A logic for planning under partial observability

A. Herzig, J. Lang, D. Longin and T. Polacsek
IRIT-UPS, F-31062 Toulouse Cedex 04, France
{herzig, lang, longin, polacsek}@irit.fr
http://www.irit.fr/~Andreas.Herzig

Abstract

We propose an epistemic dynamic logic EDL able to represent the interactions between action and knowledge that are fundamental to planning under partial observability. EDL enables us to represent incomplete knowledge, nondeterministic actions, observations, sensing actions and conditional plans; it also enables a logical expression of several frequently made assumptions about the nature of the domain, such as determinism, full observability, unobservability, or pure sensing. Plan verification corresponds to checking the validity of a given EDL formula. The allowed plans are conditional, and a key point of our framework is that a plan is meaningful if and only if the branching conditions bear on the knowledge of the agent only, and not on the real world (to which that agent may not have access); this leads us to consider "plans that reason" which may contain branching conditions referring to implicit knowledge to be evaluated at execution time.

Introduction

A large amount of work has been done recently about planning under incomplete information. By incomplete information, we mean (as usual) that the initial state of the world is not fully known and/or the actions are not deterministic. The gap between planning under complete and incomplete information relies on the role of *knowledge* in the latter and especially the *interactions between knowledge and action*. These interactions work in both ways:

- the choice of an action to perform is guided by the knowledge the agent has on the actual state of the world, and especially by the *observations* made after performing previous actions. As soon as performance of an action allows gathering knowledge by a susbsequent observation, the choice of the following action will be generally conditioned by this observation.
- performing an action may bring some more knowledge (and help the agent acting in a better way, as explained just above) by means of consecutive observations; this has led many researchers to focus and formalize *sensing actions*, which do not change the state of the world but only the agent's beliefs.

On the other hand, the planning community has recently paid a lot of attention to the role of logic in representing and solving planning problems. This includes both (1) the SATPLAN framework (Kautz & Selman 1996) and its very recent extensions to planning under incomplete knowledge (Rintanen 1999) and (2) the action description languages and their recent application to planning under incomplete knowledge (Lobo, Mendez, & Taylor 1997) (Baral, Kreinovich, & Trejo 1999). The former (1) led to powerful resolution algorithms which benefit from the theoretical and experimental results on satisfiability. The latter (2) gave birth to very expressive languages that enable reasoning with nondeterminism, minimal change, ramifications, concurrent actions, and more recently interactions between action and knowledge. But so far these approaches did not lead yet to the practical development of planning algorithms based on automated proof procedures.

The question now consists of identifying the "simplest" logic containing the notions of incomplete knowledge, nondeterministic actions, observations, conditional plan and sensing actions. Two of these words evoque two well-known families of logics:

- *dynamic logic* aims at reasoning with complex combinations of actions (including sequential and conditional actions). Decidability and complexity results, as well as automated proof procedures for the standard propositional dynamic logic and its variants are a familiar part of the logical landscape. Still, surprisingly, dynamic logic has not been much considered for planning.
- *epistemic logics* aim at reasoning with explicit knowledge of an agent. Some simple epistemic logics are computationally not harder than classical logic.

What is missing to propositional dynamic logic (PDL) so as to render it suitable for planning under incomplete information is (1) the possibility for actions to have epistemic effects and (2) the possibility to branch on epistemic conditions. This second point needs some comments: indeed, PDL enables expressing some kinds of conditional actions, but these conditional actions are not suitable for expressing conditional plans, because while a program can be supposed to know at any instant the value of any variable, this is generally not the case for an agent acting in an incomplete environment, for whom some parts of the world are hidden (see e.g. (Levesque 1996)).

