



HAL
open science

A logic for planning under partial observability (AIPS 2000)

Andreas Herzig, Jérôme Lang, Dominique Longin, Thomas Polacsek

► **To cite this version:**

Andreas Herzig, Jérôme Lang, Dominique Longin, Thomas Polacsek. A logic for planning under partial observability (AIPS 2000). 5th international conference on Artificial Intelligence Planning and Scheduling (AIPS 2000), Apr 2000, Breckenridge, CO, United States. hal-03523419

HAL Id: hal-03523419

<https://hal.science/hal-03523419v1>

Submitted on 13 Jan 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A logic for planning under partial observability

Andreas Herzig, Jérôme Lang, Dominique Longin & Thomas 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 integration of a fragment of propositional dynamic logic with an epistemic logic so as to represent logically the interactions between action and knowledge that are fundamental to planning under incomplete knowledge and partial observability. This logic, called EDL, enables us representing 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 problem such as determinism, full observability, unobservability, or pure knowledge gathering. Plan verification corresponds to testing the validity of a given formula of this logic. The allowed plans are conditional, and a key point of our framework is that a plan is meaningful if the branching conditions bear on the knowledge of the agent only, and not on the real world (to which the agent may not have access); this leads us considering “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* (which are irrelevant for planning under complete knowledge). 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 to gather knowledge by a subsequent observation, the choice of the following action will be generally conditioned by this observation. This leads to the notion of *conditional plan* (for planning under complete information).
- 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 a particular class of actions, viz. actions of observation, alias (pure) *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 (as well as model checking) and its very recent extensions to planning under incomplete knowledge and (2) the action description languages and their recent application to planning. The former (1) lead to powerful resolution algorithms which benefit from the theoretical and experimental results on satisfiability. So far, not been too much extended to planning under incomplete information (up to a very small number of exceptions) nor to the sophisticated handling of change at work in approaches of type (2). The latter (2) gives birth to very expressive languages that enable reasoning with nondeterminism, minimal change, ramifications, concurrent actions, and more recently interactions between action and knowledge. The latter approaches did not lead yet to the development of planning algorithms based on automated proof procedures. This gap between (1) and (2) is nothing but the very classical duality between expressivity and efficiency and there is no reason why it should not be successfully filled in the next years. We believe that in order to pave the way towards this objective, a necessary step is the development of logics that explicitly handle all the facets of the aforementioned interactions between action and knowledge. This paper is a first step in this direction.

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 evoke two well-known families of logics:

- *dynamic logic* aims at reasoning with complex combinations of actions (including sequential and conditional actions). It has been initially developed so as to reason about programs. Decidability and complexity results, as well as automated proof procedures for

the standard propositional dynamic logic and its variants, extensions and restrictions are a familiar part of the logical landscape. But still, and surprisingly, dynamic logic has not been much considered for planning (further on we will make it clearer why this is not as surprising as it may appear).

- *epistemic logics* aim at reasoning with explicit knowledge of an agent. Some simple epistemic logics (such as S5) are computationally not harder than classical logic.

An immediate analysis of the needs (handling the intimate interactions between action and knowledge) call for a logic including both a suitable fragment of dynamic and epistemic logic. However, to our knowledge almost nothing has yet been done for such an integration of dynamic and epistemic logic in the scope of planning under incomplete knowledge (Seegerberg), (Baltag, Moss, & Solecki 1998) and (Gerbrandy 1997).

We now go a bit deeper into the details. For the reader not familiar with dynamic logics nor with epistemic logic, the necessary background is given in Appendix. 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. (1) is clear and does not need further intuitive explanations. For (2), this is more subtle. Indeed, PDL enables expressing some kinds of conditional actions, namely, of the form if p then α else β . Are these conditional actions good candidates for expressing conditional plans? The answer is negative. While a program can be supposed to know at any instant the value of any variable, this is not the case (in general) for an agent acting in an incomplete environment, for whom some parts of the world are hidden.

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 a tiger roaring iff there is one behind the door, and opening a given door, which results 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 $princess(i)$ and $tiger(i)$ mean respectively that there is a princess (a tiger) behind door i . Consider the following plan

```

 $\pi_1$ : if  $princess(1)$ 
      then  $open(1)$ 
      else if  $princess(2)$ 
           then  $open(2)$ 
           else if  $princess(3)$ 
                (...) else if  $princess(d)$  then  $open(d)$ 

```

Clearly, this plan does not solve our planning problem, because in general the agent does not know whether $princess(i)$ 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 \mathcal{K} . A plan such as

```

 $\pi_2$ : if  $\mathcal{K}princess(1)$ 
      then  $open(1)$ 
      else if  $\mathcal{K}princess(2)$ 
           then  $open(2)$ 
           else if  $\mathcal{K}princess(3)$ 
                (...) else if  $\mathcal{K}princess(d)$  then  $open(d)$ 

```

can then be formulated. However, except when $d = 1$, it misses the goals, because the agent may ignore whether $princess(i)$ is true or not.

