A Dynamic Epistemic Framework for Conformant Planning

Quan Yu[1,2], Yanjun Li[3,4] and Yanjing Wang[*3]

[1]Dept. of Computer Science, Sun Yat-sen University, China

[2]Qiannan Normal College for Nationalities, China

[3]Department of Philosophy, Peking University, China

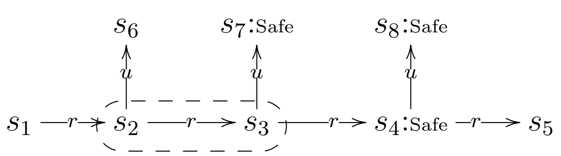
[4]Faculty of Philosophy, University of Groningen, The Netherlands

ABSTRACT

In this paper, we introduce a lightweight dynamic epistemic logical framework for automated planning under initial uncertainty. We reduce plan verification and conformant planning to model checking problems of this logic. We show that the model checking problem of the iteration-free fragment is PSPACE-complete. By using two non-standard (but equivalent) semantics, we give novel model checking algorithms to the full language and the iteration-free language.

1. INTRODUCTION

Conformant planning is the problem of finding a linear plan (a sequence of action) to achieve a goal in presence of uncertainty about the initial state (cf. [30]). For example, suppose that you are a rookie spy trapped in a foreign hotel with the following map at hand:[1]



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 circle 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 is no feedbacks/observations available during the execution of the plan.[2] 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 [10, 25], 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 [19]. In the literature, people proposed compact and implicit representations of the belief spaces, such as OBDD [14, 16, 15] and CNF [32], and different heuristics are used to guide the search for a plan [12, 7, 13, 32, 26].

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., [33]). The central philosophy of DEL takes the meaning of an action as the changes it brings to the knowledge of the agents. Intuitively, this is what we need to track the belief states during the execution of a plan[3]. Indeed, in recent years, there has been a growing interest in using DEL to handle multi-agent planning with knowledge goals (cf. e.g., [8, 24, 1, 3, 36, 27]), while the traditional AI planning focuses on the single-agent case. In particular, the event models of DEL (cf. [6]) are used to handle non-public actions that may cause different knowledge updates to different agents. In these DEL-based planning frameworks,

[*]Corresponding author

[1]It is a variant of the running example in [35].

[2]In many other cases, feedbacks may be just too 'expensive' to obtain during a plan aiming for quick actions [9].

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

states are epistemic models, actions are event models and the state transitions are implicitly encoded by the update product which computes an 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 [8, 4], that multi-agent epistemic planning is undecidable in general. Many interesting decidable fragments are found in the literature [8, 24, 36, 2], which suggest that the single-agent cases and restrictions on the form of event models are the key to decidability. However, if we focus on the single-agent case, a natural question arises: how do we compare such DEL approaches with the traditional approaches to single-agent 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. Moreover, the standard models of DEL are purely epistemic which do not encode the temporal information of available actions directly, which may limit the use 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 idea, but not the standard formalism of DEL. Our formal framework is based on the logic proposed by Wang and Li in [35], where the model is simply a transition system with initial uncertainty as in the motivating example, and the semantics of an action is a dynamic one, in the sense that it updates the uncertainty of the agent. Our contributions are summarized as follows:

- A lightweight dynamic epistemic framework with a simple language and a complete axiomatization.
- Non-trivial reduction of conformant planning to a model checking problem using our language extended with programs.
- Two novel model checking algorithms based on two alternative semantics for the proposed logic to handle the context-dependency in the original semantics.
- The complexity of model checking the iteration-free fragment of our language is PSPACE-complete. The model checking problem of the full language is in EXPTIME. Thes model checking problem of the conformant planning is in PSPACE.

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 complexity 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 4.3.

Our approach has the following advantages compared to the existing planning approaches:

- The planning goals can be specified as arbitrary formulas in an epistemic language. Extra plan constraints (e.g., what simple plans and actions to use) can be expressed explicitly by programs in the language. Therefore it may cover a richer class of (conformant) planning problems compared to the traditional AI approach where a goal is Boolean.[4]
- The plans can be specified as regular expressions with tests in terms of arbitrary EPDL formulas, which generalizes the knowledge-based programs in [18, 22].
- By reducing conformant planning to a model checking problem in an explicit logical language, we also see the subtleties in the logical specification of the planning problem. In principle, there are various model checking techniques to be applied to conformant planning based on this reduction.
- Our logical language and models are very simple compared to the standard action-model based DEL approach, yet we can encode the externally given executability of the actions in the model, inspired by epistemic temporal logic (ETL) [17, 28].
- 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.

The rest of the paper is organized as follows: We introduce our basic logical framework and its axiomatization in Section 2, and extend it in Section 3 with programs to handle the conformant planning. The complexity analysis of the model checking problem is in Section 4 and we 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 use the following language proposed in [35].

DEFINITION 2.1 (EPISTEMIC ACTION LANGUAGE ($\mathtt{EAL}$)). *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:*

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

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 agent can move forward by action a, then after doing a, ϕ holds. Through out the paper, we fix some $\mathtt{P}$ and $\mathtt{A}$, and refer to $\mathtt{EAL}^{\mathtt{A}}_{\mathtt{P}}$ by $\mathtt{EAL}$.

[4] The goal in the standard conformant planning is simply a set of different valuations of basic propositional variables. Our approach can even handle epistemic goals in negative forms, e.g., we want to make sure the agent knows something but does not know too much in the end.

The size of EAL-formulas (notation $|\varphi|$) is defined inductively: $|\top| = |p| = 1$; $|\neg\phi| = 1 + |\phi|$; $|\phi \wedge \psi| = 1 + |\phi| + |\psi|$; $|K\phi| = |[a]\phi| = 1 + |\phi|$. The set of subformulas of $\phi \in$ EAL, denoted as $sub(\phi)$, is defined as usual.

Definition 2.2 (Uncertainty map). *Given* P *and* 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 labelled 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, the 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 1. *It is crucial to notice that the designated state in a pointed uncertainty map must be one of the states in the uncertainty set.*

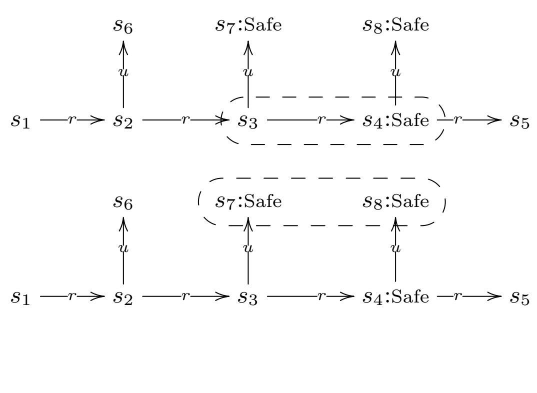
Definition 2.3 (Semantics). *Given any uncertainty map $\mathcal{M} = \langle \mathcal{S}, \{\mathcal{R}_a \mid a \in \mathtt{A}\}, \mathcal{V}, \mathcal{U}\rangle$ and any point $s \in \mathcal{U}$, the semantics is defined as follows:*

$\mathcal{M}, s \vDash \top$		*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* $\mathcal{M}, s \vDash \phi$
$\mathcal{M}, s \vDash [a]\phi$	$\iff$	$\forall t \in S : s \xrightarrow{a} t$ *implies* $\mathcal{M}\|^a, t \vDash \phi$
$\mathcal{M}, s \vDash K\phi$	$\iff$	$\forall u \in \mathcal{U} : \mathcal{M}, u \vDash \phi$

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 obtain a new uncertainty set $\mathcal{U}|^a$. Note that here we differ from [35] 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 agents 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:

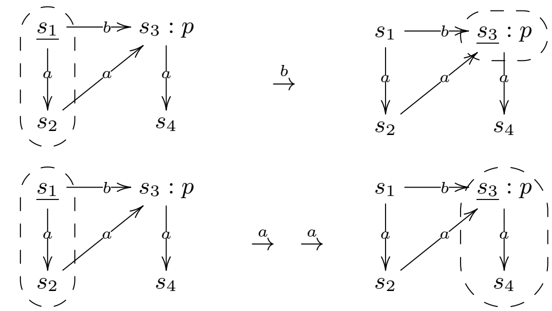


Thus we have:

- $\mathcal{M}, s_3 \vDash [r](\mathit{Safe} \wedge \neg K\mathit{Safe})$

- $\mathcal{M}, s_3 \vDash K[r][u](\mathit{Safe} \wedge K\mathit{Safe})$

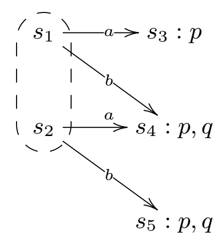
The usual global model checking algorithm for modal logics labels the states by the subformulas which are true on the states. However, this cannot work here since the truth value of epistemic formulas on the states outside $\mathcal{U}$ is simply undefined. Moreover, the exact truth value of an epistemic 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', which requires new techniques in model checking. We will make this explicit in Section 4.3 when we discuss the model checking algorithm.

Here is also an example about planning with both positive and negative epistemic goals (the agent should know something but not too much).

Example 1. *Given uncertainty map $\mathcal{M}$ depicted as follows, let the goal is Kp then both a and b are solutions. If the goal is $Kp \wedge \neg Kq$, only a is a solution.*



2.2 Axiomatization

Following the axioms proposed in [35], 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. [34]).

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 [35] is abandoned. Due to the same reason, the axiom of no miracles is also simplified.

Based on a refinement of the completeness proof in [35], we show the completeness of $\mathbb{SELA}$.

THEOREM 2.1. *$\mathbb{SELA}$ is sound and strongly compete 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 equivalent 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 $\mathbb{SELA}$ 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;
- $sR^c_a t \iff$ for any $\phi \in t$ then $\langle a\rangle\phi \in s \iff$ for any $[a]\phi \in s$ then $\phi \in t$;
- $\mathcal{V}^c(p) = \{s \mid p \in s\}$.

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

Secondly, we prove the following claims.

CLAIM 2.1. *If $\neg K\phi \in s$, then there exists $u \in \mathcal{U}^c_s$ 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 $K\psi_1, \dots, K\psi_n$ 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 $1 \le i \le n$. 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 $u \in \mathcal{U}^c_s$.