Example 1 *There are d doors. Behind exactly t of the doors, there is a tiger ($t < d$). Behind one (and only one) of the doors, there is a princess and no tiger. The agent has no prior knowledge about what is behind each of the doors (he only knows the values of d and t). The available actions are listening to what happens behind a given door, which results in hearing the tiger roaring if and only if there is one behind the door, and opening a given door, which may result in marrying the princess or being eaten by the tiger or nothing, depending on what is behind the door. The goal of the agent is to stay alive and marry the princess.*

The propositional variables $p(i)$ and $t(i)$ mean respectively that there is a princess (a tiger) behind door i. Consider the following plan

```
π1:  if p(1) then open(1)
              else if p(2) then open(2) (...)
```

Clearly, π_1 is not executable by the agent, because in general he does not know whether $p(i)$ holds or not, and therefore he is unable to branch on such a condition. The solution is to allow for *epistemic conditions* only. Suppose therefore that our language contains a modal operator **K**. A plan such as

```
π2:  if Kp(1) then open(1)
               else if Kp(2) then open(2) else (...)
```

can then be formulated. However, except when $d = 1$, it misses the goals, because the agent ignores whether $p(i)$ is true or not, therefore he will not open any door.

Thus, agents branch on epistemic conditions only, because they are able to decide whether they *know* a given formula or not, whereas they are not always able to decide whether this formula is *true* in the actual world.

This brings us to the following specificity of our logic: it enables the expression of *plans that explicitely involve a reasoning task*. For Example 1, the shortest succeeding plan (in terms of the average number of actions, given uniform probabilities for princess and tigers locations) can be expressed informally by: repeat (listen to a door and open if there is no tiger) until either the princess is delivered or the two tigers have been found; in this second case, open all remaining doors until the princess is discovered. For $d = 4$ and $t = 2$ this gives

```
π3: listen(1);
   if t(1)
   then listen(2);
        if t(2)
        then open(3); if ¬p(3) then open(4) endif
        else open(2);
             if ¬p(2)
             then listen(3); if t(3) then open(4)
                                      else open(3) endif
   else open(1); if ¬p(1) then listen(2); (...)
```

Such a plan succeeds, but it explicits all branches and it is thus space-consuming: when d and t vary, the size of a valid plan increases exponentially[1]. Now, the branching conditions of such plans are only *conjunctions of elementary observations* (those elementary observations are $t(i)$ or $\neg t(i)$ – one of these is observed after $listen(i)$ is performed), which means that, at any step of the execution, (i) the agent decides in unit time what is the next action to follow and (ii) he is not asked to reason (only to obey, to follow a fully explicited plan).

Let us now consider a plan π_4 where epistemic branching conditions are allowed. We use the following abbreviations: $KnowWherePrincess$ = $\mathbf{K}p(1) \vee \mathbf{K}p(2) \vee \ldots \vee \mathbf{K}p(d)$ and $KnowWhereTigers$ = $(\mathbf{K}t(1) \vee \mathbf{K}\neg t(1)) \wedge \ldots \wedge (\mathbf{K}t(d) \vee \mathbf{K}\neg t(d))$ and we define the procedures:

```
OpenIfKnowWherePrincess:
if Kp(1) then open(1) else if Kp(2) then
open(2)...

OpenIfKnowWhereTigers:
if K¬t(1) then open(1);
if ¬Kmarried ∧ K¬t(2) then open(2); ...

π4: listen(1); listen(2);
   if ¬KnowWherePrincess ∧ ¬KnowWhereTigers
   then listen(3);
   if KnowWherePrincess
   then OpenIfKnowWherePrincess
   else OpenIfKnowWhereTigers
```

It can be checked that π_4 reaches the goals. Interestingly, if d and t vary, there is a plan in the style of π_4 (using epistemic branching conditions) whose size is *linear* in d while any valid plan in the style of π_3 (with no epistemic branching conditions) has an *exponential* size. It is important to notice that this gain in size is counterbalanced by a loss of execution time: indeed, although π_1-like plans have a size in $\mathcal{O}(2^d)$, their execution time (assuming that actions are performed in unit time) takes only $\mathcal{O}(d)$. Contrastedly, for π_4-like plans whose size is in $\mathcal{O}(d)$, their execution requires a linear number of calls to a NP-complete oracle (to compute "KnowWherePrincess" and "KnowWhereTiger"). Therefore, a plan with epistemic branching conditions is all the more interesting as the ratio between cost of space and cost of expensive on-line execution time is high.

