Free for More: A Dynamic Epistemic Framework for Conformant Planning [1]

YANJUN LI, Peking University, China & University of Groningen, The Netherlands
QUAN YU, Sun Yat-sen University & Qiannan Normal College for Nationalities, China
YANJING WANG, Peking University, China

In this paper, we introduce a lightweight dynamic epistemic logical framework for automated planning under initial uncertainty. We generalize the standard conformant planning problem in AI in two crucial aspects: First, the planning goal can be any formula expressed in an epistemic propositional dynamic logic (EPDL); Second, procedural constraints of the desired plan specified by regular expressions can be imposed. We then reduce the problem of generalized conformant planning to the model checking problem of our logic. Although our conformant planning problem is much more general than the standard one with Boolean goals and no procedural constraints, the complexity is still PSPACE-complete which is equally hard as standard conformant planning over explicit transition systems. In other words, the generalization is "for free".

Categories and Subject Descriptors: []

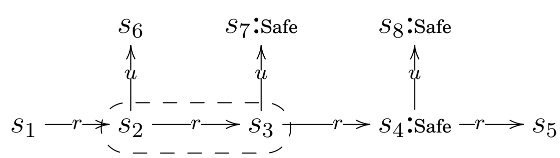
General Terms: Verification

Additional Key Words and Phrases: conformant planning, dynamic logic, epistemic logic, modal logic

ACM Reference Format:
Yanjun Li, Quan Yu and Yanjing Wang. 2015. Free for More: A Dynamic Epistemic Framework for Conformant Planning *ACM Trans. Comput. Logic* V, N, Article A (2015), 23 pages.
DOI: http://dx.doi.org/10.1145/0000000.0000000

1. INTRODUCTION

Conformant planning is the problem of finding a linear plan (a sequence of actions) to achieve a goal in presence of uncertainty about the initial state (cf. [Smith and Weld 1998]). For example, suppose that you are a rookie spy trapped in a foreign hotel with the following map at hand:[2]



[1] A preliminary version of this paper was presented at TARK2015 [Yu et al. 2015]. In this paper, we refute a crucial conjecture proposed in the preliminary version and give the precise complexity of the model checking problem of epistemic PDL over uncertainty maps.
[2] It is a variant of the running example used in [Wang and Li 2012].

Yanjing Wang is the corresponding author.
Author's addresses: Yanjun Li, Faculty of Philosophy, University of Groningen, The Netherlands; Quan Yu, Qiannan Normal College for Nationalities, China; Yanjing Wang, Department of Philosophy, Peking University, China.

DOI: http://dx.doi.org/10.1145/0000000.0000000

Now somebody spots you and sets up the alarm. In this case you need to move fast to one of the safe hiding places marked in the map (i.e., s_7, s_8 and s_4). However, since you were in panic, you lost your way and you are not sure whether you are at s_2 or s_3 (denoted by the bubble in the above graph). Now, what should you do in order to reach a safe place quickly? Clearly, merely moving r or moving u may not guarantee your safety given the uncertainty. A simple plan is to move r first and then u, since this plan will take you to a safe place, no matter where you actually are initially. This plan is *conformant* since it does not require any feedback during the execution and it should work in presence of uncertainty about the initial state. More generally, a conformant plan should also work given actions with non-deterministic effects. Such a conformant plan is crucial when there are no feedbacks/observations available during the execution of the plan, or it is just too 'expensive' in time to have feedbacks [Bonet 2010]. Note that since no information is provided during the execution, the conformant plan is simply a finite sequence of actions without any conditional moves.

As discussed in [Bonet and Geffner 2000; Palacios and Geffner 2006], conformant planning can be reduced to classical planning, the planning problem without any initial uncertainty, over the space of *belief states*. Intuitively, a belief state is a subset of the state space, which records the uncertainty during the execution of a plan, e.g., $\{s_2, s_3\}$ is an initial belief state in the above example. In order to make sure a goal is achieved eventually, it is crucial to track the transitions of belief states during the execution of the plan, and this may traverse exponentially many belief states in the size of the original state space. As one may expect, conformant planning is computationally harder than classical planning. The complexity of checking the existence of a conformant plan is EXPSPACE-complete in the size of the variables generating the state space [Haslum and Jonsson 1999]. In the literature, people proposed compact and implicit representations of the belief spaces, such as OBDD [Cimatti and Roveri 2000; Cimatti et al. 2004; Cimatti and Roveri 2011] and CNF [To et al. 2010], and different heuristics were used to guide the search for a plan, e.g., [Brafman and Hoffmann 2004; Bertoli et al. 2006; Bryce et al. 2006; Palacios and Geffner 2014].

Besides the traditional AI approaches, we can also take an epistemic-logical perspective on planning in presence of initial uncertainties, based on dynamic epistemic logic (DEL) (cf. e.g., [van Ditmarsch et al. 2007]). The central philosophy of DEL takes the meaning of an action as the change it brings to the knowledge of the agents. Intuitively, this is the transition of belief states we need to track during the execution of a plan[3]. Indeed, in recent years, there has been a growing interest in applying DEL to handle multi-agent planning with knowledge goals (cf. e.g., [Bolander and Andersen 2011; Löwe et al. 2011; Andersen et al. 2012; Aucher 2012; Yu et al. 2013; Pardo and Sadrzadeh 2013; Jensen 2014; Bolander et al. 2015; Muise et al. 2015]), while the traditional AI planning focuses on the single-agent case. In particular, the event models of DEL (cf. [Baltag and Moss 2004]) are used to handle non-public actions that may cause different knowledge updates to different agents. In these DEL-based planning frameworks, states are epistemic models, actions are event models and the state transitions are implicitly encoded by the update product which computes a new epistemic model based on an epistemic model and an event model.

One advantage of this approach is its expressiveness in handling scenarios which require reasoning about agents' higher-order knowledge about each other in presence of partially observable actions. However, this expressiveness comes at a price, as shown in [Bolander and Andersen 2011; Aucher and Bolander 2013], that multi-agent epis-

[3]Here the belief states are actually about knowledge in epistemic logic.

temic planning is undecidable in general. Many interesting decidable fragments are found in the literature [Bolander and Andersen 2011; Löwe et al. 2011; Yu et al. 2013; Andersen et al. 2015], which suggests that the restrictions on the form of event models and the single-agent cases are the key to decidability. However, if we focus on the single-agent planning, a natural question arises: how do we compare such DEL approaches with the traditional AI planning? It seems that the DEL-based approaches are more suitable for planning with actions that change (higher-order) knowledge rather than planning with fact-changing actions, although the latter type of actions can also be handled in DEL (cf. e.g., [van Benthem et al. 2006]). Moreover, the standard models of DEL are purely epistemic thus do not encode the information of available actions directly, which may limit the applicability of such approaches to planning problems based on transition systems.

In this paper, we tackle the standard single-agent conformant planning problem over transition systems, by using the ideas of DEL, but not its standard formalism. Our formal framework is based on the logic proposed by Wang and Li [2012], where the model is simply a transition system with initial uncertainty as in the motivating example, and an action is interpreted in the semantics as an update on the uncertainty of the agent. Our technical contributions are summarized as follows:

— A lightweight dynamic epistemic framework with an epistemic PDL language.
— A complete axiomatization of the program-free fragment.
— A generalization of conformant planning problem with arbitrary formulas as goals and regular expressions as procedural constraints on the desired plans.
— A reduction of generalized conformant planning to model checking of our logic.
— The model checking problem of our logic is PSPACE-complete.[4]

The last result may sound contradictory to the aforementioned result that the complexity of conformant planning is EXPSPACE-complete. Actually, the apparent 'contradiction' is due to the fact that the EXPSPACE result is based on the number of *state variables*, which require an exponential blow up to generate an explicit transition system that we use here. We will come back to this issue at the end of Section 4.2.

The most important advantage of our approach, besides the transparency of the semantics and the logic, is that we can deal with more general planning problems without paying any price in computational complexity. In the literature, the goal of a conformant planning problem is usually a formula of propositional logic, and its solution is a sequence of actions. In this paper, we generalize conformant planning by extending the goal language and constraining the possible solutions. The goal language is extended with both knowledge and program modalities, and the solution must comply with some constrains on the shape of the action sequences.

These generalizations do make sense in real-life contexts. For example, suppose that you have a toothache (p) caused by a broken tooth, and you have two choices: either replacing it with a false tooth (a) or taking some medicine to temporally relieve the pain (b). For the second option, the toothache may come again after some time (t). In this scenario, your goal can be getting rid of the pain either temporally ($\neg p$) or forever ($[t^*]\neg p$) which involves a modality about future. More precisely, you want to make sure that you *know* that it will not trouble you in the future ($K[t^*]\neg p$). Moreover, we can also handle *negative* epistemic goals, such as to make sure that you do *not* know p. It may sound strange at the first glance in a single agent setting but it does make good sense in some real-life contexts. For example, suppose that there are different drugs

[4]It was conjectured in [Yu et al. 2015] that the model checking problem was EXPTIME-complete, which is now refuted in this paper.

to cure a symptom which all come with potential side effects (p). In this scenario, you may prefer a drug which only have side-effect occasionally according to your knowledge ($\neg Kp$), over a drug which is known to have a bad side-effect for sure (Kp). We will come back to such examples more formally in the later parts of the paper.

On the other hand, constraints on the shape of the desired plan also play an important role in planning. For example, suppose that there are three means of transportation: by air (a), by bus (b) and by train (c). To get to a place C, you have several options, such as firstly riding a bus from A to B then taking a train from B to C, or flying directly from A to C. In real life, you may have constraints on the actions or subplans that you can use. For example, it might be the case that you cannot afford flying but can only afford one train ticket (not even two). Therefore, the solution should be a sequence that not only guarantees your arrival at C but also contains action b at most once and no action a at all. All such constraints on the desired plans can be expressed precisely in our approach.

In the literature, conformant planning has been generalized with epistemic goals by [Bonet 2010]. Moreover, [Bonet 2010] also considered conformant planning under the partial observability which may give the agent more information during execution of the plan, although the plan itself is still linear. However, the complexity of conformant planning with both partial observability and epistemic goals is strictly higher than the complexity of standard conformant planning [Bonet 2010]. Here, as we show in the paper, our generalized conformant planning with constraints and more general goals has exactly the same complexity as the standard conformant planning. In other words, we get these generalizations *for free*.

In sum, our approach has the following advantages compared to the existing ones:

— The planning goals can be specified as arbitrary formulas in our logic language with both epistemic and program modalities. Moreover, extra plan constraints can be expressed by regular expressions, which generalizes the knowledge-based programs in [Fagin et al. 1997; Lang and Zanuttini 2012]. Therefore it covers a richer class of (conformant) planning problems compared to the traditional AI approach where a goal is a set of valuations and no constraints on the actions.
— On the other hand, we do not pay the price in complexity. We can reduce the generalized conformant planning problem to some model checking problem of the logical language, which also reveals some subtleties. The model checking problem of the full language with programs is still PSPACE-complete.
— Our logical language and models are very simple, and we can encode the externally given executability of the actions in the model, inspired by epistemic temporal logic (ETL) proposed in [Fagin et al. 1995; Parikh and Ramanujam 1985].
— Our approach is flexible enough to provide, in the future, a unified platform to compare different planning problems under uncertainty. By studying different fragments of the logical language and model classes, we may categorize planning problems according to their complexity. Moreover, in principle, various model checking techniques can be applied to the planning problems via reductions in our framework.

The rest of the paper is organized as follows: We introduce our basic logical framework and provide an axiomatization in Section 2, and extend it in Section 3 with programs to handle the generalized conformant planning which is also introduced in the same section. The complexity analysis of the model checking problem is provided in Section 4, and we finally conclude in Section 5 with future directions.

2. BASIC FRAMEWORK

2.1. Epistemic action language

To talk about the knowledge of the agent during an execution of a plan, we start with the following simple language proposed by Wang and Li [2012].

Definition 2.1 (*Epistemic Action Language* (EAL)). Given a countable set $\mathtt{A}$ of action symbols and a countable set $\mathtt{P}$ of atomic proposition letters , the language $\mathtt{EAL}^{\mathtt{A}}_{\mathtt{P}}$ is defined as follows:[5]

$$\phi ::= \top \mid p \mid \neg\phi \mid (\phi \wedge \phi) \mid [a]\phi \mid K\phi,$$

where $p \in \mathtt{P}$, $a \in \mathtt{A}$. The following standard abbreviations are used: $\bot := \neg\top$, $\phi \vee \psi := \neg(\neg\phi \wedge \neg\psi)$, $\phi \to \psi := \neg\phi \vee \psi$, $\langle a\rangle\phi := \neg[a]\neg\phi$, $\hat{K}\phi := \neg K\neg\phi$.

