Strong Planning in the Logics of Communication and Change

Pere Pardo[1] and Mehrnoosh Sadrzadeh[2]

[1] Institut d'Investigació en Intel·ligència Artificial (IIIA - CSIC), Spain
[2] Dept. of Computer Science, University of Oxford, UK

Abstract. In this contribution we study how to adapt Backward Plan search to the Logics of Communication and Change (LCC). These are dynamic epistemic logics with common knowledge modeling the way in which announcements, sensing and world-changing actions modify the beliefs of agents or the world itself. The proposed LCC planning system greatly expands the social complexity of scenarios involving cognitive agents that can be solved. For example, goals or plans may consist of a certain distribution of beliefs and ignorance among agents. Our results include: soundness and completeness of backward planning (breadth first search), both for deterministic and strong non-deterministic planning.

1 Introduction

Practical rationality or decision-making is a key component of autonomous agents, like humans, and correspondingly has been studied at large. This research has been conducted from several fields: game theory, planning, decision theory, etc. each focusing on a different aspect (strategic decision-making, propositional means-ends analysis, and uncertainty, respectively).

While the different models are well-understood, they were (understandably) designed with a considerably low level of expressivity at the object language. For instance, game-theory does not represent the logical structure underlying the states, actions and goals; planning [5], on the other hand, represents part of it with atomic facts and negation, but it traditionally disregards other existing agents. All this contrasts with the area of logic, where logics for multi-agent systems (with increasing expressivity) have been characterized.

Specially relevant to the topic of cognitive agents are the notions of belief, action, goal, norm, and so on. The first two elements are the target of dynamic epistemic logics DEL [3], [15], [16], a recent family of logics which allow us to reason about agents' communications, observations and the usual world-changing actions. We focus on the so-called Logics of Communication and Change (LCC) [13], which generalize many previously known DEL logics, and hence include a rich variety of epistemic actions (in the DEL literature) and ontic actions (from the tradition on planning). Briefly, LCC logics are dynamic epistemic logics with common knowledge, ontic actions and several types of communicative actions (truthful or lying, public or private announcements).

Less consensus exists about representing and reasoning with motivational attitudes like goals, desires or intentions. On the one hand, logics in the BDI tradition (belief-desire-intention) [12] make them explicit in the language, e.g. one can express *agent a* *has goal* φ; in the planning tradition, though, one only makes explicit their propositional content φ (what makes φ a goal is just its membership to the set of goals). Here we adopt the second (and less expressive) representation of goals.

In the present contribution, we describe a system for planning that accepts arbitrary epistemic formulas (e.g. common knowledge) as goals or state descriptions, and with ontic/epistemic actions given by Kripke-like action models. The language of LCC logics (used to this end) is further extended with action composition $\otimes$ and choice $\cup$, in order to study planning with non-deterministic actions. In this sense, we slightly generalize on previous results in [9] and [10], by dropping a technical restriction on the precondition of non-deterministic actions, and proposing slightly different plan structures. In summary, we define a breadth first search (BFS) algorithm for strong planning in the extended LCC logics. This search method is proved to be sound and complete: its outputs are (logically) successful plans and if such a successful plan exists, the algorithm terminates with some such solution. Finally, this algorithm easily extends to optimal plan search when each action is assigned some cost for its execution.

Motivating Example. Our aim, then, is to endow LCC logic based agents with planning capacities for this logic, so they can achieve their goals in scenarios where other agents have similar cognitive and acting abilities. In particular, LCC planning seems necessary for an agent whose goals consist in (or depend on) a certain distribution of knowledge and ignorance among agents. To illustrate the kind of rational behavior an LCC planner can exhibit, consider the following example:

Example 1. Agent a placed a bet with agent b that the next coin toss would be heads (h). Agent a knows she can toss the coin and detect its outcome, or flip the coin, without agent b knowing about it. Given a sensing action that tells a whether h holds or not, a successful plan seems to be: toss the coin; if sense that h, then show h to b; otherwise flip the coin and show h.

2 Related Work

Among logics for action guidance, the family of BDI [12] and related logics for intention are possibly the more popular. While these logics usually allow for considerable expressivity w.r.t. motivational attitudes (and their interaction with beliefs), they are not completely understood at a syntactic level. In fact, the use of planning methods has been suggested for an implementation of a BDI architecture. In particular, [4] suggest the use of LCC planning for the corresponding fragment of BDI logic. In this work [4] (see also [8]), the authors study LCC forward planning based on the semantics of update models; the BFS search algorithm is shown to be complete for LCC forward planning and in addition this problem (LCC forward planning) is shown to be semi-decidable in the general multi-agent case. An extension for (single-agent) conditional plan search in AND/OR-graphs can be found in [1]. The present work addresses the multi-agent case using instead a backward search approach (in OR-graphs). The motivation for this lies in the nature of communicative actions: while forward search is based on actions that are *executable*, backward search focuses on actions that are *relevant to the current goals*. This makes a difference in LCC since many actions will exist which are everywhere executable, so forward planning will typically face the state explosion problem.

Another work along the same lines is [2] (and related papers) where regression methods are introduced for the fragment of LCC without common knowledge. Regression can also be used as a (non-incremental) planning algorithm for LCC.

3 Preliminaries: The Logics of Communication and Change

Logics for agents with epistemic and communicative abilities have been developed in the recent years, ranging from epistemic logic [7] (for individual, group or common belief or knowledge), to logics of announcements [3], [15] (public or private, honest or dishonest), and finally to incorporating ontic actions (i.e. world-changing actions) [16]. All this has been unified within the single framework of Logics of Communication and Change [13], or LCC logics, formally a dynamic extension of epistemic logic using action models. This work proposes a general (translation-based) method that provides a complete axiomatization of an LCC logic from the specification of its particular action model. Since LCC logics are built by adding dynamic action models U on top of $\mathrm{E{\cdot}PDL}$ (propositional dynamic logic PDL under an epistemic reading), we recall PDL first.

3.1 Epistemic PDL

Propositional dynamic logic [6] is a modal logic for reasoning about programs, with modalities $[\pi]$ (and $\langle\pi\rangle$) expressing *after executing program* π *it is necessarily (resp. possibly) the case that*. Using a semantics for programs π based on relations R_π (between the internal states of a machine running the program), the PDL programs π are built from basic actions a and the program constructors of composition $a; b$ (*do* a *then* b), choice $a \cup b$ (*either do* a *or* b), test $?\varphi$ (*test* φ*, and proceed if true or terminate*) and iteration a^* (*do* a*; repeat*) (the Kleene-star for the reflexive transitive closure). It was later suggested [13] that the dynamic modalities of PDL naturally admit an epistemic interpretation as well, called $\mathrm{E{\cdot}PDL}$, if we read the basic "program" $[a]$ as the modality for agent a's knowledge or belief; that is, $[a]\varphi$ reads: a *knows* φ, or a *believes* φ; and $\langle a\rangle$ reads: a *considers it possible that* φ. Note that epistemic PDL does not distinguish between knowledge and belief, as usually understood by the S5 and KD45 modal logics, respectively. And thus, at the abstract level of PDL we will indistinctly refer to $[a]$ as knowledge or belief. Within a particular model, though, we can properly refer to one or the other depending on the semantic properties, e.g. whether $[a]\varphi \to \varphi$ holds, etc.

Definition 1. *The language of* $\mathrm{E{\cdot}PDL}$*, denoted by* $\mathcal{L}_{\mathrm{E{\cdot}PDL}}$*, for a given sets of atoms* $p \in \mathsf{Var}$ *and agents* $a \in \mathsf{Ag}$ *consists of the following formulas* φ *and programs* π*:*

$$\varphi ::= p \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid [\pi]\varphi \qquad\qquad \pi ::= a \mid ?\varphi \mid \pi_1; \pi_2 \mid \pi_1 \cup \pi_2 \mid \pi^*$$

The symbols $\bot, \vee, \leftrightarrow$ and $\langle\pi\rangle$ are defined from the above as usual. Under the epistemic reading, the PDL program constructors allow us to model, among others,

$[a; b]$	*agent* a *believes that* b *believes that*	(nested belief)
$[B]$, or $[a \cup b]$	*agents in* $B = \{a, b\}$ *believe that*	(group belief)
$[B^*]$, or $[(a \cup b)^*]$	*it is common knowledge among* B *that*	(comm. knowl.)

An E·PDL model $M = (W, \langle R_a \rangle_{a \in \mathsf{Ag}}, V)$ does, as usual, contain a set of worlds W, a relation R_a in W for each agent a, and an evaluation $V : \mathsf{Var} \to \mathcal{P}(W)$).

Definition 2. *The semantics of* E·PDL *consists of models* $M = (W, \langle R_a \rangle_{a \in \mathsf{Ag}}, V)$*, containing: a set of worlds W, a relation R_a in W for each agent a, and an evaluation* $V : \mathsf{Var} \to \mathcal{P}(W)$*. This map V extends to a map* $[\![\varphi]\!]^M$ *for each formula* φ *in* $\mathcal{L}_{\mathrm{E \cdot PDL}}$*:*

$$\begin{array}{rlrl}
[\![\top]\!]^M &= W & [\![a]\!]^M &= R(a) \\
[\![p]\!]^M &= V(p) & [\![?\varphi]\!]^M &= \mathsf{Id}_{[\![\varphi]\!]} \\
[\![\neg\varphi]\!]^M &= W \setminus [\![\varphi]\!]^M & [\![\pi_1;\pi_2]\!]^M &= [\![\pi_1]\!]^M \circ [\![\pi_2]\!]^M \\
[\![\varphi_1 \wedge \varphi_2]\!]^M &= [\![\varphi_1]\!]^M \cap [\![\varphi_2]\!]^M & [\![\pi_1 \cup \pi_2]\!]^M &= [\![\pi_1]\!]^M \cup [\![\pi_2]\!]^M \\
 & & [\![\pi^*]\!]^M &= ([\![\pi]\!]^M)^*
\end{array}$$

$$[\![[\pi]\varphi]\!]^M = \{w \in W \mid \forall v((w,v) \in [\![\pi]\!]^M \Rightarrow v \in [\![\varphi]\!]^M\}$$

where $\circ$ *and* * *are the composition and reflexive transitive closure of relations.*

Notice in particular that $[\![?\bot]\!]^M = \varnothing$ and $[\![?\top]\!]^M = \mathsf{Id}_W$ (the identity relation on W). We recall the axioms/rules of E·PDL that provide a sound and complete axiomatization:

(K)	$\vdash [\pi](\varphi \to \psi) \to ([\pi]\varphi \to [\pi]\psi)$
$(test)$	$\vdash [?\varphi_1]\varphi_2 \leftrightarrow (\varphi_1 \to \varphi_2)$
$(sequence)$	$\vdash [\pi_1;\pi_2]\varphi \leftrightarrow [\pi_1][\pi_2]\varphi$
$(choice)$	$\vdash [\pi_1 \cup \pi_2]\varphi \leftrightarrow [\pi_1]\varphi \wedge [\pi_2]\varphi$
(mix)	$\vdash [\pi^*]\varphi \leftrightarrow \varphi \wedge [\pi][\pi^*]\varphi$, and
$(induction)$	$\vdash \varphi \wedge [\pi^*](\varphi \to [\pi]\varphi)) \to [\pi^*]\varphi$.
$(Modus\ ponens)$	From $\vdash \varphi_1$ and $\vdash \varphi_1 \to \varphi_2$, infer φ_2,
$(Necessitation)$	From $\vdash \varphi$, infer $\vdash [\pi]\varphi$.