CLAIM 2.2. *If $\neg[a]\phi \in s$, then there exists $t \in \mathcal{U}^c_s$ 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 $\psi_1, \dots, \psi_n$ 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 $\mathcal{U}^c_s|^a = \mathcal{U}^c_t$.*

$\subseteq$: Assuming $v \in \mathcal{U}^c_s|^a$, we need to show $v \in \mathcal{U}^c_t$, namely we need to show that $K\phi \in v \iff K\phi \in t$. Since $v \in \mathcal{U}^c_s|^a$, we have that there is $u \in \mathcal{U}^c_s$ such that $uR^c_a v$. 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 $u \in \mathcal{U}^c_s$ and axiom `T`, we have $[a]K\phi \in u$. It follows by $uR^c_a v$ 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 $v \in \mathcal{U}^c_t$, we need to show $v \in \mathcal{U}^c_s|^a$, namely there is $u \in \mathcal{U}^c_s$ such that $uR^c_a v$. 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 $1 \le i \le n$, 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 $sR^c_a t$ that $K(\neg\psi_1 \vee \cdots \vee \neg\psi_k) \in t$. Since $v \in \mathcal{U}^c_t$, 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 $1 \le i \le k$. Thus u^- is consistent. By Lindenbaum-like Lemma, there exists a MCS u extending u^-. It follows by $u^- \subseteq u$ that $u \in \mathcal{U}^c_s$ and $uR^c_a v$. We conclude that $v \in \mathcal{U}^c_s|^a$.

Finally, we will show that $\mathcal{M}^c_s, s \vDash \phi$ iff $\phi \in s$. we prove it by induction on ϕ. We only restrict our attention to the cases of $[a]\phi$ and $K\phi$; the other cases are straightforward. $\phi = [a]\psi$, if $\mathcal{M}^c_s, s \vDash [a]\psi$ but $[a]\psi \notin s$. Thus we have $\neg[a]\psi \in s$. It follows by Claim 2.2 that there is t such that $sR^c_a t$ and $\neg\psi \in t$. By IH, we have $\mathcal{M}^c_t, t \vDash \neg\psi$. By Claim 2.3, we have $\mathcal{M}^c_t = \mathcal{M}^c_s|^a$. Thus we have $\mathcal{M}^c_s|^a, t \vDash \neg\psi$. This is contrary with $\mathcal{M}^c_s, s \vDash [a]\psi$ and $sR^c_a t$. On the other hand, if $[a]\psi \in s$ but $\mathcal{M}^c_s, s \nvDash [a]\psi$. Thus there exists t such that $sR^c_a t$ and $\mathcal{M}^c_s|^a, t \vDash \neg\psi$, namely $\mathcal{M}^c_t, t \vDash \neg\psi$. By IH, we have $\neg\psi \in t$. It follows by $sR^c_a t$ that $\neg[a]\psi \in s$, contradiction. $\phi = K\psi$, if $\mathcal{M}^c_s, s \vDash K\psi$ but $K\psi \notin s$. Thus we have $\neg K\psi \in s$. It follows by Claim 2.1 that there is $u \in \mathcal{U}^c_s$ such that $\neg\psi \in u$. By IH, we have $\mathcal{M}^c_u, u \vDash \neg\psi$. It follows by $u \in \mathcal{U}^c_s$ that $\mathcal{U}^c_s = \mathcal{U}^c_u$. Thus we have $\mathcal{M}^c_s, u \vDash \neg\psi$.

This is contrary with $\mathcal{M}^c_s, s \vDash K\psi$ and $u \in \mathcal{U}^c_s$. On the other hand, if $K\psi \in s$ but $\mathcal{M}^c_s, s \nvDash K\psi$. Thus there exists $u \in \mathcal{U}^c_s$ such that $\mathcal{M}^c_s, u \vDash \neg\psi$. It follows by $\mathcal{U}^c_s = \mathcal{U}^c_u$ that $\mathcal{M}^c_u, u \vDash \neg\psi$. By IH, we have $\neg\psi \in u$. By axiom $\mathtt{T}$, it follows that $\neg K\psi \in u$. This is contrary with $K\psi \in s$ and $u \in \mathcal{U}^c_s$. $\square$

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 $[\pi]\phi \wedge \langle\pi\rangle\phi$. Given a $\mathtt{B} \subseteq \mathtt{A}$, we write $\mathtt{B}^*$ for $(\Sigma_{a\in\mathtt{B}} a)^*$. The size of EPDL formulas/programs is given by: $|[\pi]\phi| = |\pi| + |\phi|$, $|a| = 1$, $|\pi_1;\pi_2| = 1+|\pi_1|+|\pi_2|$, $|?\phi| = |\pi^*| = 1+|\phi|$, and $|\pi_1+\pi_2| = 1 + |\pi_1| + |\pi_2|$.

Given any uncertainty map $\mathcal{M} = \langle \mathcal{S}, \{\mathcal{R}_a \mid a \in \mathtt{A}\}, \mathcal{V}, \mathcal{U}\rangle$, any point $s \in \mathcal{U}$, the semantics is given by a mutual induction on ϕ and π:

$$\begin{array}{rcl}
\mathcal{M}, s \vDash \top & & \text{always} \\
\mathcal{M}, s \vDash p & \Leftrightarrow & s \in \mathcal{V}(p) \\
\mathcal{M}, s \vDash \neg\phi & \Leftrightarrow & \mathcal{M}, s \nvDash \phi \\
\mathcal{M}, s \vDash \phi \wedge \psi & \Leftrightarrow & \mathcal{M}, s \vDash \phi \text{ and } \mathcal{M}, s \vDash \psi \\
\mathcal{M}, s \vDash [\pi]\phi & \Leftrightarrow & \text{for all } \mathcal{M}', s' : (\mathcal{M}, s)[\![\pi]\!](\mathcal{M}', s') \\
& & \text{implies } \mathcal{M}', s' \vDash \phi \\
\mathcal{M}, s \vDash K\phi & \Leftrightarrow & \text{for all } t \in \mathcal{U} : \mathcal{M}, t \vDash \phi \\
(\mathcal{M}, s)[\![a]\!](\mathcal{M}', s') & \Leftrightarrow & \mathcal{M}' = \mathcal{M}|^a \text{ and } s \xrightarrow{a} s' \\
(\mathcal{M}, s)[\![?\psi]\!](\mathcal{M}', s') & \Leftrightarrow & (\mathcal{M}', s') = (\mathcal{M}, s) \text{ and } \mathcal{M}, s \vDash \psi \\
(\mathcal{M}, s)[\![\pi_1;\pi_2]\!](\mathcal{M}', s') & \Leftrightarrow & (\mathcal{M}, s)[\![\pi_1]\!] \circ [\![\pi_2]\!](\mathcal{M}', s') \\
(\mathcal{M}, s)[\![\pi_1+\pi_2]\!](\mathcal{M}', s') & \Leftrightarrow & (\mathcal{M}, s)[\![\pi_1]\!] \cup [\![\pi_2]\!](\mathcal{M}', s') \\
(\mathcal{M}, s)[\![\pi^*]\!](\mathcal{M}', s') & \Leftrightarrow & (\mathcal{M}, s)[\![\pi]\!]^\star(\mathcal{M}', s')
\end{array}$$

where $\circ, \cup, {}^\star$ at right-hand side express 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 mixed sequences, which are sequences of actions in $\mathtt{A}$ and $\phi \in$ EPDL:

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

Here are some valid formulas which are useful in our latter discussion:

$$[\pi;\pi']\phi \leftrightarrow [\pi][\pi']\phi$$
$$[\pi+\pi']\phi \leftrightarrow [\pi]\phi \wedge [\pi']\phi$$
$$[?\psi]\phi \leftrightarrow (\psi \rightarrow \phi)$$

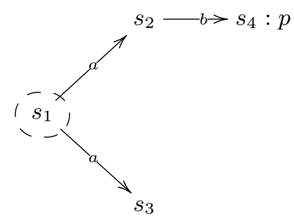
We leave the complete axiomatization of EPDL on uncertainty maps to future work.

3.2 Conformant planning via model checking EPDL

Definition 3.2 (Conformant planning). *Given an uncertainty map $\mathcal{M}$, a goal formula $\phi \in$ EPDL, and a 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 $\mathcal{M}, u \vDash (\!|a_1|\!)(\!|a_2|\!)\cdots(\!|a_n|\!)\phi$. The existence problem of conformant planning is to test whether such a sequence exists.*

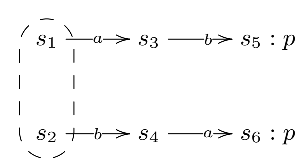
Recall that $(\!|\pi|\!)\phi$ is the shorthand of $[\pi]\phi \wedge \langle\pi\rangle\phi$. Intuitively, we want a plan which is both executable and safe w.r.t. non-deterministic actions and initial uncertainty of the agent. It is crucial to observe the difference between $(\!|a_1|\!)(\!|a_2|\!)\cdots(\!|a_n|\!)\phi$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

Example 2. *Given uncertainty map $\mathcal{M}$ depicted as follows, we have $\mathcal{M}, s_1 \vDash (\!|a;b|\!)p$ but $\mathcal{M}, s_1 \nvDash (\!|a|\!)(\!|b|\!)p$.*



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: $\mathcal{M}, u \vDash K(\!|a_1|\!)(\!|a_2|\!)\cdots(\!|a_n|\!)\phi$. 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 $\mathcal{M}, u \vDash K(\!|\mathtt{B}^*|\!)\phi$ holds. However $K(\!|\mathtt{B}^*|\!)\phi$ may hold on a model where the sequences to make sure ϕ on the states in $\mathcal{U}_M$ are different, as the following example shows:

Example 3. *Given uncertainty map $\mathcal{M}$ depicted as follows, let the goal formula be p and $\mathtt{B} = \{a, b\}$. We have $\mathcal{M}, s_1 \vDash K(\!|\mathtt{B}^*|\!)p$, but there is no solution to this conformant planning problem.*



The right formula to check for the existence of a conformant plan w.r.t. $\mathtt{B} \subseteq \mathtt{A}$ and $\phi \in$ 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, the confrmant 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 need an easy observation ($\phi(\psi/\phi)$ is obtained by replacing any occurrence of ϕ by ψ, similar for $[\![\pi(\psi/\chi)]\!]$):

