1, follows the schemings of a thief wanting to steal a precious diamond.

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 ($\neg v$), she now deliberates on how to get her hands on the diamond (d). She knows the light inside the vault is off ($\neg l$), and that the Pink Panther is on either the right (r) or left ($\neg r$) pedestal inside. Obviously, the diamond cannot be on both the right *and* left pedestal, but nonetheless the agent may be uncertain about its location. This scenario is represented by the epistemic model in Figure 1. The edge between w_1 and w_2 signifies that these worlds are indistinguishable to the agent. For visual clarity we omit reflexive edges (each world is always reachable from itself). We indicate with a string the valuation at world w, where an underlined proposition p signifies that p does *not* hold at w.

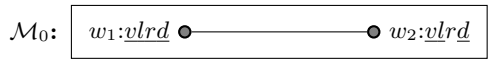

Fig. 1. The initial situation. The thief is uncertain about whether r holds.

The agent's goal is to obtain the jewel and to be outside the vault. She can enter and leave the vault, flick the light switch and snatch the contents of either the right or left pedestal. Her aim is to come up with a, possibly conditional, plan, such that she achieves her goal.

By applying DEL to scenarios such as the above, we can construct a procedure for the line of reasoning that is of interest to the thief. In the following section we recap the version of DEL relevant to our purposes. Section 3 formalises notions from planning in DEL, allowing verification of plans (using model checking) as either weak or strong solutions. In Section 4 we introduce an algorithm for plan synthesis (i.e. generation of plans). Further we show that the algorithm is terminating, sound and complete.

2 Dynamic Epistemic Logic

Dynamic epistemic logics describe knowledge and how actions change it. These changes may be epistemic (changing knowledge), ontic (changing facts) or both. The work in this paper deals only with the single-agent setting, though we briefly discuss the multi-agent setting in Section 5. As in Example 1, agent knowledge is captured by epistemic models. Changes are encoded using event models (defined below). The following concise summary of DEL is meant as a reference for the already familiar reader. The unfamiliar reader may consult [11, 12] for a thorough treatment.

Definition 1 (Epistemic Language). *Let a set of propositional symbols P be given. The language $\mathcal{L}_{\mathrm{DEL}}(P)$ is given by the following BNF:*

$$\phi ::= \top \mid p \mid \neg\phi \mid \phi \wedge \phi \mid K\phi \mid [\mathcal{E}, e]\,\phi$$

where $p \in P$, $\mathcal{E}$ denotes an event model *on $\mathcal{L}_{\mathrm{DEL}}(P)$ as (simultaneously) defined below, and $e \in D(\mathcal{E})$. K is the epistemic modality and $[\mathcal{E}, e]$ the dynamic modality. We use the usual abbreviations for the other boolean connectives, as well as for the dual dynamic modality $\langle \mathcal{E}, e\rangle\,\phi := \neg\,[\mathcal{E}, e]\,\neg\phi$. The dual of K is denoted $\tilde{K}$. $K\phi$ reads as "the (planning) agent knows ϕ" and $[\mathcal{E}, e]\,\phi$ as "after all possible executions of $(\mathcal{E}, e)$, ϕ holds".*

Definition 2 (Epistemic Models). *An* epistemic model *on $\mathcal{L}_{\mathrm{DEL}}(P)$ is a tuple $\mathcal{M} = (W, \sim, V)$, where W is a set of* worlds*, $\sim$ is an equivalence relation (the* epistemic relation*) on W, and $V : P \to 2^W$ is a* valuation*. $D(\mathcal{M}) = W$ denotes the* domain *of $\mathcal{M}$. For $w \in W$ we name $(\mathcal{M}, w)$ a* pointed epistemic model*, and refer to w as the* actual world *of $(\mathcal{M}, w)$.*

To reason about the dynamics of a changing system, we make use of *event models*. The formulation of event models we use in this paper is due to van Ditmarsch and Kooi [11]. It adds ontic change to the original formulation of [4] by adding postconditions to events.

Definition 3 (Event Models). *An* event model *on* $\mathcal{L}_{\mathrm{DEL}}(P)$ *is a tuple* $\mathcal{E} = (E, \sim, pre, post)$*, where*

- E *is a set of* (basic) events*,*
- $\sim \subseteq E \times E$ *is an equivalence relation called the* epistemic relation*,*
- $pre : E \to \mathcal{L}_{\mathrm{DEL}}(P)$ *assigns to each event a* precondition*,*
- $post : E \to (P \to \mathcal{L}_{\mathrm{DEL}}(P))$ *assigns to each event a* postcondition*.*

$D(\mathcal{E}) = E$ *denotes the* domain *of* $\mathcal{E}$*. For* $e \in E$ *we name* $(\mathcal{E}, e)$ *a* pointed event model*, and refer to* e *as the* actual event *of* $(\mathcal{E}, e)$*.*

Definition 4 (Product Update). *Let* $\mathcal{M} = (W, \sim, V)$ *and* $\mathcal{E} = (E, \sim', pre, post)$ *be an epistemic model resp. event model on* $\mathcal{L}_{\mathrm{DEL}}(P)$*. The* product update *of* $\mathcal{M}$ *with* $\mathcal{E}$ *is the epistemic model denoted* $\mathcal{M} \otimes \mathcal{E} = (W', \sim'', V')$*, where*

- $W' = \{(w, e) \in W \times E \mid \mathcal{M}, w \models pre(e)\}$*,*
- $\sim'' = \{((w, e), (v, f)) \in W' \times W' \mid w \sim v \text{ and } e \sim' f\}$*,*
- $V'(p) = \{(w, e) \in W' \mid \mathcal{M}, w \models post(e)(p)\}$ *for each* $p \in P$*.*

Definition 5 (Satisfaction Relation). *Let a pointed epistemic model* $(\mathcal{M}, w)$ *on* $\mathcal{L}_{\mathrm{DEL}}(P)$ *be given. The satisfaction relation is given by the usual semantics, where we only recall the definition of the dynamic modality:*

$$\mathcal{M}, w \models [\mathcal{E}, e]\,\phi \qquad \textit{iff } \mathcal{M}, w \models pre(e) \textit{ implies } \mathcal{M} \otimes \mathcal{E}, (w, e) \models \phi$$

where $\phi \in \mathcal{L}_{\mathrm{DEL}}(P)$ *and* $(\mathcal{E}, e)$ *is a pointed event model. We write* $\mathcal{M} \models \phi$ *to mean* $\mathcal{M}, w \models \phi$ *for all* $w \in D(\mathcal{M})$*. Satisfaction of the dynamic modality for non-pointed event models* $\mathcal{E}$ *is introduced by abbreviation, viz.* $[\mathcal{E}]\,\phi := \bigwedge_{e \in \mathcal{D}(\mathcal{E})} [\mathcal{E}, e]\,\phi$*. Furthermore,* $\langle \mathcal{E} \rangle\,\phi := \neg\,[\mathcal{E}]\,\neg\phi$*.*[1]

Throughout the rest of this paper, all languages (sets of propositional symbols) and all models (sets of possible worlds) considered are implicitly assumed to be finite.

3 Conditional Plans in DEL

One way to sum up automated planning is that it deals with the *reasoning side of acting* [13]. When planning under uncertainty, actions can be nondeterministic and the states of affairs partially observable. In the following, we present a formalism expressing planning under uncertainty in DEL, while staying true to the notions of automated planning. We consider a system similar to that of [13, sect. 17.4], which motivates the following exposition. The type of planning detailed here is *offline*, where planning is done before acting. All reasoning must therefore be based on the agent's initial knowledge.