3.2 Action Models U, e

An LCC logic will add to an E·PDL language a set of modalities $[\mathsf{U}, \mathsf{e}]$ for each pointed action model U, e with distinguished (actual) action e. These new operators $[\mathsf{U}, \mathsf{e}]$ read *after each execution of action* e *it is the case that.* An action model is a tuple $\mathsf{U} = (\mathsf{E}, \mathsf{R}, \mathsf{pre}, \mathsf{post})$ containing

- $\mathsf{E} = \{\mathsf{e}_0, \ldots, \mathsf{e}_{n-1}\}$, a set of actions
- $\mathsf{R} : \mathsf{Ag} \to (\mathsf{E} \times \mathsf{E})$, a map assigning a relation R_a to each agent $a \in \mathsf{Ag}$
- $\mathsf{pre} : \mathsf{E} \to \mathcal{L}_{\mathrm{PDL}}$, a map assigning a precondition $\mathsf{pre}(\mathsf{e})$ to each action e
- $\mathsf{post} : \mathsf{E} \times \mathsf{Var} \to \mathcal{L}_{\mathrm{PDL}}$, a map assigning a post-condition $\mathsf{post}(\mathsf{e})(p)$, or $p^{\mathsf{post}(\mathsf{e})}$, to each $\mathsf{e} \in \mathsf{E}$ and $p \in \mathsf{Var}$

Let us fix the above enumeration $\mathsf{e}_0, \ldots, \mathsf{e}_{n-1}$ which will be used throughout the paper, unless stated otherwise. During plan search, in particular, when we refine a plan with some new action, the different alternatives will be considered according to this ordering: the refinement with e_0 will be considered before the refinement with e_1, and so on.

Definition 3. *The language of the* LCC*-logic for an action model* U *extends the formulas of* E·PDL *(for the same set of variables* Var *and agents* Ag*) with modalities for pointed action models* U, e*, giving the following sets of formulas* φ *and programs* π*:*

$$\varphi ::= p \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid [\pi]\varphi \mid [\mathsf{U}, \mathsf{e}]\varphi \qquad \pi ::= a \mid ?\varphi \mid \pi_1 ; \pi_2 \mid \pi_1 \cup \pi_2 \mid \pi^*$$

The new modalities $[\mathsf{U}, \mathsf{e}]\varphi$ represent *"after the execution of* e*,* φ *will hold"*. The semantics of LCC computes $M, w \models [\mathsf{U}, \mathsf{e}]p$ in terms of the product update of M, w and U, e. This product update is (again) an E·PDL pointed model $M \circ \mathsf{U}, (w, \mathsf{e})$, with

$$M \circ \mathsf{U} = (W', \langle R'_a \rangle_{a \in \mathsf{Ag}}, V') \qquad \text{where}$$

- the set W' consists of those worlds (w, e) such that $M, w \models \mathsf{pre}(\mathsf{e})$ (so executing e will lead to the corresponding state (w, e).)
- the relation $(w, \mathsf{e}) R'_a (v, \mathsf{f})$ holds iff both $w R_a v$ and $\mathsf{e} \mathsf{R}_a \mathsf{f}$ hold; and
- the valuations are $V'(p) = \{(w, \mathsf{e}) \in W' \mid M, w \models \mathsf{post}(\mathsf{e})(p)\}$, (the truth-value of p after e depends on that of $\mathsf{post}(\mathsf{e})(p)$ before the execution)

An updated model $(W', \langle R'_a \rangle_{a \in \mathsf{Ag}}, V')$ will be denoted $(W^{M \circ \mathsf{U}}, \langle R_a^{M \circ \mathsf{U}} \rangle_{a \in \mathsf{Ag}}, V^{M \circ \mathsf{U}})$.

Example 2. Several types of announcement (that φ by agent a) can be expressed. As purely epistemic actions, they are assigned the trivial post-condition $\mathsf{post}(\cdot)(p) = p$.

- a (successful) *truthful* announcement to sub-group $X \subseteq \mathsf{Ag}$, denoted $[\mathsf{U}, \varphi!^a_X]$, with

$$\mathsf{pre}(\varphi!^a_X) = \varphi \quad \text{and} \quad \mathsf{R}_b(\varphi!^a_X, \mathsf{e}) \Leftrightarrow \begin{cases} \mathsf{e} = \varphi!^a_X & \text{if } b \in X \cup \{a\} \\ \mathsf{e} \in \{\varphi!^a_X, \mathsf{skip}\} & \text{if } b \notin X \cup \{a\} \end{cases}$$

- a (successful) *lying* announcement to X, denoted $\mathsf{U}, \varphi\dagger^a_X$, is defined by the same accessibility relation but with precondition $\mathsf{pre}(\varphi\dagger^a_X) = \neg\varphi$.

(Here skip is the null action defined $\mathsf{pre}(\mathsf{skip}) = \top$, and $\mathsf{post}(\mathsf{skip})(p) = p$.)

From here on we assume that post-conditions $\mathsf{post}(\mathsf{e})(p)$ are restricted to the elements $\{p, \top, \bot\}$, rather than $\mathsf{post}(\mathsf{e})(p)$ being an arbitrary formula. This restriction was studied in [16] for logics similar to LCC, with epistemic modalities for agents $[a]$ and group common knowledge $[B^*]$ for $B \subseteq \mathsf{Ag}$. The authors show that the logic resulting after this restriction on post-conditions is as expressive as the original where post-conditions are arbitrary formulas.

Later, we recover this expressivity by introducing a non-deterministic choice operator for actions. Let us remark that choice is more general than arbitrary post-conditions φ, since it can model the toss of a coin without describing which conditions φ would result in the coin landing heads.

This restriction makes the truth-value of p after e to be either of the following:

if $\mathsf{post}(\mathsf{e})(p) = \ldots$	then the truth-value of p after e is ...
$\top$	true (since $\top$ is always true, hence true before e)
p	its truth-value before the execution of e
$\bot$	false (since $\bot$ is always false)

3.3 Logics of Communication and Change

The PDL semantics $[\![\cdot]\!]$ for E·PDL-formulas extends to a semantics for LCC by adding:

$$[\![[\mathsf{U},\mathsf{e}]\varphi]\!]^M = \{w \in W \mid \text{ if } M, w \models \mathsf{pre}(\mathsf{e}) \text{ then } (w,\mathsf{e}) \in [\![\varphi]\!]^{M\circ\mathsf{U}}\}.$$

In [13], the authors define program transformers $T^{\mathsf{U}}_{ij}(\pi)$ that provide a mapping between E·PDL programs (see Def. 4). Given any combination of actions in a model U the transformers provide a complete set of reduction axioms, reducing LCC to E·PDL. In a sketch, the U, e-modalities are pushed inside the formula, up to the case $[\mathsf{U},\mathsf{e}]p$.

Definition 4. *Let an action model* U *with* $\mathsf{E} = \{\mathsf{e}_0,\ldots,\mathsf{e}_{n-1}\}$ *be given. The* program transformer *function* T^{U}_{ij} *is defined as follows:*

$$\begin{aligned}
T^{\mathsf{U}}_{ij}(a) &= \begin{cases} ?\mathsf{pre}(e_i); a & \text{if } \mathsf{e}_i\mathsf{R}(a)\mathsf{e}_j, \\ ?\bot & \text{otherwise}\end{cases}\\
T^{\mathsf{U}}_{ij}(?\varphi) &= \begin{cases} ?(\mathsf{pre}(\mathsf{e}_i) \wedge [\mathsf{U},\mathsf{e}_i]\varphi), & \text{if } i = j \\ ?\bot & \text{otherwise}\end{cases}\\
T^{\mathsf{U}}_{ij}(\pi_1;\pi_2) &= \textstyle\bigcup_{k=0}^{n-1}(T^{\mathsf{U}}_{ik}(\pi_1); T^{\mathsf{U}}_{kj}(\pi_2))\\
T^{\mathsf{U}}_{ij}(\pi_1\cup\pi_2) &= T^{\mathsf{U}}_{ij}(\pi_1)\cup T^{\mathsf{U}}_{ij}(\pi_2)\\
T^{\mathsf{U}}_{ij}(\pi^*) &= K^{\mathsf{U}}_{ijn}(\pi).\\
&\quad \text{where } K^{\mathsf{U}}_{ijn} \text{ is inductively defined as follows:}\\
K^{\mathsf{U}}_{ij0}(\pi) &= \begin{cases} ?\top \cup T^{\mathsf{U}}_{ij}(\pi) & \text{if } i = j \\ T^{\mathsf{U}}_{ij}(\pi) & \text{otherwise}\end{cases}\\
K^{\mathsf{U}}_{ij(k+1)}(\pi) &= \begin{cases} (K^{\mathsf{U}}_{kkk}(\pi))^* & \text{if } i = k = j \\ (K^{\mathsf{U}}_{kkk}(\pi))^*; K^{\mathsf{U}}_{kjk}(\pi) & \text{if } i = k \neq j \\ K^{\mathsf{U}}_{ikk}(\pi); (K^{\mathsf{U}}_{kkk}(\pi))^* & \text{if } i \neq k = j \\ K^{\mathsf{U}}_{ijk}(\pi) \cup (K^{\mathsf{U}}_{ikk}(\pi); (K^{\mathsf{U}}_{kkk}(\pi))^*; K^{\mathsf{U}}_{kjk}(\pi)) & \text{if } i \neq k \neq j\end{cases}
\end{aligned}$$