PROPOSITION 3.1. *If* $\vDash \psi \leftrightarrow \chi$*, then:*

(1) $\vDash \phi \leftrightarrow \phi(\psi/\chi)$*;*

(2) $[\![\pi]\!] = [\![\pi(\psi/\chi)]\!]$*.*

LEMMA 3.1. *For any* $a_1 a_2 \cdots a_n \in \mathcal{L}(\mathtt{A}^*)$*:*

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

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 $n \geq 1$. For the case of $n = 1$, we need to show that

$$\vDash K\llparenthesis a\rrparenthesis\phi \leftrightarrow \langle ?K\langle a\rangle\top; a\rangle K\phi \qquad (\circ)$$

that is to show $\mathcal{M}, s \vDash K\langle a\rangle\phi \wedge K[a]\phi$ iff $\mathcal{M}, s \vDash K\langle a\rangle\top \wedge \langle a\rangle K\phi$ for any $\mathcal{M}, s$. For the left-to-right direction, due to the fact that $\vDash K\langle a\rangle\phi \to K\langle a\rangle\top$, we only need to show that $\mathcal{M}, s \vDash \langle a\rangle K\phi$. Suppose not, it follows that $\mathcal{M}|^a, t \vDash \neg K\phi$ for any t such that $s \xrightarrow{a} t$. Due to the fact that $\mathcal{M}, s \vDash K\langle a\rangle\top$ there must exist a t such that $s \xrightarrow{a} t$ and $\mathcal{M}|^a, t \vDash \neg K\phi$. By the semantics, it follows that $\mathcal{M}|^a, t' \vDash \neg\phi$ for some $t' \in \mathcal{U}|^a$. By the definition of $\mathcal{U}|^a$, we have that there is $s' \in \mathcal{U}$ such that $s' \xrightarrow{a} t'$. Thus we have $\mathcal{M}, s' \vDash \langle a\rangle\neg\phi$. It follows that $\mathcal{M}, s \vDash \neg K[a]\phi$. This contradicts the assumption of $\mathcal{M}, s \vDash K[a]\phi$. To verify the right-to-left direction, firstly, we will show that $\mathcal{M}, s \vDash K[a]\phi$. Suppose not, then $\mathcal{M}, s' \vDash \neg[a]\phi$ for some $s' \in \mathcal{U}_{\mathcal{M}}$. It follows that $\mathcal{M}|^a, t' \vDash \neg\phi$ for some t' such that $s' \xrightarrow{a} t'$. Since $\mathcal{M}, s \vDash \langle a\rangle K\phi$, we have $\mathcal{M}|^a, t \vDash K\phi$ for some t such that $s \xrightarrow{a} t$. Because of $\mathcal{M}|^a, t' \vDash \neg\phi$, it follows by $t' \in \mathcal{U}|^a$ that $\mathcal{M}|^a, t \vDash \neg K\phi$. This contradicts that $\mathcal{M}|^a, t \vDash K\phi$. Thus we have $\mathcal{M}, s \vDash K[a]\phi$. With the assumption of $\mathcal{M}, s \vDash K\langle a\rangle\top$, it follows that $\mathcal{M}, s \vDash K\langle a\rangle\phi$.

Now, as the induction hypothesis, we assume that:

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

We need to show:

$$\begin{aligned}&\vDash K\llparenthesis a_1\rrparenthesis\llparenthesis a_2\rrparenthesis \cdots \llparenthesis a_{k+1}\rrparenthesis\phi \leftrightarrow \\ &\quad \langle ?K\langle a_1\rangle\top; a_1; \ldots ; ?K\langle a_{k+1}\rangle\top; a_{k+1}\rangle K\phi.\end{aligned}$$

By IH,

$$\begin{aligned}&\vDash K\llparenthesis a_1\rrparenthesis\llparenthesis a_2\rrparenthesis \cdots \llparenthesis a_{k+1}\rrparenthesis\phi \leftrightarrow \\ &\quad \langle ?K\langle a_1\rangle\top; a_1; \ldots ; ?K\langle a_k\rangle\top; a_k\rangle K\llparenthesis a_{k+1}\rrparenthesis\phi. \qquad (1)\end{aligned}$$

Due to $(\circ)$ and Proposition 3.1, we have

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

The conclusion is immediate by combining (1) and (2). □

THEOREM 3.1. *Given a pointed uncertainty map* $\mathcal{M}, s$*, an* EPDL *formula* ϕ *and a set* $\mathtt{B} \subseteq \mathtt{A}$*, the following two are equivalent:*

(1) There is a $\sigma = a_1 \ldots a_n \in \mathcal{L}(\mathtt{B}^*)$ *such that* $\mathcal{M}, s \vDash K\llparenthesis a_1\rrparenthesis\llparenthesis a_2\rrparenthesis \cdots \llparenthesis a_n\rrparenthesis\phi$*;*

(2) $\mathcal{M}, s \vDash \langle(\Sigma_{a\in \mathtt{B}}(?K\langle a\rangle\top; a))^*\rangle K\phi$*.*

PROOF. (1) implies (2): Assuming (1) then by Lemma 3.1, it follows that

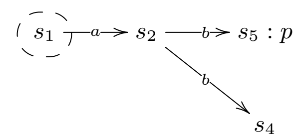
$$\mathcal{M}, s \vDash \langle ?K\langle a_1\rangle\top; a_1; \ldots ; ?K\langle a_n\rangle\top; a_n\rangle K\phi.$$

Since $\sigma \in \mathcal{L}((\Sigma_{a\in\mathtt{B}}(?K\langle a\rangle\top; a))^n)$, by the semantics, we have $\mathcal{M}, s \vDash \langle(\Sigma_{a\in\mathtt{B}}(?K\langle a\rangle\top; a))^n\rangle K\phi$. Thus we have $\mathcal{M}, s \vDash \langle(\Sigma_{a\in\mathtt{B}}(?K\langle a\rangle\top; a))^*\rangle K\phi$.

(2) implies (1): It follows by the semantics that $\mathcal{M}, s \vDash \langle(\Sigma_{a\in\mathtt{B}}(?K\langle a\rangle\top; a))^n\rangle K\phi$ for some $n \geq 0$. Thus we have $\mathcal{M}, s \vDash \langle(?K\langle a_1\rangle\top; a_1); \ldots ; (?K\langle a_n\rangle\top; a_n\rangle K\phi$ for some $a_i\mathtt{B}$ where $1 \leq i \leq n$. It follows by Lemma 3.1 that $\mathcal{M}, s \vDash K\llparenthesis a_1\rrparenthesis\llparenthesis a_2\rrparenthesis \cdots \llparenthesis a_n\rrparenthesis\phi$. □

We end this section with a remark 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 4. *Given 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* $\mathcal{M}, s_1 \nvDash \langle(\Sigma_{a\in B}(?K\langle a\rangle\top; a))^*\rangle Kp$ *with* $\mathtt{B} = a, b$*, but we could have* $\mathcal{M}, s_1 \vDash \langle(\Sigma_{a\in B}(?K\langle a\rangle\top; a))^*\rangle p$*.*



4. MODEL CHECKING EPDL: COMPLEXITY AND ALGORITHMS

In this section, we first focus on the model checking problem of the following star-free fragment of EPDL (call it $\mathtt{EPDL}^-$):

$$\begin{aligned}\phi &::= \top \mid p \mid \neg\phi \mid (\phi \wedge \phi) \mid [\pi]\phi \mid K\phi \\ \pi &::= a \mid ?\phi \mid (\pi; \pi) \mid (\pi + \pi)\end{aligned}$$

We will show that model checking $\mathtt{EPDL}^-$ is PSPACE-complete. In particular, the upper bound is shown by making use of an alternative context-dependent semantics. Then we give an EXPTIME algorithm for the model checking problem of the full EPDL inspired by another alternative semantics based on 2-dimensional models. Finally we give a PSPACE algorithm for the conformant planning problem in EPDL. Note that throughout this section, we focus on uncertainty maps with finitely many states and assume $\mathcal{R}_a = \emptyset$ for *co-finitely* many $a \in \mathtt{A}$.

4.1 Complexity of model checking EPDL$^-$

4.1.1 Lower Bound

To show the PSPACE lower bound, we provide a polynomial reduction of QBF (*quantified Boolean formula*) truth testing to the model checking problem of $\mathtt{EPDL}^-$. 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 [31]. Our method is inspired by [29] which discusses the complexity of model checking temporal logics with past operators. Surprisingly, we can use the uncertainty sets to encode 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 $Q_1 x_1 Q_2 x_2 \ldots Q_n x_n \phi(x_1, \ldots, x_n)$ where:

- For $1 \leq n \leq n, Q_i$ is $\exists$ if i is odd, and Q_i is $\forall$ if i is even.

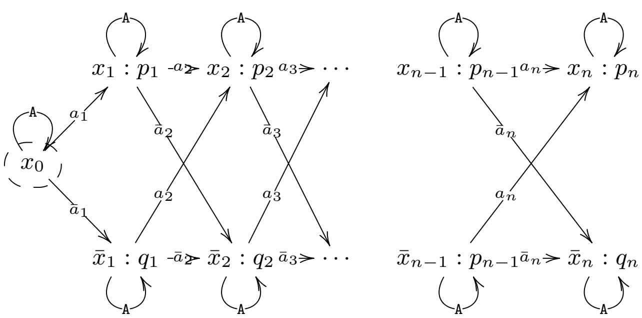
- ϕ is a propositional formula in CNF based on variables $x_1, \ldots, x_n$,

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* $\mathtt{A} = \{a_i, \bar{a}_i \mid i \geq 1\}$ *and* $\mathtt{P} = \{p_k, q_k \mid k \geq 1\}$, *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:*

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

$\mathcal{M}_n$ can be depicted as the following graph:



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

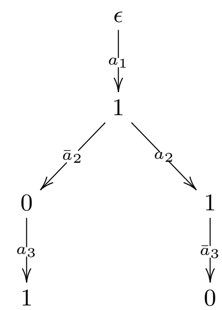
$$QT_1 \cdots QT_n \psi(\hat{K}p_1, \cdots, \hat{K}p_n, \hat{K}q_1, \cdots, \hat{K}q_n)$$

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 $\phi(x_1, \ldots, x_n)$ 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 with even level has only one successor; 3) each internal node with odd level has two successors and one successor 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.*

A V-tree τ can be depicted as the following graph:



It is not hard to see the following:

Proposition 4.1. *For each $1 \leq i \leq n$, 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 4.2. *Given $\mathcal{M}_n$, let $\sigma = A_1 \ldots A_i$ ($1 \leq i \leq n$) be a sequence of actions such that $A_k = a_k$ or $A_k = \bar{a}_k$ for each $1 \leq k \leq i$, 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 $1 \leq k \leq i$.*

Proof. We prove it by induction on i. It is obvious if $i = 1$. Next we need to show that $\mathcal{U}|^{\sigma A_{i+1}} = \{x_0, X_1, \ldots, X_{i+1}\}$. By IH, it follows that $\mathcal{U}' = \mathcal{U}|^\sigma = \{x_0, X_1, \ldots, X_i\}$ where $X_k = x_k$ if $A_k = a_k$ else $X_k = \bar{x}_k$ for each $1 \leq k \leq i$. By the definition of semantics, we have $\mathcal{U}|^{\sigma A_{i+1}} = \mathcal{U}'|^{A_{i+1}}$. It follows by the definition of $\mathcal{M}_n$ that $X_k \xrightarrow{A_{i+1}} s$ iff $s = X_k$ for each $1 \leq k < i$ and $X_i \xrightarrow{A_{i+1}} s$ iff $s = X_i$ or $s = x_{i+1}$ if $A_{i+1} = a_{i+1}$ or $s = \bar{x}_{i+1}$ if $A_{i+1} = \bar{a}_{i+1}$. Thus we have $\mathcal{U}'|^{A_{i+1}} = \{x_0, X_1, \ldots, X_{i+1}\}$ where $X_k = x_k$ if $A_k = a_k$ else $X_k = \bar{x}_k$ for each $1 \leq k \leq i+1$. $\square$