[1] Hence, $\mathcal{M}, w \models \langle \mathcal{E} \rangle\,\phi \Leftrightarrow \mathcal{M}, w \models \neg\,[\mathcal{E}]\,\neg\phi \Leftrightarrow \mathcal{M}, w \models \neg(\bigwedge_{e \in \mathcal{D}(\mathcal{E})} [\mathcal{E}, e]\,\neg\phi) \Leftrightarrow \mathcal{M}, w \models \bigvee_{e \in \mathcal{D}(\mathcal{E})} \neg\,[\mathcal{E}, e]\,\neg\phi \Leftrightarrow \mathcal{M}, w \models \bigvee_{e \in D(\mathcal{E})} \langle \mathcal{E}, e \rangle\,\phi$.

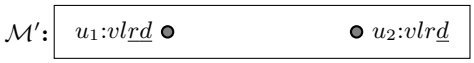

Fig. 2. A model consisting of two information cells

3.1 States and Actions: The Internal Perspective

Automated planning is concerned with achieving a certain goal state from a given initial state through some combination of available actions. In our case, states are epistemic models. These models represent situations from the perspective of the planning agent. We call this the *internal perspective*—the modeller is modelling itself. The internal perspective is discussed thoroughly in [1, 10].

Generally, an agent using epistemic models to model its own knowledge and ignorance, will not be able to point out the actual world. Consider the epistemic model $\mathcal{M}_0$ in Figure 1, containing two indistinguishable worlds w_1 and w_2. Regarding this model to be the planning agent's own representation of the initial state of affairs, the agent is of course not able to point out the actual world. It is thus natural to represent this situation as a non-pointed epistemic model. In general, when the planning agent wants to model a future (imagined) state of affairs, she does so by a non-pointed model.

The equivalence classes (wrt. $\sim$) of a non-pointed epistemic model are called the *information cells* of that model (in line with the corresponding concept in [5]). We generally identify any equivalence class $[w]_\sim$ of a model $\mathcal{M}$ with the submodel it induces, that is, we identify $[w]_\sim$ with $\mathcal{M} \restriction [w]_\sim$. We also use the expression *information cell* on $\mathcal{L}_{\mathrm{DEL}}(P)$ to denote any connected epistemic model on $\mathcal{L}_{\mathrm{DEL}}(P)$, that is, any epistemic model consisting of a single information cell. All worlds in an information cell satisfy the same K-formulas (formulas of the form $K\phi$), thus representing the same situation as seen from the agent's internal perspective. Each information cell of a (non-pointed) epistemic model represents a possible state of knowledge of the agent.

Example 2. Recall that our jewel thief is at the planning stage, with her initial information cell $\mathcal{M}_0$. She realises that entering the vault and turning on the light will reveal the location of the Pink Panther. Before actually performing these actions, she can rightly reason that they will lead her to know the location of the diamond, though whether that location is left or right cannot be determined (yet).

Her representation of the possible outcomes of going into the vault and turning on the light is the model $\mathcal{M}'$ in Figure 2. The information cells $\mathcal{M}' \restriction \{u_1\}$ and $\mathcal{M}' \restriction \{u_2\}$ of $\mathcal{M}'$ are exactly the two distinguishable states of knowledge the jewel thief considers possible prior turning the light on in the vault.

In the DEL framework, actions are naturally represented as event models. Due to the internal perspective, these are also taken to be non-pointed. For instance, in a coin toss action, the agent cannot beforehand point out which side will land face up.

Example 3. Continuing Example 2 we now formalize the actions available to our thieving agent as the event models in Figure 3. We use the same conven-

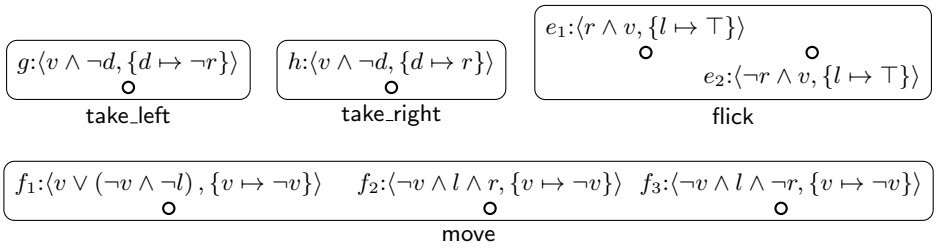

Fig. 3. Event models representing the actions of the thief

tions for edges as we did for epistemic models. For a basic event e we label it $\langle pre(e), post(e)\rangle$.[2]

The agent is endowed with four actions: take_left, resp. take_right, represent trying to take the diamond from the left, resp. right, pedestal; the diamond is obtained only if it is on the chosen pedestal. Both actions require the agent to be inside the vault and not holding the diamond. flick requires the agent to be inside the vault and turns the light on. Further, it reveals which pedestal the diamond is on. move represents the agent moving in or out of the vault, revealing the location of the diamond provided the light is on.

It can be seen that the epistemic model $\mathcal{M}'$ in Example 2 is the result of two successive product updates, namely $\mathcal{M}_0 \otimes \mathsf{move} \otimes \mathsf{flick}$.

3.2 Applicability, Plans and Solutions

Reasoning about actions from the initial state as in Example 3 is exactly what planning is all about. We have however omitted an important component in the reasoning process, one which is crucial. The notion of *applicability* in automated planning dictates when the outcomes of an action are defined. The idea translates to DEL by insisting that no world the planning agent considers possible is eliminated by the product update of an epistemic model with an event model.

Definition 6 (Applicability). *An event model $\mathcal{E}$ is said to be* applicable *in an epistemic model $\mathcal{M}$ if $\mathcal{M} \models \langle \mathcal{E}\rangle \top$.*

This concept of applicability is easily shown to be equivalent with the one defined in [10] when restricting the latter to the single-agent case. However, for our purposes of describing plans as formulas, we need to express applicability as formulas as well. The discussion in [15, sect. 6.6] also notes this aspect, insisting that actions must be meaningful. The same sentiment is expressed by our notion of applicability.

The situation in Example 2 calls for a way to express conditional plans. Clearly, our agent can only snatch the jewel from the correct pedestal conditioned on how events unfold when she acts. To this end we introduce a language for conditional plans allowing us to handle such contingencies.

[2] For a proposition p whose truth value does not change in e we assume the identity mapping $post(e)(p) = p$, as is also the convention in automated planning.

Definition 7 (Plan Language). *Given a finite set* A *of event models on* $\mathcal{L}_{\mathrm{DEL}}(P)$*, the* plan language $\mathcal{L}_{\mathrm{P}}(P, \mathsf{A})$ *is given by:*

$$\pi ::= \mathcal{E} \mid \mathsf{skip} \mid \mathsf{if}\ K\phi\ \mathsf{then}\ \pi\ \mathsf{else}\ \pi \mid \pi; \pi$$

where $\mathcal{E} \in \mathsf{A}$ *and* $\phi \in \mathcal{L}_{\mathrm{DEL}}(P)$*. We name members* π *of this language* plans*, and use* $\mathsf{if}\ K\phi\ \mathsf{then}\ \pi$ *as shorthand for* $\mathsf{if}\ K\phi\ \mathsf{then}\ \pi\ \mathsf{else}\ \mathsf{skip}$*.*

The reading of the plan constructs are "do $\mathcal{E}$", "do nothing", "if $K\phi$ then π, else π'", and "first π then π'" respectively. Note that the condition of the if-then-else construct is required to be a K-formula. This is to ensure that the planning agent can only make her choices of actions depend on worlds that are distinguishable to her (cf. the discussion of the internal perspective in Section 3.1). The idea is similar to the *meaningful plans* of [15], where branching is only allowed on *epistemically interpretable formulas*.

An alternative way of specifying conditional plans is *policies*, where (in our terminology) each information cell maps to an event model [13, Sect. 16.2]. There are slight differences between the expressiveness of conditional plans and policies (e.g. policies can finitely represent repetitions); our main motivation for not using policies is that it would require an enumeration of each information cell of the planning domain.