A calculus for the LCC logic of a given action model U is given by the following:

the axioms and rules for E·PDL	
$[\mathsf{U},\mathsf{e}]\top \leftrightarrow \top$	(top)
$[\mathsf{U},\mathsf{e}]p \leftrightarrow (\mathsf{pre}(\mathsf{e}) \to \mathsf{post}(e)(p))$	(atoms)
$[\mathsf{U},\mathsf{e}]\neg\varphi \leftrightarrow (\mathsf{pre}(\mathsf{e}) \to \neg[\mathsf{U},\mathsf{e}]\varphi)$	(negation)
$[\mathsf{U},\mathsf{e}](\varphi_1 \wedge \varphi_2) \leftrightarrow ([\mathsf{U},\mathsf{e}]\varphi_1 \wedge [\mathsf{U},\mathsf{e}]\varphi_2)$	(conjunction)
$[\mathsf{U},\mathsf{e}_i][\pi]\varphi \leftrightarrow \bigwedge_{j=0}^{n-1}[T^{\mathsf{U}}_{ij}(\pi)][\mathsf{U},\mathsf{e}_j]\varphi$	(E·PDL-programs)
if $\vdash \varphi$ then $\vdash [\mathsf{U},\mathsf{e}]\varphi$	(Necessitation)

The completeness for this calculus is shown by reducing LCC to E·PDL. The translation, simultaneously defined for formulas $t(\cdot)$ and programs $r(\cdot)$ is

$$
\begin{array}{ll}
t(\top) & = \top \\
t(p) & = p \\
t(\neg\varphi) & = \neg t(\varphi) \\
t(\varphi_1 \wedge \varphi_2) & = t(\varphi_1) \wedge t(\varphi_2) \\
t([\pi]\varphi) & = [r(\pi)]t(\varphi) \\
t([\mathsf{U},\mathsf{e}]\top) & = \top \\
t([\mathsf{U},\mathsf{e}]p) & = t(\mathsf{pre}(\mathsf{e})) \to p^{\mathsf{post}(\mathsf{e})} \\
t([\mathsf{U},\mathsf{e}]\neg\varphi) & = t(\mathsf{pre}(\mathsf{e})) \to \neg t([\mathsf{U},\mathsf{e}]\varphi) \\
t([\mathsf{U},\mathsf{e}](\varphi_1 \wedge \varphi_2)) & = t([\mathsf{U},\mathsf{e}]\varphi) \wedge t([\mathsf{U},\mathsf{e}]\varphi_2) \\
t([\mathsf{U},\mathsf{e}_i][\pi]\varphi) & = \bigwedge_{j=0}^{n-1}[T_{ij}^{\mathsf{U}}(r(\pi))]t([\mathsf{U},\mathsf{e}_j]\varphi) \\
t([\mathsf{U},\mathsf{e}][\mathsf{U},\mathsf{e}']\varphi) & = t([\mathsf{U},\mathsf{e}]t([\mathsf{U},\mathsf{e}']\varphi))
\end{array}
\qquad
\begin{array}{ll}
r(a) & = a \\
r(B) & = B \\
r(?\varphi) & = ?t(\varphi) \\
r(\pi_1;\pi_2) & = r(\pi_1);r(\pi_2) \\
r(\pi_1 \cup \pi_2) & = r(\pi_1) \cup r(\pi_2) \\
r(\pi^*) & = (r(\pi))^*
\end{array}
$$

These translation functions t and r will be part of the backward planning algorithms presented in the next sections.

4 Backward Deterministic Planning in LCC

We proceed to introduce search algorithms for planning domains expressible in some LCC logic. In this section we study the deterministic case. The first step is to adapt the basic elements of planning systems:

- the goal and initial state are formulas of E·PDL (the static fragment of LCC).
- the set of available actions $A \subseteq \mathsf{E}$, among those in the action model U
- an available action is a pointed action model U,e where $\mathsf{e} \in A$

A deterministic plan is an executable sequence of actions in A that necessarily leads from any initial state to some goal state.

As we said, the proposed search methods for LCC planning are based on the above reduction of LCC into E·PDL. Given a (goal) formula φ for the current plan π and some action e, we want to compute the minimal conditions ψ (upon an arbitrary state) that would make φ to hold after e. After refinement of π with e, this minimal condition ψ will be the new goal replacing φ. More formally, we say $\psi \in \mathcal{L}_{\mathrm{PDL}}$ is the *weakest precondition* for a formula $[\mathsf{U},\mathsf{e}]\varphi$, iff (in LCC)

$$\models \psi \leftrightarrow [\mathsf{U},\mathsf{e}]\varphi.$$

This notion generalizes the definition in classical planning of open goals after refinement. Recall in classical planning, the different variables (or literals) p, q are logically independent, so the total effects of an action simply decompose into the individual effects w.r.t. each variable.

The weakest precondition for e to cause an arbitrary formula φ is the formula:

$$t([\mathsf{U},\mathsf{e}]\varphi \wedge \langle \mathsf{U},\mathsf{e}\rangle\top)$$

extracted from the reduction to E·PDL by way of translation using t, r. Indeed, the correctness of the translation based on t, r makes

$$\models t([\mathsf{U}, \mathsf{e}]\varphi \wedge \langle \mathsf{U}, \mathsf{e}\rangle\top) \leftrightarrow [\mathsf{U}, \mathsf{e}]\varphi \wedge \langle \mathsf{U}, \mathsf{e}\rangle\top$$

These functions t, r can then be seen as goal-transforming functions: a current goal φ is mapped into $t([\mathsf{U}, \mathsf{e}]\varphi \wedge \langle \mathsf{U}, \mathsf{e}\rangle\top)$, which becomes the new goal after we refine the plan with e.

Definition 5. *Given some* LCC *logic for an action model* U*, a* planning domain *is a triple* $\mathbb{M} = (\varphi_T, A, \varphi_G)$*, where* φ_T, φ_G *are consistent* E·PDL *formulas describing, resp., the initial and goal states; and* $A \subseteq \mathsf{E}$ *is the subset of a actions available to the agent.*

A solution *to* $\mathbb{M}$ *is a sequence* $\mathsf{f}_1, \ldots, \mathsf{f}_m \in A^{<\omega}$ *of actions in A, such that*

$$\models \varphi_T \to [\mathsf{U}, \mathsf{f}_1] \ldots [\mathsf{U}, \mathsf{f}_m]\varphi_G \quad \textit{and} \quad \models \varphi_T \to \langle \mathsf{U}, \mathsf{f}_1\rangle \ldots \langle \mathsf{U}, \mathsf{f}_m\rangle\top$$

The subset $A \subseteq \mathsf{E}$ denotes those actions that are actually available to our planner-executor agent a. The reason to distinguish A from E is that some other agent $b \in \mathsf{Ag}$ might attribute our agent a some abilities which a does not actually possess, or b might fail to attribute a some of her actual abilities (and attribute her instead a decaffeinated version of some of these abilities). Thus, on the one hand, we want to distinguish the beliefs of b after an execution of some action e as depending on how b interpret this action e. On the other, we want to make explicit which abilities does our agent possess, in order to build realistic plans.

From here on, π will denote a deterministic plan, i.e. a sequence of actions e in decreasing order of execution (rather than an arbitrary epistemic PDL program as before). Plans are denoted by a pair (*action sequence*, *open goals*)

Definition 6. *Given some planning domain* $\mathbb{M} = (\varphi_T, A, \varphi_G)$*, the (initial) empty plan is the pair* $\pi_\varnothing = (\varnothing, \varphi_G)$ *and if* $\pi = (\pi, \varphi_{\mathsf{goals}(\pi)})$ *is a plan, then* $\pi(\mathsf{e}) = (\pi^\frown\langle \mathsf{e}\rangle, \varphi_{\mathsf{goals}}(\pi(\mathsf{e})))$*, defined by the goal* $\varphi_{\mathsf{goals}(\pi(\mathsf{e}))} = t([\mathsf{U}, \mathsf{e}]\varphi_{\mathsf{goals}(\pi)} \wedge \langle \mathsf{U}, \mathsf{e}\rangle\top)$*, is also a plan. A plan* π *is a* leaf *iff* $\varphi_{\mathsf{goals}(\pi(\mathsf{e}))}$ *is inconsistent, or* $\models \varphi_{\mathsf{goals}(\pi(\mathsf{e}))} \to \varphi_{\mathsf{goals}(\pi)}$*.*

Leafs are plans not worth considering, either because (a) when we add the last action refinement e, the resulting plan demands an inconsistent precondition $\varphi_{\mathsf{goals}(\pi(\mathsf{e}))}$ (and hence the plan cannot be executed) or (b) because e does not contribute to delete part of the previous goals $\varphi_{\mathsf{goals}(\pi)}$. The search space for the proposed planning algorithm (see below) is the set sequences $(\mathsf{f}_1, \ldots, \mathsf{f}_m) \in A^{<\omega}$. (These sequences are read in decreasing order of execution, i.e. as the sequence of operators $\mathsf{U}, \mathsf{f}_m, \ldots, \mathsf{U}, \mathsf{f}_1$.) Then, the planning algorithm explores just a fragment of this space, since it will not bother to generate/evaluate further refinements of leaf plans. A breadth first search (henceforth, BFS) algorithm for deterministic planning in LCC is given in Figure 1.

Actions $\mathsf{e} \in \mathsf{E}$, as defined above, are deterministic, in the sense that $\models [\mathsf{U}, \mathsf{e}]\varphi \vee \psi \leftrightarrow ([\mathsf{U}, \mathsf{e}]\varphi \vee [\mathsf{U}, \mathsf{e}]\psi)$. Thus, deterministic plans consist of actions $\mathsf{e} \in \mathsf{E}$ in our current action models U. Later we will extend LCC with composition $\otimes$ and choice $\cup$ to study the non-deterministic case. There we will fully recover the expressivity of actions

```
Input : $\mathbb{M} = (\varphi_T, A, \varphi_G)$.
LET Plans $= \langle \pi_\varnothing \rangle$ and $\pi = \pi_\varnothing$
WHILE $\not\models \varphi_T \to \varphi_{\mathsf{goals}(\pi)}$
    DELETE $\pi$ FROM Plans
    SET Plans = Plans$^\frown\langle\, \pi(\mathsf{e}) \mid \mathsf{e} \in A$ and $\pi(\mathsf{e})$ not a leaf $\rangle$
    SET $\pi =$ the first element of Plans
Output : $\pi$    (i.e. the sequence $[\mathsf{U}, \mathsf{e}_1] \ldots [\mathsf{U}, \mathsf{e}_k]$)
```

Fig. 1. BFS algorithm for backward deterministic planning in LCC

defined by arbitrary post-conditions $p^{\mathsf{post}(\mathsf{e})} = \varphi$ of [13], i.e. actions with conditional effects: *if* φ *then (after* e*)* p. The first contribution of this paper is the following result:[1]

Theorem 1. *BFS is sound and complete for* LCC *backward planning: the output* π *of the algorithm in Fig. 1 is a solution for* $(\varphi_T, A, \varphi_G)$*; conversely, if a solution exists, then the algorithm terminates (with a solution output).*

5 An Extension of LCC with Action Composition and Choice

In this section we propose an extension of LCC logic with bounded composition and choice, denoted $\mathrm{LCC}_{\cup\otimes n}$. To this end, we first expand any LCC logic with the composition of at most n actions, denoted $\otimes n$, and later we add choice $\cup$. Both operations map two actions e, f to a new action denoted, resp., $\mathsf{e} \otimes \mathsf{f}$ and $\mathsf{e} \cup \mathsf{f}$, interpreted as follows:

– $\mathsf{e} \otimes \mathsf{f}$ models an execution of e followed by an execution of f, and
– $\mathsf{e} \cup \mathsf{f}$ models non-deterministic actions: each execution of $\mathsf{e} \cup \mathsf{f}$ either instantiates as an execution of e or as an execution of f.