$K\phi$ says that the agent knows that ϕ, and $[a]\phi$ expresses that if the agent can move forward by action a, then after doing a, ϕ holds. Throughout the paper, we fix some $\mathtt{P}$ and $\mathtt{A}$, and refer to $\mathtt{EAL}^{\mathtt{A}}_{\mathtt{P}}$ by $\mathtt{EAL}$.

Definition 2.2 (*Uncertainty map*). Given $\mathtt{P}$ and $\mathtt{A}$, a (multimodal) Kripke model $\mathcal{N}$ is a tuple $\langle \mathcal{S}, \{\mathcal{R}_a \mid a \in \mathtt{A}\}, \mathcal{V}\rangle$, where $\mathcal{S}$ is a non-empty set of states, $\mathcal{R}_a \subseteq \mathcal{S} \times \mathcal{S}$ is a binary relation labeled by a, $\mathcal{V} : \mathcal{S} \to 2^{\mathtt{P}}$ is a valuation function. An uncertainty map $\mathcal{M}$ is a Kripke model $\langle \mathcal{S}, \{\mathcal{R}_a \mid a \in \mathtt{A}\}, \mathcal{V}\rangle$ with a non-empty set $\mathcal{U} \subseteq \mathcal{S}$. Given an uncertainty map $\mathcal{M}$, we refer to its components by $\mathcal{S}_{\mathcal{M}}$, $\mathcal{R}_{a\mathcal{M}}$, $\mathcal{V}_{\mathcal{M}}$, and $\mathcal{U}_{\mathcal{M}}$. A pointed uncertainty map $\mathcal{M}, s$ is an uncertainty map $\mathcal{M}$ with a designated state $s \in \mathcal{U}_{\mathcal{M}}$. We write $s \xrightarrow{a} t$ for $(s, t) \in \mathcal{R}_a$.

Intuitively, a Kripke model encodes a map (transition system) and the uncertainty set $\mathcal{U}$ encodes the uncertainty that the agent has about where he is in the map. The graph mentioned at the beginning of the introduction is a typical example of an uncertainty map. Note that there may be non-deterministic transitions in the model, i.e., there may be $t_1 \neq t_2$ such that $s \xrightarrow{a} t_1$ and $s \xrightarrow{a} t_2$ for some s, t_1, t_2.

Remark 2.3. It is crucial to notice that the designated state in a pointed uncertainty map must be one of the states in the uncertainty set.

Definition 2.4 (*Semantics*). Given any uncertainty map $\mathcal{M} = \langle \mathcal{S}, \{\mathcal{R}_a \mid a \in \mathtt{A}\}, \mathcal{V}, \mathcal{U}\rangle$ and any state $s \in \mathcal{U}$, the semantics is defined as follows:

$\mathcal{M}, s \vDash \top$		always
$\mathcal{M}, s \vDash p$	$\iff$	$s \in \mathcal{V}(p)$
$\mathcal{M}, s \vDash \neg\phi$	$\iff$	$\mathcal{M}, s \nvDash \phi$
$\mathcal{M}, s \vDash \phi \wedge \psi$	$\iff$	$\mathcal{M}, s \vDash \phi$ and $\mathcal{M}, s \vDash \psi$
$\mathcal{M}, s \vDash [a]\phi$	$\iff$	$\forall t \in S : s \xrightarrow{a} t$ implies $\mathcal{M}\vert^a, t \vDash \phi$
$\mathcal{M}, s \vDash K\phi$	$\iff$	$\forall u \in \mathcal{U} : \mathcal{M}, u \vDash \phi$

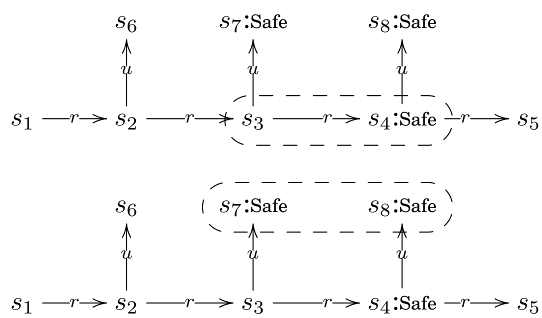
where $\mathcal{M}|^a = \langle \mathcal{S}, \{\mathcal{R}_a \mid a \in \mathtt{A}\}, \mathcal{V}, \mathcal{U}|^a\rangle$ and $\mathcal{U}|^a = \{r' \mid \exists r \in \mathcal{U}$ such that $r \xrightarrow{a} r'\}$. We say ϕ is valid (notation: $\vDash \phi$) if it is true on all the pointed uncertainty maps. For a action sequence $\sigma = a_1 \dots a_n$, we write $\mathcal{U}|^\sigma$ for $(\dots((\mathcal{U}|^{a_1})|^{a_2})\dots)|^{a_n}$. and write $\mathcal{M}|^\sigma$ for $(\dots((\mathcal{M}|^{a_1})|^{a_2})\dots)|^{a_n}$.

Intuitively, the agent 'carries' the uncertainty set with him when moving forward and obtains a new uncertainty set $\mathcal{U}|^a$. Note that here we differ from [Wang and Li

[5]We do need unboundedly many action symbols to encode the desired problem in the later discussion of model checking complexity.

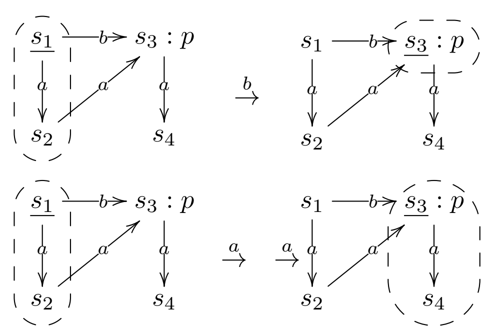
2012] where the updated uncertainty set is further refined according to what the agent can observe at the new state. For conformant planning, we do not consider the observational power of the agent during the execution of a plan.

Let us call the model mentioned in the introduction $\mathcal{M}$, it is not hard to see that $\mathcal{M}|^r$ and $(\mathcal{M}|^r)|^u$ are as follows:



Thus we have $\mathcal{M}, s_3 \vDash [r](\textit{Safe} \wedge \neg K\textit{Safe})$ and $\mathcal{M}, s_3 \vDash K[r][u](\textit{Safe} \wedge K\textit{Safe})$

The usual global model checking algorithm for modal logics labels the states with the subformulas that are true on the states. However, this cannot work here since the truth value of epistemic formulas on the states outside $\mathcal{U}$ is simply undefined. Moreover, the exact truth value of an epistemic formula on a state depends on 'how you get there', as the following example shows (the underlined states mark the actual states):



Let the left-hand-side model be $\mathcal{M}$ then it is clear that $\mathcal{M}|^b, s_3 \vDash Kp$ while $\mathcal{M}|^{aa}, s_3 \nvDash Kp$ thus $\mathcal{M}, s_1 \vDash \langle b \rangle Kp \wedge \langle a \rangle \langle a \rangle \neg Kp$. This shows that the truth value of an epistemic subformula w.r.t. a state in the model is somehow 'context-dependent': it depends on the uncertainty set.

2.2. Axiomatization

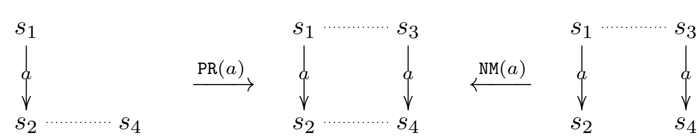
Following the axioms proposed in [Wang and Li 2012], we give the following axiomatization for `EAL` w.r.t. our semantics:

System $\mathbb{SELA}$

Axioms		**Rules**	
TAUT	all axioms of propositional logic	MP	$\dfrac{\phi, \phi \to \psi}{\psi}$
DISTK	$K(p \to q) \to (Kp \to Kq)$	NECK	$\dfrac{\phi}{K\phi}$
DIST(a)	$[a](p \to q) \to ([a]p \to [a]q)$	NEC(a)	$\dfrac{\phi}{[a]\phi}$
T	$Kp \to p$	SUB	$\dfrac{\phi(p)}{\phi(\psi)}$
4	$Kp \to KKp$		
5	$\neg Kp \to K\neg Kp$		
PR(a)	$K[a]p \to [a]Kp$		
NM(a)	$\langle a\rangle Kp \to K[a]p$		

where a ranges over A, p, q range over P. PR$(\cdot)$ and NM$(\cdot)$ denote the axioms of *perfect recall* and *no miracles* respectively (cf. [Wang and Cao 2013]).

Note that since we do not assume that the agent can observe the available actions, the axiom OBS(a) : $K\langle a\rangle\top \vee K\neg\langle a\rangle\top$ in [Wang and Li 2012] is abandoned. Due to the same reason, the axiom of no miracles is also simplified. Intuitively, PR(a) means that the agent can tell t from t_0 since he perfectly remembers that he can tell states resulting in t from states resulting in t_0. NM(a) means there are no such miracles that the agent cannot distinguish two states initially but nevertheless he can distinguish the states resulting from executing the same action on these two states. These properties can be depicted as follows.



We show the completeness of $\mathbb{SELA}$ using a more direct proof strategy based on canonical model compared to the one used in [Wang and Li 2012].

THEOREM 2.5. *$\mathbb{SELA}$ is sound and strongly complete w.r.t.* EAL *on uncertainty maps.*

PROOF. To prove that $\mathbb{SELA}$ is sound on uncertainty maps, we need to show that all the axioms are valid and all the inference rules preserve validity. Since the uncertainty set in an UM denotes an equivalence class, axioms T, 4 and 5 are valid; due to the semantics, the validity of axioms PR$(\cdot)$ and NM$(\cdot)$ can be proved step by step; others can be proved as usual.

To prove that $\mathbb{SELA}$ is strongly complete on uncertainty maps, we only need to show that every $\mathbb{SELA}$-consistent set of formulas is satisfiable on some uncertainty map. The proof idea is that we construct an uncertainty map consisting of maximal $\mathbb{SELA}$-consistent sets (MCSs), and then with the Lindenbaum-like lemma that every $\mathbb{SELA}$-consistent set of formulas can be extended in to a MCS (we omit the proof here), we only need to prove that every formula holds on the MCS to which it belongs.

Firstly, we construct a canonical Kripke model $\mathcal{N}^c = \langle \mathcal{S}^c, \{\mathcal{R}^c_a \mid a \in \mathtt{A}\}, \mathcal{V}^c\rangle$ as follows:

— $\mathcal{S}^c$ is the set of all MCSs;
— $s\mathcal{R}^c_a t \iff \langle a\rangle\phi \in s$ for any $\phi \in t$ (equivalently $\phi \in t$ for any $[a]\phi \in s$);

— $\mathcal{V}^c(p) = \{s \mid p \in s\}$.

Given $s \in \mathcal{S}^c$, we define $\mathcal{U}_s^c = \{u \in \mathcal{S}^c \mid K\phi \in s \text{ iff } K\phi \in u\}$, and it is obvious that $s \in \mathcal{U}_s^c$. Thus we have that for each $s \in \mathcal{S}^c$, $\mathcal{M}_s^c = \langle \mathcal{N}^c, \mathcal{U}_s^c \rangle$ is an uncertainty map, and $\mathcal{M}_s^c, s$ is a pointed uncertainty map.

Secondly, we prove the following claim.

CLAIM 2.1. *If $\neg K\phi \in s$, then there exists $u \in \mathcal{U}_s^c$ such that $\neg\phi \in u$.*

Let u^- be $\{K\psi \mid K\psi \in s\} \cup \{\neg\phi\}$. Then u^- is consistent. For suppose not, there are $K\psi_1, \ldots, K\psi_n$ such that $\vdash K\psi_1 \wedge \cdots \wedge K\psi_n \to \phi$. By rule NECK and axiom DISTK, it follows that $\vdash KK\psi_1 \wedge \cdots \wedge KK\psi_n \to K\phi$. It follows by axiom 4 that $KK\psi_i \in s$ for each $1 \leq i \leq n$. Thus we have $K\phi \in s$. This is contrary with $\neg K\phi \in s$. We conclude that u^- is consistent. By Lindenbaum-like Lemma, there exists a MCS u extending u^-. Now since $u^- \subseteq u$, it is clear that $K\psi \in s$ implies $K\psi \in u$ for any ψ. On the other hand, if $K\psi \notin s$ then $K\neg K\psi \in s$ by axiom 5. Therefore $K\neg K\psi \in u$, and thus $K\psi \notin u$ by axiom T. It follows that $u \in \mathcal{U}_s^c$.