Definition 8 (Translation). *We define a* strong translation $[\![\cdot]\!]_s \cdot$ *and a* weak translation $[\![\cdot]\!]_w \cdot$ *as functions from* $\mathcal{L}_{\mathrm{P}}(P, \mathsf{A}) \times \mathcal{L}_{\mathrm{DEL}}(P)$ *into* $\mathcal{L}_{\mathrm{DEL}}(P)$ *by:*

$$\begin{aligned}
&[\![\mathcal{E}]\!]_s\, \phi := \langle \mathcal{E} \rangle \top \wedge [\mathcal{E}]\, K\phi \\
&[\![\mathcal{E}]\!]_w\, \phi := \langle \mathcal{E} \rangle \top \wedge \tilde{K}\, \langle \mathcal{E} \rangle\, K\phi \\
&[\![\mathsf{skip}]\!].\, \phi := \phi \\
&[\![\mathsf{if}\ \phi'\ \mathsf{then}\ \pi\ \mathsf{else}\ \pi']\!].\, \phi := (\phi' \to [\![\pi]\!].\, \phi) \wedge (\neg\phi' \to [\![\pi']\!].\, \phi) \\
&[\![\pi; \pi']\!].\, \phi := [\![\pi]\!].\, ([\![\pi']\!].\, \phi)
\end{aligned}$$

Plans describe the manner in which actions are carried out. We interpret plans π relative to a formula ϕ and want to answer the question of whether or not π achieves ϕ. Using Definition 8 we can answer this question by verifying truth of the DEL formula provided by the translations. This is supported by the results of Section 4. We concisely read $[\![\pi]\!]_s\, \phi$ as "π achieves ϕ", and $[\![\pi]\!]_w\, \phi$ as "π may achieve ϕ" (elaborated below). By not specifying separate semantics for plans our framework is kept as simple as possible. Note that applicability (Definition 6) is built into the translations through the occurrence of the conjunct $\langle \mathcal{E} \rangle \top$ in both the strong translation $[\![\mathcal{E}]\!]_s\, \phi$ and the weak translation $[\![\mathcal{E}]\!]_w\, \phi$.

The difference between the two translations relate to the *robustness* of plans: $[\![\pi]\!]_s\, \phi$, resp. $[\![\pi]\!]_w\, \phi$, means that every step of π is applicable and that following π always leads, resp. may lead, to a situation where ϕ is known.

Definition 9 (Planning Problems and Solutions). *Let* P *be a finite set of propositional symbols. A planning problem on* P *is a triple* $\mathcal{P} = (\mathcal{M}_0, \mathsf{A}, \phi_g)$ *where*

- $\mathcal{M}_0$ *is an information cell on* $\mathcal{L}_{\mathrm{DEL}}(P)$ *called the* initial state*.*
- A *is a finite set of event models on* $\mathcal{L}_{\mathrm{DEL}}(P)$ *called the* action library*.*
- $\phi_g \in \mathcal{L}_{\mathrm{DEL}}(P)$ *is the* goal (formula)*.*

We say that a plan $\pi \in \mathcal{L}_{\mathrm{P}}(P, \mathsf{A})$ *is a* strong solution *to* $\mathcal{P}$ *if* $\mathcal{M}_0 \models [\![\pi]\!]_s \phi_g$*, a* weak solution *if* $\mathcal{M}_0 \models [\![\pi]\!]_w \phi_g$ *and not a solution otherwise.*

Planning problems are defined with the sentiment we've propagated in our examples up until now. The agent is presently in $\mathcal{M}_0$ and wishes ϕ_g to be the case. To this end, she reasons about the actions (event models) in her action library A, creating a conditional plan. Using model checking, she can verify whether this plan is either a weak or strong solution, since plans translate into formulas of $\mathcal{L}_{\mathrm{DEL}}(P)$. Further, [11] gives reduction axioms for DEL-formulas, showing that any formula containing the dynamic modality can be expressed as a formula in (basic) epistemic logic. Consequently, plan verification can be seen simply as epistemic reasoning about $\mathcal{M}_0$.

Example 4. We continue our running example by discussing it formally as a planning problem and considering the solutions it allows. The initial state is still $\mathcal{M}_0$, and the action library $\mathsf{A} = \{\mathsf{flick}, \mathsf{move}, \mathsf{take\_left}, \mathsf{take\_right}\}$. We discuss the plans below and their merit for our thief.

- $\pi_1 = \mathsf{flick}; \mathsf{move}; \mathsf{if}\ Kr\ \mathsf{then\ take\_right\ else\ take\_left}; \mathsf{move}$
- $\pi_2 = \mathsf{move}; \mathsf{take\_right}; \mathsf{move}$
- $\pi_3 = \mathsf{move}; \mathsf{flick}; \mathsf{take\_right}; \mathsf{move}$
- $\pi_4 = \mathsf{move}; \mathsf{flick}; \mathsf{if}\ Kr\ \mathsf{then\ take\_right\ else\ take\_left}; \mathsf{move}$

We consider two planning problems varying only on the goal formula, $\mathcal{P}_1 = (\mathcal{M}_0, \mathsf{A}, d \wedge \neg v)$ and $\mathcal{P}_2 = (\mathcal{M}_0, \mathsf{A}, \tilde{K}d \wedge \neg v)$. In $\mathcal{P}_1$ her goal is to obtain the diamond and be outside the vault, whereas in $\mathcal{P}_2$ she wishes to be outside the vault *possibly* having obtained the diamond.

Let $\pi_1' = \mathsf{move}; \mathsf{if}\ Kr\ \mathsf{then\ take\_right\ else\ take\_left}; \mathsf{move}$ and note that $\pi_1 = \mathsf{flick}; \pi_1'$. Using the strong translation of π_1, we get $\mathcal{M}_0 \models [\![\pi_1]\!]_s \phi_g$ iff $\mathcal{M}_0 \models \langle \mathsf{flick} \rangle \top \wedge [\mathsf{flick}] [\![\pi_1']\!]_s \phi_g$. As $\mathcal{M}_0 \models \langle \mathsf{flick} \rangle \top$ does not hold, π_1 is not a solution. This is expected, since flicking the switch in the initial state is not an applicable action. Verifying that π_2 is a strong solution to $\mathcal{P}_2$ amounts to checking if $\mathcal{M}_0 \models [\![\pi_2]\!]_s \tilde{K}d \wedge \neg v$ which translates to

$$\mathcal{M}_0 \models \langle \mathsf{move} \rangle \top \wedge [\mathsf{move}] \Big(\langle \mathsf{take\_right} \rangle \top \wedge [\mathsf{take\_right}] \Big(\langle \mathsf{move} \rangle \top \wedge [\mathsf{move}] \Big(\tilde{K}d \wedge \neg v \Big) \Big) \Big)$$

With the same approach we can conclude that π_2 is not a solution to $\mathcal{P}_1$, π_3 is a weak solution to $\mathcal{P}_1$ and $\mathcal{P}_2$, and π_4 is a strong solution to $\mathcal{P}_1$ and $\mathcal{P}_2$.

4 Plan Synthesis

We now show how to synthesise conditional plans for solving planning problems. To synthesise plans, we need a mechanism for coming up with formulas characterising information cells for if-then-else constructs to branch on. Inspired by [7, 8], these are developed in the following. Proofs are omitted, as they are straightforward and similar to proofs in the aforementioned references.

Definition 10 (Characterising Formulas). *Let $\mathcal{M} = (W, \sim, V)$ denote an information cell on $\mathcal{L}_{\mathrm{DEL}}(P)$. We define for all $w \in W$ a formula ϕ_w by: $\phi_w = \bigwedge_{p \in V(w)} p \wedge \bigwedge_{p \in P - V(w)} \neg p$. We define the* characterising formula *for $\mathcal{M}$, $\delta_{\mathcal{M}}$, as follows: $\delta_{\mathcal{M}} = K(\bigwedge_{w \in W} \tilde{K}\phi_w \wedge K \bigvee_{w \in W} \phi_w)$.*

Lemma 1. *Let $\mathcal{M}$ be an information cell on $\mathcal{L}_{\mathrm{DEL}}(P)$. Then for all epistemic models $\mathcal{M}' = (W', \sim', V')$ and all $w' \in W'$ we have that $(\mathcal{M}', w') \models \delta_{\mathcal{M}}$ if and only if there exists a $w \in \mathcal{D}(\mathcal{M})$ such that $(\mathcal{M}, w) \leftrightarroweq (\mathcal{M}', w')$.*[3]

For purposes of synthesis, we use the product update solely on non-pointed epistemic and event models. Lemma 2 shows that satisfaction of the dynamic modality for non-pointed event models in non-pointed epistemic models relates to the product update in the obvious way.