For the composition of actions, the resulting action models are shown equivalent to a bounded number of updates with the previous simple actions. The logic of the former action updates, denoted $\mathrm{LCC}_{\otimes n}$ reduces to the corresponding LCC logic.

Then we introduce choice $\cup$ into these models $\mathsf{U}^{\leq n}$. The semantics for non-deterministic actions $\mathsf{e} \cup \mathsf{f}$ is presented in terms of multi-pointed models (w, e) and (w, f), one for each possible realization of the former action. Again we extend the language and axioms accordingly for this logic $\mathrm{LCC}_{\cup\otimes n}$, and reduce this logic again to E·PDL. In the next section, we will study non-deterministic planning problems in terms of plan solutions expressible in this $\mathrm{LCC}_{\cup\otimes n}$ logics.

5.1 Update with the Product of n Actions in U^n

To define the composition of actions, we simply consider the product of an action model by itself, $\mathsf{U}_1 \otimes \cdots \otimes \mathsf{U}_k$, for each $k \leq n$. Here n denotes the maximum number of

[1] Proofs for results in this paper can be found at the first author's webpage `www.iiia.csic.es/en/individual/pere-pardo`.

compositions allowed in the resulting logic $\mathrm{LCC}_{\otimes n}$. An obvious requirement is that these action models are defined for the same set of variables Var and agents Ag.

We define first action models of the form $\mathsf{U}^n = \mathsf{U}_1 \otimes \cdots \otimes \mathsf{U}_n$ and study them from a semantic point of view. This action model U^n just contains arbitrary products of exactly n actions: $\mathsf{f}_1 \otimes \cdots \otimes \mathsf{f}_n$.

Note that, in the next definition, the pre' functions of the product action model U^n are defined in terms of the corresponding functions pre from U, and pre' from $\mathsf{U}^2, \ldots, \mathsf{U}^{n-1}$. From here on, we let $\overrightarrow{\mathsf{f}}$ denote some sequence $\mathsf{f}_1 \otimes \cdots \otimes \mathsf{f}_k$, also written $\mathsf{f}_1, \ldots, \mathsf{f}_k$, for an appropriate k.

Definition 7. *Let* $\mathsf{U} = (\mathsf{E}, \mathsf{R}, \mathsf{pre}, \mathsf{post})$ *be an action model. We define the* product action model

$$\mathsf{U}^n \;=\; (\mathsf{E}', \mathsf{R}', \mathsf{pre}', \mathsf{post}')$$

inductively as follows:

$$\begin{aligned}
\mathsf{E}' &= \mathsf{E}^n = \{(\mathsf{f}_1, \ldots, \mathsf{f}_n) \mid \mathsf{f}_1, \ldots, \mathsf{f}_n \in \mathsf{E}\} \\
\mathsf{R}'_a &= \{\langle(\mathsf{e}, \ldots, \mathsf{e}'), (\mathsf{f}, \ldots, \mathsf{f}')\rangle \mid \mathsf{e}\mathsf{R}_a\mathsf{f} \textit{ and } \ldots \textit{ and } \mathsf{e}'\mathsf{R}_a\mathsf{f}'\} \\
\mathsf{pre}'(\mathsf{e} \otimes \mathsf{f}\,) &= \mathsf{pre}(\mathsf{e}) \wedge [\mathsf{U}, \mathsf{e}]\mathsf{pre}(\,\mathsf{f}\,) \qquad \textit{for the case } n = 2 \\
\mathsf{pre}'(\mathsf{f}_1 \otimes \overrightarrow{\mathsf{f}}\,) &= \mathsf{pre}(\mathsf{e}) \wedge [\mathsf{U}, \mathsf{e}]\mathsf{pre}(\,\overrightarrow{\mathsf{f}}\,) \\
\mathsf{post}'(\mathsf{f}_1 \otimes \cdots \otimes \mathsf{f}_n) &= \begin{cases} \mathsf{post}(\mathsf{f}_k)(p) & \textit{if } \mathsf{post}(\mathsf{f}_k)(p) \neq p = \\ & \qquad = \mathsf{post}(\mathsf{f}_{k+1})(p) = \ldots = \mathsf{post}(\mathsf{f}_n)(p) \\ \mathsf{post}(\mathsf{f}_1)(p) & \textit{if } \mathsf{post}(\mathsf{f}_1)(p) = \ldots = \mathsf{post}(\mathsf{f}_n)(p) = p \end{cases}
\end{aligned}$$

More formally, in Def. 7 we should rather define inductively (from the case $n = 2$)

$$\mathsf{pre}'(\mathsf{e} \otimes \overrightarrow{\mathsf{f}}\,) = \mathsf{pre}(\mathsf{e}) \wedge t([\mathsf{U}, \mathsf{e}]\mathsf{pre}'(\,\overrightarrow{\mathsf{f}}\,))$$

in order to comply with the condition upon action models: $\mathsf{pre} : \mathsf{E} \to \mathcal{L}_{\mathrm{PDL}}$. But for the sake of simplicity, we will keep the above notation. Also note that in U^n the product of actions $\mathsf{f} \otimes \cdots \otimes \mathsf{f}'$ treats p just as the latest action in this tuple satisfying $\mathsf{post}(\cdot)(p) \neq p$ (i.e. the latest action non-trivial w.r.t. p). Finally, observe that some combinations $\mathsf{e} \otimes \mathsf{f}$ in the product action model will never be applicable, e.g. when $\models [\mathsf{U}, \mathsf{e}]\neg\mathsf{pre}(\mathsf{f})$. For the purpose of planning, one can forget about the existence of these actions in the resulting model $\mathsf{U} \otimes \mathsf{U}$.

It can be seen by direct inspection that the so-called product action model U^n is indeed an action model, provided U is. Moreover, the update of an E·PDL model M by a product action model, say $\mathsf{U} \otimes \mathsf{U}$, reduces to a sequence of updates with the simpler action model, e.g. $(M \circ \mathsf{U}) \circ \mathsf{U}$. With more detail, updating a state w with an action $\mathsf{e} \otimes \mathsf{f}$ is semantically equivalent to updating w with e first, and then updating again with f. We first check this is the case for $\mathsf{U}^2 = \mathsf{U} \otimes \mathsf{U}$.

Lemma 1. *We have the following isomorphism*

$$M \circ (\mathsf{U} \otimes \mathsf{U}) \cong (M \circ \mathsf{U}) \circ \mathsf{U}.$$

This isomorphism extends to the valuations of arbitrary formulas and programs.

Corollary 1. *For each formula φ in the language of* $\mathsf{U} \otimes \mathsf{U}$*:*

$$(w, (\mathsf{e}, \mathsf{f})) \in [\![\varphi]\!]^{M \circ \mathsf{U}^2} \Leftrightarrow ((w, \mathsf{e}), \mathsf{f}) \in [\![\varphi]\!]^{(M \circ \mathsf{U}) \circ \mathsf{U}}$$

Also, note that the proof of Lemma 1 does not depend upon the assumption that the two action models are the same. More generally, we have the following result for different action models U, U'.

Corollary 2. *Let* U, U' *be action models defined on the same sets of variables* Var *and agents* Ag*. Then,* $M \circ (\mathsf{U} \otimes \mathsf{U}') \cong (M \circ \mathsf{U}) \circ \mathsf{U}'$*. Moreover,* $[\![\varphi]\!]^{M \circ (\mathsf{U} \otimes \mathsf{U}')} = [\![\varphi]\!]^{(M \circ \mathsf{U}) \circ \mathsf{U}'}$*, for each φ in the language of* $\mathsf{U} \otimes \mathsf{U}'$*.*

Before proceeding to the generalization of this lemma, we need the claim that the update with an action model U preserves isomorphisms.

Lemma 2. *If $M \cong M'$ are isomorphic epistemic models, and* U *is an action model, then* $M \circ \mathsf{U} \cong M' \circ \mathsf{U}$*.*

The previous Corollary 2 for the basic case $n = 2$ extends to an arbitrary finite number $n \geq 2$ of actions $\mathsf{f}_1, \ldots, \mathsf{f}_n$. That is, it extends to updates with products of arbitrary n actions taken from a given action model U.

Corollary 3. *We have* $M \circ \mathsf{U}^n \cong (M \circ \mathsf{U}_1) \cdots \circ \mathsf{U}_n$

5.2 Update with the Produce of $\leq n$ Actions in $\mathsf{U}^{\leq n}$

Finally, we can define the action model $\mathsf{U}^{\leq n}$ for the product of at most n actions (from a fixed action model U) in terms of the product action models $\mathsf{U}, \mathsf{U}^2, \ldots, \mathsf{U}^n$ previously defined.