Given $\sigma = A_1 \ldots A_n$ where A_i is a_i or $\bar{a}_i$ for each $1 \leq i \leq n$, let $g(\sigma) = x_n$ if $A_n = a_n$ and $g(\sigma) = \bar{x}_n$ if $A_n = \bar{a}_n$. By Proposition 4.2, 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 4.3. *For each $1 \leq i \leq n$, 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. If it is the case $i = 1$, we begin the proof with the left-to-right direction. Thus we have that there exists a state t such that $x_0 \xrightarrow{A_1} t$ and $\mathcal{M}_k|^{A_1}, t \vDash p_1 \vee q_1$ and $\mathcal{M}_k|^{A_1}, t \vDash QT_2 \ldots QT_n \psi$ where A_1 is a_1 or $\bar{a}_1$. If $\mathcal{M}_k|^{A_1}, t \vDash p_1$, it follows by the definition of $\mathcal{M}_k$ that A_1 is a_1 and t is x_1. We can construct V-tree τ as $\epsilon \xrightarrow{a_1} 1$, and we have $\mathcal{M}_k|^{a_1}, x_1 \vDash QT_2 \ldots QT_n \psi$. If $\mathcal{M}_k|^{A_1}, t \vDash q_1$, then we have $A_1 = \bar{a}_1$ and $t = \bar{x}_1$. We can construct τ as $\epsilon \xrightarrow{\bar{a}_1} 0$, and we also have $\mathcal{M}_k|^{\bar{a}_1}, \bar{x}_1 \vDash QT_2 \ldots QT_n \psi$. For the right-to-left direction, it is obvious. Next we need to show that $\mathcal{M}_k, x_0 \vDash QT_1 \ldots QT_{i+1} QT_{i+2} \ldots QT_n \psi$ iff 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 each $\sigma' \in path(\tau')$. By IH, we only need to show that there exists a V-tree τ with depth i such that $\mathcal{M}_k|^{\sigma}, g(\sigma) \vDash QT_{i+1} \dots QT_n\psi$ for each $\sigma \in path(\tau)$ iff there exists a V-tree τ' with depth $i+1$ such that $\mathcal{M}_k|^{\sigma'}, g(\sigma') \vDash QT_{i+2} \dots QT_n\psi$ for each $\sigma' \in path(\tau')$. There are two situations: $i+1$ is even or odd.

If $i+1$ is even, to verify the right-to-left direction, we get a V-tree τ' by extending τ with two successors 0 and 1 to each leaf of τ, and labelling edge to new leaf 0 as $\bar{a}_{i+1}$ and labelling edge to 1 as a_{i+1}. Next we need to show that for each $\sigma' \in path(\tau')$, $\mathcal{M}_k|^{\sigma'}, g(\sigma') \vDash QT_{i+2} \dots QT_n\psi$. By the definition of $g(\sigma)$ and $g(\sigma')$, we have $g(\sigma) \overset{A_{i+1}}{\to} g(\sigma')$ where A_{i+1} is a_{i+1} or $\bar{a}_{i+1}$ such that $\sigma' = \sigma A_{i+1}$. Since $\mathcal{M}_k|^{\sigma}, g(\sigma) \vDash [(a_{i+1} + \bar{a}_{i+1})][?(p_{i+1} \vee q_{i+1})]QT_{i+2} \dots QT_n\psi$ and $g(\sigma) \overset{A_{i+1}}{\to} g(\sigma')$, it follows that $\mathcal{M}_k|^{\sigma'}, g(\sigma') \vDash [?(p_{i+1} \vee q_{i+1})]QT_{i+2} \dots QT_n\psi$. It is obvious that $\mathcal{M}_k|^{\sigma'}, g(\sigma') \vDash p_{i+1} \vee q_{i+1}$. Thus we have $\mathcal{M}_k|^{\sigma'}, g(\sigma') \vDash QT_{i+2} \dots QT_n\psi$. To verify the left-to-right direction, we get τ by cutting the $i+1$ level of τ'. Next we need to show that $\mathcal{M}_k|^{\sigma}, g(\sigma) \vDash [(a_{i+1} + \bar{a}_{i+1})][?(p_{i+1} \vee q_{i+1})]QT_{i+2} \dots QT_n\psi$ for each $\sigma \in path(\tau)$. By the semantics, for each t such that $g(\sigma) \overset{A_{i+1}}{\to} t$ and $\mathcal{M}_k|^{\sigma A_{i+1}}, t \vDash p_{i+1} \vee q_{i+1}$, we need to show $\mathcal{M}_k|^{\sigma A_{i+1}}, t \vDash QT_{i+2} \dots QT_n\psi$. By the definition of $\mathcal{M}_k$ and $\mathcal{M}_k|^{\sigma A_{i+1}}, t \vDash p_{i+1} \vee q_{i+1}$, it follows that t is x_{i+1} or $\bar{x}_{i+1}$. Thus there exists $\sigma' \in path(\tau')$ such that $\sigma' = \sigma A_{i+1}$ and $t = g(\sigma')$. By assumption, we have $\mathcal{M}_k|^{\sigma'}, g(\sigma') \vDash QT_{i+2} \dots QT_n\psi$. Thus we have for each $\sigma \in path(\tau)$, $\mathcal{M}_k|^{\sigma}, g(\sigma) \vDash QT_{i+1} \dots QT_n\psi$.

If $i+1$ is odd, we firstly verify the right-to-left direction. By assumption, it follows that for each $\sigma \in path(\tau)$, there exist t such that $g(\sigma) \overset{A_{i+1}}{\to} t$ and $\mathcal{M}_k|^{\sigma A_{i+1}}, t \vDash p_{i+1} \vee q_{i+1}$ and $\mathcal{M}_k|^{\sigma A_{i+1}}, t \vDash QT_{i+2} \dots QT_n\psi$. By the definition of $\mathcal{M}_k$ and $\mathcal{M}_k|^{\sigma A_{i+1}}, t \vDash p_{i+1} \vee q_{i+1}$, we have t is x_{i+1} or $\bar{x}_{i+1}$. Then we get τ' by extending each $\sigma \in path(\tau)$. If t is x_{i+1}, we add a successor 1 and label the new edge as a_{i+1}. If t is $\bar{x}_{i+1}$, we add a successor 0 and label the new edge as $\bar{a}_{i+1}$. Then $\sigma' = \sigma A_{i+1}$ and $t = g(\sigma')$. Thus we have $\mathcal{M}_k|^{\sigma'}, g(\sigma') \vDash QT_{i+2} \dots QT_n\psi$. To verify the left-to-right direction, we get τ by cutting the $i+1$ level of τ'. Next we need to show that for each $\sigma \in path(\tau)$, $\mathcal{M}_k|^{\sigma}, g(\sigma) \vDash \langle(a_{i+1} + \bar{a}_{i+1}); ?(p_{i+1} \vee q_{i+1})\rangle QT_{i+2} \dots QT_n\psi$. By the construction of τ, we know that for each $\sigma \in path(\tau)$, there exists $\sigma' \in path(\tau')$ such that $g(\sigma) \overset{A_{i+1}}{\to} g(\sigma')$, $\mathcal{M}_k|^{\sigma'}, g(\sigma') \vDash p_{i+1} \vee q_{i+1}$ and $\mathcal{M}_k|^{\sigma'}, g(\sigma') \vDash QT_{i+2} \dots QT_n\psi$. Thus we have $\mathcal{M}_k|^{\sigma}, g(\sigma) \vDash QT_{i+1} \dots QT_n\psi$. $\square$

THEOREM 4.1. *The following two are equivalent:*

- $\alpha = Q_1x_1Q_2x_2 \dots Q_nx_n\phi(x_1, \dots, x_n)$ *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 4.1 and 4.3, 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 $1 \le i \le n$. 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 $\sigma(x_i) = 1$ iff $\mathcal{M}_n|^{\sigma}, g(\sigma) \vDash \hat{K}p_i$ and $\mathcal{M}_n|^{\sigma}, g(\sigma) \vDash \hat{K}p_i$ iff $\mathcal{M}_n|^{\sigma}, g(\sigma) \vDash \neg\hat{K}q_i$. By the definition of τ, we know that $\sigma = A_1 \dots A_n$ where A_i is a_i or $\bar{a}_i$ for each $1 \le i \le n$.

Firstly, we will show that $\mathcal{M}_n|^{\sigma}, g(\sigma) \vDash \hat{K}p_i$ if and only if $\mathcal{M}_n|^{\sigma}, g(\sigma) \vDash \neg\hat{K}q_i$. To verify the right-to-left direction, if $\mathcal{M}_n|^{\sigma}, g(\sigma) \vDash \hat{K}p_i$, it follows by the definition of $\mathcal{M}_n$ that $x_i \in \mathcal{U}|^{\sigma}$. Then it must be the case that a_i occurs in σ. Suppose not, $\bar{a}_i$ occurs in σ. It follows by Proposition 4.2, $\mathcal{U}|^{\sigma} = \{x_0, A_1, \dots, A_{i-1}, \bar{x}_i, A_{i+1}, \dots, A_n\}$. This is contrary with $x_i \in \mathcal{U}|^{\sigma}$. Thus it must be that a_i occurs in σ. It follows by Proposition 4.2 that $\mathcal{U}|^{\sigma} = \{x_0, A_1, \dots, A_{i-1}, x_i, A_{i+1}, \dots, A_n\}$. Thus $\bar{x}_i \notin \mathcal{U}|^{\sigma}$. By the definition of $\mathcal{M}_n$ and the semantics, we have $\mathcal{M}_n|^{\sigma}, g(\sigma) \vDash \neg\hat{K}q_i$. To verify the left-to-right direction, $\mathcal{M}_n|^{\sigma}, g(\sigma) \vDash \neg\hat{K}q_i$ implies that $\bar{x}_i \notin \mathcal{U}|^{\sigma}$. For the similar reason as above, it must be the case that $\bar{a}_i$ does not occur in σ. Thus we have that a_i occurs in σ. It follows by Proposition 4.2 that $x_i \in \mathcal{U}|^{\sigma}$. Thus we have $\mathcal{M}_n|^{\sigma}, g(\sigma) \vDash \hat{K}p_i$.