CLAIM 2.2. *If $\neg[a]\phi \in s$, then there exists $t \in \mathcal{S}_s^c$ such that $s \xrightarrow{a} t$ and $\neg\phi \in t$.*

Let t^- be $\{\psi \mid [a]\psi \in s\} \cup \{\neg\phi\}$. Then t^- is consistent. For suppose not, there are $\psi_1, \ldots, \psi_n$ such that $\vdash \psi_1 \wedge \cdots \wedge \psi_n \to \phi$. By rule NEC$(a)$ and axiom DIST(a), it follows that $\vdash [a](\psi_1 \wedge \cdots \wedge \psi_n) \to [a]\phi$. By $\vdash [a]\psi_1 \wedge \cdots \wedge [a]\psi_n \to [a](\psi_1 \wedge \cdots \wedge \psi_n)$, it follows that $\vdash [a]\psi_1 \wedge \cdots \wedge [a]\psi_n \to [a]\phi$. Thus we have $[a]\phi \in s$. This is contrary with $\neg[a]\phi \in s$. We conclude that t^- is consistent. By Lindenbaum-like Lemma, there exists a MCS t extending t^-. It follows by $t^- \subseteq t$ that $s \xrightarrow{a} t$ and $\neg\phi \in t$.

CLAIM 2.3. *If $s \xrightarrow{a} t$, then we have $\mathcal{U}_s^c|^a = \mathcal{U}_t^c$.*

$\subseteq$: Assuming $v \in \mathcal{U}_s^c|^a$, we need to show $v \in \mathcal{U}_t^c$, namely we need to show that $K\phi \in v \iff K\phi \in t$. Since $v \in \mathcal{U}_s^c|^a$, we have that there is $u \in \mathcal{U}_s^c$ such that $uR_a^c v$. If $K\phi \in t$, it follows by axiom 4 that $KK\phi \in t$. Thus we have $\langle a \rangle KK\phi \in s$. By axiom NM$(a)$, it follows that $K[a]K\phi \in s$. By $u \in \mathcal{U}_s^c$ and axiom T, we have $[a]K\phi \in u$. It follows by $uR_a^c v$ that $K\phi \in v$. If $K\phi \notin t$, we have $\neg K\phi \in t$. By axiom 5, we have $K\neg K\phi \in t$. Similarly, we have $\neg K\phi \in v$. Thus we have $K\phi \notin v$.

$\supseteq$: Assuming $v \in \mathcal{U}_t^c$, we need to show $v \in \mathcal{U}_s^c|^a$, namely there is $u \in \mathcal{U}_s^c$ such that $uR_a^c v$. Let u^- be $\{K\phi \mid K\phi \in s\} \cup \{\langle a \rangle\psi \mid \psi \in v\}$. Then u^- is consistent. For suppose not, we have $\vdash K\phi_1 \wedge \cdots \wedge K\phi_n \to [a]\neg\psi_1 \vee \cdots \vee [a]\neg\psi_k$ for some n and k. Since $\vdash [a]\neg\psi_1 \vee \cdots \vee [a]\neg\psi_k \to [a](\neg\psi_1 \vee \cdots \vee \neg\psi_k)$, we have $\vdash K\phi_1 \wedge \cdots \wedge K\phi_n \to [a](\neg\psi_1 \vee \cdots \vee \neg\psi_k)$. By rule NECK and axiom DISTK, we have $\vdash KK\phi_1 \wedge \cdots \wedge KK\phi_n \to K[a](\neg\psi_1 \vee \cdots \vee \neg\psi_k)$. Since $KK\phi_i \in s$ for each $1 \leq i \leq n$, we have $K[a](\neg\psi_1 \vee \cdots \vee \neg\psi_k) \in s$. By axiom PR$(a)$, it follows that $[a]K(\neg\psi_1 \vee \cdots \vee \neg\psi_k) \in s$. It follows by $sR_a^c t$ that $K(\neg\psi_1 \vee \cdots \vee \neg\psi_k) \in t$. Since $v \in \mathcal{U}_t^c$, by axiom T, we have $\neg\psi_1 \vee \cdots \vee \neg\psi_k \in v$. This is contrary with $\psi_i \in v$ for each $1 \leq i \leq k$. Thus u^- is consistent. By Lindenbaum-like Lemma, there exists a MCS u extending u^-. It follows by $u^- \subseteq u$ that $u \in \mathcal{U}_s^c$ and $uR_a^c v$. We conclude that $v \in \mathcal{U}_s^c|^a$.

Finally, we will show that $\mathcal{M}_s^c, s \vDash \phi$ iff $\phi \in s$. we prove it by induction on ϕ. We only focus on the case of $K\phi$ and $[a]\phi$; the others are obvious. By Claim 2.1 and induction hypothesis, it is straightforward that $\mathcal{M}_s^c, s \vDash K\phi$ iff $K\phi \in s$. With Claim 2.3, it follows that $\mathcal{M}_t^c = \mathcal{M}_s^c|^a$ if $s \xrightarrow{a} t$. Moreover, by Claim 2.1 and induction hypothesis, it can be shown that $\mathcal{M}_s^c, s \vDash [a]\phi$ iff $[a]\phi \in s$. $\square$

3. AN EXTENSION OF EAL FOR CONFORMANT PLANNING

3.1. Epistemic PDL over uncertainty maps

In this section we extend the language of EAL with programs in propositional dynamic logic and use this extended language to express the existence of a conformant plan.

Definition 3.1 (*Epistemic PDL*). The *Epistemic PDL Language* (EPDL) is defined as follows:

$$\phi ::= \top \mid p \mid \neg\phi \mid (\phi \wedge \phi) \mid [\pi]\phi \mid K\phi$$
$$\pi ::= a \mid ?\phi \mid (\pi;\pi) \mid (\pi+\pi) \mid \pi^*$$

where $p \in \mathtt{P}$, $a \in \mathtt{A}$. We use $(\!|\pi|\!)\phi$ to denote $[\pi]\phi \wedge \langle\pi\rangle\phi$, which is logically equivalent to $[\pi]\phi \wedge \langle\pi\rangle\top$. Given a finite $\mathtt{B} \subseteq \mathtt{A}$, we write $\mathtt{B}^*$ for $(\Sigma_{a\in\mathtt{B}}a)^*$, i.e., the iteration over the 'sum' (non-deterministic choices) of all the action symbols in B. We use $\Pi_{\mathtt{A}}$ to denote all the programs. The size of EPDL formulas/programs is given by: $|[\pi]\phi| = |\pi| + |\phi|$, $|p| = |a| = 1$, $|\pi_1;\pi_2| = 1 + |\pi_1| + |\pi_2|$, $|?\phi| = |\neg\phi| = |K\phi| = 1 + |\phi|$, $|\phi \wedge \psi| = 1 + |\phi| + |\psi|$, $|\pi_1 + \pi_2| = 1 + |\pi_1| + |\pi_2|$ and $|\pi^*| = 1 + |\pi|$.

Given any uncertainty map $\mathcal{M} = \langle \mathcal{S}, \{\mathcal{R}_a \mid a \in \mathtt{A}\}, \mathcal{V}, \mathcal{U}\rangle$, any state $s \in \mathcal{U}$, the semantics is given by a mutual induction on ϕ and π (we only show the case about $[\pi]\phi$, other cases are as in EAL):

$$\begin{array}{rcl}
\mathcal{M}, s \vDash [\pi]\phi & \Leftrightarrow & \text{for all } \mathcal{M}', s' : (\mathcal{M}, s)[\![\pi]\!](\mathcal{M}', s') \\
 & & \text{implies } \mathcal{M}', s' \vDash \phi \\
(\mathcal{M}, s)[\![a]\!](\mathcal{M}', s') & \Leftrightarrow & \mathcal{M}' = \mathcal{M}|^a \text{ and } s \xrightarrow{a} s' \\
(\mathcal{M}, s)[\![?\psi]\!](\mathcal{M}', s') & \Leftrightarrow & (\mathcal{M}', s') = (\mathcal{M}, s) \text{ and } \mathcal{M}, s \vDash \psi \\
(\mathcal{M}, s)[\![\pi_1;\pi_2]\!](\mathcal{M}', s') & \Leftrightarrow & (\mathcal{M}, s)[\![\pi_1]\!] \circ [\![\pi_2]\!](\mathcal{M}', s') \\
(\mathcal{M}, s)[\![\pi_1+\pi_2]\!](\mathcal{M}', s') & \Leftrightarrow & (\mathcal{M}, s)[\![\pi_1]\!] \cup [\![\pi_2]\!](\mathcal{M}', s') \\
(\mathcal{M}, s)[\![\pi^*]\!](\mathcal{M}', s') & \Leftrightarrow & (\mathcal{M}, s)[\![\pi]\!]^\star(\mathcal{M}', s')
\end{array}$$

where $\circ, \cup, ^\star$ at the right-hand side denote the usual composition, union and reflexive transitive closure of binary relations respectively. Clearly this semantics coincides with the semantics of EAL on EAL formulas.

Note that each program π can be viewed as a set of computation sequences, which are sequences of actions in A and tests with $\phi \in \mathtt{EPDL}$:

$$\begin{array}{l}
\mathcal{L}(a) = \{a\} \\
\mathcal{L}(?\phi) = \{?\phi\} \\
\mathcal{L}(\pi;\pi') = \{\sigma\eta \mid \sigma \in \mathcal{L}(\pi) \text{ and } \eta \in \mathcal{L}(\pi')\} \\
\mathcal{L}(\pi+\pi') = \mathcal{L}(\pi) \cup \mathcal{L}(\pi') \\
\mathcal{L}(\pi^*) = \{\epsilon\} \cup \bigcup_{n>0}(\mathcal{L}(\underbrace{\pi\cdots\pi}_{n})) \text{ where } \epsilon \text{ is the empty sequence}
\end{array}$$

We leave the complete axiomatization of EPDL on uncertainty maps to future work.

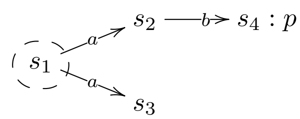
3.2. Generalized conformant planning via model checking EPDL

Let PL be the propositional language based on P. We can now define the conformant planning problem formally.

Definition 3.2 (*Standard conformant planning*). Given an uncertainty map $\mathcal{M}$, a goal formula $\phi \in \mathtt{PL}$, and a finite set $\mathtt{B} \subseteq \mathtt{A}$, the conformant planning problem is to find a finite (possibly empty) sequence $\sigma = a_1a_2\cdots a_n \in \mathcal{L}(\mathtt{B}^*)$ such that for each $u \in \mathcal{U}_{\mathcal{M}}$ we have $\mathcal{M}, u \vDash (\!|a_1|\!)(\!|a_2|\!)\cdots(\!|a_n|\!)\phi$, i.e., $\mathcal{M}, v \vDash K(\!|a_1|\!)(\!|a_2|\!)\cdots(\!|a_n|\!)\phi$ for any $v \in \mathcal{U}_{\mathcal{M}}$. The existence problem of conformant planning is to test whether such a sequence exists.

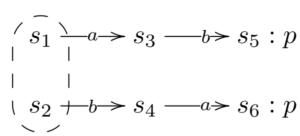
Recall that $\llparenthesis\pi\rrparenthesis\phi$ is the shorthand of $[\pi]\phi \wedge \langle\pi\rangle\top$. Intuitively, we want a plan which is both executable and safe w.r.t. non-deterministic actions and initial uncertainty of the agent. It is crucial to observe the difference between $\llparenthesis a_1\rrparenthesis\llparenthesis a_2\rrparenthesis\cdots\llparenthesis a_n\rrparenthesis\phi$ and $\llparenthesis a_1; a_2; \cdots; a_n\rrparenthesis\phi$ by the following example:

Example 3.3. Given the uncertainty map $\mathcal{M}$ depicted as follows, we have $\mathcal{M}, s_1 \vDash \llparenthesis a; b\rrparenthesis p$ but $\mathcal{M}, s_1 \nvDash \llparenthesis a\rrparenthesis\llparenthesis b\rrparenthesis p$. Intuitively there is no conformant plan to make sure p at s_1.