An epistemic dynamic logic

Language of EDL

The language of epistemic dynamic logic EDL is constructed from a set of atomic formulas VAR, a set of atomic actions ACT_0, the classical logic operators $\rightarrow, \wedge, \vee, \neg$, the epistemic operator **K**, the dynamic operator $[.]$, and the action operators λ and ;. The formula $[\alpha]q$ is read "after the execution of the action α, q is true". Note that we allow for nested epistemic and dynamic operators. λ is the action "do nothing". The complex action $\alpha;\beta$ is read "execute α and then β". $\langle\alpha\rangle A$ is an abbreviation of $\neg([\alpha]\neg A)$, and $[\texttt{if } A \texttt{ then } \alpha \texttt{ else } \beta]C$ is an abbreviation of $(A \rightarrow [\alpha]C) \wedge (\neg A \rightarrow [\beta]C)$.

An EDL formula is

- *objective* iff it does not contain any modality;
- *static* iff it does not contain any dynamic modality;

[1] This is because the evaluation of the condition "the d tigers have been found" after having listened to k doors needs counting the tigers heard so far, which needs explicitly listing the $\binom{k}{t}$ corresponding cases.

- an *epistemic atom* iff it is of the form $\mathbf{K}A$, where A is any EDL formula;
- an *epistemic formula* iff it is formed from epistemic atoms and the connectives of classical logic.

For instance, $a \vee (b \wedge c)$ is objective; $\mathbf{K}(a \vee \neg\mathbf{K}(a \to [\alpha]b))$ is an epistemic atom; $\mathbf{K}(a \vee \neg\mathbf{K}(a \to [\alpha]b)) \vee \neg\mathbf{K}[\beta]c$ is an epistemic formula; $a \vee \neg\mathbf{K}(b \wedge \mathbf{K}c)$ is static but not epistemic; $\mathbf{K}a \vee \neg\mathbf{K}(b \wedge \mathbf{K}c)$ is both static and epistemic.

Semantics of EDL

The semantics of EDL is in terms of possible worlds (states). We interpret the knowledge of the agent at a possible world w by a set of worlds associated to w. Actions are interpreted as transition relations on worlds.

We define a model for EDL as a quadruple $M = \langle W, R_{\mathbf{K}}, \{R_\alpha : \alpha \in ACT_0\}, V\rangle$ where W is a set of possible worlds, $R_{\mathbf{K}} \subseteq W \times W$ and every $R_\alpha \subseteq W \times W$ is an accessibility relation[2], and V associates to each world an interpretation. We require

- $R_{\mathbf{K}}$ to be an equivalence relation on W,
- $R_\lambda\{w\} = \{w\}$,
- $R_{\alpha;\beta} = R_\alpha \circ R_\beta$,
- $R_\alpha \circ R_{\mathbf{K}} \subseteq R_{\mathbf{K}} \circ R_\alpha$.

The truth conditions are defined as usual, in particular:

- $\models_{M,w} \mathbf{K}A$ if $\models_{M,v} A$ for every state $v \in R_{\mathbf{K}}(w)$
- $\models_{M,w} [\alpha]A$ if $\models_{M,w'} A$ for every state $w' \in R_\alpha(w)$

Logical consequence (with global axioms) is noted $\models$.

Axiomatization of EDL

Our axiomatisation of EDL contains that of classical logic together with modal logics S5 for knowledge and K for actions.

N(**K**)	$\dfrac{A}{\mathbf{K}A}$
N($[\alpha]$)	$\dfrac{A}{[\alpha]A}$
K(**K**)	$(\mathbf{K}A \wedge \mathbf{K}(A \to C)) \to \mathbf{K}C$
T(**K**)	$\mathbf{K}A \to A$
5(**K**)	$\neg\mathbf{K}A \to \mathbf{K}\neg\mathbf{K}A$
Def(λ)	$[\lambda]A \leftrightarrow A$
Def($\alpha;\beta$)	$[\alpha;\beta]A \leftrightarrow [\alpha][\beta]A$
K($[\alpha]$)	$([\alpha]A \wedge [\alpha](A \to C)) \to [\alpha]C$
Acq($[\alpha]$, **K**)	$\mathbf{K}[\alpha]C \to \mathbf{K}[\alpha]\mathbf{K}C$

All the axioms are standard, except Acq($[\alpha]$, **K**) which means that if the agent knows what will be true after an action then he does not loose this knowledge after the action. Note that Acq($[\alpha]$, **K**) could be replaced by $\mathbf{K}[\alpha]C \to [\alpha]\mathbf{K}C$ to which it is equivalent given the other axioms.