Definition 8. *Let* U *be an action model and let* $\mathsf{U}_1 = \ldots = \mathsf{U}_n (= \mathsf{U})$ *be n different copies of* U*, denoted* $\mathsf{U}_k = (\mathsf{E}_k, \mathsf{R}_k, \mathsf{pre}_k, \mathsf{post}_k)$ *for each* $1 \leq k \leq n$*. We define* $\mathsf{U}^{\leq n} = (\mathsf{E}^{\leq n}, \mathsf{R}^{\leq n}, \mathsf{pre}^{\leq n}, \mathsf{post}^{\leq n})$ *as follows*

$$\mathsf{E}^{\leq n} = \bigcup_{k \leq n} \mathsf{E}_k \qquad \mathsf{pre}^{\leq n} = \bigcup_{k \leq n} \mathsf{pre}_k$$
$$\mathsf{R}^{\leq n}(a) = \bigcup_{k \leq n} \mathsf{R}_k(a) \quad \mathsf{post}^{\leq n} = \bigcup_{k \leq n} \mathsf{post}_k$$

In parallel, the sequence of at most n updates on a model M*, denoted*

$$(M \circ \mathsf{U}_1) \cdots \circ \mathsf{U}_{\leq n} = (W^{(M \circ \mathsf{U}_1) \cdots \circ \mathsf{U}_{\leq n}}, R^{(M \circ \mathsf{U}_1) \cdots \circ \mathsf{U}_{\leq n}}, V^{(M \circ \mathsf{U}_1) \cdots \circ \mathsf{U}_{\leq n}})$$

can be defined in a straightforward way from each product action model $(M \circ \mathsf{U}_1) \cdots \circ \mathsf{U}_k$*.*

$$W^{(M \circ \mathsf{U}_1) \cdots \circ \mathsf{U}_{\leq n}} = \bigcup_{k \leq n} W^{(M \circ \mathsf{U}_1) \cdots \circ \mathsf{U}_k}$$
$$R^{(M \circ \mathsf{U}_1) \cdots \circ \mathsf{U}_{\leq n}}(a) = \bigcup_{k \leq n} R^{(M \circ \mathsf{U}_1) \cdots \circ \mathsf{U}_k}(a)$$
$$V^{(M \circ \mathsf{U}_1) \cdots \circ \mathsf{U}_{\leq n}} = \bigcup_{k \leq n} V^{(M \circ \mathsf{U}_1) \cdots \circ \mathsf{U}_k}$$

It can be observed that $\mathsf{U}^{\leq n}$ is an action model; and also that $(M \circ \mathsf{U}_1) \cdots \circ \mathsf{U}_{\leq n}$ is an E·PDL model. Moreover, we can extend Corollary 3 to the present case:

Corollary 4. *If* U *is an action model, then*

$$M \circ \mathsf{U}^{\leq n} \cong (M \circ \mathsf{U}_1) \cdots \circ \mathsf{U}_{\leq n}$$

5.3 The Logic $\text{LCC}_{\otimes n}$ of the Action Model $\mathsf{U}^{\leq n}$

Let U be again a fixed action model and consider the corresponding product action model $\mathsf{U}^{\leq n}$. The language $\mathcal{L}_{\text{LCC}_{\otimes n}}$ of the logic $\text{LCC}_{\otimes n}$ for this action model $\mathsf{U}^{\leq n}$ is simply the language of LCC, but now with action modalities of the form $[\mathsf{U}^{\leq n}, \mathsf{f}_1 \otimes \cdots \otimes \mathsf{f}_k]$, for each $\mathsf{f}_1 \otimes \cdots \otimes \mathsf{f}_k \in \mathsf{E}^{\leq n}$ in the present action model $\mathsf{U}^{\leq n}$.

The semantics of updates with pointed action model $\mathsf{U}^{\leq n}$, $(\mathsf{f}_1, \ldots, \mathsf{f}_k)$ is also that of simple action models U. In the present case, we have

$$M, w \models [\mathsf{U}^n, \mathsf{e} \otimes \cdots \otimes \mathsf{f}]\varphi \text{ iff } M, w \models \mathsf{pre}(\mathsf{e} \otimes \cdots \otimes \mathsf{f}) \text{ implies}$$
$$M \circ \mathsf{U}^n, (w, (\mathsf{e} \otimes \cdots \otimes \mathsf{f})) \models \varphi$$

A complete axiom system for $\text{LCC}_{\otimes n}$, the logic of (bounded) product action models $\mathsf{U}^{\leq n}$, is obtained by extending the previous LCC axioms and rules with reduction axioms for the new product actions $\mathsf{f}_1 \otimes \cdots \otimes \mathsf{f}_k$.

the LCC reduction axioms and rules for $[\mathsf{U}, \mathsf{e}]\varphi$ formulas with $\varphi \in \mathcal{L}_{\text{LCC}_{\otimes n}}$

plus

$[\mathsf{U}^{\leq n}, (\mathsf{f}_1, \mathsf{f}_2, \ldots, \mathsf{f}_k)]\varphi \leftrightarrow [\mathsf{U}^{\leq n}, \mathsf{f}_1][\mathsf{U}^{\leq n}, (\mathsf{f}_2, \ldots, \mathsf{f}_k)]\varphi$ (Product)

Fig. 2. The axioms and rules for $\text{LCC}_{\otimes n}$

These axioms suffice for the introduction of composition. They induce again a translation function t which splits product actions $[\mathsf{U}^{\leq n}, \mathsf{e} \otimes \mathsf{f}]$ into a sequence of updates $[\mathsf{U}^{\leq n}, \mathsf{e}][\mathsf{U}^{\leq n}, \mathsf{f}]$ and proceeds as the translation for LCC for the remaining cases.

Lemma 3. *The product axiom is sound:*

$$\models [\mathsf{U}^{\leq n}, \mathsf{f}_1 \otimes \mathsf{f}_2 \otimes \cdots \otimes \mathsf{f}_k]\varphi \;\leftrightarrow\; [\mathsf{U}^{\leq n}, \mathsf{f}_1][\mathsf{U}^{\leq n}, \mathsf{f}_2 \otimes \cdots \otimes \mathsf{f}_n]\varphi$$

As we said, we extend the previous translation $\mathcal{L}_{\text{LCC}} \to \mathcal{L}_{\text{E}\cdot\text{PDL}}$ into a translation $\mathcal{L}_{\text{LCC}_{\otimes n}} \to \mathcal{L}_{\text{E}\cdot\text{PDL}}$ with the help of an additional clause

$$t([\mathsf{U}^{\leq n}, (\mathsf{f}_1, \mathsf{f}_2, \ldots, \mathsf{f}_n)]\varphi) = t([\mathsf{U}^{\leq n}, \mathsf{f}_1]t([\mathsf{U}^{\leq n}, (\mathsf{f}_2, \ldots, \mathsf{f}_k)]\varphi))$$

Theorem 2. *For each formula $\varphi \in \mathcal{L}_{\text{LCC}_{\otimes n}}$, we have*

$$\models \varphi \Leftrightarrow \vdash \varphi$$

Proof. ($\Leftarrow$) Soundness is established by the corresponding result for LCC in [13] plus the above result for the reduction axiom for product actions. These results also establish the correctness of the extended translation function: each formula in $\text{LCC}_{\otimes n}$ is logically equivalent (in LCC) to an E·PDL-formula $t(\varphi)$.

($\Rightarrow$) E·PDL is complete, and each formula in $\mathcal{L}_{\text{LCC}_{\otimes n}}$ is equivalent to some $\mathcal{L}_{\text{E}\cdot\text{PDL}}$ formula.

In addition, the LCC reduction axioms that would correspond to product modalities (except for the case of E·PDL-programs) are also sound.

Proposition 1. *Except for the* LCC *axiom on* E·PDL*-programs, the* LCC *reduction axioms are sound for product action modalities* $[\mathsf{U}^{\leq n}, \mathsf{f}_1 \otimes \cdots \otimes \mathsf{f}_k]$ *are sound.*

In contrast to the previous section on deterministic planning, we cannot fix a priori which action model $\mathsf{U}^{\leq n}$ (and logic) are we working with, when solving a given planning domain based on U. It is only after the planning algorithm terminates with a solution, that we (a posteriori) discover for which n the action model $\mathsf{U}^{\leq n}$ (actually $\mathsf{U}^{\cup\leq n}$, see below) will suffice to check that this plan is indeed a solution. Non-deterministic solutions are more naturally expressed if we further extend the logics $\mathrm{LCC}_{\otimes n}$ with non-deterministic choice.

5.4 $\mathbf{LCC}_{\cup\otimes n}$: Choice and Non-deterministic Actions

In this section we extend the LCC-logics of bounded composition with the operator choice, that maps some pairs of actions e, f into a new action $\mathsf{e} \cup \mathsf{f}$. The latter expression denotes an action with indeterminate effects: an execution of $\mathsf{e} \cup \mathsf{f}$ will turn either as an execution of e or as an execution of f. It is an external agent, the environment (nature) in principle, who chooses the particular outcome after each execution of $\mathsf{e} \cup \mathsf{f}$. (This is called *demonic* non-determinism, in opposition to so-called *angelic* non-determinism where the planner agent itself selects a course of actions e rather than another one f, if both are executable.) Choice will be indistinctly represented as follows E_d, $\{\mathsf{e}, \ldots, \mathsf{f}\}$ or $\mathsf{e} \cup \ldots \cup \mathsf{f}$.

The language of $\mathrm{LCC}_{\cup\otimes n}$ adds to that of $\mathrm{LCC}_{\otimes n}$ a clause for action modalities of the form

$$[\mathsf{U}^{\leq n}, \mathsf{E}_d]\varphi$$

where $\mathsf{E}_d \subseteq \mathsf{E}^{\leq n}$ is an arbitrary (but non-empty) set of product actions $(\mathsf{f}_1 \otimes \cdots \otimes \mathsf{f}_k)$. The new actions, say,

$$\begin{aligned} \mathsf{E}_d &= \{(\mathsf{f}_1 \otimes \cdots \otimes \mathsf{f}_k), \ldots, (\mathsf{f}'_1 \otimes \cdots \otimes \mathsf{f}'_{k'})\} \quad \text{are also denoted} \\ &= (\mathsf{f}_1 \otimes \cdots \otimes \mathsf{f}_k) \cup \ldots \cup (\mathsf{f}'_1 \otimes \cdots \otimes \mathsf{f}'_{k'}). \end{aligned}$$

The presence of post-conditions in LCC actions prevents us from modeling the new non-deterministic actions, e.g. $\mathsf{e} \cup \mathsf{f}$, as full-fledged actions in the action model (as we did for product $\mathsf{e} \otimes \mathsf{f} \in \mathsf{E}^{\leq n}$). The problem is that for actions like *tossing a coin*, the post-condition for *heads*, say the variable h, will be at each execution either $\top$ or $\bot$; hence the post-condition for h is not a unique formula, and post cannot be a map.

This contrasts with the match between $\mathsf{U}^{\leq n}$ and $\mathrm{LCC}_{\otimes n}$ above, and also with the purely epistemic action models [3]. In these logics, each action operator in the language is associated an element in the action model. In this sense, even if our set of actions in the model is the same $\mathsf{E}^{\leq n}$ that we had for $\mathrm{LCC}_{\otimes n}$ logics, each constructible non-deterministic plans will be shown "equivalent" to some E_d modality. For example, the plan -informally written as- $\mathsf{e} \otimes (\mathsf{f} \cup \mathsf{f}')$ will be associated the modality $[\mathsf{U}, (\mathsf{e} \otimes \mathsf{f}) \cup (\mathsf{e} \otimes \mathsf{f}')]$.