Next we will show that $\sigma(x_i) = 1$ iff $\mathcal{M}_n|^{\sigma}, g(\sigma) \vDash \hat{K}p_i$. To verify the right-to-left direction, $\sigma(x_i) = 1$ implies that $A_i = a_i$. It follows by Proposition 4.2 that $x_i \in \mathcal{U}|^{\sigma}$. Thus we have $\mathcal{M}_n|^{\sigma}, g(\sigma) \vDash \hat{K}p_i$. To verify the left-to-right direction, we will show that $\sigma(x_i) = 0$ implies $\mathcal{M}_n|^{\sigma}, g(\sigma) \vDash \hat{K}q_i$. It follows by the definition of $\sigma(x_i) = 0$ that $A_i = \bar{a}_i$. It follows by Proposition 4.2 that $\bar{x}_i \in \mathcal{U}|^{\sigma}$. Thus we have $\mathcal{M}_n|^{\sigma}, g(\sigma) \vDash \hat{K}q_i$. $\square$

This gives us the desired lower bound:

THEOREM 4.2. *The model checking problem for* $\mathtt{EPDL}^-$ *is* PSPACE-*hard.*

4.1.2 Upper Bound

In this section we give a non-trivial model checking algorithm for $\mathtt{EPDL}^-$ inspired by an equivalent semantics.

As we mentioned earlier, the semantics of $\mathtt{EPDL}$ is context-dependent: reaching the same state through different paths may affect the truth value of an epistemic subformula. This means that the usual global model checking algorithm for modal logics may not work here. In order to establish the upper bound, we first give the following equivalent semantics to $\mathtt{EPDL}^-$ which makes the context dependency explicit in order to facilitate a local model checking algorithm. The idea is to keep the model intact but record the scope of action modalities in order to compute the right uncertainty set for epistemic subformulas. Similar idea appeared in [34] to give an alternative semantics of public announcement logic.

DEFINITION 4.3. *Given an uncertainty map $\mathcal{M} = \langle \mathcal{S}, \{\mathcal{R}_a \mid a \in \mathtt{A}\}, \mathcal{V}, \mathcal{U}\rangle$ and any point $s \in \mathcal{S}$, the satisfaction relation $\Vdash$ is defined using the auxiliary satisfaction relation $\Vdash_\sigma$ and auxiliary relation $\overset{\omega}{\to}_{\sigma}$, where σ is a finite (possibly empty) sequence of actions in* $\mathtt{A}$:

$\mathcal{M}, s \Vdash \phi$	$\Leftrightarrow \mathcal{M}, s \Vdash_\epsilon \phi$
$\mathcal{M}, s \Vdash_\sigma \top$	$\Leftrightarrow$ *always*
$\mathcal{M}, s \Vdash_\sigma p$	$\Leftrightarrow p \in \mathcal{V}(s)$
$\mathcal{M}, s \Vdash_\sigma \neg\phi$	$\Leftrightarrow \mathcal{M}, s \nVdash_\sigma \phi$
$\mathcal{M}, s \Vdash_\sigma \phi \wedge \psi$	$\Leftrightarrow \mathcal{M}, s \Vdash_\sigma \phi$ *and* $\mathcal{M}, s \Vdash_\sigma \psi$
$\mathcal{M}, s \Vdash_\sigma K\phi$	$\Leftrightarrow$ *for all* $v \in \mathcal{U}\|^\sigma : \mathcal{M}, v \Vdash_\sigma \phi$
$\mathcal{M}, s \Vdash_\sigma \langle\pi\rangle\phi$	$\Leftrightarrow$ *there exists* $\omega \in \mathcal{L}(\pi)$ *and* $t \in \mathcal{S}$ *such that* $s \xrightarrow{\omega_\sigma} t$ *and* $\mathcal{M}, t \Vdash_{\sigma r(\omega)} \phi$
$s \xrightarrow{\epsilon_\sigma} t$	$\Leftrightarrow s = t$
$s \xrightarrow{(a\omega')_\sigma} t$	$\Leftrightarrow$ *there exists* s' *such that* $s \xrightarrow{a} s'$ *and* $s' \xrightarrow{\omega'_{(\sigma a)}} t$
$s \xrightarrow{(?\phi\omega')_\sigma} t$	$\Leftrightarrow \mathcal{M}, s \Vdash_\sigma \phi$ *and* $s \xrightarrow{\omega'_\sigma} t$

where $r(\omega)$ *is the sequence of actions obtained by eliminating all the tests in* ω.

Note that ω in the above definition is a finite sequence of actions and EPDL$^-$-tests, while σ is test-free.

We first prove a simple proposition regarding the decomposition of $\xrightarrow{\omega_\sigma}$.

PROPOSITION 4.4. *Given an uncertainty map* $\mathcal{M}$ *and sequences of actions and tests* η, ω, ω' *such that* $\eta = \omega\omega'$, *we have* $(s,t) \in \xrightarrow{\eta_\sigma}$ *iff* $(s,t) \in \xrightarrow{\omega_\sigma} \circ \xrightarrow{\omega'_{\sigma r(\omega)}}$ *for any sequence of actions* σ.

PROOF. We prove it by induction on $|\eta|$. If $|\eta| \leq 2$, it is obvious by the definition. If $|\eta| > 2$, there are two cases, that is, $\eta = a\eta'$ or $\eta = ?\phi\eta'$.

Case $\eta = a\eta'$: We have $\omega = a\omega''$ for some initial segment ω'' of η', and $(s,t) \in \xrightarrow{(a\eta')_\sigma}$ iff there exists s' such that $s \xrightarrow{a} s'$ and $(s',t) \in \xrightarrow{\eta'_{\sigma a}}$. By IH, we have $\xrightarrow{\eta'_{\sigma a}} = \xrightarrow{\omega''_{\sigma a}} \circ \xrightarrow{\omega'_{\sigma a r(\omega'')}}$. Thus we have $(s',t) \in \xrightarrow{\eta'_{\sigma a}}$ iff there exists t' such that $(s',t') \in \xrightarrow{\omega''_{\sigma a}}$ and $(t',t) \in \xrightarrow{\omega'_{\sigma a r(\omega'')}}$. By definition, we have that $s \xrightarrow{a} s'$ and $(s',t') \in \xrightarrow{\omega''_{\sigma a}}$ iff $(s,t') \in \xrightarrow{a\omega''_\sigma}$. Thus we have $(s,t) \in \xrightarrow{a\omega''_\sigma} \circ \xrightarrow{\omega'_{\sigma a r(\omega'')}}$, namely $(s,t) \in \xrightarrow{\omega_\sigma} \circ \xrightarrow{\omega'_{\sigma r(\omega)}}$.

Case $\eta = ?\phi\eta'$: We have $\omega = ?\phi\omega''$ for some initial segment ω'' of η', and $(s,t) \in \xrightarrow{(?\phi\eta')_\sigma}$ iff $\mathcal{M}, s \Vdash_\sigma \phi$ and $s \xrightarrow{\eta'_\sigma} t$. By IH, we have $s \xrightarrow{\eta'_\sigma} t$ iff $(s,t) \in \xrightarrow{\omega''_\sigma} \circ \xrightarrow{\omega'_{\sigma r(\omega'')}}$. Thus we have there exists s' such that $(s,s') \in \xrightarrow{\omega''_\sigma}$ and $(s',t) \in \xrightarrow{\omega'_{\sigma r(\omega'')}}$. This follows that $(s,s') \in \xrightarrow{(?\phi\omega'')_\sigma}$, and $(s,t) \in \xrightarrow{(?\phi\omega'')_\sigma} \circ \xrightarrow{\omega'_{\sigma r(?\phi\omega'')}}$, namely $(s,t) \in \xrightarrow{\omega_\sigma} \circ \xrightarrow{\omega'_{\sigma r(\omega)}}$. $\square$

In the following we show that $\Vdash$ coincides with $\vDash$.

THEOREM 4.3. *Given an uncertainty map* $\mathcal{M}$ *and an action sequence* σ, *if* $\mathcal{U}|^\sigma \neq \emptyset$, *we have that for each* $s \in \mathcal{U}|^\sigma$,

(i) $\mathcal{M}|^\sigma, s[\![\pi]\!]\mathcal{M}', s'$ *iff there exists* $\omega \in \mathcal{L}(\pi)$ *such that* $\mathcal{M}' = \mathcal{M}|^{\sigma r(\omega)}$ *and* $s \xrightarrow{\omega_\sigma} s'$,

(ii) $\mathcal{M}|^\sigma, s \vDash \phi$ *iff* $\mathcal{M}, s \Vdash_\sigma \phi$.

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

Case $\pi_1;\pi_2$: First, we show the direction from left to right. It follows by assumption that there is pointed uncertainty map $\mathcal{N}, t$ such that $\mathcal{M}|^\sigma, s[\![\pi_1]\!]\mathcal{N}, t$ and $\mathcal{N}, t[\![\pi_2]\!]\mathcal{M}', s'$. By IH, we have that there exists $\omega \in \mathcal{L}(\pi_1)$ such that $\mathcal{N} = \mathcal{M}|^{\sigma r(\omega)}$ and $s \xrightarrow{\omega_\sigma} t$. Since $\mathcal{N}, t$ is a pointed uncertainty map and $\mathcal{N} = \mathcal{M}|^{\sigma r(\omega)}$, we have $t \in \mathcal{U}|^{\sigma r(\omega)}$. By IH and $\mathcal{M}|^{\sigma r(\omega)}, t[\![\pi_2]\!]\mathcal{M}', s'$, we have that there exists $\omega' \in \mathcal{L}(\pi_2)$ such that $\mathcal{M}|^{\sigma r(\omega) r(\omega')} = \mathcal{M}|^{\sigma r(\omega\omega')} = \mathcal{M}'$ and $t \xrightarrow{\omega'_{\sigma r(\omega)}} s'$. By Proposition 4.4, it follows that $\omega\omega' \in \mathcal{L}(\pi_1;\pi_2)$ and $s \xrightarrow{(\omega\omega')_\sigma} s'$. Right to left: If there exists $\omega \in \mathcal{L}(\pi_1;\pi_2)$ such that $\mathcal{M}' = \mathcal{M}|^{\sigma r(\omega)}$ and $s \xrightarrow{\omega_\sigma} s'$, we need to show that $\mathcal{M}|^\sigma, s[\![\pi]\!]\mathcal{M}', s'$. It follows by $\omega \in \mathcal{L}(\pi_1;\pi_2)$ that there are $\rho \in \mathcal{L}(\pi_1)$ and $\rho' \in \mathcal{L}(\pi_2)$ such that $\omega = \rho\rho'$. Thus we have $\mathcal{M}' = \mathcal{M}|^{\sigma r(\rho)}|^{r(\rho')}$ and $s \xrightarrow{(\rho\rho')_\sigma} s'$. By Proposition 4.4, this follows that there exists t such that $s \xrightarrow{\rho_\sigma} t$ and $t \xrightarrow{\rho'_{\sigma r(\rho)}} s'$. It follows by $\mathcal{U}|^{\sigma r(\omega)} \neq \emptyset$ that $\mathcal{U}|^{\sigma r(\rho)} \neq \emptyset$. By IH, it follows that $\mathcal{M}|^\sigma, s[\![\pi_1]\!]\mathcal{M}|^{\sigma r(\rho)}, t$ and $\mathcal{M}|^{\sigma r(\rho)}, t[\![\pi_2]\!]\mathcal{M}|^{\sigma r(\rho) r(\rho')}, s'$, then $\mathcal{M}|^\sigma, s[\![\pi_1;\pi_2]\!]\mathcal{M}', s'$.