Lemma 2. *Let $\mathcal{M}$ be an epistemic model and $\mathcal{E}$ an event model. Then $\mathcal{M} \models [\mathcal{E}]\phi$ iff $\mathcal{M} \otimes \mathcal{E} \models \phi$.*

Proof. $\mathcal{M} \models [\mathcal{E}]\phi \;\Leftrightarrow\;$ for all $w \in \mathcal{D}(\mathcal{M}) : \mathcal{M}, w \models [\mathcal{E}]\phi \Leftrightarrow$
for all $w \in \mathcal{D}(\mathcal{M}) : \mathcal{M}, w \models \bigwedge_{e \in \mathcal{D}(\mathcal{E})} [\mathcal{E}, e]\phi \Leftrightarrow$
for all $(w, e) \in \mathcal{D}(\mathcal{M}) \times \mathcal{D}(\mathcal{E}) : \mathcal{M}, w \models [\mathcal{E}, e]\phi \Leftrightarrow$
for all $(w, e) \in \mathcal{D}(\mathcal{M}) \times \mathcal{D}(\mathcal{E}) : \mathcal{M}, w \models pre(e)$ implies $\mathcal{M} \otimes \mathcal{E}, (w, e) \models \phi \Leftrightarrow$
for all $(w, e) \in D(\mathcal{M} \otimes \mathcal{E}) : \mathcal{M} \otimes \mathcal{E}, (w, e) \models \phi \;\Leftrightarrow\; \mathcal{M} \otimes \mathcal{E} \models \phi$.

4.1 Planning Trees

When synthesising plans, we explicitly construct the search space of the problem as a labelled AND-OR tree, a familiar model for planning under uncertainty [13]. Our AND-OR trees are called *planning trees*.

Definition 11. *A* planning tree *is a finite, labelled* AND-OR *tree in which each node n is labelled by an epistemic model $\mathcal{M}(n)$, and each edge (n, m) leaving an* OR*-node is labelled by an event model $\mathcal{E}(n, m)$.*

Planning trees for planning problems $\mathcal{P} = (\mathcal{M}_0, \mathsf{A}, \phi_g)$ are constructed as follows. Let the initial planning tree T_0 consist of just one OR-node $root(T_0)$ with $\mathcal{M}(root(T_0)) = \mathcal{M}_0$ (the root labels the initial state). A planning tree for $\mathcal{P}$ is then any tree that can be constructed from T_0 by repeated applications of the following non-deterministic tree expansion rule.

Definition 12 (Tree Expansion Rule). *Let T be a planning tree for a planning problem $\mathcal{P} = (\mathcal{M}_0, \mathsf{A}, \phi_g)$. The tree expansion rule is defined as follows. Pick an* OR*-node n in T and an event model $\mathcal{E} \in \mathsf{A}$ applicable in $\mathcal{M}(n)$ with the proviso that $\mathcal{E}$ does not label any existing outgoing edges from n. Then:*

1. *Add a new node m to T with $\mathcal{M}(m) = \mathcal{M}(n) \otimes \mathcal{E}$, and add an edge (n, m) with $\mathcal{E}(n, m) = \mathcal{E}$.*

[3] Here $(\mathcal{M}, w) \leftrightarroweq (\mathcal{M}', w)$ denotes that $(\mathcal{M}, w)$ and $(\mathcal{M}', w)$ are bisimilar according to the standard notion of bisimulation on pointed epistemic models.

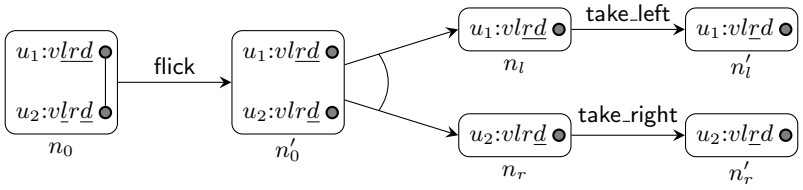

Fig. 4. Planning tree for a variant of the Pink Panther problem.

2. *For each information cell $\mathcal{M}'$ in $\mathcal{M}(m)$, add an* OR*-node m' with $\mathcal{M}(m') = \mathcal{M}'$ and add the edge (m, m').*

The tree expansion rule is similar in structure to—and inspired by—the expansion rules used in tableau calculi, e.g. for modal and description logics [14]. Note that the expansion rule applies only to OR-nodes, and that an applicable event model can only be used once at each node.

Considering single-agent planning a two-player game, a useful analogy for planning trees are game trees. At an OR-node n, the agent gets to pick any applicable action $\mathcal{E}$ it pleases, winning if it ever reaches an epistemic model in which the goal formula holds (see the definition of solved nodes further below). At an AND-node m, the environment responds by picking one of the information cells of $\mathcal{M}(m)$—which of the distinguishable outcomes is realised when performing the action.

Example 5. In Fig. 4 is a planning tree for a variant of the Pink Panther planning problem, this one where the thief is already inside the vault. The root is n_0. Three applications of the tree expansion rule have been made, the labels on edges indicating the chosen action. n_0, n_l and n_r are OR-nodes. n'_0, n'_l and n'_r are AND-nodes. The child nodes of the latter two AND-nodes have been omitted, as their information cell is the same as that of their parent nodes. Pay particular attention to how flick reveals the location of the diamond. In the initial state, $\mathcal{M}(n_0) \models \neg Kr \wedge \neg K\neg r$, while $\mathcal{M}(n'_0) \models Kr \vee K\neg r$, $\mathcal{M}(n_l) \models K\neg r$ and $\mathcal{M}(n_r) \models Kr$.

Without restrictions on the tree expansion rule, even very simple planning problems might be infinitely expanded. Finiteness of trees (and therefore termination) is ensured by the following blocking condition.

$\mathcal{B}_1$ The tree expansion rule may not be applied to a node n for which there exists an ancestor node m with $\mathcal{M}(m) \leftrightarroweq \mathcal{M}(n)$.[4]

A planning tree for a planning problem $\mathcal{P}$ is called $\mathcal{B}_1$-*saturated* if no more expansions are possible satisfying condition $\mathcal{B}_1$.

Lemma 3 (Termination). *Any procedure that builds a $\mathcal{B}_1$-saturated planning tree for a planning problem $\mathcal{P}$ by repeated application of the tree expansion rule terminates.*

[4] Here $\mathcal{M}(m) \leftrightarroweq \mathcal{M}(n)$ denotes that $\mathcal{M}(m)$ and $\mathcal{M}(n)$ are bisimilar according to the standard notion of bisimulation between non-pointed epistemic models.

Proof. Planning trees built by repeated application of the tree expansion rule are finitely branching: the action library is finite, and every epistemic model has only finitely many information cells. Furthermore, condition $\mathcal{B}_1$ ensures that no branch has infinite length: there only exists finitely many mutually non-bisimilar epistemic models over any given finite set of propositional symbols [10]. König's Lemma now implies finiteness of the planning tree.