[2]We shall sometimes identify $R_{\mathbf{K}}$ and R_α with mappings $R_{\mathbf{K}} : W \longrightarrow 2^W$ and $R_\alpha : W \longrightarrow 2^W$.

Action and information

Our language is sufficiently expressive to distinguish explicitly between the ontic and the epistemic effects of actions, where ontic (respectively epistemic) effects are meant to be effects on the physical world only (resp. on the epistemic state of the agent only). Two particular classes of actions are *uninformative actions* whose effects are purely ontic and *purely informative actions* whose effects are purely epistemic. Intuitively, uninformative actions cannot bring any new knowledge, which means that everything that is known after the action is performed could be predicted before it was performed. The other way round, purely informative actions do not change the world. In our example, the actions $listen(i)$ are purely informative, while the actions $open(i)$ are not.

A key hypothesis of our logic is that any action α can be decomposed into two actions, namely $\alpha = \alpha^e \circ \alpha^o$, where α^e is purely informative and α^o uninformative. $open(i)$ can be written as $open(i)^e \circ open(i)^o$, where its uninformative component $open(i)^o$ has the effect of making the agent married, eaten, or none of both, *without him being aware*, and its informative component $open(i)^e$ has the effect to make the agent *learn* whether he gets married, eaten or none of both.

Given that $\alpha = \alpha^e \circ \alpha^o$, what is the relation between α^o and α^e? Ideally, all changes brought about by α^o are perceived through α^e. We call actions of this kind (i.e., actions informing the agent about all changes they cause) *fully informant*. Formally, α is fully informant iff for any objective formula A, we have that

$$A \to [\alpha](\neg A \to \mathbf{K}\neg A)$$

holds. Purely informative actions α are fully informant because the ontic part is empty, i.e. $\alpha^o = \lambda$. In our example $open(i)$ is fully informant because it makes the agent aware of the change of truth value of $eaten$ or $married$ when it occurs.

Noticeably, these properties (purely informative, uninformative, fully informant) are about *individual actions*. Nevertheless, global properties of environments (domain descriptions) can be captured from local properties of actions. An *environment* Σ consists of a nonlogical theory T describing the general laws of the world and what is known about the effects of actions, together with the description of what is known about the initial state Σ_{Init}. An environment Σ is *fully observable* iff (i) each $\alpha \in ACT_0$ is fully informant, and (ii) Σ_{Init} is epistemically complete[3]. Σ is *unobservable* iff each $\alpha \in ACT_0$ is uninformative. Σ is *purely informative* iff each $\alpha \in ACT_0$ is purely informative. Σ is *deterministic* iff each $\alpha \in ACT_0$ is deterministic.

Uninformative actions

We characterize uninformative actions by two axioms.

DetEpi($[\alpha]$, **K**) $\langle\alpha\rangle\mathbf{K}A \to [\alpha]\mathbf{K}A$

Con($[\alpha]$, **K**) $[\alpha]\mathbf{K}C \to ([\alpha]\bot \vee \mathbf{K}[\alpha]C)$

The first expresses that uninformative actions are epistemically deterministic, in the sense that if *there is* a way of executing α such that A is known afterwards, then A should be

[3]i.e. if $\Sigma_{Init} \models \mathbf{K}A \vee \mathbf{K}\neg A$ for all objective A

known after every possible execution of α. This is natural, given that α does not bring any new knowledge (in particular about the way it has been executed). The second says that the epistemic effects of α are known before hand. Semantically, the axioms correspond to the conditions

- If $w', w'' \in R_\alpha(w)$ then $R_{\mathbf{K}}(w') = R_{\mathbf{K}}(w'')$
- If $R_\alpha(w) \neq \emptyset$ and $w_1 R_{\mathbf{K}} \circ R_\alpha w_2$ then $w_1 R_\alpha \circ R_{\mathbf{K}} w_2$

It can be proved that these two axioms together are equivalent to the more compact criterion of uninformativeness $\neg\mathbf{K}[\alpha]A \rightarrow [\alpha]\neg\mathbf{K}A$. It says that the agent cannot observe anything after α is performed: indeed, for any formula A, if he cannot predict before α is performed that A will hold after α, then he will not know A after α is performed. Acq($[\alpha]$, **K**) together with Con($[\alpha]$, **K**) gives us the equivalence $[\alpha]\mathbf{K}C \equiv ([\alpha]\bot \vee \mathbf{K}[\alpha]C)$.