As suggested in [13] non-deterministic actions are introduced with the help of multi-pointed semantics.

Definition 9. *Given an epistemic model M and an action model U, let $W_d \subseteq W$ and $\mathsf{E}_d = \{\mathsf{f}_1, \ldots, \mathsf{f}_k\} \subseteq \mathsf{E}$. Then M, W_d and U, E_d are multi-pointed models. We define*

$$\begin{array}{lll} M, W_d \models \varphi & \textit{iff} & M, w \models \varphi \quad \textit{for each } w \in W_d \\ M, w \models [\mathsf{U}, \mathsf{E}_d]\varphi & \textit{iff} & M \circ \mathsf{U}, \{(w, \mathsf{f}), \ldots, (w, \mathsf{f}')\} \models \varphi \\ & & \textit{for each } (w, \mathsf{f}), \ldots, (w, \mathsf{f}') \in W^{M \circ \mathsf{U}} \textit{ with } \mathsf{f}, \ldots, \mathsf{f}' \in \mathsf{E}_d \end{array}$$

In other words, this semantics for $[\mathsf{U}, \mathsf{E}_d]$ modalities simply amounts to the semantics of the operators $[\mathsf{U}, \mathsf{f}]$ for each $\mathsf{f} \in \mathsf{E}_d$. That is,

$$M, w \models [\mathsf{U}, \mathsf{E}_d]\varphi \quad \text{iff} \quad \text{for each } f \in \mathsf{E}_d, \;\; M, w \models \mathsf{pre}(\mathsf{f}) \text{ implies } M \circ \mathsf{U}, (w, \mathsf{f}) \models \varphi$$

For the reasons pointed above, non-deterministic actions $\mathsf{e} \cup \mathsf{f}$ or E_d are not actions in the action model, only their components e and f are. In other words, the action model is just $\mathsf{U}^{\leq n}$. In summary, we just add the modalities $[\mathsf{U}, \mathsf{E}_d]$ and expand the semantics to the multi-pointed case, rather than expanding the action models themselves.

In [13], the additional reduction axiom listed next is suggested for non-deterministic choice. Here we add it to the previous system $\mathrm{LCC}_{\otimes n}$:

$$\boxed{\begin{array}{lr} \text{the reduction axioms and rules of } \mathrm{LCC}_{\otimes n} & \\ \qquad\qquad\qquad\qquad \text{plus} & \\ [\mathsf{U}, \mathsf{E}_d]\varphi \;\leftrightarrow\; \bigwedge_{\mathsf{e} \in \mathsf{E}_d} [\mathsf{U}, \mathsf{e}]\varphi & (\textit{choice}) \end{array}}$$

Fig. 3. The axioms and rules for $\mathrm{LCC}_{\cup\otimes n}$

It is straightforward that the reduction axiom (*choice*) for $[\mathsf{U}, \mathsf{E}_d]\varphi$ is sound w.r.t. the semantics above. This allows us to extend once more the translation function t from $\mathrm{LCC}_{\otimes n}$ to $\mathrm{LCC}_{\cup\otimes n}$ with the clause

$$t([\mathsf{U}^{\leq n}, \mathsf{e} \cup \ldots \cup \mathsf{f}]\varphi) \;=\; t([\mathsf{U}^{\leq n}, \mathsf{e}]\varphi) \wedge \ldots \wedge t([\mathsf{U}^{\leq n}, \mathsf{f}]\varphi)$$

The resulting translation function t splits the new modalities $[\mathsf{U}, \mathsf{E}_d]$ and then proceeds as in the case of $\mathrm{LCC}_{\otimes n}$. The soundness of the axiom (*choice*) preserves the soundness of the expanded translation function, again reducing the language of $\mathrm{LCC}_{\cup\otimes n}$ to that of E·PDL and giving the next completeness result.

Corollary 5. *The logic* $\mathrm{LCC}_{\cup\otimes n}$ *is sound and complete.*

Fact 1. *The* LCC *axioms for* $[\mathsf{U}, \mathsf{e}]$ *that do not involve preconditions* $\mathsf{pre}(\cdot)$ *are also sound for* $[\mathsf{U}, \mathsf{e} \cup \mathsf{f}]$ *modalities. That is, all the* LCC *axioms except for (*atoms*) and (*partial functionality*).*

Also notice that the executability of non-deterministic actions $\mathsf{e} \cup \mathsf{f}$ only requires that some action e or f (or both) is executable.

Lemma 4. *The following holds:* $\models \langle \mathsf{U}, \mathsf{E}_d \rangle \top \leftrightarrow \bigvee_{\mathsf{e} \in \mathsf{E}_d} \mathsf{pre}(\mathsf{e})$.

6 Non-deterministic Plans in LCC

Now we turn into non-deterministic planning, for planning domains containing actions with disjunctive effects are available to the agent, e.g.

$$\models [\mathsf{U}, \mathsf{f}_0 \cup \mathsf{f}_1]\, p \vee q, \quad \text{but with} \quad \not\models [\mathsf{U}, \mathsf{f}_0 \cup \mathsf{f}_1] p \;\; \text{and} \;\; \not\models [\mathsf{U}, \mathsf{f}_0 \cup \mathsf{f}_1] q$$

as given by the post-conditions postconditions $\mathsf{post}(\mathsf{f}_0)(p) = \mathsf{post}(\mathsf{f}_1)(q) = \top$, and $\mathsf{post}(\mathsf{f}_0)(q) = q$ and $\mathsf{post}(\mathsf{f}_1)(p) = p$).

In particular, we focus on strong non-deterministic planning. Recall a strong solution for a given planning domain is a plan such that all of its possible executions in the initial state lead to a goal state. Thus, ignoring preconditions, the above action $\mathsf{f}_0 \cup \mathsf{f}_1$ is a strong solution to $(\varphi_T, \{\mathsf{f}_0 \cup \mathsf{f}_1\}, \varphi_G)$, for the goal $\varphi_G = p \vee q$; and it is a weak solution when the goal is $\varphi_G = p$.)

Example 3. Consider the action *toss a coin*. This can be seen as a non-deterministic choice between the two deterministic actions of *toss heads* and *toss tails*. Let (resp.) toss_h and $\mathsf{toss}_{\neg h}$ denote these actions, with assigned post-conditions

$$\mathsf{post}(\mathsf{toss}_h) : h \longmapsto \top, \quad \text{and} \quad \mathsf{post}(\mathsf{toss}_{\neg h}) : h \longmapsto \bot$$

Note that the executing agent a cannot distinguish whether she executes toss_h or $\mathsf{toss}_{\neg h}$ (at least until the coin has landed and the agent proceeds to observe the result). This indistinguishability, formally given by $\mathsf{R}_a(\mathsf{toss}_h, \mathsf{toss}_{\neg h})$ and viceversa, is called *run-time* indistinguishability in [4]. Even if the agent intends the toss to result in heads (i.e. the agent intends toss_h), the action really available to a is

$$\mathsf{toss}_h \cup \mathsf{toss}_{\neg h} \qquad \text{computed as} \quad \bigcup \{\mathsf{e} \in \mathsf{E} \mid \mathsf{R}_a(\mathsf{toss}_h, \mathsf{e})\}$$

Randomness is not essential feature to non-deterministic actions, as the next example illustrates.

Example 4. Consider for instance, the action of pressing a button on the wall, which will switch the light *on* or *off* (the latter denoting $\neg$*on*). Let the corresponding deterministic actions be denoted on and off, defined by similar post-conditions:

$$\mathsf{post}(\mathsf{on}) : \mathit{on} \longmapsto \top, \quad \text{and} \quad \mathsf{post}(\mathsf{off}) : \mathit{on} \longmapsto \bot$$

In contrast to the coin example, these two actions have different (in fact, mutually inconsistent) preconditions:

$$\mathsf{pre}(\mathsf{on}) = \mathit{off} \quad \text{and} \quad \mathsf{pre}(\mathsf{off}) = \mathit{on}$$

Suppose first our executing agent a is blind (or blind-folded), so she cannot distinguish on from off at run-time (during execution). See Figure 4 (Top). Notice that $\mathsf{on} \cup \mathsf{off}$ has a trivial precondition: $\mathit{on} \vee \neg \mathit{on}$, given by $\mathsf{pre}(\mathsf{on}) \vee \mathsf{pre}(\mathsf{off})$.

Secondly, suppose instead that the agent can see (or has been told) whether the light is initially *on*, Figure 4(Mid). She knows which of the two actions $\mathsf{on}^\star$ or $\mathsf{off}^\star$ is executable (has a true precondition), so we can model them separately as two deterministic actions.

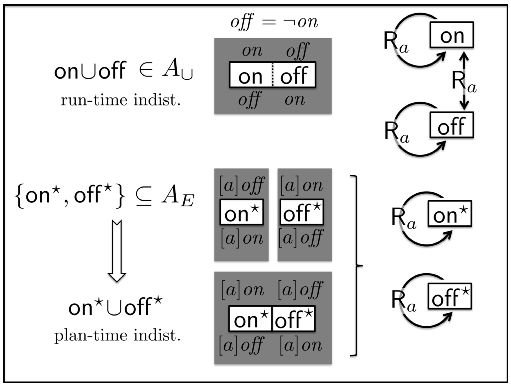

Fig. 4. (Top) A blind agent pressing the light button: $\mathsf{on} \cup \mathsf{off}$. (Mid) Switching the light on (while seeing): $\mathsf{on}^\star$. Similarly for $\mathsf{off}^\star$. (Bottom) Pressing the light button (while seeing), during the planning phase.

Along this line, the planner agent a might not know (during planning) whether she will find the light *on* or *off*, when she switches it (this being a planned action). Figure 4 (Bottom). This is called *plan-time* indistinguishability in [4], since only at execution time the agent will know whether whether she is going to turn the light on or off. This kind of actions, modeled as a choice $\mathsf{on}^\star \cup \mathsf{off}^\star$.

After this review on the effects of partial observability of states and actions, we proceed to the task of plan search. As these examples show, the previous notions of *available actions* A, *plan* and *solution* must be redefined for the present non-deterministic case. For the sake of simplicity, we will only consider the choice between two actions $\mathsf{f}_0 \cup \mathsf{f}_1$. The definitions and results in this paper can be generalized to the choice of finitely many actions $\mathsf{f}_0 \cup \mathsf{f}_1 \cup \cdots \cup \mathsf{f}_k$.