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 or not in the actual world.

This brings us to the following specificity of our logic: it enables the expression of *plans that explicitly involve a reasoning task*. Let us consider the tiger and princess example with 4 doors and 2 tigers. Described in classical conditional planning formalisms, this problem would give for instance the following plan:

```

 $\pi_3$ :  $listen(1)$ ;
      if  $tiger(1)$ 
      then  $listen(2)$ ;
           if  $tiger(2)$ 
           then  $open(3)$ ; if  $\neg princess(3)$  then  $open(4)$ 
           else  $open(2)$ ;
                if  $\neg princess(2)$ 
                then  $listen(3)$ ;
                     if  $tiger(3)$  then  $open(4)$  else  $open(3)$ 
      else  $open(1)$ ;
           if  $\neg princess(1)$ 
           then  $listen(2)$ ;
                if  $tiger(2)$ 
                then  $listen(3)$ ;
                     if  $tiger(3)$  then  $open(4)$  else  $open(3)$ 
           else  $open(2)$ 

```

It can be checked easily that π_3 achieves the goal with certainty. However, it explicit all branches and is thus space-consuming. If d and t vary then the size of a valid plan increases exponentially. Now, such plans have *conjunctions of elementary observations as branching conditions* ($tiger(1)$ or $\neg tiger(1)$ is observed after $listen(1)$ is performed, etc.), which means that, at any step of its execution, (i) the agent decides in unit time what is the next action to follow and (ii) she is not asked to reason (only to obey, to follow a fully explicit plan).

Let us now consider the following plan π_4 for the same problem, where epistemic branching conditions are allowed. We use the following abbreviations:

$KnowWherePrincess = \mathcal{K}princess(1) \vee \dots \vee \mathcal{K}princess(d)$

$KnowWhereTigers = (\mathcal{K}tiger(1) \vee \mathcal{K}\neg tiger(1)) \wedge \dots \wedge (\mathcal{K}tiger(d) \vee \mathcal{K}\neg tiger(d))$

and we define the following procedures:

OpenIfKnowWherePrincess:

```
if  $\mathcal{K}princess(1)$ 
then  $open(1)$ 
else if  $\mathcal{K}princess(2)$ 
then  $open(2)$ 
else (...)
```

OpenIfKnowWhereTigers:

```
if  $\mathcal{K}\text{-tiger}(1)$  then  $open(1)$ ;
if  $\neg married \wedge \mathcal{K}\text{-tiger}(2)$  then  $open(2)$ ;
(...)
if  $\neg married \wedge \mathcal{K}\text{-tiger}(d)$  then  $open(d)$ 
```