Purely informative actions

Purely informative actions do not change the world but only the knowledge; they are characterized by the axiom

Pres($[\alpha]$) $A \rightarrow [\alpha]A$ if A is an objective formula.

Semantically, this corresponds to the condition

- if $w_1 R_\alpha w_2$ then $V_{w_1} = V_{w_2}$

It follows from Acq($[\alpha]$, **K**) and standard modal principles that purely informative actions do not diminish the knowledge of an agent.

Proposition 1 *Let A be an objective formula and α a purely informative action. Then*

Pres($[\alpha]$, **K**) $\boldsymbol{K}A \rightarrow [\alpha]\boldsymbol{K}A$

is provable from EDL.

If A is subjective then this does not necessarily hold, in particular if A expresses ignorance. For example $\mathbf{K}\neg\mathbf{K}A \rightarrow [\alpha]\mathbf{K}\neg\mathbf{K}A$ cannot be accepted, given that $\mathbf{K}\neg\mathbf{K}A \leftrightarrow \neg\mathbf{K}A$ is valid in our logic of knowledge S5.

A solution to the Frame Problem

We must solve the Frame Problem in order to put to work our logic. Basically, we could integrate any solution into our framework, given that we have analysed the epistemic effects of an action in terms of its ontic effects on possible worlds. Scherl and Levesque e.g. used Reiter's solution to the Frame Problem, and applied regression as a reasoning method (Scherl & Levesque 1993). We adopt the solution in (Castilho, Gasquet, & Herzig 1999) based on dependence relations, which can be taken over without modifications and which we briefly recall here.

We associate to every atomic action α the set of atomic formulas it influences. Formally, we suppose given a dependency function $DEP : ACT_0 \longrightarrow VAR$. $p \in DEP(\alpha)$ means that α may change the truth value of the atom p. The other way round, if $p \notin DEP(\alpha)$ then α does not change the truth value of p. In other words, DEP represents frame axioms in an economic way. This is expressed by the generic frame axiom

Pres$_{DEP}$($[\alpha]$) if $p \notin DEP(\alpha)$ then $p \rightarrow [\alpha]p$ and $\neg p \rightarrow [\alpha]\neg p$

Semantically, the axiom corresponds to the condition

- For all $w, w' \in W$ and $p \in VAR$, if $w' \in R_\alpha(w)$ and $p \notin DEP(\alpha)$ then $\models_{M,w} p$ iff $\models_{M,w'} p$.

Given a dependence relation DEP, $\models_{DEP}$ is the corresponding extension of theEDL consequence relation.

As a particular case, purely informative actions verify $DEP(\alpha) = \emptyset$. Thus, we have for instance the frame axioms $t(1) \rightarrow [listen(1)]t(1)$ and $p(2) \rightarrow [listen(1)]p(2)$. Combining all these atomic frame axioms by principles of classical logic, we obtain the following.

Proposition 2 *Let A be objective, and let* atm(A) *be the set of atoms of A. If* atm$(A) \cap DEP(\alpha) = \emptyset$ *then:*

- $\models_{DEP} A \rightarrow [\alpha]A$;
- $\models_{DEP} \boldsymbol{K}A \rightarrow [\alpha]\boldsymbol{K}A$.

Plan verification in EDL

The set of *meaningful plans* is the smallest set such that

- α is a meaningful plan for every $\alpha \in ACT_0 \cup \{\lambda\}$;
- if π and π' are meaningful plans and A is an epistemic formula then $\pi;\pi'$ and `if` A `then` π `else` π' are meaningful plans.

Intuitively, a meaningful plan is a plan whose branching conditions are "epistemically interpretable", which means that the agent can decide whether the branching condition holds or not (which would not necessarily be the case if the formula were not epistemic, cf. example).