Case $\pi_1 + \pi_2$: We first show the direction from left to right. If $\mathcal{M}|^\sigma, s[\![\pi_1 + \pi_2]\!]\mathcal{M}', s'$, we have $\mathcal{M}|^\sigma, s[\![\pi_1]\!]\mathcal{M}', s'$ or $\mathcal{M}|^\sigma, s[\![\pi_2]\!]\mathcal{M}', s'$. Suppose $\mathcal{M}|^\sigma, s[\![\pi_1]\!]\mathcal{M}', s'$, it follows by IH that there exists $\omega \in \mathcal{L}(\pi_1) \subseteq \mathcal{L}(\pi_1 + \pi_2)$ such that $\mathcal{M}' = \mathcal{M}|^{\sigma r(\omega)}$ and $s \xrightarrow{\omega_\sigma} s'$. Right to Left: If there exists $\omega \in \mathcal{L}(\pi_1 + \pi_2)$, it follows that $\omega \in \mathcal{L}(\pi_1)$ or $\omega \in \mathcal{L}(\pi_2)$. Suppose $\omega \in \mathcal{L}(\pi_1)$, since $\mathcal{M}' = \mathcal{M}|^{\sigma r(\omega)}$ and $s \xrightarrow{\omega_\sigma} s'$, it follows by IH that $\mathcal{M}|^\sigma, s[\![\pi_1]\!]\mathcal{M}', s'$. This follows that $\mathcal{M}|^\sigma, s[\![\pi_1 + \pi_2]\!]\mathcal{M}', s'$.

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

Case $\langle\pi\rangle\phi$: We have $\mathcal{M}|^\sigma, s \vDash \langle\pi\rangle\phi$ if and only if there is pointed uncertainty map $\mathcal{M}', s'$ such that $\mathcal{M}|^\sigma, s[\![\pi]\!]\mathcal{M}', s'$ and $\mathcal{M}', s' \vDash \phi$. By (i), it follows that $\mathcal{M}|^\sigma, s[\![\pi]\!]\mathcal{M}', s'$ iff there exists $\omega \in \mathcal{L}(\pi)$ such that $\mathcal{M}' = \mathcal{M}|^{\sigma r(\omega)}$ and $s \xrightarrow{\omega_\sigma} s'$. By IH, it follows that $\mathcal{M}|^{\sigma r(\omega)}, s' \vDash \phi$ iff $\mathcal{M}, s' \Vdash_{\sigma r(\omega)} \phi$. Thus we have $\mathcal{M}, s \Vdash \langle\pi\rangle\phi$. $\square$

Let σ be ϵ, we have the equivalence of $\Vdash$ and $\vDash$.

COROLLARY 4.1. *Given pointed uncertainty map* $\mathcal{M}, s$, *we have* $\mathcal{M}, s \vDash \phi$ *iff* $\mathcal{M}, s \Vdash \phi$ *for each* $\phi \in$ *EPDL*$^-$.

This alternative semantics induces a natural algorithm to compute the truth value of an EPDL$^-$ formula w.r.t. to a pointed uncertainty map. The idea is to recursively call a function $MC(\mathcal{M}, s, \sigma, \phi)$ which returns the truth value of a subformula ϕ on state s given the context of σ while keeping $\mathcal{M}$ intact. Note that, we do not need to compute all the $MC(\mathcal{M}, s, \sigma, \phi)$ for each σ and each subformula ϕ. The only tricky part comes when evaluating $\langle\pi\rangle\phi$ formulas since it is too space consuming to compute the whole set of $\mathcal{L}(\pi)$ in the search of the right ω. Instead, we can generate one by one in some lexicographical order all the possible sequences up to a bound based on the atomic actions and tests occurring in the formula, and then test whether it belongs to the program π. Note that in this way, we can use the space repeatedly, and the membership testing of $\mathcal{L}(\pi)$ is not expensive (NLOGSPACE-complete according to [20]).

In the appendix we present three algorithms based on matrix representation of the model: Algorithm 1 computes the uncertainty set $\mathcal{U}|^\sigma$; Algorithm 2 computes $\xrightarrow{w_\sigma}$ and 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 the length of the formula and for each call polynomial space suffices. The detailed algorithms and complexity analysis can be found in the appendix. It is not hard to show the following:

THEOREM 4.4 (UPPER BOUND). *The model checking problem of* EPDL$^-$ *is in* PSPACE. *Thus it is* PSPACE-*complete.*

4.2 Upper Bounds for model checking EPDL

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

DEFINITION 4.4 (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.[5]

DEFINITION 4.5 (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:*

$\mathfrak{M}, s \Vdash \top$	*always*
$\mathfrak{M}, s \Vdash p$	$\Leftrightarrow s \in \mathcal{V}(p)$
$\mathfrak{M}, s \Vdash \neg\phi$	$\Leftrightarrow \mathfrak{M}, s \nVdash \phi$
$\mathfrak{M}, s \Vdash \phi \wedge \psi$	$\Leftrightarrow \mathfrak{M}, s \Vdash \phi$ *and* $\mathfrak{M}, s \Vdash \psi$
$\mathfrak{M}, s \Vdash K\phi$	$\Leftrightarrow \forall u \in \mathcal{S} : s \sim u$ *implies* $\mathfrak{M}, u \Vdash \phi$
$\mathfrak{M}, s \Vdash [\pi]\phi$	$\Leftrightarrow \forall t \in S : s \xrightarrow{\pi} t$ *implies* $\mathfrak{M}, t \Vdash \phi$
$\xrightarrow{a}$	$= \mathcal{R}_a$
$\xrightarrow{?\phi}$	$= \{(s, s) \mid \mathfrak{M}, s \Vdash \phi\}$
$\xrightarrow{\pi_1;\pi_2}$	$= \xrightarrow{\pi_1} \circ \xrightarrow{\pi_2}$
$\xrightarrow{\pi_1+\pi_2}$	$= \xrightarrow{\pi_1} \cup \xrightarrow{\pi_2}$
$\xrightarrow{\pi^*}$	$= (\xrightarrow{\pi})^\star$

where $\circ, \cup,$ * 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 4.6. *Given any Kripke model $\mathcal{M} = \langle \mathcal{S}, \{\mathcal{R}_a \mid a \in \mathtt{A}\}, \mathcal{V}\rangle$, we define the ETS model $\mathcal{M}^\bullet$ as follows:*

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

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

[5]Here we abuse the notation $\Vdash$ to denote the new semantics. Note that it is different from the alternative semantics in the previous section.

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

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

PROPOSITION 4.5. *Given any map $\mathcal{M}$, we have*

(i) $\mathcal{M}^\Gamma, s[\![\pi]\!]\mathcal{M}^\Delta, t$ iff $s_\Gamma \xrightarrow{\pi} t_\Delta$ in $\mathcal{M}^\bullet$;[6]

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

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 $\pi_1;\pi_2$, $\pi_1 + \pi_2$ and π^*; the other cases are straightforward.

Case $\pi_1;\pi_2$: We have $\mathcal{M}^\Gamma, s[\![\pi_1;\pi_2]\!]\mathcal{M}^\Delta, t$ if and only if $\mathcal{M}^\Gamma, s[\![\pi_1]\!]\mathcal{M}^{\Gamma'}, s'$ and $\mathcal{M}^{\Gamma'}, s'[\![\pi_2]\!]\mathcal{M}^\Delta, t$ 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 $s_\Gamma \xrightarrow{\pi_1;\pi_2} t_\Delta$. The case for $\pi_1 + \pi_2$ is similar.

Case π^*: By induction on n, it can be proved that for each n, $\mathcal{M}^\Gamma, s[\![\pi^n]\!]\mathcal{M}^\Delta, t$ if and only if $s_\Gamma \xrightarrow{\pi^n} t_\Delta$. This follows that $\mathcal{M}^\Gamma, s[\![\pi^*]\!]\mathcal{M}^\Delta, t$ 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 $\mathcal{M}^\Gamma, s \vDash [\pi]\phi$ but $\mathcal{M}^\bullet, s_\Gamma \nVdash [\pi]\phi$, then $\mathcal{M}^\bullet, t_\Delta \nVdash \phi$ for some $t_\Delta \in \mathcal{S}^\bullet$ such that $s_\Gamma \xrightarrow{\pi} t_\Delta$. By IH, this follows $\mathcal{M}^\Delta, t \nvDash \phi$ and $\mathcal{M}^\Gamma, s[\![\pi]\!]\mathcal{M}^\Delta, t$. This is contradictory with the assumption that $\mathcal{M}^\Gamma, s \vDash [\pi]\phi$. If $\mathcal{M}^\bullet, s_\Gamma \Vdash [\pi]\phi$ but $\mathcal{M}^\Gamma, s \nvDash [\pi]\phi$, it follows that $\mathcal{N}, t \nvDash \phi$ for some pointed uncertainty map $\mathcal{N}, t$ such that $\mathcal{M}^\Gamma, s[\![\pi]\!]\mathcal{M}^\Delta, t$ for some $\Delta \in 2^{\mathcal{S}}$. Thus we have $\mathcal{M}^\Gamma, s[\![\pi]\!]\mathcal{M}^\Delta, t$ and $\mathcal{M}^\Delta, t \nvDash \phi$. By IH, we have $s_\Gamma \xrightarrow{\pi} t_\Delta$ and $\mathcal{M}^\bullet, t_\Delta \nVdash \phi$. This is contradictory with $\mathcal{M}^\bullet, s_\Gamma \Vdash [\pi]\phi$. □

COROLLARY 4.2. *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 we can have a model checking method via model checking EPDL over ETS models.