From here on, we abstract from any particular bound n upon the length of plans, so in the following we will just write the action model as U rather than as a fixed action model $\mathsf{U}^{\leq n}$. With this remark in mind, recall the set of action sequences definable in $\mathrm{LCC}_{\cup\otimes n}$ is any sequence of action modalities

$$[\mathsf{U}, \mathsf{E}_1] \ldots [\mathsf{U}, \mathsf{E}_k] \quad \text{(also written } (\mathsf{E}_0, \ldots, \mathsf{E}_k))$$

Concerning the basic actions available to the agent, we have: (1) a set A_e of actions e from E; and (2) a set of $A_\cup$ containing pairs of actions, denoted $\mathsf{e} \cup \mathsf{f}$, with again $\mathsf{e} \in \mathsf{E}$ and $\mathsf{f} \in \mathsf{E}$. For an example of these basic actions, we have $\mathsf{on}^\star$in A_E and $\mathsf{on}^\star \cup \mathsf{off}^\star$ and $\mathsf{toss}_h \cup \mathsf{toss}_{\neg h}$ in $A_\cup$. The following definition replace the old set A from Definition 5 by the new set $A_\mathsf{E} \cup A_\cup$.

Definition 10. *A* non-deterministic planning domain *in* U *is a triple*

$$\mathbb{M} = (\varphi_T, A_\mathsf{E} \cup A_\cup, \varphi_G)$$

with $A_\mathsf{E} \subseteq \mathsf{E}$*, and* $A_\cup \subseteq \mathsf{E} \times \mathsf{E}$.

Not all of the above action sequences $[\mathsf{U}, \mathsf{E}_1] \ldots [\mathsf{U}, \mathsf{E}_k]$ in the language of $\mathrm{LCC}_{\cup\otimes n}$ denote action sequences that are available to the agent according to a planning domain $\mathbb{M}$. The latter sub-class is defined next.

Definition 11. *We say* $[\mathsf{U}, \mathsf{e}]$ *and* $[\mathsf{U}, \mathsf{e} \cup \mathsf{f}]$ *are* $\mathbb{M}$*-sequences whenever* $\mathsf{e} \in A_{\mathsf{E}}$ *and* $\mathsf{e} \cup \mathsf{f}$ *in* $A_{\cup}$. *Moreover, if* $\mathsf{e}' \otimes \cdots \otimes \mathsf{e}''$ *and* $\mathsf{f}' \otimes \cdots \otimes \mathsf{f}''$ *are elements of* $A_{\mathsf{E}}^{<\omega}$ *and* $\mathsf{e} \cup \mathsf{f} \in A_{\cup}$ *satisfies* $(\mathsf{e}, \mathsf{f}), (\mathsf{f}, \mathsf{e}) \notin \mathsf{R}_a$, *then*

$$[\mathsf{U}, (\mathsf{e} \otimes \mathsf{e}' \otimes \cdots \otimes \mathsf{e}'') \cup (\mathsf{f} \otimes \mathsf{f}' \otimes \cdots \otimes \mathsf{f}'')] \quad \textit{is an } \mathbb{M}\textit{-sequence}$$

Finally, any finite sequence $[\mathsf{U}, \mathsf{E}_k] \ldots [\mathsf{U}, \mathsf{E}_1]$ *of* $\mathbb{M}$*-sequences is an* $\mathbb{M}$*-sequence.*

The idea of $\mathbb{M}$-sequences is to minimally constrain (within the limits of $\mathcal{L}_{\mathrm{LCC}_{\cup\otimes n}}$) how much freedom an agent is allowed after executing a non-deterministic action $\mathsf{e} \cup \mathsf{f}$ (while preserving epistemic control):

– if the components e and f are run-time indistinguishable according to R_a, the next action after executing $\mathsf{e} \cup \mathsf{f}$ must be uniquely specified (though it can be another non-deterministic action),
– if the components e and f are run-time distinguishable, one can execute alternative (deterministic) actions, say e' or f', depending on whether the execution of $\mathsf{e} \cup \mathsf{f}$ instantiated, resp., as e or as f.

Example 5. (Cont'd) Recall the sets of available actions $A_{\mathsf{E}} = \varnothing$ and $A_{\cup} = \{\mathsf{toss}_h \cup \mathsf{toss}_{\neg h}\}$ from Example 3. Read the tossing action as causing the coin to land into agent a's hand. And expand these sets with a sensing action in $A_{\cup}$ (feeling in your hand whether the coin landed heads) and a flip (into heads) action in A_{E}:

$$\begin{array}{lcc}
\mathsf{feel}_h \cup \mathsf{feel}_{\neg h} & \mathsf{pre}(\mathsf{feel}_h) = h & \mathsf{pre}(\mathsf{feel}_{\neg h}) = \neg h \\
 & \mathsf{post}(\mathsf{feel}_h) = \mathsf{id}_{\mathsf{Var}} & \mathsf{post}(\mathsf{feel}_{\neg h}) = \mathsf{id}_{\mathsf{Var}} \\
 & \mathsf{R}_a(\mathsf{feel}_h, \mathsf{feel}_{\neg h}) & \mathsf{R}_a(\mathsf{feel}_{\neg h}, \mathsf{feel}_h) \\
\mathsf{flip}_h & \mathsf{pre}(\mathsf{flip}_h) = \neg h & \\
 & \mathsf{post}(\mathsf{flip}_h) : h \mapsto \top &
\end{array}$$

Then, the following is an $\mathbb{M}$-sequence leading to a *heads* result in any execution.

$$[\mathsf{U}, \mathsf{toss}_h \cup \mathsf{toss}_{\neg h}]\ [\mathsf{U}, (\mathsf{feel}_h \cup (\mathsf{feel}_{\neg h} \otimes \mathsf{flip}_h)]$$

tossing the coin, sensing it, and if tails flip it to heads

Definition 12. *We say that an* $\mathbb{M}$*-sequence* $[\mathsf{U}, \mathsf{E}_1], \ldots, [\mathsf{U}, \mathsf{E}_r]$ *is a solution to the planning domain* $\mathbb{M} = (\varphi_T, A_{\mathsf{E}} \cup A_{\cup}, \varphi_G)$ *iff*

$$\begin{array}{ll}
\models \varphi_T \to [\mathsf{U}, \mathsf{E}_1] \ldots [\mathsf{U}, \mathsf{E}_r]\varphi_G & (\text{success}) \\
\models \varphi_T \to \langle \mathsf{U}, \mathsf{E}_1 \rangle \ldots \langle \mathsf{U}, \mathsf{E}_r \rangle \top & (\text{executability})
\end{array}$$

It can be shown that the $\mathbb{M}$-sequence from Ex. 5 is a solution for the planning domain

$$\mathbb{M} = (\ \top,\ \{\ \mathsf{toss}_h \cup \mathsf{toss}_{\neg h},\ \mathsf{feel}_h \cup \mathsf{feel}_{\neg h},\ \mathsf{flip}_h, \mathsf{skip}\ \},\ [(a \cup b)^*]h\ )$$

7 A Search Algorithm for Non-deterministic Planning in LCC

Let us then proceed to the study of search algorithms for arbitrary planning domains $\mathbb{M}$. These planning algorithms search for solutions in the space of plans, defined below. The idea is to reduce a non-deterministic plan into a sequence of pairs of deterministic plans, each pair motivated by the introduction of a non-deterministic action. These plans are a triple consisting of: (1) a (possibly empty) $\mathbb{M}$-sequence $[\mathsf{U}, \mathsf{E}_k], \ldots [\mathsf{U}, \mathsf{E}_1]$, (possibly) prefixed by an operator-like expression $[\mathsf{U}, \cdot]$ (denoting the operator under construction); and formulas for (2) an initial state and (3) open goals corresponding to (1).

$$\text{plan } \pi \;=\; (\text{ operator} + \mathbb{M}\text{-sequence}, \;\; \text{init. state } \varphi_{\mathsf{init}(\pi)}, \;\; \text{open goals } \varphi_{\mathsf{goals}(\pi)} \;)$$

Again we abuse notation and refer to (1) with the label π of the plan it belongs to.

Definition 13. *Given a planning domain* $\mathbb{M} = (\varphi_T, A_{\mathsf{E}} \cup A_{\cup}, \varphi_G)$, *the* empty plan *for* $\mathbb{M}$ *is the pair* $\pi_\varnothing = (\varnothing, \varphi_G)$. *For a given plan* $\pi_k = [\mathsf{U}, \mathsf{E}_k] \ldots [\mathsf{U}, \mathsf{E}_1]$ *and its refinement with some* $\mathsf{e} \in A_{\mathsf{E}}$, *denoted* $\pi = \pi_k(\mathsf{e}) = [\mathsf{U}, \mathsf{e}]\pi_k$, *we define the refinements* $\pi(\cdot)$ *with* $\mathsf{f} \in A_{\mathsf{E}}$ *or a run-time dist. action* $\mathsf{f} \cup \mathsf{f}' \in A_{\cup}$ *as:*

$$\begin{array}{rlrl}
\pi(\mathsf{f}) &= [\mathsf{U}, \mathsf{f} \otimes \mathsf{e}]\pi_k & \pi(\mathsf{f} \cup \mathsf{f}') &= [\mathsf{U}, (\mathsf{f} \otimes \mathsf{e}) \cup (\mathsf{f}' \otimes \mathbf{x})]\pi_k \\
\varphi_{\mathsf{init}(\pi(\mathsf{f}))} &= \varphi_T & \varphi_{\mathsf{init}(\pi(\mathsf{f} \cup \mathsf{f}'))} &= \text{“}[\mathsf{U}, \mathsf{f}'](\cdot)\text{”} \\
\varphi_{\mathsf{goals}(\pi(\mathsf{f}))} &= t([\mathsf{U}, \mathsf{f}]\varphi_{\mathsf{goals}(\pi)} \wedge \langle \mathsf{U}, \mathsf{f} \rangle \top) & \varphi_{\mathsf{goals}(\pi(\mathsf{f} \cup \mathsf{f}'))} &= \varphi_{\mathsf{goals}(\pi_k)}
\end{array}$$