Definition 13 (Solved Nodes). *Let T be any (not necessarily saturated) planning tree for a planning problem $\mathcal{P} = (\mathcal{M}_0, \mathsf{A}, \phi_g)$. By recursive definition, a node n in T is called* solved *if one of the following holds:*

- $\mathcal{M}(n) \models \phi_g$ *(the node satisfies the goal formula).*
- n *is an* OR*-node having at least one solved child.*
- n *is an* AND*-node having all its children solved.*

Continuing the game tree analogy, we see that a solved node corresponds is one for which there exists a winning strategy. Regardless of the environment's choice, the agent can achieve its goal. Let T and $\mathcal{P}$ be as above. Below we show that when a node n is solved, it is possible to construct a (strong) solution to the planning problem $(\mathcal{M}(n), \mathsf{A}, \phi_g)$. In particular, if the root node is solved, a strong solution to $\mathcal{P}$ can be constructed. As it is never necessary to expand a solved node, nor any of its descendants, we can augment the blocking condition $\mathcal{B}_1$ in the following way.

$\mathcal{B}_2$ The tree expansion rule may not be applied to a node n if one of the following holds: 1) n is solved; 2) n has a solved ancestor; 3) n has an ancestor node m with $\mathcal{M}(m) \leftrightarroweq \mathcal{M}(n)$.

In the following, we will assume that all planning trees have been built according to $\mathcal{B}_2$. One consequence is that a solved OR-node has exactly one solved child. We make use of this in the following definition.

Definition 14 (Plans for Solved Nodes). *Let T be any planning tree for $\mathcal{P} = (\mathcal{M}_0, \mathsf{A}, \phi_g)$. For each solved node n in T, a plan $\pi(n)$ is defined recursively by:*

- *if $\mathcal{M}(n) \models \phi_g$, then $\pi(n) = \mathsf{skip}$.*
- *if n is an* OR*-node and m its solved child, then $\pi(n) = \mathcal{E}(n, m); \pi(m)$.*
- *if n is an* AND*-node with children $m_1, \ldots, m_k$, then $\pi(n) =$*
  $\mathsf{if}\ \delta_{\mathcal{M}(m_1)}\ \mathsf{then}\ \pi(m_1)\ \mathsf{else\ if}\ \ \delta_{\mathcal{M}(m_2)}\ \mathsf{then}\ \pi(m_2)\ \mathsf{else}\ \cdots\ \mathsf{if}\ \delta_{\mathcal{M}(m_k)}\ \mathsf{then}\ \pi(m_k)$

Example 6. For the goal of achieving the diamond, $\phi_g = d$, we have that the root n_0 of the planning tree of Figure 4 is solved, as both n_l' and n_r' satisfy the goal formula. Definition 14 gives us $\pi(n_0) = \mathsf{flick}; \mathsf{if}\ \delta_{\mathcal{M}(n_l)}\ \mathsf{then\ take\_left}; \mathsf{skip\ else\ if}\ \delta_{\mathcal{M}(n_r)}\ \mathsf{then\ take\_right}; \mathsf{skip}$. This plan can easily be shown to be a strong solution to the planning problem of achieving d from the initial state $\mathcal{M}(n_0)$. In our soundness result below, we show that plans of solved roots are always strong solutions to their corresponding planing problems.

Theorem 1 (Soundness). *Let T be a planning tree for a problem $\mathcal{P}$ such that $root(T)$ is solved. Then $\pi(root(T))$ is a strong solution to $\mathcal{P}$.*

Proof. We need to prove that $\pi(root(T))$ is a strong solution to $\mathcal{P}$, that is, $\mathcal{M}_0 \models [\![\pi(root(T))]\!]_s\, \phi_g$. Since $\mathcal{M}_0$ is the label of the root, this can be restated as $\mathcal{M}(root(T)) \models [\![\pi(root(T))]\!]_s\, \phi_g$. To prove this fact, we will prove the following stronger claim:

– For each solved node n in T, $\mathcal{M}(n) \models [\![\pi(n)]\!]_s\, \phi_g$.

We prove this by induction on the height of n. The base case is when n is a leaf. Since n is solved, we must have $\mathcal{M}(n) \models \phi_g$. In this case $\pi(n) = \mathsf{skip}$. From $\mathcal{M}(n) \models \phi_g$ we can conclude $\mathcal{M}(n) \models [\![\mathsf{skip}]\!]_s\, \phi_g$, that is, $\mathcal{M}(n) \models [\![\pi(n)]\!]_s\, \phi_g$. This covers the base case. For the induction step, assume that for all solved nodes m of height $< h$, $\mathcal{M}(m) \models [\![\pi(m)]\!]_s\, \phi_g$. Let n be an arbitrary solved node n of height h. We then need to show $\mathcal{M}(n) \models [\![\pi(n)]\!]_s\, \phi_g$. We have two cases to consider, depending on whether n is an AND- or an OR-node.

Case 1: n is an AND-node. Let $m_1, \ldots, m_k$ be the children of n. By definition, all of these are solved. We have $\pi(n) = \mathsf{if}\ \delta_{\mathcal{M}(m_1)}\ \mathsf{then}\ \pi(m_1)\ \mathsf{else\ if}\ \delta_{\mathcal{M}(m_2)}\ \mathsf{then}\ \pi(m_2)\ \mathsf{else}\ \cdots\ \mathsf{if}\ \delta_{\mathcal{M}(m_k)}\ \mathsf{then}\ \pi(m_k)\ \mathsf{else\ skip}$. The induction hypothesis gives us $\mathcal{M}(m_i) \models [\![\pi(m_i)]\!]_s\, \phi_g$ for all $i = 1, \ldots, k$.

Claim (1). $\mathcal{M}(n) \models \delta_{\mathcal{M}(m_i)} \rightarrow [\![\pi(m_i)]\!]_s\, \phi_g$, for all $i = 1, \ldots, k$.

Proof. Let $w \in \mathcal{D}(\mathcal{M}(n))$ be chosen arbitrarily. We then need to prove that if $\mathcal{M}(n), w \models \delta_{\mathcal{M}(m_i)}$ then $\mathcal{M}(n), w \models [\![\pi(m_i)]\!]_s\, \phi_g$. Assuming $\mathcal{M}(n), w \models \delta_{\mathcal{M}(m_i)}$, we get from Lemma 1 that there must be a $w' \in \mathcal{D}(\mathcal{M}(m_i))$ such that $\mathcal{M}(m_i), w' \leftrightarrows \mathcal{M}(n), w$. Since $\mathcal{M}(m_i) \models [\![\pi(m_i)]\!]_s\, \phi_g$, in particular we get $\mathcal{M}(m_i), w' \models [\![\pi(m_i)]\!]_s\, \phi_g$, and thus $\mathcal{M}(n), w \models [\![\pi(m_i)]\!]_s\, \phi_g$.

Claim (2). $\mathcal{M}(n) \models \bigvee_{i=1,\ldots,k} \delta_{\mathcal{M}(m_i)}$.

Proof. Let $w \in \mathcal{D}(\mathcal{M}(n))$ be chosen arbitrarily. We then need to prove that $\mathcal{M}(n), w \models \vee_{i=1,\ldots,k} \delta_{\mathcal{M}(m_i)}$. Since $w \in \mathcal{D}(\mathcal{M}(n))$ it must belong to one of the information cells of $\mathcal{M}(n)$, that is, $w \in \mathcal{D}(\mathcal{M}(m_j))$ for some j. Thus $\mathcal{M}(n), w \leftrightarrows \mathcal{M}(m_j), w$. From Lemma 1 we then get $\mathcal{M}(n), w \models \delta_{\mathcal{M}(m_j)}$, and thus $\mathcal{M}(n), w \models \vee_{i=1,\ldots,k} \delta_{\mathcal{M}(m_i)}$.