A *plan verification problem* $\mathcal{V}$ is defined by a 5-tuple $\langle T, \Sigma_{Init}, DEP, G, \pi\rangle$ where

- $T = \langle S, E, X\rangle$ is an EDL theory composed of a set of state axioms S expressing static laws of the domain, laws about the effects of actions E, and executability laws X. Static laws are static formulas; effect laws are formulas of the form $A \rightarrow [\alpha]C$ with A objective and C an epistemic formula; executability laws are formulas of the form $A \leftrightarrow \langle\alpha\rangle\top$ with A objective.
- Σ_{Init} is an epistemic atom;
- DEP is a dependency function;
- G is a static formula (the goal);
- π is an meaningful plan.

Given a plan verification problem $\mathcal{V}$, π is said to be

- *executable* for $\mathcal{V}$ iff $T \models_{DEP} \Sigma_{Init} \rightarrow \langle\pi\rangle\top$ holds.
- *valid* for $\mathcal{V}$ iff $T \models_{DEP} \Sigma_{Init} \rightarrow [\pi]G$ holds.

The validity problem in EDL is PSPACE-hard, and the consequence problem is EXPTIME-hard. The reason is that EDL extends modal logic K, where these problems are PSPACE- and EXPTIME-complete, respectively. Nevertheless, the complexity of the much more specific plan verification problem in EDL is much lower:

Proposition 3 (complexity of plan verification) PLAN VERIFICATION *in EDL is Π_2^p-complete.*

Note that if branching conditions were restricted to elementary conjunctions of observations instead of any epistemic conditions then the problem would be "only" coNP-complete.

It is worth investigating what plan validation becomes when some specific assumptions are made about observability.

Unobservable environments When the environment is unobservable, we can show by induction that uninformativeness extends to any meaningful plan: for any meaningful plan π and any objective formula A,

$$\neg\mathbf{K}[\pi]A \to [\pi]\neg\mathbf{K}A$$

holds, which leads to the following intuitive result: *if there is a meaningful valid plan π for $\mathcal{V} = \langle T, \Sigma_{Init}, DEP, G\rangle$ then there is a nonbranching (i.e., without tests) meaningful valid plan for $\mathcal{V}$.*

Fully observable environments When the environment is fully observable, we can show by induction that for any meaningful plan π and any objective formula A, we have $[\pi](\mathbf{K}A \vee \mathbf{K}\neg A)$ holds, and that the epistemic operator is needless, which is expressed intuitively by the following result: *if there is a meaningful valid plan π for $\mathcal{V}$ then there is an epistemic-free (e.g., without any occurrence of* **K***) meaningful valid plan for $\mathcal{V}$.* Thus, a fragment of PDL is sufficient for capturing plan verification in fully observable environments.

Purely epistemic environments When the environment is fully epistemic, all actions are commutative and knowledge preserving. This leads to the following result: if there is a meaningful valid plan π for $\mathcal{P}$ then $\alpha_1; \alpha_2; ...; \alpha_n$ is a valid plan for $\mathcal{P}$, where $ACT_0 = \{\alpha_1, ..., \alpha_n\}$.

We end up the section with some considerations on plan existence. Informally, the plan existence problem reads: given $\mathcal{P} = \langle T, \Sigma_{Init}, DEP, G\rangle$, is there a plan π for $\mathcal{P}$? Similar to verification problems, we check whether there is a proof of $T, \neg G \models_{DEP} \neg\Sigma_{Init}$. If this is the case, then we can associate a meaningful executable plan π to $\mathcal{P}$. π can then be checked for validity (which will not always be the case, in particular when actions are nondeterministic). π is certainly only a first step towards a plan. We leave this issue to further research.