Given $\mathcal{M}$ and ϕ, to *verify* whether $\sigma \in \mathcal{L}(\pi)$ is a conformant plan can be formulated as the model checking problem: $\mathcal{M}, u \vDash K\llparenthesis a_1\rrparenthesis\llparenthesis a_2\rrparenthesis\cdots\llparenthesis a_n\rrparenthesis\phi$. On the other hand, the existence problem of a conformant plan is more complicated to formulate: it asks whether there *exists* a $\sigma \in \mathcal{L}(\mathtt{B}^*)$ such that it can be verified as a conformant plan. The simple-minded attempt would be to check whether $\mathcal{M}, u \vDash K\langle\mathtt{B}^*\rangle\phi$ holds. Despite the $\langle\cdot\rangle$-vs.-$\llparenthesis\cdot\rrparenthesis$ distinction, $K\langle\mathtt{B}^*\rangle\phi$ may hold on a model where the sequences to guarantee ϕ on different states in $\mathcal{U}_M$ are different, as the following example shows:

Example 3.4. Given the uncertainty map $\mathcal{M}$ depicted as follows, let the goal formula be p and $\mathtt{B} = \{a, b\}$. We have $\mathcal{M}, s_1 \vDash K\langle\mathtt{B}^*\rangle p$, but there is no solution to this conformant planning problem.



The correct formula to check for the existence of a conformant plan w.r.t. the actions in $\mathtt{B} \subseteq \mathtt{A}$ and a goal formula $\phi \in \mathtt{EPDL}$ is:

$$\theta_{\mathtt{B},\phi} = \langle(\Sigma_{a\in\mathtt{B}}(?K\langle a\rangle\top; a))^*\rangle K\phi.$$

For example, if $\mathtt{B} = \{a_1, a_2\}$ then $\theta_{\mathtt{B},\phi} = \langle((?K\langle a_1\rangle\top; a_1) + (?K\langle a_2\rangle\top; a_2))^*\rangle K\phi$. Intuitively, a conformant plan consists of actions that are always executable given the uncertainty of the agent (guaranteed by the guard $K\langle a\rangle\top$). In the end the plan should also make sure that ϕ must hold given the uncertainty of the agent (guaranteed by $K\phi$). In the following, we will prove that this formula is indeed correct.

First, we observe that the rule of substitution of equivalents is valid ($\phi(\psi/\chi)$ is obtained by replacing any occurrence of χ by ψ, similar for $[\![\pi(\psi/\chi)]\!]$):

PROPOSITION 3.5. *If* $\vDash \psi \leftrightarrow \chi$, *then we have* $\vDash \phi \leftrightarrow \phi(\psi/\chi)$ *and* $[\![\pi]\!] = [\![\pi(\psi/\chi)]\!]$.

PROPOSITION 3.6. $\vDash K\llparenthesis a\rrparenthesis\phi \leftrightarrow \langle ?K\langle a\rangle\top; a\rangle K\phi$

PROOF. Since $\vDash K\llparenthesis a\rrparenthesis\phi \leftrightarrow (K[a]\phi \wedge K\langle a\rangle\phi)$ and $\vDash (K\langle a\rangle\top \wedge \langle a\rangle K\phi) \leftrightarrow \langle ?K\langle a\rangle\top; a\rangle K\phi$, we only need to show that $\vDash (K[a]\phi \wedge K\langle a\rangle\phi) \leftrightarrow (K\langle a\rangle\top \wedge \langle a\rangle K\phi)$.

Left to right:
(L1) $\vDash K[a]\phi \to [a]K\phi$, by validity of Axiom $\mathtt{PR}(a)$
(L2) $\vDash K\langle a\rangle\phi \to \langle a\rangle\top \wedge K\langle a\rangle\top$, by semantics
(L3) $\vDash \langle a\rangle\top \wedge [a]K\phi \to \langle a\rangle K\phi$, by semantics
(L4) $\vDash K[a]\phi \wedge K\langle a\rangle\phi \to K\langle a\rangle\top \wedge \langle a\rangle K\phi$, by (L1)-(L3)
Right to left:
(R1) $\vDash \langle a\rangle K\phi \to K[a]\phi$, by validity of Axiom $\mathtt{NM}(a)$

(R2) $\vDash K[a]\phi \wedge K\langle a\rangle\top \to K\langle a\rangle\phi$, by semantics
(R3) $\vDash K\langle a\rangle\top \wedge \langle a\rangle K\phi \to K[a]\phi \wedge K\langle a\rangle\phi$, by R(1)-R(2) $\square$

LEMMA 3.7. *For any $a_1a_2\cdots a_n \in \mathcal{L}(\mathtt{A}^*)$, any $\phi \in$ EPDL:*

$$\vDash K\llparenthesis a_1\rrparenthesis\llparenthesis a_2\rrparenthesis\cdots\llparenthesis a_n\rrparenthesis\phi \leftrightarrow \langle ?K\langle a_1\rangle\top; a_1; \dots; ?K\langle a_n\rangle\top; a_n\rangle K\phi$$

PROOF. It is trivial when $n = 0$ (i.e., the sequence is ϵ), since the claim then boils down to $K\phi \leftrightarrow K\phi$. We prove the non-trivial cases by induction on $n \geq 1$. When $n = 1$, it follows from Proposition 3.6. Now, as the induction hypothesis, we assume that:

$$\vDash K\llparenthesis a_1\rrparenthesis\llparenthesis a_2\rrparenthesis\cdots\llparenthesis a_k\rrparenthesis\phi \leftrightarrow \langle ?K\langle a_1\rangle\top; a_1; \dots; ?K\langle a_k\rangle\top; a_k\rangle K\phi.$$

We need to show:

$$\vDash K\llparenthesis a_1\rrparenthesis\llparenthesis a_2\rrparenthesis\cdots\llparenthesis a_{k+1}\rrparenthesis\phi \leftrightarrow \langle ?K\langle a_1\rangle\top; a_1; \dots; ?K\langle a_{k+1}\rangle\top; a_{k+1}\rangle K\phi.$$

By IH,

$$\vDash K\llparenthesis a_1\rrparenthesis\llparenthesis a_2\rrparenthesis\cdots\llparenthesis a_{k+1}\rrparenthesis\phi \leftrightarrow \langle ?K\langle a_1\rangle\top; a_1; \dots; ?K\langle a_k\rangle\top; a_k\rangle K\llparenthesis a_{k+1}\rrparenthesis\phi. \qquad (1)$$

Due to Propositions 3.5 and 3.6, we have

$$\begin{aligned}&\vDash\langle ?K\langle a_1\rangle\top; a_1; \dots; ?K\langle a_k\rangle\top; a_k\rangle K\llparenthesis a_{k+1}\rrparenthesis\phi \leftrightarrow\\ &\quad\langle ?K\langle a_1\rangle\top; a_1; \dots; ?K\langle a_n\rangle\top; a_k\rangle\langle ?K\langle a_{k+1}\rangle\top; a_{k+1}\rangle K\phi. \ (2)\end{aligned}$$

The conclusion is immediate by combining (1) and (2). $\square$

Note that, the above proof does not require that $\phi \in$ PL.

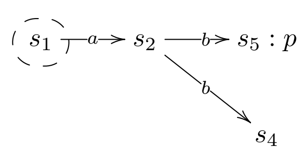
The following theorem follows from the above lemma.

THEOREM 3.8. *Given a pointed uncertainty map $\mathcal{M}, s$, a PL formula ϕ and a finite set $\mathtt{B} \subseteq \mathtt{A}$, the following two are equivalent:*

(1) There is a $\sigma = a_1 \dots a_n \in \mathcal{L}(\mathtt{B}^*)$ such that $\mathcal{M}, s \vDash K\llparenthesis a_1\rrparenthesis\llparenthesis a_2\rrparenthesis\cdots\llparenthesis a_n\rrparenthesis\phi$;
(2) $\mathcal{M}, s \vDash \langle(\Sigma_{a\in\mathtt{B}}(?K\langle a\rangle\top; a))^*\rangle K\phi$.

We would like to emphasize that the K operator right before ϕ in the definition of $\theta_{\mathtt{B},\phi}$ cannot be omitted, as demonstrated by the following example:

Example 3.9. Given the uncertainty map $\mathcal{M}$ depicted as follows, let the goal formula be p. As we can see, there is no solution to this conformant planning problem. Indeed $\mathcal{M}, s_1 \nvDash \langle(\Sigma_{a\in\mathtt{B}}(?K\langle a\rangle\top; a))^*\rangle Kp$ with $\mathtt{B} = \{a, b\}$, but we could have $\mathcal{M}, s_1 \vDash \langle(\Sigma_{a\in\mathtt{B}}(?K\langle a\rangle\top; a))^*\rangle p$.



3.3. Generalized conformant planning

In this subsection, we generalize the standard conformant planning and reduce it to the model checking problem of EPDL too.

The generalization is two-fold, as we mentioned in the introduction: on the goal and on the constraint on the desired plan. Formally:

Definition 3.10 (*Generalized conformant planning*). Given an uncertainty map $\mathcal{M}$, a goal formula $\phi \in$ EPDL, and a test-free (i.e., $?\phi$-free) $\pi \in \Pi_{\mathtt{A}}$, the generalized conformant planning problem is to find a finite (possibly empty) sequence $\sigma = a_1 \cdots a_n \in \mathcal{L}(\pi)$ such that for each $u \in \mathcal{U}_{\mathcal{M}}$ we have $\mathcal{M}, u \vDash (\!|a_1|\!)(\!|a_2|\!) \cdots (\!|a_n|\!)\phi$. The existence problem of conformant planning is to test whether such a sequence exists.

Note that the standard conformant planning is actually the generalized conformant planning w.r.t. $(\Sigma_{a \in \mathtt{B}} a)^*$ with the restriction on the PL goal formulas.

Let t be the translation of test-free programs such that each atomic action a is replaced by $(?K\langle a\rangle\top; a)$:

$$
\begin{aligned}
&t(a) = (?K\langle a\rangle\top; a) \\
&t(\pi;\pi') = t(\pi); t(\pi') \\
&t(\pi + \pi') = t(\pi) + t(\pi') \\
&t(\pi^*) = (t(\pi))^*
\end{aligned}
$$

It is not hard to show:

THEOREM 3.11. *Given a pointed uncertainty map $\mathcal{M}, s$, an EPDL formula ϕ and a test-free program $\pi \in \Pi_{\mathtt{A}}$, the following two are equivalent:*

(1) There is an action sequence $a_1 \ldots a_n \in \mathcal{L}(\pi)$ such that $\mathcal{M}, s \vDash K(\!|a_1|\!)(\!|a_2|\!) \cdots (\!|a_n|\!)\phi$;
(2) $\mathcal{M}, s \vDash \langle t(\pi)\rangle K\phi$.

PROOF SKETCH. Let π be a test-free program, then it can be shown by induction on π that $a_1 \cdots a_n \in \mathcal{L}(\pi)$ if and only if $t(a_1) \cdots t(a_n) \in \mathcal{L}(t(\pi))$. Moreover, it follows by Lemma 3.7 that $\vDash K(\!|a_1|\!)(\!|a_2|\!) \cdots (\!|a_n|\!)\phi \leftrightarrow \langle t(a_1); \cdots; t(a_n)\rangle K\phi$. Please note that the proof of Lemma 3.7 does not depend on the shape of ϕ. Therefore, (1) and (2) are equivalent. □

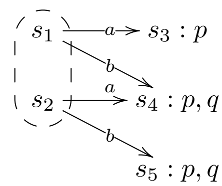
Note that Theorem 3.8 is now a special case of the above theorem where a particular $\pi = (\Sigma_{a \in \mathtt{B}} a)^*$ is fixed.

The new conformant planning problem is clearly more general, but do we really need this generality? First of all, let us see what we can express with EPDL formulas as goals.

Having positive epistemic goals such as Kq is nothing more than having the goal q since according to Theorem 3.11, the model checking problems w.r.t. to goals q or Kq are equivalent due to the fact that $\vDash Kq \leftrightarrow KKq$. However, we can also express negative epistemic goals such as $\neg Kq$ which is not expressible in the standard setting: $K\neg Kq$ is logically equivalent to $\neg Kq$ but not equivalent to any $K\phi$ formula where $\phi \in$ PL. In fact, a negative epistemic goal can be viewed as a set of sets of possible states, e.g., $\neg Kq$ amounts to a set of uncertainty sets that intersect with the set of $\neg q$ worlds.

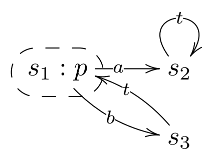
However, careful reader may wonder why we would ever want a negative epistemic goal at all. It definitely makes sense in the multi-agent setting: you may want a plan to know whether p but at the same time not to let your enemy know it as well. Actually, it also makes perfect sense in the single-agent setting. Suppose that there are two possible causes s_1 and s_2 for the same symptom of a patient, but you are not sure which is the actual cause. Now suppose you also have two drugs a and b which can both alleviate the symptom (p). However, drug b is bound to have a bad side-effect (q) but drug a only has it in the case of cause s_2. Which drug do you prefer? It seems that we would like to avoid the side-effect when possible, which amounts to the goal of $p \wedge \neg Kq$ or equivalently $p \wedge \hat{K}\neg q$. Here is a concrete example.