From (1) and (2), we now get:

$$\mathcal{M}(n) \models \bigwedge_{i=1,\ldots,k} (\delta_{\mathcal{M}(m_i)} \rightarrow [\![\pi(m_i)]\!]_s\, \phi_g) \wedge \bigvee_{i=1,\ldots,k} \delta_{\mathcal{M}(m_i)} \Rightarrow$$

$$\mathcal{M}(n) \models \bigwedge_{i=1,\ldots,k} \big(\delta_{\mathcal{M}(m_i)} \wedge \bigwedge_{j=1,\ldots,i-1} \neg\delta_{\mathcal{M}(m_j)} \rightarrow [\![\pi(m_i)]\!]_s\, \phi_g\big) \wedge \big(\bigwedge_{i=1,\ldots,k} \neg\delta_{\mathcal{M}(m_i)} \rightarrow [\![\mathsf{skip}]\!]_s\, \phi_g\big) \Rightarrow$$

$$\begin{aligned}
\mathcal{M}(n) \models\ & (\delta_{\mathcal{M}(m_1)} \rightarrow [\![\pi(m_1)]\!]_s\, \phi_g)\ \wedge (\neg\delta_{\mathcal{M}(m_1)} \rightarrow \\
& (\delta_{\mathcal{M}(m_2)} \rightarrow [\![\pi(m_2)]\!]_s\, \phi_g) \wedge (\neg\delta_{\mathcal{M}(m_2)} \rightarrow \\
& \quad \cdots \\
& \quad\ (\delta_{\mathcal{M}(m_k)} \rightarrow [\![\pi(m_k)]\!]_s\, \phi_g) \wedge (\neg\delta_{\mathcal{M}(m_k)} \rightarrow \\
& \quad\quad [\![\mathsf{skip}]\!]_s\, \phi_g) \cdots) \Rightarrow \\
\mathcal{M}(n) \models\ & [\![\mathsf{if}\ \delta_{\mathcal{M}(m_1)}\ \mathsf{then}\ \pi(m_1)\ \mathsf{else}
\end{aligned}$$

$$\mathsf{if}\ \delta_{\mathcal{M}(m_2)}\ \mathsf{then}\ \pi(m_2)\ \mathsf{else}$$
$$\ldots$$
$$\mathsf{if}\ \delta_{\mathcal{M}(m_k)}\ \mathsf{then}\ \pi(m_k)\ \mathsf{else}$$
$$\mathsf{skip}]\!]\ \phi_g \Rightarrow$$
$\mathcal{M}(n) \models [\![\pi(n)]\!]_s\, \phi_g$.

Case 2: n **is an or-node.** Here we have $\pi(n) = \mathcal{E}(n,m);\pi(m)$ for the solved child m of n. The induction hypothesis gives $\mathcal{M}(m) \models [\![\pi(m)]\!]_s\, \phi_g$, and hence $\mathcal{M}(m) \models K\,[\![\pi(m)]\!]_s\, \phi_g$. We now show $\mathcal{M}(n) \models [\![\pi(n)]\!]_s\, \phi_g$. Since, by definition, $\mathcal{M}(m) = \mathcal{M}(n) \otimes \mathcal{E}(n,m)$, we get $\mathcal{M}(n) \otimes \mathcal{E}(n,m) \models K\,[\![\pi(m)]\!]_s\, \phi_g$. We can now apply Lemma 2 to conclude $\mathcal{M}(n) \models [\mathcal{E}(n,m)]\,K\,[\![\pi(m)]\!]_s\, \phi_g$. By definition, $\mathcal{E}(n,m)$ must be applicable in $\mathcal{M}(n)$, that is, $\mathcal{M}(n) \models \langle\mathcal{E}(n,m)\rangle\,\top$. Thus we now have $\mathcal{M}(n) \models \langle\mathcal{E}(n,m)\rangle\,\top \wedge [\mathcal{E}(n,m)]\,K\,[\![\pi(m)]\!]_s\, \phi_g$. Using Definition 8, we can rewrite this as $\mathcal{M}(n) \models [\![\mathcal{E}(n,m)]\!]_s\, [\![\pi(m)]\!]_s\, \phi_g$. Using Definition 8 again, we get $\mathcal{M}(n) \models [\![\mathcal{E}(n,m);\pi(m)]\!]_s\, \phi_g$, and thus finally $\mathcal{M}(n) \models [\![\pi(n)]\!]_s\, \phi_g$, as required.

Theorem 2 (Completeness). *If there is a strong solution to the planning problem $\mathcal{P} = (\mathcal{M}_0, \mathsf{A}, \phi_g)$, then a planning tree T for $\mathcal{P}$ can be constructed, such that root(T) is solved.*

Proof. We first prove the following claim.

Claim (1). If ($\mathsf{if}\ \phi\ \mathsf{then}\ \pi_1\ \mathsf{else}\ \pi_2$) is a strong solution to $\mathcal{P} = (\mathcal{M}_0, \mathsf{A}, \phi_g)$, then so is π_1 or π_2.

Proof. Assume ($\mathsf{if}\ \phi\ \mathsf{then}\ \pi_1\ \mathsf{else}\ \pi_2$) is a strong solution to $(\mathcal{M}_0, \mathsf{A}, \phi_g)$, that is, $\mathcal{M}_0 \models [\![\mathsf{if}\ \phi\ \mathsf{then}\ \pi_1\ \mathsf{else}\ \pi_2]\!]_s\, \phi_g$. Then, by definition, $\mathcal{M}_0 \models (\phi \to [\![\pi_1]\!]_s\, \phi_g) \wedge (\neg\phi \to [\![\pi_2]\!]_s\, \phi_g)$. Since $\mathcal{M}_0$ is an information cell, and ϕ is a K-formula, we must have either $\mathcal{M}_0 \models \phi$ or $\mathcal{M}_0 \models \neg\phi$. Thus we get that either $\mathcal{M}_0 \models [\![\pi_1]\!]_s\, \phi_g$ or $\mathcal{M}_0 \models [\![\pi_2]\!]_s\, \phi_g$, as required.

Note that we have $[\![\mathsf{skip};\pi]\!]_s\, \phi_g = [\![\mathsf{skip}]\!]_s\, ([\![\pi]\!]_s\, \phi_g) = [\![\pi]\!]_s\, \phi_g$. Thus, we can without loss of generality assume that no plan contains a subexpression of the form $\mathsf{skip};\pi$. The length of a plan π, denoted $|\pi|$, is defined recursively by: $|\mathsf{skip}| = 1$; $|\mathcal{E}| = 1$; $|\mathsf{if}\ \phi\ \mathsf{then}\ \pi_1\ \mathsf{else}\ \pi_2| = |\pi_1| + |\pi_2|$; $|\pi_1;\pi_2| = |\pi_1| + |\pi_2|$.

Claim (2). Let π be a strong solution to $\mathcal{P} = (\mathcal{M}_0, \mathsf{A}, \phi_g)$ with $|\pi| \geq 2$. Then there exists a strong solution of the form $\mathcal{E};\pi'$ with $|\mathcal{E};\pi'| \leq |\pi|$.