Example

How we can handle our running example in our logic? For the sake of readability, suppose $d = 2$ and $t = 1$. A meaningful valid plan is

$\pi_{2,1} = listen(1);$ `if` $\mathbf{K}t(1)$ `then` $open(2)$ `else` $open(1)$

Proving the validity of $\pi_{2,1}$ amounts to proving that the formula $T \models_{DEP} \Sigma_{Init} \to [\pi_{2,1}](married \wedge alive)$ is a theorem, where the initial situation Σ_{Init} is

$\Sigma_{Init} = alive \wedge ((t(1) \wedge \neg t(2) \wedge p(2)) \vee (t(2) \wedge \neg t(1) \wedge p(1)))$

and T is the nonlogical theory of the domain, consisting of the following set T of effect axioms:

$$\begin{aligned} T = \{\; & t(i) \to [listen(i)]\mathbf{K}t(i),\ \neg t(i) \to [listen(i)]\mathbf{K}\neg t(i), \\ & p(i) \to [open(i)]married,\ t(i) \to [open(i)]\neg alive, \\ & (\neg p(i) \wedge \neg married) \to [open(i)]\neg married, \\ & (\neg t(i) \wedge alive) \to [open(i)]alive, \\ & \langle listen(i)\top\rangle,\ \langle open(i)\top\rangle\}. \end{aligned}$$

(where we suppose $i \in \{1, 2\}$).

Moreover, let $DEP(listen(i)) = \emptyset$ and $DEP(open(i)) = \{married, alive\}$. (The dependence-based solution to the Frame Problem requires the last two conditional frame axioms of T.) We establish that $T \models_{DEP} \Sigma_{Init} \to [\pi_{2,1}](married \wedge alive)$ by proving $T \models_{DEP} (t(1) \wedge \neg t(2) \wedge p(2) \wedge alive) \to [\pi_{2,1}](married \wedge alive)$ and $T \models_{DEP} (t(2) \wedge \neg t(1) \wedge p(1) \wedge alive) \to [\pi_{2,1}](married \wedge alive)$. Then the disjunction of the respective antecedens is nothing but Σ_{Init}, and since $(\mathbf{K}t(1) \wedge [open(2)](alive \wedge married)) \vee (\mathbf{K}\neg t(1) \wedge [open(1)](alive \wedge married))$ is equivalent to $[listen(1);$ `if` $\mathbf{K}t(1)$ `then` $open(2)$ `else` $open(1)](alive \wedge married)$, putting things together we obtain what we wanted.

Related work and conclusion

There is a significant amount of related work about the interactions between action and knowledge, both in the KR and the planning communities. Combining knowledge and action in a logical framework comes back to the work of (Moore 1985) who provided a theory of action including knowledge-producing actions. Building on this theory, (Scherl & Levesque 1993) represent knowledge-producing actions in the situation calculus by means of an explicit accessibility relation between situations, treated as an ordinary fluent, that corresponds to our epistemic accessibility relation. (Levesque 1996) then uses this knowledge fluents to represent complex plans involving, like ours, nondeterminism, observations and branching (and also loops, that we did not consider). He points out that the executability of a plan requires that the agent *needs to know how to execute it*, which implies that branching conditions must involve knowledge and not objective facts whose truth may not be accessible to the agent. On the one hand, in our logic, consisting of a fragment of propositional PDL extended with epistemic modalities to represent the effects of actions, the interactions between dynamic and epistemic modalities enable a simple representation of various observability assumptions; on the other hand, by using the situation calculus, Levesque handles more easily than us *value tests* returning the value of a variable (for instance, he is able to represent in a simple way a plan such as `search Mary's phone number in the phonebook and then dial it` whereas we cannot do it unless we write down a finite but unreasonable amount of propositional formulas). This approach was extended in (Lakemeyer & Levesque 1998) so as to introduce the *only knowing* modality. (Bacchus & Petrick 1998) point out the practical impossibility to generate explicit conditional plans, because they get too large, and thus advocate for the need of reasoning about knowledge during plan execution, which is one of the key points of our logic. Their representation model makes use of an epistemic modality. Our approach could be thought of as being complementary to theirs, because we provide a simple way to represent various kinds of interactions between knowledge and action while they focus on the practical computation of the effects of a plan containing sensing actions and knowledge preconditions. (Geffner & Wainer 1998) provide a general language enabling representing nondeterministic actions, sensing actions, observations and conditional plans. Their notion of executable policy is very similar to our notion of meaningful plan, though it is not expressed the same way technically speaking: to avoid generating unreasonably large policies (which happens whenever all accessible belief states are explicitely considered), they express policies on states rather than on belief states; a policy is then said to be executable in a belief state Bel if, roughly speaking, it assigns equivalent actions to all states considered possible in Bel. We choose another way to escape representing explicitly conditional plans, namely by calling for reasoning tasks during execution. (Lobo, Mendez, & Taylor 1997) extend Gelfond and Lifschitz' language $\mathcal{A}$ for reasoning about action so as to represent knowledge effects. An interesting notion in their approach is *knowledge removing actions* that may affect the knowledge the agent has on a fluent. These knowledge removing actions (such as `toss`) can be easily handled in EDL.