Example 3.12 (*negative epistemic goal*). Given the uncertainty map $\mathcal{M}$ depicted as follows, let the goal be p then both a and b are conformant plans. If the goal is $p \wedge \neg Kq$, only a is a good plan.



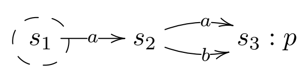
Besides negative epistemic goals, it is also useful to have goals involving actions and even programs. Consider the following example:

Example 3.13 (*goal with program*). Given the uncertainty map $\mathcal{M}$ depicted as follows. p denotes the proposition that a tooth hurts. Now you have two choices: you can either replace it with a false tooth (a) or fix the problem temporally without the replacement (b). The trouble for the second option is that it may go wrong again in some time (t). What would you choose? If your goal is $[t^*]\neg p$, which means free of worries forever, then a is clearly a better plan.



Now, let us consider the other aspect of our generalization: the constraints on the plan. Normally, the conformant plan is only an action sequence to guarantee the agent achieving the goal. However, in real-life scenarios, we may require that the plan should have a special form. For example, we would like the plan is such an action sequence that a occurs at most once since the cost of doing a is so high that we cannot afford twice.

Example 3.14 (*constraints on plan*). Given the uncertainty map $\mathcal{M}$ depicted as follows. p denotes the place you want to go. There are two kinds of transportation: by bus (a) or by walking (b). However, you can afford taking a bus only one time. Therefore, the solution should be a sequence in $\mathcal{L}(\pi)$ with $\pi = b^*; a; b^* + b^*$. To check if there is a solution for this generalized conformant planning, by Theorem 3.11 we only need to check $\mathcal{M}, s_1 \vDash \langle t(\pi)\rangle Kp$. It is easy to see that under this constraint only $a; b$ is a plan.



As we mentioned, there are also other ways to generalize conformant planning. For example, [Bonet 2010] extends conformant planning with both partial observability and epistemic formulas, but this generalization does not come for free. It is shown in [Bonet 2010] that the complexity of conformant planning with partial observability and epistemic formulas is higher than standard conformant planning. As we will show in the next section, our generalization is *for free* in complexity.

4. MODEL CHECKING EPDL: COMPLEXITY AND ALGORITHMS

In this section, we show that model checking EPDL on uncertainty maps is PSPACE-complete. Thus the generalized conformant planning problem is also PSPACE-complete based on the fact that the standard conformant planning is already PSPACE hard over transition systems. In particular, the lower bound is obtained by a reduction from QBF (*quantified Boolean formula*) truth testing to the model checking problem borrowing some ideas from temporal logic with past operators. The upper bound is shown by making use of an equivalent alternative semantics based on the model space over pointed uncertainty maps. It is somehow surprising that a PSPACE algorithm suffices to model check EPDL formulas over this potential exponentially larger model.

Note that throughout this section, we focus on uncertainty maps with finitely many states and assume $\mathcal{R}_a = \emptyset$ for *cofinitely* many $a \in \mathtt{A}$.

4.1. Lower Bound

To show the PSPACE lower bound, we provide a polynomial reduction of QBF truth testing to the model checking problem of EPDL. Note that to determine whether a given QBF (even in prenex normal form based on a conjunctive normal form) is true or not is known to be PSPACE-complete [Stockmeyer and Meyer 1973]. Our method is inspired by [Schnoebelen 2003] which discusses the complexity of model checking temporal logics with past operators. Surprisingly, we can use the uncertainty sets to record the 'past' and use the dual of the knowledge operator to 'go back' to the past. This intuitive idea will become more clear in the proof.

QBF formulas are $Q_1 x_1 Q_2 x_2 \dots Q_n x_n \phi(x_1, \dots, x_n)$ where:

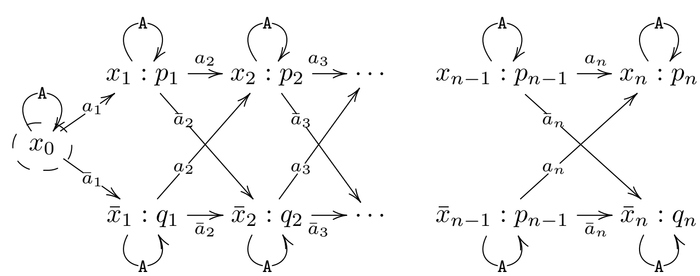
— For $1 \le i \le n, Q_i$ is '$\exists$' if i is odd, and Q_i is '$\forall$' if i is even.
— ϕ is a propositional formula in CNF based on variables $x_1, \dots, x_n$,

For each such QBF α with n variables, we need to find a pointed model $\mathcal{M}_n, x_0$ and a formula θ_α such that α is true iff $\mathcal{M}_n, x_0 \vDash \theta_\alpha$. The model $\mathcal{M}_n$ is defined below.

Definition 4.1. Let $\mathtt{A} = \{a_i, \bar{a}_i \mid i \ge 1\}$ and $\mathtt{P} = \{p_k, q_k \mid k \ge 1\}$, the uncertainty map $\mathcal{M}_n = \langle \mathcal{S}, \{\mathcal{R}_a \mid a \in \mathtt{A}\}, \mathcal{V}, \mathcal{U} \rangle$ is defined as:

— $\mathcal{S} = \{x_0\} \cup \{x_i \mid 1 \le i \le n\} \cup \{\bar{x}_i \mid 1 \le i \le n\}$
— $\mathcal{V}(x_0) = \emptyset$, and $\mathcal{V}(x_i) = \{p_i\}, \mathcal{V}(\bar{x}_i) = \{q_i\}$ for $1 \le i \le n$.
— $\xrightarrow{a_i} = \{(s,s) \mid s \in \mathcal{S}\} \cup \{(x_{i-1}, x_i), (\bar{x}_{i-1}, x_i)\}$
— $\xrightarrow{\bar{a}_i} = \{(s,s) \mid s \in \mathcal{S}\} \cup \{(x_{i-1}, \bar{x}_i), (\bar{x}_{i-1}, \bar{x}_i)\}$
— $\mathcal{U} = \{x_0\}$

$|\mathcal{M}_n|$ is linear in n and can be depicted as the following:



Given $\alpha = Q_1 x_1 Q_2 x_2 \dots Q_n x_n \phi(x_1, \dots, x_n)$, the formula θ_α is defined as

$$QT_1 \cdots QT_n \psi(\hat{K}p_1, \cdots, \hat{K}p_n, \hat{K}q_1, \cdots, \hat{K}q_n)$$

where QT_i is $\langle (a_i + \bar{a}_i); ?(p_i \vee q_i) \rangle$ if i is odd and QT_i is $[(a_i + \bar{a}_i); ?(p_i \vee q_i)]$ if i is even, and ψ is obtained from $\phi(x_1, \dots, x_n)$ by replacing each x_i with $\hat{K}p_i$ and $\neg x_i$ with $\hat{K}q_i$.

To ease the latter proof, we first define the valuation tree below.

Definition 4.2 (*V-tree*). A V-tree τ is a rooted tree such that 1) each node is 0 or 1 (except the root ϵ); 2) each internal node in an even level has only one successor; 3) each internal node in an odd level has two successors: one is 0 and the other one is 1; 4) each edge to node 0 of level i is labelled $\bar{a}_i$; 5) each edge to node 1 of level i is labelled a_i. Given a V-tree with depth n, a path σ is a sequence of $A_1 \ldots A_n$ where $A_i = a_i$ or $A_i = \bar{a}_i$. A path σ can also be seen as a valuation assignment for $x_1, \ldots, x_n$ with the convention that $\sigma(x_i) = 1$ if a_i occurs in σ, and $\sigma(x_i) = 0$ if $\bar{a}_i$ occurs in σ. Let $path(\tau)$ be the set of all paths of τ.

As an example, a V-tree τ can be depicted as below:

$$\begin{tikzcd}[row sep=tiny] & & 1 \arrow[r, "\bar{a}_3"] & 0 \\ \epsilon \arrow[r, "a_1"] & 1 \arrow[ur, "a_2"] \arrow[dr, "\bar{a}_2"'] & & \\ & & 0 \arrow[r, "a_3"] & 1 \end{tikzcd}$$

It is not hard to see the following:

PROPOSITION 4.3. *For each $1 \le i \le n$, we have $\alpha = Q_1 x_1 \ldots Q_i x_i Q_{i+1} x_{i+1} \ldots Q_n x_n \phi$ is true iff there exists a V-tree τ with depth i such that $\sigma(Q_{i+1} x_{i+1} \ldots Q_n x_n \phi) = 1$ for each $\sigma \in path(\tau)$ (σ is viewed as a valuation).*

Now let us see the update result of running a path $\sigma \in path(\tau)$ on $\mathcal{M}_n$.

PROPOSITION 4.4. *Given $\mathcal{M}_n$, let $\sigma = A_1 \ldots A_i$ $(1 \le i \le n)$ be a sequence of actions such that $A_k = a_k$ or $A_k = \bar{a}_k$ for each $1 \le k \le i$, then we have $\mathcal{U}|^{\sigma} = \{x_0, X_1, \ldots, X_i\}$ where $X_k = x_k$ if $A_k = a_k$ else $X_k = \bar{x}_k$ for each $1 \le k \le i$.*

PROOF. We prove it by induction on i. It is obvious if $i = 1$. Next we need to show the proposition holds when $i+1$. By IH, it follows that $\mathcal{U}' = \mathcal{U}|^{A_1 \ldots A_i} = \{x_0, X_1, \ldots, X_i\}$ where $X_k = x_k$ if $A_k = a_k$ else $X_k = \bar{x}_k$ for each $1 \le k \le i$. Since $\mathcal{U}'|^{A_{i+1}} = \{x_0, X_1, \ldots, X_{i+1}\}$ where $X_{i+1} = x_{i+1}$ or $X_{i+1} = \bar{x}_{i+1}$. Therefore, we only need to show that if $A_{i+1} = a_{i+1}$ then $X_{i+1} = x_{i+1}$ else $X_{i+1} = \bar{x}_{i+1}$. If $A_{i+1} = a_{i+1}$, since $X_i \overset{a_{i+1}}{\to} x_{i+1}$, we have $X_{i+1} = x_{i+1}$. If $A_{i+1} \neq a_{i+1}$, it follows that $A_{i+1} = \bar{a}_{i+1}$. since $X_i \overset{\bar{a}_{i+1}}{\to} \bar{x}_{i+1}$, we have $X_{i+1} = \bar{x}_{i+1}$. □

Given $\sigma = A_1 \ldots A_n$ where A_i is a_i or $\bar{a}_i$ for each $1 \le i \le n$, let $g(\sigma) = x_n$ if $A_n = a_n$ and $g(\sigma) = \bar{x}_n$ if $A_n = \bar{a}_n$. By Proposition 4.4, we always have $g(\sigma) \in \mathcal{U}_{\mathcal{M}_k}|^{\sigma}$ with $k > n$. Thus given $\mathcal{M}_k$ and $\sigma = A_1 \ldots A_n$ and $k > n$, $\mathcal{M}_k|^{\sigma}, g(\sigma)$ is a pointed uncertainty map.

PROPOSITION 4.5. *For each $1 \le i \le n$, we have $\mathcal{M}_k, x_0 \vDash QT_1 \ldots QT_i QT_{i+1} \ldots QT_n \psi$ iff there exists a V-tree τ with depth i such that $\mathcal{M}_k|^{\sigma}, g(\sigma) \vDash QT_{i+1} \ldots QT_n \psi$ for each $\sigma \in path(\tau)$, where $k > n$ and $g(\sigma)$ is the state corresponds to the last edge of σ, e.g., $g(a_1 \bar{a}_2) = \bar{x}_2$.*

PROOF. We prove it by induction on i. It is obvious when $i = 1$. Next we will show that the proposition holds when $i+1$. By IH,we only need to show that there exists a V-tree τ with depth i such that $\mathcal{M}_k|^{\sigma}, g(\sigma) \vDash QT_{i+1} \ldots QT_n \psi$ for all $\sigma \in path(\tau)$, if and only if, there exists a V-tree τ' with depth $i+1$ such that $\mathcal{M}_k|^{\sigma'}, g(\sigma') \vDash QT_{i+2} \ldots QT_n \psi$ for all $\sigma' \in path(\tau')$.