Proof. Proof by induction on $|\pi|$. The base case is $|\pi| = 2$. We have two cases, $\pi = \mathsf{if}\ \phi\ \mathsf{then}\ \pi_1\ \mathsf{else}\ \pi_2$ and $\pi = \pi_1;\pi_2$, both with $|\pi_1| = |\pi_2| = 1$. If π is the latter, it already has desired the form. If $\pi = \mathsf{if}\ \phi\ \mathsf{then}\ \pi_1\ \mathsf{else}\ \pi_2$ we have by Claim 1 that either π_1 or π_2 is a strong solution to $\mathcal{P}$. Thus also either $\pi_1;\mathsf{skip}$ or $\pi_2;\mathsf{skip}$ is a strong solution to $\mathcal{P}$, and both of these have length $|\pi|$. This completes the base case. For the induction step, we assume that if π', with $|\pi'| < l$, is a strong solution to a planning problem $\mathcal{P}'$, then there exists is a strong solution of the form $(\mathcal{E};\pi'')$, with $|\mathcal{E};\pi''| \leq |\pi'|$. Now consider a plan π of length l which is a strong solution to $\mathcal{P}$. We again have two cases to consider,

$\pi = \mathsf{if}\ \phi\ \mathsf{then}\ \pi_1\ \mathsf{else}\ \pi_2$ and $\pi = \pi_1; \pi_2$. If $\pi = \pi_1; \pi_2$ is a strong solution to $\mathcal{P}$, then π_1 is a strong solution to the planning problem $\mathcal{P}' = (\mathcal{M}_0, \mathsf{A}, [\![\pi_2]\!]_s\, \phi_g)$, as $\mathcal{M}_0 \models [\![\pi_1; \pi_2]\!]_s\, \phi_g \Leftrightarrow \mathcal{M}_0 \models [\![\pi_1]\!]_s\, [\![\pi_2]\!]_s\, \phi_g$. Clearly $|\pi_1| < l$, so the induction hypothesis gives that there is a strong solution $(\mathcal{E}; \pi_1')$ to $\mathcal{P}'$, with $|\mathcal{E}; \pi_1'| \leq |\pi_1|$. Then, $a; \pi_1'; \pi_2$ is a strong solution to $\mathcal{P}$ and we have $|a; \pi_1'; \pi_2| = |a; \pi_1'| + |\pi_2| \leq |\pi_1| + |\pi_2| = |\pi|$. If $\pi = \mathsf{if}\ \phi\ \mathsf{then}\ \pi_1\ \mathsf{else}\ \pi_2$ is a strong solution to $\mathcal{P}$, then we have by Claim 1 that either π_1 or π_2 is a strong solution to $\mathcal{P}$. With both $|\pi_1| < l$ and $|\pi_2| < l$, the induction hypothesis gives the existence a strong solution $\mathcal{E}; \pi'$, with $|\mathcal{E}; \pi'| \leq |\pi|$. This completes the proof of the claim.

We now prove the theorem by induction on $|\pi|$, where π is a strong solution to $\mathcal{P} = (\mathcal{M}_0, \mathsf{A}, \phi_g)$. We need to prove that there exists a planning tree T for $\mathcal{P}$ in which the root is solved. Let T_0 denote the planning tree for $\mathcal{P}$ only consisting of its root node with label $\mathcal{M}_0$. The base case is when $|\pi| = 1$. Here, we have two cases, $\pi = \mathsf{skip}$ and $\pi = \mathcal{E}$. In the first case, the planning tree T_0 already has its root solved, since $\mathcal{M}_0 \models [\![\mathsf{skip}]\!]_s\, \phi_g \Leftrightarrow \mathcal{M}_0 \models \phi_g$. In the second case $\pi = \mathcal{E}$. Since π is a strong solution to $\mathcal{P}$, we have $\mathcal{M}_0 \models [\![\mathcal{E}]\!]_s\, \phi_g$, that is, $\mathcal{M}_0 \models \langle \mathcal{E} \rangle \top \wedge [\mathcal{E}]\, K \phi_g$. Thus $\mathcal{E}$ is applicable in $\mathcal{M}_0$ meaning that we can apply the tree expansion rule to T_0, which will produce an AND-node m with $a(root(T_0), m) = \mathcal{E}$ and $\mathcal{M}(m) = \mathcal{M}_0 \otimes \mathcal{E}$. Call the expanded tree T_1. Since we have $\mathcal{M}_0 \models [\mathcal{E}] K \phi_g$, Lemma 2 gives us $\mathcal{M}_0 \otimes \mathcal{E} \models K \phi_g$, that is, $\mathcal{M}(m) \models K \phi_g$, and hence $\mathcal{M}(m) \models \phi_g$. This implies that $\mathcal{M}(m)$ and thus $root(T_1)$ is solved. The base case is hereby completed.

For the induction step, assume that a planning tree with solved root can be constructed for problems with strong solutions of length $< l$. Let π be a strong solution to $\mathcal{P}$ with $|\pi| = l$. By Claim 2, there exists a strong solution of the form $\mathcal{E}; \pi'$ with $|\mathcal{E}; \pi'| \leq |\pi|$. As $\mathcal{M}_0 \models [\![\mathcal{E}; \pi']\!]_s\, \phi_g \Leftrightarrow \mathcal{M}_0 \models [\![\mathcal{E}]\!]_s\, [\![\pi']\!]_s\, \phi_g \Leftrightarrow \mathcal{M}_0 \models \langle \mathcal{E} \rangle \top \wedge [\mathcal{E}] K([\![\pi']\!]_s\, \phi_g)$, the tree expansion rule can be applied by picking $\mathcal{E}$ and $\mathcal{M}_0$. This produces the AND-node m with $\mathcal{E}(n, m) = \mathcal{E}$ and $\mathcal{M}(m) = \mathcal{M}_0 \otimes \mathcal{E}$. $m_1, \ldots, m_k$ are the children of m, and $\mathcal{M}(m_i) = \mathcal{M}_i$ the information cells in $\mathcal{M}(m)$. From $\mathcal{M}_0 \models [\mathcal{E}] K([\![\pi']\!]_s\, \phi_g)$ we get $\mathcal{M}_0 \otimes \mathcal{E} \models K\, [\![\pi']\!]_s\, \phi_g$, using Lemma 2. This implies $\mathcal{M}_i \models K\, [\![\pi']\!]_s\, \phi_g$, and hence $\mathcal{M}_i \models [\![\pi']\!]_s\, \phi_g$, for each information cell $\mathcal{M}_i$ of $\mathcal{M}(m) = \mathcal{M}_0 \otimes \mathcal{E}$. Thus π' must be a strong solution to each of the planning problems $\mathcal{P}_i = (\mathcal{M}_i, \mathcal{A}, \phi_g)$. As $|\pi'| < |\mathcal{E}; \pi'| \leq l$, the induction hypothesis gives that planning trees T_i with solved roots can be constructed for each $\mathcal{P}_i$. Let T denote T_0 expanded with $m, m_1, \ldots, m_k$, and each T_i be the subtree rooted at m_i. Then each of the nodes m_i are solved in T, and in turn both m and $root(T)$ are solved.

4.2 Strong Planning Algorithm

With all the previous in place, we now have an algorithm for synthesising strong solutions for planning problems $\mathcal{P}$, given as follows.

StrongPlan($\mathcal{P}$)
1 Let T be the plan. tree only consisting of $root(T)$ labelled by the init. state of $\mathcal{P}$.
2 Repeatedly apply the tree expansion rule of $\mathcal{P}$ to T until it is $\mathcal{B}_2$-saturated.
3 If $root(T)$ is solved, return $\pi(root(T))$, otherwise return FAIL.

Theorem 3. StrongPlan($\mathcal{P}$) *is a terminating, sound and complete algorithm for producing strong solutions to planning problems. Soundness means that if* StrongPlan($\mathcal{P}$) *returns a plan, it is a strong solution to* $\mathcal{P}$*. Completeness means that if* $\mathcal{P}$ *has a strong solution,* StrongPlan($\mathcal{P}$) *will return one.*

Proof. Termination comes from Lemma 3 (with $\mathcal{B}_1$ replaced by the stronger condition $\mathcal{B}_2$), soundness from Theorem 1 and completeness from Theorem 2 (given any two saturated planning trees T_1 and T_2 for the same planning problem, the root node of T_1 is solved iff the root node of T_2 is).

4.3 Weak Planning Algorithm

With few changes, the machinery already in place gives an algorithm for synthesising weak solutions. Rather than requiring all children of an AND-node be solved, we require only one. This corresponds to the notion of weak, defined in Definition 8. Only one possible execution need lead to the goal.

Definition 15 (Weakly Solved Nodes). *A node n is called* weakly solved *if either $\mathcal{M}(n) \models \phi_g$ or n has at least one weakly solved child.*

We keep the tree expansion rule, but make use of a new blocking condition $\mathcal{B}_3$ using Definition 15 rather than Definition 13.