Let us now consider the following plan (again for $d = 4$ and $t = 2$):

```
 $\pi_4$ : $listen(1)$ ;  $listen(2)$ ;
if  $\neg KnowWherePrincess \wedge \neg KnowWhereTigers$ 
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 $\mathcal{O}(d)$. In contrast, for π_3 -like plans whose size is in $\mathcal{O}(d)$, their execution requires a linear number of calls to a NP-complete oracle. Therefore, a plan with epistemic branching conditions (EBC plan for short) 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 EDL is the fusion of an epistemic logic and a dynamic logic. Hintikka first proposed to analyse of knowledge and belief in terms of possible states (Hintikka 1962). In this paper we are only interested in *strong knowledge* rather than in belief (since merely believing a fact would not ensure the truth of this fact and therefore would be in principle helpless so as to prove that a given plan certainly succeeds).

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 \mathcal{K} , the dynamic operator $[\cdot]$, 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 $[\text{if } A \text{ then } \alpha \text{ else } \beta]C$ 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;
- an *epistemic atom* iff it is of the form $\mathcal{K}A$, where A is any EDL formula;
- an *epistemic formula* iff it is formed from epistemic atoms and the usual connectives.

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

Semantics of EDL

The semantics of EDL is in terms of possible 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 $M = \langle W, R_{\mathcal{K}}, \{R_{\alpha} : \alpha \in ACT_0\}, V \rangle$ where W is a set of possible worlds, $R_{\mathcal{K}} \subseteq W \times W$ and every $R_{\alpha} \subseteq W \times W$ is an accessibility relation¹, and V associates to each world an interpretation. We require

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

The truth conditions are defined as usual, in particular:

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

Axiomatization of EDL

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

$N(\mathcal{K})$	$\frac{A}{\mathcal{K}A}$
$N([\alpha])$	$\frac{A}{[\alpha]A}$
$K(\mathcal{K})$	$(\mathcal{K}A \wedge \mathcal{K}(A \rightarrow C)) \rightarrow \mathcal{K}C$
$T(\mathcal{K})$	$\mathcal{K}A \rightarrow A$
$5(\mathcal{K})$	$\neg \mathcal{K}A \rightarrow \mathcal{K}\neg \mathcal{K}A$
$\text{Def}(\lambda)$	$[\lambda]A \leftrightarrow A$
$\text{Def}(\alpha;\beta)$	$[\alpha;\beta]A \leftrightarrow [\alpha][\beta]A$
$K([\alpha])$	$([\alpha]A \wedge [\alpha](A \rightarrow C)) \rightarrow [\alpha]C$
$\text{Acq}([\alpha], \mathcal{K})$	$\mathcal{K}[\alpha]C \rightarrow \mathcal{K}[\alpha]\mathcal{K}C$

¹We shall sometimes identify $R_{\mathcal{K}}$ and R_{α} with mappings $R_{\mathcal{K}} : W \rightarrow 2^W$ and $R_{\alpha} : W \rightarrow 2^W$.

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

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. As a consequence, our logic is a helpful language for expressing some specific conditions about observability, since observability is primarily a matter of interactions between ontic and epistemic effects. Intuitively, uninformative actions cannot any 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

$$A \rightarrow [\alpha](\neg A \rightarrow \mathcal{K}\neg A)$$

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* can be captured from local properties of actions. An **environment** Σ consists in 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 infor-

mant, and (ii) Σ_{Init} is complete²;

- *unobservable* iff each $\alpha \in ACT_0$ is uninformative;
- *ontically static* iff each $\alpha \in ACT_0$ is purely informative;
- *deterministic* iff each $\alpha \in ACT_0$ is deterministic.

Uninformative actions

We characterize uninformative actions by two axioms.

$$\text{DetEpi}([\alpha], \mathcal{K}) \quad \langle \alpha \rangle \mathcal{K}A \rightarrow [\alpha]\mathcal{K}A$$

$$\text{Con}([\alpha], \mathcal{K}) \quad [\alpha]\mathcal{K}C \rightarrow ([\alpha]\perp \vee \mathcal{K}[\alpha]C)$$

The first expresses that uninformative actions are epistemically deterministic. 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 $R_\mathcal{K}(w') = R_\mathcal{K}(w'')$
- If $R_\alpha(w) \neq \emptyset$ and $w_1 R_\mathcal{K} \circ R_\alpha w_2$ then $w_1 R_\alpha \circ R_\mathcal{K} w_2$

It can be proved that these two axioms together are equivalent to the more compact criterion of uninformativity $\neg \mathcal{K}[\alpha]A \rightarrow [\alpha]\neg \mathcal{K}A$. 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 is performed, then he will not know A after α is performed. $\text{Acq}([\alpha], \mathcal{K})$, $\text{Con}([\alpha], \mathcal{K})$ gives us the equivalence $[\alpha]\mathcal{K}C \equiv ([\alpha]\perp \vee \mathcal{K}[\alpha]C)$.

Purely informative actions

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

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

Semantically, this corresponds to the condition

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

It follows by 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*

$$\text{Pres}([\alpha], \mathcal{K}) \quad \mathcal{K}A \rightarrow [\alpha]\mathcal{K}A$$

is provable from EDL.

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

A purely informative action α typically makes an agent evolve from $\neg \mathcal{K}A \wedge \neg \mathcal{K}\neg A$ to either $\mathcal{K}A$ or $\mathcal{K}\neg A$ for some objective formula A . We might thus identify α with the action of *testing whether A* , denoted by $A??$. Note that the empty action, equivalent to testing whether \top , is a (degenerate) purely informative action.

In (Herzig, Lang, & Polacsek 1999) we analysed such testing actions $A??$ as the nondeterministic composition

²i.e. if $\Sigma_{Init} \models \mathcal{K}A \vee \mathcal{K}\neg A$ for all objective A