PROPOSITION 4.6. *The model checking problem of* EPDL *on uncertainty maps is in* EXPTIME.

PROOF. Given an uncertainty map $\mathcal{M} = \langle \mathcal{N}, \mathcal{U}\rangle$, the construction of ETS $\mathcal{N}^\bullet$ can be done in exponential time in the size of $\mathcal{N}$. By modifying the algorithm for PDL in [23], we can get an algorithm to check EPDL formula ϕ on $\mathcal{N}^\bullet$ w.r.t. $\Vdash$, and its time complexity is $O(|\phi|^2 \cdot |\mathcal{N}^\bullet|^3)$. Thus, the time complexity of model checking ϕ on $\mathcal{M}$ is bounded by $O(|\phi|^2 \cdot |\mathcal{S}_\mathcal{N}|^3 \cdot 2^{3|\mathcal{S}_\mathcal{N}|-3})$. □

We conjecture that the model checking problem of full EPDL is EXPTIME-complete, and leave the lower bound to the extended version of this paper.

4.3 Complexity of conformant planning

In the rest of this section, let us look at the complexity of conformant planning in terms of EPDL model checking. Although the model checking problem of full EPDL is likely to be EXPTIME-complete, the complexity of model checking

[6]Cf. the definition of $\xrightarrow{\pi}$ in Def. 4.5.

the EPDL formula which encodes the conformant planning problem (cf. Theorem 3.1) is in PSPACE if the goal formula is program-free. More precisely, we can show the following:

THEOREM 4.5. *The problem of model checking* EPDL *formulas in the shape of* $\langle(\Sigma_{a\in B}(?K\langle a\rangle\top;a))^*\rangle K\phi$, *where* ϕ *is an epistemic formula (i.e. program-free) and* $B\subseteq A$, *is in* PSPACE.

PROOF. (Sketch) Note that $(\sum_{a\in\mathtt{B}}(?K\langle a\rangle\top;a))^*$ is a special program which has only simple epistemic tests depending on the structure of the underlying Kripke model. Now given a Kripke model $\mathcal{N}$ and a set $\mathtt{B}\subseteq\mathtt{A}$ we can define an ETS model $\mathcal{N}^\circ$ similar to $\mathcal{N}^\bullet$ but with a different definition for the action relations:

$$\mathcal{R}^\circ_a=\{(s_\Gamma,t_\Delta)\mid s\xrightarrow{a}t,\Delta=\Gamma|^a,\forall u\in\Gamma\exists v\ st.\ u\xrightarrow{a}v.\}$$

Note that the extra condition guarantees that the action a is always executable w.r.t. the whole Γ, thus fulfilling the test $K\langle a\rangle\top$. Now we can have an analog of Corollary 4.2, and reduce the problem of checking $\langle\mathcal{N},\mathcal{U}\rangle,s\vDash(\sum_{a\in\mathtt{B}}(?K\langle a\rangle\top;a))^*K\phi$ to the reachability problem in $\mathcal{N}^\circ$: whether there is a path from $s_\mathcal{U}$ in $\mathcal{N}^\circ$ such that it can reach a state $t_{\mathcal{U}'}$ where $K\phi$ holds. Since ϕ is $[\pi]$-free, we can check it easily given $\mathcal{U}'$ using polynomial space, thus the main task is to find the reachable $t_{\mathcal{U}'}$. Note that, in the size of $\mathcal{N}$, there are exponentially many such $t_{\mathcal{U}'}$ and the maximal length of the plan is also exponential. However, we do not need to build the whole $\mathcal{N}^\circ$ and the bisection-like algorithm behind the proof of Savitch's Theorem will do the job.[7] More precisely, we first pick up a $t_{\mathcal{U}'}$, and then run the recursive bisection method to see whether $t_{\mathcal{U}'}$ is reachable from $s_\mathcal{U}$ within $2^{|\mathcal{N}|}$ steps. The depth of the recursion is bounded by $log_2(2^{|\mathcal{N}|})=|\mathcal{N}|$ and at each recursion we need to record the choice of the state which can be encoded by a $(0,1)$-vector using $log_2(2^{|\mathcal{N}|})=|\mathcal{N}|$ space (plus one bit to record the result). Moreover, at the bottom of the recursion we only need to verify one step reachability, i.e., whether two states in $\mathcal{N}^\circ$ are linked by $\mathcal{R}^\circ_a$, without building the whole $\mathcal{N}^\circ$. Thus the whole procedure of model checking can be done using polynomial space. □

As we mentioned in the introduction, the conformant planning problems in the AI literature are usually given by using state variables and actions with precondition and (conditional) effects, rather than explicit transition systems. The corresponding 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 [21, 9]. Actually, the complexity result of Theorem 4.5 can be strengthened to PSPACEcomplete based on the corresponding complexity result in the AI literature.

However, not all the transition systems can be generated in this way since the preconditions and postconditions are purely propositional and thus cannot distinguish two states that share the same valuation. Thus in a transition system we may allow multiple states with the same valuation but different available actions and this may handle scenarios where some external factors about the states are not modelled by basic propositions.

[7] A similar algorithm was used to pinpoint complexity of the conformant planning in AI, cf.[21].

5. CONCLUSIONS AND FUTURE WORK

In this work we first introduce the logical language EAL over uncertainty maps and axiomatize it completely. EAL is then extended to EPDL with programs to specify conformant and conditional plans. We show that the conformant planning problems can be reduced to model checking problems of EPDL. Finally we showed that model checking star-free EPDL over uncertainty maps is PSPACE-complete and model checking the full fragment is in EXPTIME. On the other hand, model checking the conformant planning problem is in PSPACE.

Note that our EPDL is a powerful language which can already express conditional plans. 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 have been already incorporated in [35], which can be extended to general feedbacks discussed in the literature of contingent planning (cf. e.g., [11]). 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 [5].

6. REFERENCES

[1] Mikkel Birkegaard Andersen, Thomas Bolander, and Martin Holm Jensen. Conditional epistemic planning. In *Logics in Artificial Intelligence*, pages 94–106. Springer, 2012.

[2] Mikkel Birkegaard Andersen, Thomas Bolander, and Martin Holm Jensen. Don't plan for the unexpected: Planning based on plausibility models. *Logique et Analyse*, 1(1), 2014.

[3] Guillaume Aucher. Del-sequents for regression and epistemic planning. *Journal of Applied Non-Classical Logics*, 22(4):337–367, 2012.

[4] Guillaume Aucher and Thomas Bolander. Undecidability in epistemic planning. In *IJCAI*, pages 27–33, 2013.

[5] Christer Bäckström and Peter Jonsson. All pspace-complete planning problems are equal but some are more equal than others. In *Proceedings of the Fourth Annual Symposium on Combinatorial Search, SOCS 2011, Castell de Cardona, Barcelona, Spain, July 15.16, 2011*, 2011.

[6] Alexandru. Baltag and Larry Moss. Logics for epistemic programs. *Synthese*, 139:165–224, March 2004.

[7] Piergiorgio Bertoli, Alessandro Cimatti, Marco Roveri, and Paolo Traverso. Strong planning under partial observability. *Artif. Intell.*, 170(4-5):337–384, 2006.

[8] Thomas Bolander and M. Birkegaard Andersen. Epistemic planning for single and multi-agent systems. *Journal of Applied Non-Classical Logics*, 21(1):9–34, 2011.

[9] Blai Bonet. Conformant plans and beyond: Principles and complexity. *Artif. Intell.*, 174(3-4):245–269, 2010.

[10] Blai Bonet and Hector Geffner. Planning with incomplete information as heuristic search in belief space. In *Proceedings of the Fifth International Conference on Artificial Intelligence Planning Systems, Breckenridge, CO, USA, April 14-17, 2000*, pages 52–61, 2000.

[11] Blai Bonet and Hector Geffner. Width and complexity of belief tracking in non-deterministic conformant and contingent planning. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, July 22-26, 2012, Toronto, Ontario, Canada.*, 2012.

[12] Ronen I. Brafman and Jörg Hoffmann. Conformant planning via heuristic forward search: A new approach. In *Proceedings of the Fourteenth International Conference on Automated Planning and Scheduling (ICAPS 2004), June 3-7 2004, Whistler, British Columbia, Canada*, pages 355–364, 2004.

[13] Daniel Bryce, Subbarao Kambhampati, and David E. Smith. Planning graph heuristics for belief space search. *J. Artif. Intell. Res. (JAIR)*, 26:35–99, 2006.

[14] Alessandro Cimatti and Marco Roveri. Conformant planning via symbolic model checking. *J. Artif. Intell. Res. (JAIR)*, 13:305–338, 2000.

[15] Alessandro Cimatti and Marco Roveri. Conformant planning via symbolic model checking. *CoRR*, abs/1106.0252, 2011.

[16] Alessandro Cimatti, Marco Roveri, and Piergiorgio Bertoli. Conformant planning via symbolic model checking and heuristic search. *Artif. Intell.*, 159(1-2):127–206, 2004.

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

[18] R. Fagin, J. Y. Halpern, Y. Moses, and M. Y. Vardi. Knowledge-based programs. *Distributed Computing*, 10(4):199–225, July 1997.

[19] Patrik Haslum and Peter Jonsson. Some results on the complexity of planning with incomplete information. In *Recent Advances in AI Planning, 5th European Conference on Planning, ECP'99, Durham, UK, September 8-10, 1999, Proceedings*, pages 308–318, 1999.

[20] Tao Jiang and B. Ravikumar. A note on the space complexity of some decision problems for finite automata. *Information Processing Letters*, 40:25–31, October 1991.

[21] Jon Kleinberg and Éva Tardos. *Algorithm Design*. Addison-Wesley, 2005.

[22] Jérôme Lang and Bruno Zanuttini. Knowledge-based programs as plans - the complexity of plan verification. In *ECAI 2012 - 20th European Conference on Artificial Intelligence. System Demonstrations Track, Montpellier, France, August 27-31 , 2012*, pages 504–509, 2012.

[23] Martin Lange. Model checking propositional dynamic logic with all extras. *Journal of Applied Logic*, 4:39–49, 2006.

[24] Benedikt Löwe, Eric Pacuit, and Andreas Witzel. Del planning and some tractable cases. In *Logic, Rationality, and Interaction*, pages 179–192. Springer, 2011.

[25] Héctor Palacios and Hector Geffner. Compiling uncertainty away: Solving conformant planning problems using a classical planner (sometimes). In *Proceedings, The Twenty-First National Conference on Artificial Intelligence and the Eighteenth Innovative Applications of Artificial Intelligence Conference, July 16-20, 2006, Boston, Massachusetts, USA*, pages 900–905, 2006.