These approaches focus on representing actions and conditional plans involving knowledge preconditions and effects, and checking whether a given plan reaches the goal. Up to know, little has been done in order to generate plans having knowledge preconditions. (Rintanen 1999) extends the planning as satisfiability framework to planning under incomplete knowledge by means of Quantified Boolean Formulae. (Boutilier & Poole 1996) provide a propositional-like representation and resolution framework for POMDPs. None of these works makes use of epistemic nor dynamic modalities.

Lastly, a few authors developed logical systems integrating dynamic and epistemic modalities, but not from a planning perspective. (Del Val, Maynard-Reid II, & Shoham 1997) study from a logical perspective the relations between what the agent perceive and what they believe; this is much related to our logical expressions of observability assumptions in EDL. (Fagin *et al.* 1995) have a language with temporal 'next' and 'always' operators instead of action operators. They have axioms of perfect recall similar to our Acq($[\alpha]$, **K**) (axioms KT1, KT2). They do not integrate a solution to the Frame Problem into their approach. Also slightly related to our work is (Meyer, van der Hoek, & van der Linder 1994) who consider tests as epistemic updates.

Apart from the handling of plan existence, further work includes the study of the complexity of validity for the full logic EDL (so far we only have a complexity result for *plan verification* in EDL) and next, the complexity of plan existence with epistemic preconditions, which would complete the panorama of complexity results for planning under incomplete knowledge (Littman 1997; Baral, Kreinovich, & Trejo 1999).

References

Bacchus, F., and Petrick, R. 1998. Modeling an agent's incomplete knowledge during planning and execution. In *Proc. KR'98*, 423–443.

Baral, C.; Kreinovich, V.; and Trejo, R. 1999. Computational complexity of planning and approximate planning in the presence of incompleteness. In *Proc. of IJCAI'99*, 948–953.

Boutilier, C., and Poole, D. 1996. Computing optimal policies for partially observable decision processes using compact representations. In *Proceedings of AAAI'96*, 1168–1175.

Castilho, M. A.; Gasquet, O.; and Herzig, A. 1999. Formalizing action and change in modal logic I: the frame problem. *J. of Logic and Computation* 9(5).

Del Val, A.; Maynard-Reid II, P.; and Shoham, Y. 1997. Qualitative reasoning about perception and belief. In *Proc. IJCAI'97*, 508–513.

Fagin, R.; Halpern, J.; Moses, Y.; and Vardi, M. 1995. *Reasoning about knowledge*. MIT Press.

Geffner, H., and Wainer, J. 1998. Modeling action, knowledge and control. In *Proc. ECAI'98*, 532–536.

Kautz, H., and Selman, B. 1996. Planning as satisfiability. In *Proc. AAAI'96*, 1139–1146.

Lakemeyer, G., and Levesque, H. 1998. AOL: a logic of acting, sensing, knowing, and only knowing. In *Proc. KR'98*, 316–327.

Levesque, H. J. 1996. What is planning in the presence of sensing? In *Proc. AAAI'96*, 1139–1146.

Littman, M. 1997. Probabilistic planning: representation and complexity. In *Proc. of AAAI'97*, 748–754.

Lobo, J.; Mendez, G.; and Taylor, S. R. 1997. Adding knowledge to the action description language a. In *Proc. AAAI'97*, 454–459.

Meyer, J.-J.; van der Hoek, W.; and van der Linder, B. 1994. Test as epistemic updates. In *Proc. ECAI'94*.

Moore, R. 1985. A formal theory of knowledge and action. In *Formal Theories of the Commensense World*. Norwood. 319–358.

Rintanen, J. 1999. Constructing plans by a theorem prover. *JAIR* 10:323–352.

Scherl, R., and Levesque, H. 1993. The frame problem and knowledge-producing actions. In *Proc. AAAI'93*, 689–695.