of verifying that A (denoted by $A?$) and verifying that $\neg A$. Formally, $A??$ is an abbreviation of $A? \cup (\neg A)?$. Such an action $A?$ of verifying that A is an extension of standard PDL testing actions to the epistemic context. We consider that verifying that A is inexecutable if A is false: $\neg A \rightarrow [A?] \perp$.

Deterministic actions

In some domains actions are deterministic: every possible execution of an action will always lead to the same result.

$$\text{Det}([\alpha]) \quad \neg[\alpha]C \rightarrow [\alpha]\neg C$$

Semantically we have

- $R_\alpha(w)$ is a singleton or the empty set.

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 $DEP : ACT_0 \rightarrow VAR$. $p \in DEP(\alpha)$ 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 axiom

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

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 the EDL 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.*

If $DEP(\alpha) = \emptyset$ then $\models_{DEP} A \rightarrow [\alpha]A$.

If $\text{atm}(A)$ is the set of atoms of A and $\text{atm}(A) \cap DEP(\alpha) = \emptyset$ then $\models_{DEP} \mathcal{K}A \rightarrow [\alpha]\mathcal{K}A$.

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; effects 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 a 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.

While the validity problems in EDL is PSPACE-hard (because EDL extends the logic K where validity is PSPACE-complete), the complexity of the much more specific plan verification problem in EDL is 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 and determinism.

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 ,

$$\langle \pi \rangle \mathcal{K}A \rightarrow \mathcal{K}[\pi]A$$

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

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 $[\pi](\mathcal{K}A \vee \mathcal{K}\neg A)$ 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{P} then there is an epistemic-free (e.g.,*

without any occurrence of \mathcal{K}) meaningful valid plan for \mathcal{P} . Thus, a fragment of PDL is sufficient for capturing plan verification in fully observable environments.

Fully 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 $\alpha_1; \alpha_2; \dots; \alpha_n$ is a valid plan for \mathcal{P} , where $ACT_0 = \{\alpha_1, \dots, \alpha_n\}$.

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

$\pi_{2,1} = \text{listen}(1); \text{if } \mathcal{K}\text{tiger}(1) \text{ then } \text{open}(2) \text{ else } \text{open}(1)$

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

$\Sigma_{Init} = \text{alive} \wedge ((\text{tiger}(1) \wedge \neg \text{tiger}(2) \wedge \text{princess}(2)) \vee (\text{tiger}(2) \wedge \neg \text{tiger}(1) \wedge \text{princess}(1)))$ and T is the non-logical theory of the domain, consisting of the following set T of effect axioms³:

$$T = \{ \begin{array}{l} \text{tiger}(i) \rightarrow [\text{listen}(i)]\mathcal{K}\text{tiger}(i), \\ \neg \text{tiger}(i) \rightarrow [\text{listen}(i)]\mathcal{K}\neg \text{tiger}(i), \\ \text{princess}(i) \rightarrow [\text{open}(i)]\text{married}, \\ \text{tiger}(i) \rightarrow [\text{open}(i)]\neg \text{alive}, \\ (\neg \text{princess}(i) \wedge \neg \text{married}) \rightarrow [\text{open}(i)]\neg \text{married}, \\ (\neg \text{tiger}(i) \wedge \text{alive}) \rightarrow [\text{open}(i)]\text{alive}, \\ \langle \text{listen}(i) \rangle \top, \langle \text{open}(i) \rangle \top. \end{array}$$

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

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

³We identify T with the conjunction of its elements.

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, our logic, consisting of a fragment of propositional PDL extended with epistemic modalities to represent the effects of actions, is decidable, and furthermore 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 expressed the same way technically speaking: to avoid generating unreasonably large policies (which happens whenever all accessible belief states are explicitly 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 repre-

senting 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. (Bonet & Geffner 2000) have developed a general language which enables a high-level representation of Partially Observable Markov Decision Processes, which could probably be extended with reasoning tasks at run time. (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 $\text{Acq}([\alpha], \mathcal{K})$ (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 of the extension of EDL so as to deal with 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).

Acknowledgements Thanks to Philippe Balbiani for helpful discussions.

References

- Bacchus, F., and Petrick, R. 1998. Modeling an agent's incomplete knowledge during planning and execution. In *Proc. KR'98*, 423–443.
- Baltag, A.; Moss, L.; and Solecki, S. 1998. The logic of public announcements, common knowledge and private suspicions. In *Proc. TARK'98*.
- 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.
- Bonet, B., and Geffner, H. 2000. Planning and control in artificial intelligence: A unifying perspective. *Applied Intelligence*. to appear.
- 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.
- Gerbrandy, J. 1997. Dynamic epistemic logic. Technical report, ILLC, Amsterdam.
- Herzig, A.; Lang, J.; and Polacsek, T. 1999. Knowledge, actions, and tests. In Bell, J., ed., *IJCAI'99 Workshop on Practical Reasoning and Rationality*.
- Hintikka, J. K. K. 1962. *Knowledge and belief*. Ithaca, N.Y.: Cornell University Press.
- 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 Commonsense 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.
- Segerberg, K. Two traditions in the logic of belief: bringing them together. to appear.