Left-to-Right: Since τ' is a V-tree with depth $i+1$, we can get the V-tree τ with depth i by cutting the leavies and the last edge of the V-tree τ'. Then we only need to show that $\mathcal{M}_k|^{\sigma}, g(\sigma) \vDash QT_{i+1} \ldots QT_n \psi$ for all $\sigma \in path(\tau)$. Since for each $\sigma \in path(\tau)$ there exists $\sigma' \in path(\tau')$ such that $\sigma' = \sigma a_{i+1}$ or $\sigma' = \sigma \bar{a}_{i+1}$, and $\mathcal{M}_k|^{\sigma'}, g(\sigma') \vDash QT_{i+2} \ldots QT_n \psi$ $(\triangle)$, next we will show $\mathcal{M}_k|^{\sigma}, g(\sigma) \vDash QT_{i+1} \ldots QT_n \psi$ no matter $i+1$ is odd or even.

— $i+1$ is odd. If $\sigma' = \sigma a_{i+1} \in path(\tau')$, it follows that $g(\sigma') = x_{i+1}$. Thus we have $g(\sigma) \overset{a_{i+1}}{\to} x_{i+1}$ and $\mathcal{M}_k|^{\sigma a_{i+1}}, x_{i+1} \vDash QT_{i+2} \ldots QT_n\psi$. What is more, because of $p_{i+1} \in \mathcal{V}(x_{i+1})$, we have $\mathcal{M}_k|^{\sigma}, g(\sigma) \vDash \langle (a_{i+1} + \bar{a}_{i+1}); ?(p_{i+1} \vee q_{i+1}) \rangle \ldots QT_n\psi$, namely $\mathcal{M}_k|^{\sigma}, g(\sigma) \vDash QT_{i+1} \ldots QT_n\psi$. Similarly, we can also get $\mathcal{M}_k|^{\sigma}, g(\sigma) \vDash QT_{i+1} \ldots QT_n\psi$ if $\sigma' = \sigma\bar{a}_{i+1}$.

— $i+1$ is even. It follows that both σa_{i+1} and $\sigma\bar{a}_{i+1}$ are members of $path(\tau')$. Assume that $\mathcal{M}_k|^{\sigma}, g(\sigma) \nvDash QT_{i+1} \ldots QT_n\psi$, namely $\mathcal{M}_k|^{\sigma}, g(\sigma) \nvDash [(a_{i+1} + \bar{a}_{i+1})](p_{i+1} \vee q_{i+1} \to QT_{i+2} \ldots QT_n\psi)$. Since $g(\sigma) \overset{a_{i+1}}{\to} x_{i+1}$ and $g(\sigma) \overset{\bar{a}_{i+1}}{\to} \bar{x}_{i+1}$, it follows that $\mathcal{M}_k|^{\sigma a_{i+1}}, x_{i+1} \nvDash QT_{i+2} \ldots QT_n\psi$ or $\mathcal{M}_k|^{\sigma\bar{a}_{i+1}}, \bar{x}_{i+1} \nvDash QT_{i+2} \ldots QT_n\psi$. However, each of them are contradictary with $(\triangle)$. Therefore, we have $\mathcal{M}_k|^{\sigma}, g(\sigma) \vDash QT_{i+1} \ldots QT_n\psi$.

Left-to-Right: Since τ is a V-tree with depth i and $\mathcal{M}_k|^{\sigma}, g(\sigma) \vDash QT_{i+1} \ldots QT_n\psi$ for all $\sigma \in path(\tau)$, we can get the V-tree τ' with depth $i+1$ by extending each $\sigma \in path(\tau)$. There are two situations:

— $i+1$ is odd. Since $\mathcal{M}_k|^{\sigma}, g(\sigma) \vDash \langle (a_{i+1} + \bar{a}_{i+1}); ?(p_{i+1} \vee q_{i+1}) \rangle \ldots QT_n\psi$, it follows that $\mathcal{M}_k|^{\sigma}, g(\sigma) \vDash \langle a_{i+1} \rangle \langle ?(p_{i+1} \vee q_{i+1}) \rangle \ldots QT_n\psi$ or $\mathcal{M}_k|^{\sigma}, g(\sigma) \vDash \langle \bar{a}_{i+1} \rangle \langle ?(p_{i+1} \vee q_{i+1}) \rangle \ldots QT_n\psi$. If it is the case of $\mathcal{M}_k|^{\sigma}, g(\sigma) \vDash \langle a_{i+1} \rangle \langle ?(p_{i+1} \vee q_{i+1}) \rangle \ldots QT_n\psi$, we extend σ to be σa_{i+1}, and it is obvious that $\mathcal{M}_k|^{\sigma a_{i+1}}, g(\sigma a_{i+1}) \vDash QT_{i+2} \ldots QT_n\psi$. If it is the case of $\mathcal{M}_k|^{\sigma}, g(\sigma) \vDash \langle \bar{a}_{i+1} \rangle \langle ?(p_{i+1} \vee q_{i+1}) \rangle \ldots QT_n\psi$, we extend σ to be $\sigma\bar{a}_{i+1}$, and it is obvious that $\mathcal{M}_k|^{\sigma\bar{a}_{i+1}}, g(\sigma\bar{a}_{i+1}) \vDash QT_{i+2} \ldots QT_n\psi$.

— $i+1$ is even. We split eahc σ in to σa_{i+1} and $\sigma\bar{a}_{i+1}$. By assumption, we know that $\mathcal{M}_k|^{\sigma}, g(\sigma) \vDash [(a_{i+1} + \bar{a}_{i+1}); ?(p_{i+1} \vee q_{i+1})] \ldots QT_n\psi$, namely $\mathcal{M}_k|^{\sigma}, g(\sigma) \vDash [(a_{i+1} + \bar{a}_{i+1})](p_{i+1} \vee q_{i+1} \to QT_{i+2} \ldots QT_n\psi)$. What is more, because of $\mathcal{M}_k|^{\sigma a_{i+1}}, g(\sigma a_{i+1}) \vDash (p_{i+1} \vee q_{i+1}$ and $\mathcal{M}_k|^{\sigma\bar{a}_{i+1}}, g(\sigma\bar{a}_{i+1}) \vDash (p_{i+1} \vee q_{i+1}$, it follows that $\mathcal{M}_k|^{\sigma a_{i+1}}, g(\sigma a_{i+1}) \vDash QT_{i+2} \ldots QT_n\psi$ and $\mathcal{M}_k|^{\sigma\bar{a}_{i+1}}, g(\sigma\bar{a}_{i+1}) \vDash QT_{i+2} \ldots QT_n\psi$.

□

THEOREM 4.6. *The following two are equivalent:*

— $\alpha = Q_1x_1Q_2x_2 \ldots Q_nx_n\phi(x_1, \ldots, x_n)$ *is true*

— $\mathcal{M}_n, x_0 \vDash QT_1 \cdots QT_n\psi(\hat{K}p_1 \cdots \hat{K}p_n, \hat{K}q_1 \cdots \hat{K}q_n)$ *in which* ψ *is obtained from* ϕ *by replacing each* x_i *with* $\hat{K}p_i$ *and* $\neg x_i$ *with* $\hat{K}q_i$.

PROOF. By Propositions 4.3 and 4.5, we only need to show that given V-tree τ with depth n, $\sigma(\phi) = 1$ if and only if $\mathcal{M}_n|^{\sigma}, g(\sigma) \vDash \psi$ for each $\sigma \in path(\tau)$. Since ϕ is in CNF, ψ is also in CNF-like form obtained by replacing each x_i with $\hat{K}p_i$ and each $\neg x_i$ with $\hat{K}q_i$ for $1 \leq i \leq n$. Thus we only need to show that $\sigma(x_i) = 1$ iff $\mathcal{M}_n|^{\sigma}, g(\sigma) \vDash \hat{K}p_i$, and $\sigma(\neg x_i) = 1$ iff $\mathcal{M}_n|^{\sigma}, g(\sigma) \vDash \hat{K}q_i$. Since $\sigma(x_i) = 1$ iff $\sigma(\neg x_i) = 0$, we only need to show that $\mathcal{M}_n|^{\sigma}, g(\sigma) \vDash \hat{K}p_i$ iff $\mathcal{M}_n|^{\sigma}, g(\sigma) \vDash \neg\hat{K}q_i$, and $\sigma(x_i) = 1$ iff $\mathcal{M}_n|^{\sigma}, g(\sigma) \vDash \hat{K}p_i$.

Firstly, we will show that $\mathcal{M}_n|^{\sigma}, g(\sigma) \vDash \hat{K}p_i$ iff $\mathcal{M}_n|^{\sigma}, g(\sigma) \vDash \neg\hat{K}q_i$. Because of $\mathcal{V}(p_i) = \{x_i\}$, we know that $\mathcal{M}_n|^{\sigma}, g(\sigma) \vDash \hat{K}p_i$ iff $x_i \in \mathcal{U}|^{\sigma}$. It follows by Proposition 4.4 that $x_i \in \mathcal{U}|^{\sigma}$ iff $\bar{x}_i \notin \mathcal{U}|^{\sigma}$. Because of $\mathcal{V}(q_i) = \{\bar{x}_i\}$, we know that $\bar{x}_i \notin \mathcal{U}|^{\sigma}$ iff $\mathcal{M}_n|^{\sigma}, g(\sigma) \vDash \neg\hat{K}q_i$. Therefore, we have $\mathcal{M}_n|^{\sigma}, g(\sigma) \vDash \hat{K}p_i$ iff $\mathcal{M}_n|^{\sigma}, g(\sigma) \vDash \neg\hat{K}q_i$.

Next, we will show that $\sigma(x_i) = 1$ iff $\mathcal{M}_n|^{\sigma}, g(\sigma) \vDash \hat{K}p_i$. By the definition of τ, we know that $\sigma = A_1 \ldots A_n$ where A_i is a_i or $\bar{a}_i$ for each $1 \leq i \leq n$. By our convention, we know that $\sigma(x_i) = 1$ iff $A_i = a_i$. It follows by Proposition 4.4 that $A_i = a_i$ iff $x_i \in \mathcal{U}|^{\sigma}$. Because of $\mathcal{V}(p_i) = \{x_i\}$, we know that $x_i \in \mathcal{U}|^{\sigma}$ iff $\mathcal{M}_n|^{\sigma}, g(\sigma) \vDash \hat{K}p_i$. Therefore, we have $\sigma(x_i) = 1$ iff $\mathcal{M}_n|^{\sigma}, g(\sigma) \vDash \hat{K}p_i$. □

This gives us the desired lower bound:

THEOREM 4.7. *The model checking problem for* EPDL *is* PSPACE*-hard.*

Note that the PSPACE-hardness also holds for the star-free fragment of EPDL since θ_α does not use the Kleene star.

4.2. Upper Bound

In this section, we give a PSPACE model checking method for EPDL via model checking EPDL over two-dimensional models with both epistemic and action relations. Let us first define such models.

Definition 4.8 (*Epistemic Temporal Structure*). An *Epistemic Temporal Structure* (ETS) is a Kripke model with both epistemic and action relations. Formally, an ETS model $\mathfrak{M}$ is a tuple $\langle \mathcal{S}, \{\mathcal{R}_a \mid a \in \mathtt{A}\}, \sim, \mathcal{V}\rangle$, where $\mathcal{R}_a$ is a binary relation on $\mathcal{S}$, $\sim$ is an equivalence relation on $\mathcal{S}$ and $\mathcal{V} : \mathcal{S} \to 2^{\mathtt{P}}$ is a valuation function.

Now we define an alternative semantics of EPDL over ETSs.

Definition 4.9 (*ETS Semantics*). Given any ETS model $\mathfrak{M} = \langle \mathcal{S}, \{\mathcal{R}_a \mid a \in \mathtt{A}\}, \sim, \mathcal{V}\rangle$ and any state $s \in \mathcal{S}$, the satisfaction relation for EPDL formulas is defined as follows (the Boolean cases are as in the standard modal logic):

$$
\begin{array}{|lcl|}
\hline
\mathfrak{M}, s \Vdash K\phi & \Leftrightarrow & \forall u \in \mathcal{S} : s \sim u \text{ implies } \mathfrak{M}, u \Vdash \phi \\
\mathfrak{M}, s \Vdash [\pi]\phi & \Leftrightarrow & \forall t \in S : s \xrightarrow{\pi} t \text{ implies } \mathfrak{M}, t \Vdash \phi \\
\xrightarrow{a} & = & \mathcal{R}_a \\
\xrightarrow{?\phi} & = & \{(s, s) \mid \mathfrak{M}, s \Vdash \phi\} \\
\xrightarrow{\pi_1;\pi_2} & = & \xrightarrow{\pi_1} \circ \xrightarrow{\pi_2} \\
\xrightarrow{\pi_1+\pi_2} & = & \xrightarrow{\pi_1} \cup \xrightarrow{\pi_2} \\
\xrightarrow{\pi^*} & = & (\xrightarrow{\pi})^\star \\
\hline
\end{array}
$$

where $\circ, \cup, ^\star$ at right-hand side denote the usual composition, union and reflexive transitive closure of binary relations respectively.