[26] Héctor Palacios and Hector Geffner. Compiling uncertainty away in conformant planning problems with bounded width. *CoRR*, abs/1401.3468, 2014.

[27] Pere Pardo and Mehrnoosh Sadrzadeh. Strong planning in the logics of communication and change. In *Declarative Agent Languages and Technologies X*, pages 37–56. Springer, 2013.

[28] Rohit. Parikh and R. Ramanujam. Distributed processes and the logic of knowledge. In *Proceedings of Conference on Logic of Programs*, pages 256–268, London, UK, 1985. Springer-Verlag.

[29] Philippe Schnoebelen. The complexity of temporal logic model checking. In Philippe Balbiani, Nobu-Yuki Suzuki, Frank Wolter, and Michael Zakharyaschev, editors, *Selected Papers from the 4th Workshop on Advances in Modal Logics (AiML'02)*, pages 393–436, Toulouse, France, 2003. King's College Publication. Invited paper.

[30] David E. Smith and Daniel S. Weld. Conformant graphplan. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence and Tenth Innovative Applications of Artificial Intelligence Conference, AAAI 98, IAAI 98, July 26-30, 1998, Madison, Wisconsin, USA.*, pages 889–896, 1998.

[31] Larry J Stockmeyer and Albert R Meyer. Word problems requiring exponential time (preliminary report). In *Proceedings of the fifth annual ACM symposium on Theory of computing*, pages 1–9. ACM, 1973.

[32] Son Thanh To, Tran Cao Son, and Enrico Pontelli. A new approach to conformant planning using cnf*. In *Proceedings of the 20th International Conference on Automated Planning and Scheduling, ICAPS 2010, Toronto, Ontario, Canada, May 12-16, 2010*, pages 169–176, 2010.

[33] Hans van Ditmarsch, Wiebe van der Hoek, and Barteld Kooi. *Dynamic epistemic logic*. Springer, 2007.

[34] Yanjing Wang and Qinxiang Cao. On axiomatizations of public announcement logic. *Synthese*, 190:103–134, 2013.

[35] Yanjing Wang and Yanjun Li. Not all those who wander are lost: Dynamic epistemic reasoning in navigation. In *Advances in Modal Logic*, pages 559–580, 2012.

[36] Quan Yu, Ximing Wen, and Yongmei Liu. Multi-agent

epistemic explanatory diagnosis via reasoning about actions. In *IJCAI*, pages 1183–1190, 2013.

APPENDIX

A. ALGORITHMS FOR EPDL$^-$

Definition A.1 (Matrix representation). *Let $\mathtt{B}_{n\times m}$ denote a (0,1)-matrix of size $n \times m$. A matrix $\mathtt{B}_{n\times 1}$, or $\mathtt{B}_n$ for short, is called a vector. Given finite uncertainty map $\mathcal{M}$, its domain $\mathcal{S}$ can be linearly ordered as $\{s_1, \cdots, s_n\}$. Thus $\mathcal{M}$ can be represented by a set $\{\mathtt{B}^a_{n\times n} \mid a \in \mathtt{A}\}$ of adjacency matrices for accessibility relation, a vector $\mathtt{B}^{\mathcal{U}}_n$ for $\mathcal{U}$ and a set $\{\mathtt{B}^p_n \mid p \in \mathtt{P}\}$ of vectors for atomic propositions.*

Definition A.2. *Given (0,1)-matrices $\mathtt{B}'_{n\times k}, \mathtt{B}_{k\times m}$, their product $\mathtt{B}''_{n\times m}$ is defined as: $\mathtt{B}''_{n\times m}[i,j] = 1$ iff there exists $r \le n$ such that $\mathtt{B}'_{n\times k}[i,r] = \mathtt{B}_{k\times m}[r,j] = 1$ for all $1 \le i \le n, 1 \le j \le m$.*

The following algorithms are to check whether ϕ holds on pointed uncertainty map $\mathcal{M}, s$ by Definition 4.3. The main algorithm (Algorithm 3) recursively calls itself for each non-trivial subformula of ϕ. The complex cases are for the subformulas in the form of $\langle\pi\rangle\phi$ and $K\phi$. By Definition 4.3, to check $\mathcal{M}, s \Vdash_\sigma \langle\pi\rangle\phi$, we need to make sure that there exists a sequence $\omega \in \mathcal{L}(\pi)$ and a state $t \in \mathcal{S}$ such that $s \overset{\omega_\sigma}{\to} t$ and $\mathcal{M}, t \Vdash_{\sigma r(\omega)} \phi$. Since π is star-free, $|\omega| \le |\pi|$ for each $\omega \in \mathcal{L}(\pi)$. It is clear that we cannot compute and store the whole set of $\mathcal{L}(\pi)$ within polynomial space. Instead, *one by one* we generate all the possible sequences that are shorter than $|\pi|$ and are formed from the alphabet of π (cf. line 14) and check whether they are in $\mathcal{L}(\pi)$. We can order the possible sequences lexicographically according to an ordering of the basic actions and tests in *Sig*, and compute the next sequence merely from the current one using function *next*. *memb_chec(ω, π)* checks whether it is the case $\omega \in \mathcal{L}(\pi)$. If $\omega \in \mathcal{L}(\pi)$, we need to check whether there exists $s_j \in \mathcal{S}_{\mathcal{M}}$ such that $s \overset{\omega_\sigma}{\to} s_j$ (Algorithm 2) and $\mathcal{M}, s_j \Vdash_{\sigma r(\omega)} \phi$, where $r(\omega)$ is the test-free subsequence of ω which is easy to compute. For the case of $K\phi$, we need to calculate the state set $\mathcal{U}|^\sigma$ (Algorithm 1).

Algorithm 1: Function CNU($\mathcal{U}, \sigma$): Calculate the the new uncertainty set $\mathcal{U}|^\sigma$

input : $\mathcal{U}$, σ
output: $\mathtt{B}^{\mathcal{U}|^\sigma}_n$

$A \leftarrow \mathtt{B}^{\mathcal{U}}_n$;
for $i \leftarrow 1$ **to** $|\sigma|$ **do**
  $A \leftarrow A \times \mathtt{B}^{\sigma[i]}_{n\times n}$;
return A;

B. COMPLEXITY ANALYSIS

We suppose $|\mathcal{S}_{\mathcal{M}}| = n$ and $|\phi| = k$. Algorithm 1 uses one variable A to record the uncertainty set which requires $O(n)$ space. Note that there is a mutual recursion in Algorithm 2 and 3, but the depth of the overall recursion is bounded by k. In Algorithm 2, the variable consuming the most of the space is the matrix $\mathtt{B}_{n\times n}$ recording the (intermediate) relation.

Algorithm 2: Function $PW(\omega, \sigma)$: Calculate the binary relation $\overset{\omega_\sigma}{\to}$

input : Mixed sequence ω, action sequence σ
output: $\mathtt{B}_{n\times n}$

switch ω_σ **do**
  case ϵ_σ **return** $Matrix(\{(s,s) \mid s \in \mathcal{S}\})$ `/*` $Matrix(R)$ `is the` $(0,1)$`-matrix representation of the binary relation` R `*/`;
  case $(?\phi\omega')_\sigma$
    return $Matrix(\{(s,s) \mid \mathrm{MC}(\mathcal{M}, s, \sigma, \phi) = \text{true}\}) \times \mathrm{PW}(\omega', \sigma)$;
  case $(a\omega')_\sigma$ **return** $\mathtt{B}^a_{n\times n} \times \mathrm{PW}(\omega', \sigma a)$;

Algorithm 3: Function MC($\mathcal{M}, s, \sigma, \phi$): Model checking algorithm for $\mathtt{EPDL}^-$

input : The pointed uncertainty map $(\mathcal{M}, s)$, sequence of actions σ, $\phi \in \mathtt{EPDL}^-$.
output: true if $\mathcal{M}, s \Vdash_\sigma \phi$.

1 **switch** ϕ **do**
2   **case** p
3   **if** $p \in \mathcal{V}(s)$ **then**
4   **return** true;
5   **else**
6   **return** false;
7   **case** $\neg\varphi$
8   **return not** $MC(\mathcal{M}, s, \sigma, \varphi)$
9   **case** $\varphi_1 \wedge \varphi_2$
10   **return** MC($\mathcal{M}, s, \sigma, \varphi_1$) **and** MC($\mathcal{M}, s, \sigma, \varphi_2$);
11   **case** $\langle\pi\rangle\varphi$
12   Let *Sig* be the array consisting of atomic programs and formulas in π ordered according to their first appearances;
13     $\omega \leftarrow Sig[1]$ `/*` ω `is the candidate sequence we want to test */`;
14   **while** $|\omega| \le |\pi|$ **do**
15   **if** *memb_chec(ω, π)* **then**
16   **for** i = *1* **to** $\mathcal{S}_{\mathcal{M}}$ **do**
17   **if** $(s, s_i) \in PW(\omega, \sigma)$ **then**
18   **if** $MC(\mathcal{M}, s_j, \sigma r(\omega), \varphi)$ **then**
19   **return** true;
20       $\omega \leftarrow next(\omega, Sig)$ `/* calculate the next sequence lexicographically according to the order` Sig `*/`;
21   **return** false;
22   **case** $K\varphi$
23     $\mathtt{B}^{\mathcal{U}|^\sigma}_n = \mathrm{CNU}(\mathcal{U}, \sigma)$ `/* calculate the vector representation of` $\mathcal{U}|^\sigma$ `*/`
24   **for** m = *1* **to** $|\mathcal{S}_{\mathcal{M}}|$ **do**
25   **if** $(\mathtt{B}^{\mathcal{U}|^\sigma}_n)_m = 1$ **and** $MC(\mathcal{M}, s_m, \sigma, \varphi) =$ *false* **then**
26   **return** false;
27   **return** true;

Since σ and ω are also variables in the main algorithm and $|\omega| + |\sigma| \leq k$ due to the construction in Algorithm 3, the space usage of Algorithm 2 before the recursive calls of PW and MC is bounded by $O(k+n^2)$. For Algorithm 3, the most space-demanding part is the $\langle\pi\rangle\phi$ case, where we need to store π, Sig, and keep track one ω and one state s in the loop, which are bounded by either k or s. Moreover, according to [20], the complexity of *memb_chec* is NLOGSPACE-complete in the size of Sig, i.e., the alphabet of π which is bounded again by k. Thus before calling MC nor PW again in the $\langle\pi\rangle\phi$ case, the space requirement is at most linear in both k and n, which is less demanding than PW for each recursion. Recall that the overall recursion depth of MC (and PW) is bounded by k thus the space usage of the whole algorithm is bounded by $O(k(k+n^2)) = O(k^2 + kn^2)$.