Given a plan π *of the form* $\pi = (\, [\mathsf{U}, (\mathsf{f} \otimes \mathsf{e}) \cup (\mathsf{f}' \otimes \mathbf{x} \otimes \mathsf{e}')]\pi_k, \text{“}[\mathsf{U}, \mathsf{f}'](\cdot)\text{”}, \varphi_{\mathsf{goals}(\pi)} \,)$, *and an action* $\mathsf{e}'' \in A_{\mathsf{E}}$ *we define the refinement* $\pi(\mathsf{e}'')$ *as*

$$\pi(\mathsf{e}'') = \begin{cases} [\mathsf{U}, (\mathsf{f} \otimes \mathsf{e}) \cup (\mathsf{f}' \otimes \mathbf{x} \otimes \mathsf{e}'' \otimes \mathsf{e})]\pi_k & \textit{if } \not\models [\mathsf{U}, \mathsf{f}'][\mathsf{U}, \mathsf{e}'' \otimes \mathsf{e}]\varphi_{\mathsf{goals}(\pi_k)} \\ [\mathsf{U}, (\mathsf{f} \otimes \mathsf{e}) \cup (\mathsf{f}' \otimes \mathsf{e}'' \otimes \mathsf{e})] & \textit{otherwise} \end{cases}$$

$$\varphi_{\mathsf{init}(\pi(\mathsf{e}''))} = \begin{cases} \varphi_{\mathsf{init}(\pi(\mathsf{e}''))} & \textit{if } \not\models [\mathsf{U}, \mathsf{f}'][\mathsf{U}, \mathsf{e}'' \otimes \mathsf{e}]\varphi_{\mathsf{goals}(\pi_k)} \\ \varphi_T & \textit{otherwise} \end{cases}$$

$$\varphi_{\mathsf{goals}(\pi(\mathsf{e}''))} = \begin{cases} t([\mathsf{U}, \mathsf{e}'']\varphi_{\mathsf{goals}(\pi)} \wedge \langle \mathsf{U}, \mathsf{e}'' \rangle \top) & \textit{if } \not\models [\mathsf{U}, \mathsf{f}'][\mathsf{U}, \mathsf{e}'' \otimes \mathsf{e}]\varphi_{\mathsf{goals}(\pi_k)} \\ t([\mathsf{U}, (\mathsf{f} \otimes \mathsf{e}) \cup (\mathsf{f}' \otimes \mathsf{e}'' \otimes \mathsf{e})]\varphi_{\mathsf{goals}(\pi_k)} & \\ \quad \wedge \langle \mathsf{U}, (\mathsf{f} \otimes \mathsf{e}) \cup (\mathsf{f}' \otimes \mathsf{e}'' \otimes \mathsf{e}) \rangle \top) & \textit{otherwise} \end{cases}$$

Finally, if $\mathsf{f} \cup \mathsf{f}'$ *is run-time indistinguishable to the agent, i.e.* $(\mathsf{f}, \mathsf{f}'), (\mathsf{f}', \mathsf{f}) \in \mathsf{R}_a$, *we define the refinement of* π_k *with* $\mathsf{f} \cup \mathsf{f}'$ *as:*

$$\begin{aligned}
\pi(\mathsf{f} \cup \mathsf{f}') &= [\mathsf{U}, \mathsf{f} \cup \mathsf{f}']\pi_k \\
\varphi_{\mathsf{init}(\pi(\mathsf{f} \cup \mathsf{f}'))} &= \varphi_T \\
\varphi_{\mathsf{goals}(\pi(\mathsf{f} \cup \mathsf{f}'))} &= t([\mathsf{U}, \mathsf{f} \cup \mathsf{f}']\varphi_{\mathsf{goals}(\pi_k)} \wedge \langle \mathsf{U}, \mathsf{f} \cup \mathsf{f}' \rangle \top)
\end{aligned}$$

Given a plan π *and a refinement of it* $\pi(\cdot)$, *we say* $\pi(\cdot)$ *is a* leaf *iff either* $\varphi_{\pi(\cdot)}$ *is inconsistent or* $\models \varphi_{\mathsf{goals}(\pi(\cdot))} \to \varphi_{\mathsf{goals}(\pi)}$. *The Terminating Condition for a plan* π *is*

$$\varphi_{\mathsf{init}(\pi)} = \varphi_T \quad \textit{and} \quad \models \varphi_{\mathsf{init}(\pi)} \to \varphi_{\mathsf{goals}(\pi)}$$

After a run-time indistinguishable action, e.g. coin tossing, conditional plans can be made depending on the outcome of an observation. Let us finally address the properties of non-deterministic planning based on BFS.

```
Input : $\mathbb{M} = (\varphi_T, A_\mathsf{E} \cup A_\cup, \varphi_G)$.
LET Plans $= \langle \pi_\varnothing \rangle$ and $\pi = \pi_\varnothing$
WHILE $\pi$ does not satisfy Terminating Condition
    DELETE $\pi$ FROM Plans
    SET Plans $=$ Plans$^\frown\langle\, \pi' \mid \pi'$ refines $\pi$ and $\pi'$ not a leaf $\rangle$.
    SET $\pi =$ the first element of Plans
Output : $\pi$    (i.e. the $\mathbb{M}$-sequence defined by $\pi$)
```

Fig. 5. BFS algorithm for backward non-deterministic planning in $\mathrm{LCC}_{\cup\otimes n}$

Theorem 3. *Let the output of the BFS algorithm in Fig. 5 be* $[\mathsf{U}, \mathsf{E}_1] \dots [\mathsf{U}, \mathsf{E}_k]$ *for a planning domain* $\mathbb{M}$. *Then,* $[\mathsf{U}, \mathsf{E}_1] \dots [\mathsf{U}, \mathsf{E}_k]$ *is an* $\mathbb{M}$*-sequence and a solution for* $\mathbb{M}$.

Theorem 4. *For a given planning domain* $\mathbb{M}$, *if some* $\mathbb{M}$*-sequence exists that is a solution to* $\mathbb{M}$, *then the BFS algorithm in Fig. 5 terminates (with a solution).*

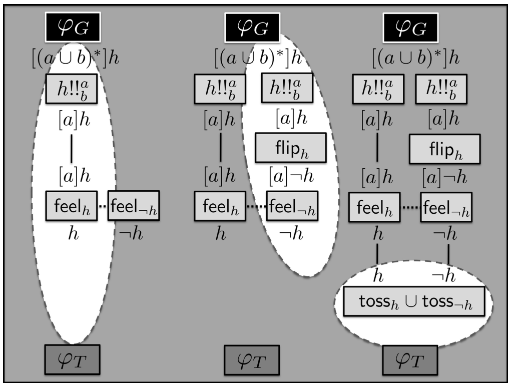

Fig. 6. Plan search in Example 1. Incremental construction of a solution for the coin example.

Example 6. Recall Example 1, where the planner agent a must show heads, denoted h, to win the prize. The action flip_h is secret in the sense of $(\mathsf{flip}_h, \mathsf{skip}) \in \mathsf{R}_b$, i.e. agent b believes nothing is happening; this secrecy is known by a provided flip_h is only R_a-related to itself. The construction of a solution is shown in Figure 6, where: (Left) a deterministic plan is being built, consisting of a's demonstration $h!!^a_b$ that h to b (with a knowing a priori that h); a plan-time indistinguishable action $\mathsf{feel}_h \cup \mathsf{feel}_{\neg h}$ is added. (Center) The planner proceeds to solve the rightmost case where $\mathsf{feel}_{\neg h}$ is executed (due to a $\neg h$ state). This planning sub-problem is solved by a flip_h action, followed by the same demonstration $h!!^a_b$. (Right) Finally, the algorithm stops after adding the run-time indistinguishable action of tossing $\mathsf{toss}_h \cup \mathsf{toss}_{\neg h}$. Note the remaining of the plan is executable no matter the result of the coin toss. The slightly different plan construction from [10] can also be built with two deterministic sensing actions (for h and $\neg h$).

8 Conclusions and Future Work

We presented backward planning algorithms for a planner-reasoner agent enabling her to find deterministic or (non-deterministic) strong plans in multi-agent scenarios. We considered dynamic epistemic logics with ontic actions, further extended with composition and choice. Planners in these logics are sensitive to others' beliefs and may contain communications and observations as well as the usual fact-changing actions. As for future work, we would like to study more complex plan structures, or new kinds of actions like belief revision announcements. Another direction would be the study of (logical) heuristics to improve the performance of LCC planners.

Acknowledgements. This work has been funded by projects AT (CSD 2007-022), AR-INF (TIN2009-14704-C03-03); and grants 2009-SGR-1434 and EPSRC EP/J002607/1.

References

1. Andersen, M.B., Bolander, T., Jensen, M.H.: Conditional Epistemic Planning. In: del Cerro, L.F., Herzig, A., Mengin, J. (eds.) JELIA 2012. LNCS (LNAI), vol. 7519, pp. 94–106. Springer, Heidelberg (2012)
2. Aucher, G.: DEL-sequents for progression. Journal of Applied Non-Classical Logics 21(3-4), 289–321 (2011)
3. Baltag, A., Moss, L., Solecki, S.: The logic of public announcements, common knowledge and private suspicions. In: Proc. of 7th Conf. TARK 1998, pp. 43–56 (1998)
4. Bolander, T., Andersen, M.: Epistemic planning for single- and multi-agent systems. Journal of Applied Non-Classical Logics 21(1), 9–34 (2011)
5. Ghallab, M., Nau, D., Traverso, P.: Automated Planning: Theory and Practice. Morgan Kaufmann (2004)
6. Harel, D., Kozen, D., Tiuryn, J.: Dynamic Logic. MIT Press, Massachusetts (2000)
7. Hintikka, J.: Knowledge and belief: an introduction to the logic of the two notions. Cornell University Press (1962)
8. Löwe, B., Pacuit, E., Witzel, A.: Planning based on dynamic epistemic logic (2010)
9. Pardo, P., Sadrzadeh, M.: Planning in the Logics of Communication and Change. In: Proc. of AAMAS 2012 (2012)
10. Pardo, P., Sadrzadeh, M.: Backward Planning in the Logics of Communication and Change. In: Proc. of Agreement Technologies AT 2012 (2012)
11. Pearl, J.: Heuristics: Intelligent Search Strategies for Computer Problem Solving. Addison-Wesley (1984)
12. Rao, A., Georgeff, M.: Modeling rational agents within a BDI-architecture. In: Proc. of Principles of Knowledge Representation and Reasoning (KR), pp. 473–484 (1991)
13. van Benthem, J., van Eijck, J., Kooi, B.: Logics of Communication and Change. Information and Computation 204, 1620–1662 (2006)
14. van der Hoek, W., Wooldridge, M.: Tractable Multiagent Planning for Epistemic Goals. In: Proc. of AAMAS 2002, pp. 1167–1174 (2002)
15. van Ditmarsch, H., van der Hoek, W., Kooi, B.: Dynamic Epistemic Logic. Springer (2008)
16. van Ditmarsch, H., Kooi, B.: Semantic results for ontic and epistemic change. In: Bonanno, van der Hoek, Wooldridge (eds.) LOFT 7, pp. 87–117 (2008)