We can turn a Kripke model without the epistemic relation into an ETS model by essentially considering all the possible uncertainty sets.

Definition 4.10. Given any Kripke model $\mathcal{N} = \langle \mathcal{S}, \{\mathcal{R}_a \mid a \in \mathtt{A}\}, \mathcal{V}\rangle$, we define the ETS model $\mathcal{N}^\bullet$ as follows:

$$
\begin{array}{lcl}
\mathcal{S}^\bullet & = & \{s_\Gamma \mid s \in \mathcal{S}, \Gamma \in 2^{\mathcal{S}}, s \in \Gamma\} \\
\mathcal{R}_a^\bullet & = & \{(s_\Gamma, t_\Delta) \mid s \xrightarrow{a} t, \Delta = \Gamma|^a\} \\
\sim^\bullet & = & \{(s_\Gamma, t_\Delta) \mid \Gamma = \Delta\} \\
\mathcal{V}^\bullet(s_\Gamma) & = & \mathcal{V}(s)
\end{array}
$$

where $\Gamma|^a = \{t \in \mathcal{S} \mid \exists s \in \Gamma \text{ such that } s \xrightarrow{a} t\}$. For any Kripke model $\mathcal{N}$ and any $\Gamma \in 2^{\mathcal{S}} \backslash \{\emptyset\}$, let $\mathcal{N}^\Gamma$ be the uncertainty map $\langle \mathcal{N}, \Gamma\rangle$.

Note that each s_Γ can be viewed as an uncertainty set (Γ) with a designated state (s), and the definition of $\mathcal{R}_a$ captures the update in the $\vDash$ semantics of EPDL, and $\mathcal{N}^\bullet$ unravels all the updates in a whole picture. Note that the size of $\mathcal{N}^\bullet$ is $|\mathcal{S}| \cdot 2^{|\mathcal{S}|-1}$ where $\mathcal{S}$ is the set of states of $\mathcal{N}$.

Now we can show that $\vDash$ and $\Vdash$ coincide w.r.t. uncertainty map $\mathcal{N}^\Gamma$ and ETS model $\mathcal{N}^\bullet$.

PROPOSITION 4.11. *Given any map $\mathcal{N}$, we have*

(i) $\mathcal{N}^\Gamma, s[\![\pi]\!]\mathcal{N}^\Delta, t$ iff $s_\Gamma \xrightarrow{\pi} t_\Delta$ in $\mathcal{N}^\bullet$ (Cf. the definition of $\xrightarrow{\pi}$ in Def. 4.9);

(ii) $\mathcal{N}^\Gamma, s \vDash \phi$ *iff* $\mathcal{N}^\bullet, s_\Gamma \Vdash \phi$.

PROOF. The proof is by simultaneous induction on π and ϕ (due to the test actions). For (i), we will only focus on the non-trivial cases of $\pi_1;\pi_2$ and π^*; the other cases are straightforward.

Case $\pi_1;\pi_2$: We have $\mathcal{N}^\Gamma, s[\![\pi_1;\pi_2]\!]\mathcal{N}^\Delta, t$ if and only if $\mathcal{N}^\Gamma, s[\![\pi_1]\!]\mathcal{N}^{\Gamma'}, s'$ and $\mathcal{N}^{\Gamma'}, s'[\![\pi_2]\!]\mathcal{N}^\Delta, t$ for some $\Gamma' \in 2^{\mathcal{S}}$. By IH, this amounts to $s_\Gamma \xrightarrow{\pi_1} s'_{\Gamma'}$ and $s'_{\Gamma'} \xrightarrow{\pi_2} t_\Delta$. Thus, we have $s_\Gamma \xrightarrow{\pi_1;\pi_2} t_\Delta$.

Case π^*: By induction on n, it can be proved that for each n, $\mathcal{N}^\Gamma, s[\![\pi^n]\!]\mathcal{N}^\Delta, t$ if and only if $s_\Gamma \xrightarrow{\pi^n} t_\Delta$. This follows that $\mathcal{N}^\Gamma, s[\![\pi^*]\!]\mathcal{N}^\Delta, t$ if and only if $s_\Gamma \xrightarrow{\pi^*} t_\Delta$.

For (ii), we will only focus on the case of $[\pi]\phi$; the other cases are straightforward.

Case $[\pi]\phi$: If $\mathcal{N}^\Gamma, s \vDash [\pi]\phi$ but $\mathcal{N}^\bullet, s_\Gamma \nVdash [\pi]\phi$, then $\mathcal{N}^\bullet, t_\Delta \nVdash \phi$ for some $t_\Delta \in \mathcal{S}^\bullet$ such that $s_\Gamma \xrightarrow{\pi} t_\Delta$. By IH, this follows $\mathcal{N}^\Delta, t \nvDash \phi$ and $\mathcal{N}^\Gamma, s[\![\pi]\!]\mathcal{N}^\Delta, t$. This is contradictory with the assumption that $\mathcal{N}^\Gamma, s \vDash [\pi]\phi$. If $\mathcal{N}^\bullet, s_\Gamma \Vdash [\pi]\phi$ but $\mathcal{N}^\Gamma, s \nvDash [\pi]\phi$, it follows that $\mathcal{N}^\Gamma, s[\![\pi]\!]\mathcal{N}^\Delta, t$ and $\mathcal{N}^\Delta, t \nvDash \phi$ for some $\Delta \in 2^{\mathcal{S}}$ and $t \in \Delta$. By IH, we have $s_\Gamma \xrightarrow{\pi} t_\Delta$ and $\mathcal{N}^\bullet, t_\Delta \nVdash \phi$. This is contradictory with $\mathcal{N}^\bullet, s_\Gamma \Vdash [\pi]\phi$. $\square$

COROLLARY 4.12. *Given an uncertainty map* $\mathcal{M} = \langle \mathcal{N}, \mathcal{U} \rangle$ *and* $s \in \mathcal{U}$, *we have* $\mathcal{M}, s \vDash \phi$ *iff* $\mathcal{N}^\bullet, s_\mathcal{U} \Vdash \phi$.

Based on the above corollary, in order to check $\mathcal{M}, s \vDash \phi$, we only need to check $\mathcal{N}^\bullet, s_\mathcal{U} \Vdash \phi$ which is rather standard for PDL-like logics with PTIME algorithm (cf. e.g., [Lange 2006]). However, to build $\mathcal{N}^\bullet$ we need to exponentially blow up the state space and this led us to conjecture in [Yu et al. 2015] that the complexity of model checking EPDL over uncertainty maps could be EXPTIME-hard. Luckily, the exponential blow-up is avoidable based on the following important observation:

> *We do not need to compute the whole $\mathcal{N}^\bullet$ beforehand, the only thing to be computed is the one-step transition between two states in $\mathcal{N}^\bullet$ when needed, and this will give us the* PSPACE *algorithm.*

Let us start from some obvious observation: if a node is reachable from another one in a graph then the length of the shortest path connecting the two should not exceed the cardinality of the graph.

PROPOSITION 4.13. *Let $\mathcal{N}$ be a Kripke model with cardinality n, and $\mathcal{N}^\bullet$ is constructed as Definition 4.10. We have that for each π, $s_\Gamma \xrightarrow{\pi^*} t_\Delta$ iff $s_\Gamma \xrightarrow{\pi^k} t_\Delta$ where $k \leq n \cdot 2^{n-1}$ ($n \cdot 2^{n-1}$ is the cardinality of the domain of $\mathcal{N}^\bullet$).*

Given a Kripke model $\mathcal{N}$, a state set $\Gamma \subseteq \mathcal{S}$, a state $s \in \Gamma$ and a formula ϕ, next we will give PSPACE algorithms for checking $\mathcal{N}^\bullet, s_\Gamma \Vdash \phi$.

Suppose that $|\mathcal{S}_\mathcal{N}| = n$ and $|\phi| = m$. Algorithm 1 checks the atomic program whether $(s_\Gamma, t_\Delta) \in \mathcal{R}^\bullet_a$. Algorithm 1 uses variables Γ' and $\mathcal{R}_a$ to calculate the state set $\Gamma|^a$. Therefore it requires $O(n^2)$ space.

Algorithm 2 below checks whether $s_\Gamma \xrightarrow{\pi^k} t_\Delta$ in $\mathcal{N}^\bullet$ for a given k. For Algorithm 2, the most space-demanding part is the $k > 1$ case using the midpoints recursively. Because of $k \leq n \cdot 2^{n-1}$, the depth of the recursion is bounded by $O(n \cdot log\ n)$.

Algorithm 3 is the main model checking algorithm. Note that Algorithms 2 and 3 involve mutual recursion of each other due to the tests in programs. However, the depth of the recursion is bounded by a polynomial of $|\phi| \cdot |\mathcal{N}|$, and for each call polynomial space suffices. Except the recursion step, the space usage of Algorithms 2 and

Algorithm 1: Function $BR(s_\Gamma, t_\Delta, a)$: Check whether $(s_\Gamma, t_\Delta) \in \mathcal{R}^\bullet_a$.

input : s_Γ, t_Δ, a
output: true if $(s_\Gamma, t_\Delta) \in \mathcal{R}^\bullet_a$

/* Firstly, we calculate $\Gamma|^a$. */
Let Γ' be an empty set;
foreach $u \in \Gamma$ **do**
  foreach $v \in \mathcal{S}$ **do**
    if $(u, v) \in \mathcal{R}_a$ **then** put v in Γ';
if $\Delta = \Gamma'$ *and* $(s, t) \in \mathcal{R}_a$ **then return** true **else return** false;

Algorithm 2: Function $CR(s_\Gamma, t_\Delta, \pi, k)$: Check whether $s_\Gamma \xrightarrow{\pi^k} t_\Delta$ in $\mathcal{N}^\bullet$.

input : s_Γ, t_Δ, π, k
output: true if $s_\Gamma \xrightarrow{\pi^k} t_\Delta$ in $\mathcal{N}^\bullet$

if $k = 0$ **then return** $s = t \wedge \Gamma = \Delta$;
else if $k = 1$ **then**
  switch π **do**
    case a BR(s_Γ, t_Δ, a) ;
    case $?\psi$
      if $s = t$ *and* $\Gamma = \Delta$ **then return** MC($\mathcal{N}, s_\Gamma, \psi$) **else return** false;
    case $\pi_1 + \pi_2$ **return** $CR(s_\Gamma, t_\Delta, \pi_1, 1) \vee CR(s_\Gamma, t_\Delta, \pi_2, 1)$;
    case $\pi_1; \pi_2$
      foreach $\Gamma' \subseteq \mathcal{S}_\mathcal{N}$ **do**
        foreach $s' \in \Gamma'$ **do**
          if $CR(s_\Gamma, s'_{\Gamma'}, \pi_1, 1) \wedge CR(s'_{\Gamma'}, t_\Delta, \pi_2, 1)$ **then return** true ;
      return false;
    case π^* /* By Proposition 4.13, we only need to check π^i. */
      for $i \leftarrow 0$ **to** $n \cdot 2^{n-1}$ **do**
        if $CR(s_\Gamma, t_\Delta, \pi, i)$ **then return** true;
      return false;
else
  foreach $\Gamma' \subseteq \mathcal{S}_\mathcal{N}$ **do**
    foreach $s' \in \Gamma'$ **do**
      if $CR(s_\Gamma, s'_{\Gamma'}, \pi, \lfloor k/2 \rfloor) \wedge CR(s'_{\Gamma'}, t_\Delta, \pi, \lceil k/2 \rceil)$ **then return** true ;
  return false;

3 is bounded by $O(n^2)$. Thus the overall running space of Algorithm 3 is bounded by $O(m \cdot n^3 \cdot log\ n)$.

THEOREM 4.14. *The model checking problem of* EPDL *on uncertainty maps is in* PSPACE.

As we mentioned in the introduction, the conformant planning problems in the AI literature are usually given by using state variables and actions with preconditions and (conditional) effects, rather than explicit transition systems. The corresponding

Algorithm 3: Function MC($\mathcal{N}, s_\Gamma, \phi$): Model checking algorithm for EPDL (Boolean cases omitted)

input : $\mathcal{N}, s_\Gamma, \phi$
output: true if $\mathcal{N}^\bullet, s_\Gamma \Vdash \phi$.

switch ϕ **do**
  case $\langle\pi\rangle\varphi$
    foreach $\Delta \subseteq \mathcal{S}_\mathcal{N}$ **do**
      foreach $t \in \Delta$ **do**
        if *CR*($s_\Gamma, t_\Delta, \pi, 1$) *and MC*($\mathcal{N}, t_\Delta, \varphi$) **then return** true ;
    return false;
  case $K\varphi$
    foreach $t \in \Gamma$ **do**
      if *MC*($\mathcal{N}, t_\Gamma, \varphi$) *is false* **then return** false ;
    return true;

explicit transition system can be generated by taking all the possible valuations of the state variables as the state space (an exponential blow-up), and computing the transitions among the valuations according to the preconditions and the postconditions of the actions. In terms of the size of explicit transition systems, our above result is consistent with the EXPSPACE complexity result in the AI literature for conformant planning with Boolean and modal goals [Kleinberg and Tardos 2005; Bonet 2010].