Definition 16 (Plans for Weakly Solved Nodes). *Let T be any planning tree for $\mathcal{P} = (\mathcal{M}_0, \mathsf{A}, \phi_g)$. For each weakly solved node n in T, a plan $\pi_w(n)$ is defined recursively by:*

- *if $\mathcal{M}(n) \models \phi_g$, then $\pi_w(n) = \mathsf{skip}$*
- *if n is an* OR*-node and m its weakly solved child, then $\pi_w(n) = \mathcal{E}(n, m); \pi_w(m)$*
- *if n is an* AND*-node and m its weakly solved child, then $\pi_w(n) = \pi_w(m)$*

The algorithm for weak planning is defined as follows.

WeakPlan($\mathcal{P}$)
1 Let T be the plan. tree only consisting of $root(T)$ labelled by the init. state of $\mathcal{P}$.
2 Repeatedly apply the tree expansion rule of $\mathcal{P}$ to T until it is $\mathcal{B}_3$-saturated.
3 If $root(T)$ is weakly solved, return $\pi_w(root(T))$, otherwise return FAIL.

Theorem 4. WeakPlan($\mathcal{P}$) *is a terminating, sound and complete algorithm for producing weak solutions to planning problems.*

5 Related and Future Work

In this paper, we have presented a syntactic characterisation of weak and strong solutions to epistemic planning problems, that is, we have characterised solutions as formulas. [10] takes a semantic approach to strong solutions for epistemic planning problems. In their work plans are sequences of actions, requiring conditional choice of actions at different states to be encoded in the action structure itself. We represent choice explicitly, using a language of conditional plans.

An alternative to our approach of translating conditional plans into formulas of DEL would be to translate plans directly into (complex) event models. This is the approach taken in [3], where they have a language of epistemic programs similar to our language of plans (modulo the omission of ontic actions). Using this approach in a planning setting, one could translate each possible plan π into the corresponding event model $\mathcal{E}(\pi)$, check its applicability, and check whether $\mathcal{M}_0 \otimes \mathcal{E}(\pi) \models \phi_g$ (the goal is satisfied in the product update of the initial state with the event model). However, even for a finite action library, there are infinitely many distinct plans, and thus infinitely many induced event models to consider when searching for a solution. To construct a terminating planning algorithm with this approach, one would still have to limit the plans considered (e.g. by using characterising formulas), and also develop a more involved loop-checking mechanism working at the level of plans. Furthermore, our approach more obviously generalises to algorithms for replanning, which is current work.

The meaningful plans of [15, chap. 2] are reminiscent of the work in this paper. Therein, plan verification is cast as validity of an EDL-consequence in a given system description. Like us, they consider single-agent scenarios, conditional plans, applicability and incomplete knowledge in the initial state. Unlike us, they consider only deterministic actions. In the multi-agent treatment [15, chap. 4], action laws are translated to a fragment of DEL with only public announcements and public assignments, making actions singleton event models. This means foregoing nondeterminism and therefore sensing actions.

Planning problems in [16] are solved by producing a sequence of pointed event models where an external variant of applicability (called *possible at*) is used. Using such a formulation means outcomes of actions are fully determined, making conditional plans and weak solutions superfluous. As noted by the authors, and unlike our framework, their approach does not consider factual change. We stress that [10, 16, 15] all consider the multi-agent setting which we have not treated here.

In our work so far, we haven't treated the problem of where domain formulations come from, assuming just that they are given. Standardised description languages are vital if modal logic-based planning is to gain wide acceptance in the planning community. Recent work worth noting in this area includes [6], which presents a specification language for the multi-agent belief case.

As suggested by our construction of planning trees, there are several connections between our approach and two-player imperfect information games. First, product updates imply perfect recall [9]. Second, when the game is at a node belonging to an information set, the agent knows a proposition only if it holds throughout the information set; corresponding to our use of information cells. Finally, the strong solutions we synthesise are very similar to mixed strategies. A strong solution caters to any information cell (contingency) it may bring about, by selecting exactly one sub-plan for each [2].

Our work naturally relates to [13], where the notions of strong and weak solutions are found. Their belief states are sets of states which may be partioned by observation variables. Our partition of epistemic models into infor-

mation cells follows straight from the definition of product update. A clear advantage in our approach is that actions encode both nondetermism and partial observability. [17] shows that for conditional planning (prompted by nondeterministic actions) in partially observable domains the *plan existence problem* is 2-EXP-complete (plans must succeed with probability 1; i.e. be strong solutions). StrongPlan($\mathcal{P}$) implicitly answers the same question for $\mathcal{P}$ (it gives a strong solution if one exists). Reductions between the two decision problem variants would give a complexity measure of our approach, and also formally link conditional epistemic planning with the approaches used in automated planning.

We would like to do plan verification and synthesis in the multi-agent settings. We believe that generalising the notions introduced in this paper to multi-pointed epistemic and event models are key. Plan synthesis in the multi-agent setting is undecidable [10], but considering restricted classes of actions as is done in [16] seems a viable route for achieving decidable multi-agent planning. Another interesting area is to consider modalities such as plausibility and preferences. This would allow an agent to plan for (perhaps only) the most likely outcomes of its own actions and the preferred actions taken by other agents in the system. This could then be combined with the possibility of doing replanning, as mentioned above.

References

1. Aucher, G.: An internal version of epistemic logic. Studia Logica 94(1), 1–22 (2010)
2. Aumann, R., Hart, S. (eds.): Handbook of Game Theory with Economic Applications. Elsevier (1992)
3. Baltag, A., Moss, L.S.: Logics for Epistemic Programs. Synthese 139, 165–224 (2004)
4. Baltag, A., Moss, L.S., Solecki, S.: The logic of public announcements and common knowledge and private suspicions. In: TARK-98. pp. 43–56 (1998)
5. Baltag, A., Smets, S.: A qualitative theory of dynamic interactive belief revision. In: Bonanno, G., van der Hoek, W., Wooldridge, M. (eds.) Logic and the Foundations of Game and Decision Theory (LOFT7). Texts in Logic and Games, vol. 3, pp. 13–60. Amsterdam University Press (2008)
6. Baral, C., Gelfond, G., Pontelli, E., Son, T.C.: An action language for reasoning about beliefs in multi-agent domains. In: Proceedings of the 14th International Workshop on Non-Monotonic Reasoning (2012)
7. Barwise, J., Moss, L.: Vicious circles. CSLI Publications (1996)
8. van Benthem, J.: Dynamic odds and ends. Technical Report ML-1998-08, University of Amsterdam (1998)
9. van Benthem, J.: Games in dynamic-epistemic logic. Bulletin of Economic Research 53(4), 219–48 (2001)
10. Bolander, T., Andersen, M.B.: Epistemic planning for single- and multi-agent systems. Journal of Applied Non-Classical Logics 21, 9–34 (2011)
11. van Ditmarsch, H., Kooi, B.: Semantic results for ontic and epistemic change. In: LOFT 7. pp. 87–117. Amsterdam University Press (2008)
12. Ditmarsch, H.v., van der Hoek, W., Kooi, B.: Dynamic Epistemic Logic. Springer (2007)

13. Ghallab, M., Nau, D.S., Traverso, P.: Automated Planning: Theory and Practice. Morgan Kaufmann (2004)
14. Horrocks, I., Hustadt, U., Sattler, U., Schmidt, R.: Computational modal logic. In: Handbook of Modal Logic. Elsevier (2006)
15. de Lima, T.: Optimal Methods for Reasoning about Actions and Plans in Multi-Agents Systems. Ph.D. thesis, IRIT, University of Toulouse 3, France (2007)
16. Löwe, B., Pacuit, E., Witzel, A.: DEL planning and some tractable cases. Lecture Notes in Computer Science 6953 (2011)
17. Rintanen, J.: Complexity of planning with partial observability. In: Zilberstein, S., Koehler, J., Koenig, S. (eds.) ICAPS. pp. 345–354. AAAI (2004)