However, not all the transition systems can be generated in this way since the preconditions and postconditions are (usually) purely propositional and thus two states that share the same valuation must have the same executable actions. In an arbitrary transition system, multiple states with the same valuation may have different available actions due to some underlying protocol or other (external) factors not modelled by basic propositions.

As a simple corollary based on the complexity of the standard conformant planning:

COROLLARY 4.15. *Generalized conformant planning is* PSPACE*-complete.*

5. CONCLUSIONS AND FUTURE WORK

In this work we first introduce the logical language EAL interpreted on uncertainty maps and give a complete axiomatization. EAL is then extended to EPDL with programs to specify conformant and conditional plans. We also generalize conformant planning with arbitrary EPDL formulas as goals and test-free programs as plan constraints. We show that the generalized conformant planning problem can be reduced to certain model checking problem of EPDL. Finally we show that model checking EPDL over uncertainty maps is PSPACE-complete, and thus the generalized conformant planning problem is also PSPACE-complete.

Note that our EPDL is a powerful language which can already express conditional plans, e.g. $(?p; a{+}?\neg p; b); c$. This suggests that we can use the very EPDL language to *verify* plans in contingent planning w.r.t. a variant of the semantics which can handle feedbacks during the execution. In fact, observational power about the availability of the actions has been already incorporated in [Wang and Li 2012], which can be extended to general feedbacks discussed in the literature of contingent planning (cf. e.g., [Bonet and Geffner 2012]). On the other hand, to check the *existence* of a conditional plan, we are not sure whether EPDL is expressive enough, as subtleties may arise as in the case of conformant planning. We leave the contingent planning to future work.

Another natural extension is to go probabilistic, and reduce the probabilistic planning over MDP to some model checking problem of the probabilistic version of our EPDL. Our ultimate goal is to cast all the standard AI planning problems into one unified logical framework in order to facilitate careful comparison and categorization. We will then see clearly how the form of the goal formula, the constructor of the plan, and the observational ability matter in the theoretical and practical complexity of planning, in line with the research pioneered in [Bäckström and Jonsson 2011].

Finally there is an obvious link with the logic of "knowing how" proposed by Wang [2015]: "knowing how" may mean that you have a good plan to achieve your goal. Different planning problems may induce different meanings of "knowing how" which in turn lead to different logics. We leave the explicit connections to future work.

ACKNOWLEDGMENTS

Yanjun Li thanks the support from China Scholarship Council. Quan Yu is supported by NSF Grant No.61463044 and Grant No.[2014]7421 from the Joint Fund of the NSF of Guizhou province of China. Yanjing Wang acknowledges the support from ROCS of SRF by Education Ministry of China and the NSSF key project 12&ZD119.

REFERENCES

Mikkel Birkegaard Andersen, Thomas Bolander, and Martin Holm Jensen. 2012. Conditional epistemic planning. In *Logics in Artificial Intelligence*. Springer, 94–106. DOI:http://dx.doi.org/10.1007/978-3-642-33353-8_8

Mikkel Birkegaard Andersen, Thomas Bolander, and Martin Holm Jensen. 2015. Don't plan for the unexpected: Planning based on plausibility models. (2015). To appear in *Logique et Analyse*.

Guillaume Aucher. 2012. DEL-sequents for regression and epistemic planning. *Journal of Applied Non-Classical Logics* 22, 4 (2012), 337–367. DOI:http://dx.doi.org/10.1080/11663081.2012.736703

Guillaume Aucher and Thomas Bolander. 2013. Undecidability in Epistemic Planning. In *IJCAI*. 27–33. https://hal.inria.fr/hal-00824653

Christer Bäckström and Peter Jonsson. 2011. All PSPACE-Complete Planning Problems Are Equal but Some Are More Equal than Others. In *SOCS 2011*. http://www.aaai.org/ocs/index.php/SOCS/SOCS11/paper/view/4009

Alexandru. Baltag and Larry Moss. 2004. Logics for epistemic programs. *Synthese* 139 (March 2004), 165–224. DOI:http://dx.doi.org/10.1023/B:SYNT.0000024912.56773.5e

Piergiorgio Bertoli, Alessandro Cimatti, Marco Roveri, and Paolo Traverso. 2006. Strong planning under partial observability. *Artificial Intelligence* 170, 4-5 (2006), 337–384. DOI:http://dx.doi.org/10.1016/j.artint.2006.01.004

Thomas Bolander and M. Birkegaard Andersen. 2011. Epistemic planning for single and multi-agent systems. *Journal of Applied Non-Classical Logics* 21, 1 (2011), 9–34. DOI:http://dx.doi.org/10.3166/jancl.21.9-34

Thomas Bolander, Martin Holm Jensen, and François Schwarzentruber. 2015. Complexity Results in Epistemic Planning. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, Qiang Yang and Michael Wooldridge (Eds.). AAAI Press, 2791–2797. http://ijcai.org/papers15/Abstracts/IJCAI15-395.html

Blai Bonet. 2010. Conformant plans and beyond: Principles and complexity. *Artificial Intelligence.* 174, 3-4 (2010), 245–269. DOI:http://dx.doi.org/10.1016/j.artint.2009.11.001

Blai Bonet and Hector Geffner. 2000. Planning with Incomplete Information as Heuristic Search in Belief Space. In *ICAPS 2000*. 52–61. DOI:http://dx.doi.org/10.1.1.38.8535

Blai Bonet and Hector Geffner. 2012. Width and Complexity of Belief Tracking in Non-Deterministic Conformant and Contingent Planning. In *AAAI 2012*. http://www.aaai.org/ocs/index.php/AAAI/AAAI12/paper/view/5022

Ronen I. Brafman and Jörg Hoffmann. 2004. Conformant Planning via Heuristic Forward Search: A New Approach. In *ICAPS 2004*. 355–364. DOI:http://dx.doi.org/10.1016/j.artint.2006.01.003

Daniel Bryce, Subbarao Kambhampati, and David E. Smith. 2006. Planning Graph Heuristics for Belief Space Search. *Journal of Artificial Intelligence Research* 26 (2006), 35–99. DOI:http://dx.doi.org/10.1613/jair.1869

Alessandro Cimatti and Marco Roveri. 2000. Conformant Planning via Symbolic Model Checking. *Journal of Artificial Intelligence Research* 13 (2000), 305–338. DOI:http://dx.doi.org/10.1613/jair.774

Alessandro Cimatti and Marco Roveri. 2011. Conformant Planning via Symbolic Model Checking. *CoRR* abs/1106.0252 (2011). http://arxiv.org/abs/1106.0252

Alessandro Cimatti, Marco Roveri, and Piergiorgio Bertoli. 2004. Conformant planning via symbolic model checking and heuristic search. *Artificial Intelligence* 159, 1-2 (2004), 127–206. DOI:http://dx.doi.org/10.1016/j.artint.2004.05.003

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

R. Fagin, J. Y. Halpern, Y. Moses, and M. Y. Vardi. 1997. Knowledge-based programs. *Distributed Computing* 10, 4 (4 July 1997), 199–225. DOI:http://dx.doi.org/10.1007/s004460050038

Patrik Haslum and Peter Jonsson. 1999. Some Results on the Complexity of Planning with Incomplete Information. In *ECP 1999*. 308–318. DOI:http://dx.doi.org/10.1007/10720246_24

Martin Holm Jensen. 2014. *Epistemic and Doxastic Planning*. Ph.D. Dissertation. Technical University of Denmark.

Jon Kleinberg and Éva Tardos. 2005. *Algorithm Design*. Addison-Wesley.

Jérôme Lang and Bruno Zanuttini. 2012. Knowledge-Based Programs as Plans - The Complexity of Plan Verification. In *ECAI 2012*. 504–509. DOI:http://dx.doi.org/10.3233/978-1-61499-098-7-504

Martin Lange. 2006. Model checking propositional dynamic logic with all extras. *Journal of Applied Logic* 4, 1 (2006), 39–49. DOI:http://dx.doi.org/10.1016/j.jal.2005.08.002

Benedikt Löwe, Eric Pacuit, and Andreas Witzel. 2011. DEL planning and some tractable cases. In *LORI 2011*. Springer, 179–192. DOI:http://dx.doi.org/10.1007/978-3-642-24130-7_13

Christian Muise, Vaishak Belle, Paolo Felli, Sheila McIlraith, Tim Miller, Adrian R Pearce, and Liz Sonenberg. 2015. Planning Over Multi-Agent Epistemic States: A Classical Planning Approach. In *The 29th AAAI Conference on Artificial Intelligence*.

Héctor Palacios and Hector Geffner. 2006. Compiling Uncertainty Away: Solving Conformant Planning Problems using a Classical Planner (Sometimes). In *AAAI 2006*. 900–905. http://www.aaai.org/Library/AAAI/2006/aaai06-142.php

Héctor Palacios and Hector Geffner. 2014. Compiling Uncertainty Away in Conformant Planning Problems with Bounded Width. *CoRR* abs/1401.3468 (2014). http://arxiv.org/abs/1401.3468

Pere Pardo and Mehrnoosh Sadrzadeh. 2013. Strong Planning in the Logics of Communication and Change. In *Declarative Agent Languages and Technologies X*. Springer, 37–56. DOI:http://dx.doi.org/10.1007/978-3-642-37890-4_3

Rohit. Parikh and R. Ramanujam. 1985. Distributed Processes and the Logic of Knowledge. In *Proceedings of Conference on Logic of Programs*. Springer-Verlag, London, UK, 256–268. DOI:http://dx.doi.org/10.1007/3-540-15648-8_21

Philippe Schnoebelen. 2003. The Complexity of Temporal Logic Model Checking. In *AiML 2002*, Philippe Balbiani, Nobu-Yuki Suzuki, Frank Wolter, and Michael Zakharyaschev (Eds.). King's College Publication, Toulouse, France, 393–436. http://www.lsv.ens-cachan.fr/Publis/PAPERS/PDF/Sch-aiml02.pdf Invited paper.

David E. Smith and Daniel S. Weld. 1998. Conformant Graphplan. In *AAAI 1998*. 889–896. http://www.aaai.org/Library/AAAI/1998/aaai98-126.php

Larry J Stockmeyer and Albert R Meyer. 1973. Word problems requiring exponential time (Preliminary Report). In *STOC 1973*. ACM, 1–9. DOI:http://dx.doi.org/10.1145/800125.804029

Son Thanh To, Tran Cao Son, and Enrico Pontelli. 2010. A New Approach to Conformant Planning Using CNF*. In *ICAPS 2010*. 169–176. http://www.aaai.org/ocs/index.php/ICAPS/ICAPS10/paper/view/1461

J. van Benthem, J. van Eijck, and B. Kooi. 2006. Logics of Communication and Change. *Information and Computation* 204, 11 (2006), 1620–1662.

Hans van Ditmarsch, Wiebe van der Hoek, and Barteld Kooi. 2007. *Dynamic epistemic logic*. Springer. DOI:http://dx.doi.org/10.1007/978-1-4020-5839-4

Yanjing Wang. 2015. A Logic of Knowing How. In *Proceedings of LORI 2015*.

Yanjing Wang and Qinxiang Cao. 2013. On axiomatizations of public announcement logic. *Synthese* 190 (2013), 103–134. Issue 1S. DOI:http://dx.doi.org/10.1007/s11229-012-0233-5

Yanjing Wang and Yanjun Li. 2012. Not All Those Who Wander Are Lost: Dynamic Epistemic Reasoning in Navigation.. In *AiML 2012*. 559–580. http://www.aiml.net/volumes/volume9/Wang-Li.pdf

Quan Yu, Yanjun Li, and Yanjing Wang. 2015. A Dynamic Epistemic Framework for Conformant Planning. In *Proceedings of TARK'15*.

Quan Yu, Ximing Wen, and Yongmei Liu. 2013. Multi-Agent Epistemic Explanatory Diagnosis via Reasoning about Actions. In *IJCAI 2013*. 1183–1190. http://www.aaai.org/ocs/index.php/IJCAI/IJCAI13/paper/view/6631