On the Impact of Modal Depth in Epistemic Planning

Tristan Charrier
Université de Rennes 1
IRISA
tristan.charrier@irisa.fr

Bastien Maubert
Università degli studi di Napoli
Federico II
bastien.maubert@gmail.com

François Schwarzentruber
ENS Rennes / IRISA
francois.schwarzentruber
@ens-rennes.fr

Abstract

Epistemic planning is a variant of automated planning in the framework of dynamic epistemic logic. In recent works, the epistemic planning problem has been proved to be undecidable when preconditions of events can be epistemic formulas of arbitrary complexity, and in particular arbitrary modal depth. It is known however that when preconditions are propositional (and there are no postconditions), the problem is between PSPACE and EXPSPACE. In this work we bring two new pieces to the picture. First, we prove that the epistemic planning problem with propositional preconditions and without postconditions is in PSPACE, and is thus PSPACE-complete. Second, we prove that very simple epistemic preconditions are enough to make the epistemic planning problem undecidable: preconditions of modal depth at most two suffice.

	no postconditions	with postconditions
$d = 0$	PSPACE-complete	Decidable
$d \leq 1$	?	Undecidable
$d \leq 2$	Undecidable	Undecidable $\Downarrow$
unbounded	Undecidable $\not\Rightarrow$	Undecidable $\Downarrow$

Table 1: Overview (d: modal depth; gray: this paper).

1 Introduction

A key objective in artificial intelligence is to develop autonomous agents able to plan their actions towards achieving their goals, and to reason about their own and other agents' knowledge. Planning, which consists in finding a sequence of actions to reach a given objective from an initial situation, is a central research domain in artificial intelligence. Concerning reasoning about knowledge, dynamic epistemic logic (DEL) is now recognised as a very promising framework [van Ditmarsch *et al.*, 2007]. Recently, planning and dynamic epistemic logic have been combined in the so-called epistemic planning problem [Bolander and Andersen, 2011].

In DEL, events can deal with high-order reasoning. For instance we may model the following event:

> "Anne receives a letter revealing that φ is true and Bob knows that Anne receives the truth value of φ but Anne is unsure whether Bob knows that or not."

In DEL, φ is called a *precondition*: φ needs to be true for this event to occur, and therefore its occurence brings the information that φ is true. This event is purely informative, but DEL also allows physical (ontic) effects on the world; these are referred to as *postconditions*. One natural question is: how does the nesting of knowledge in pre- and postconditions impact the complexity of the epistemic planning problem?

On the one hand, when only propositional preconditions are used, such as φ = "Bob is married", the problem is decidable if postconditions are also propositional [Yu *et al.*, 2013]. In this case it is in k-EXPTIME, where k is the maximal modal depth of goal formulas [Aucher *et al.*, 2014]; if there are no postconditions (events are purely epistemic), the problem is in EXPSPACE [Bolander *et al.*, 2015].

On the other hand, epistemic preconditions such as φ = "Bob considers it possible that Anne knows that Bob does not know that it is raining" yield undecidability: if propositional postconditions are allowed, then the problem is already undecidable with preconditions of modal depth one [Bolander and Andersen, 2011]. It is also known to be undecidable without postconditions, if we allow for preconditions of unbounded modal depth [Aucher and Bolander, 2013]. See Table 1 for a summary of results about epistemic planning.

In this paper, our contribution is twofold:

1. With propositional preconditions and no postconditions, epistemic planning is in PSPACE (Theorem 1). The key point is that in this case events commute [Löwe *et al.*, 2011]. This allows for a succinct representation of tuples of events, and we build upon a model checking procedure from [Aucher and Schwarzentruber, 2013] to devise a polynomial space decision procedure.

2. Epistemic planning without postconditions is already undecidable with preconditions of modal depth two (Theorem 2). The proof, by reduction from the halting problem for two counter machines, refines the one given in [Aucher and Bolander, 2013], which requires preconditions with unbounded modal depth. By designing more involved gadgets to code the configurations and instructions of the machines, we manage to bound the modal depth of preconditions.

We first recall the background on epistemic planning in Section 2. We establish our two contributions, described above, in Section 3 and Section 4 respectively. We briefly discuss future work in Section 5.

2 Background on epistemic planning

In this section, we recall the necessary background about dynamic epistemic logic and epistemic planning.

2.1 Dynamic epistemic logic

Let AP be a countably infinite set of *atomic propositions*, and let $Ag = \{1, \ldots, n\}$ be a finite set of *agents*. The epistemic language $\mathcal{L}_{\mathbf{EL}}$ is the language of propositional logic extended with one knowledge modality for each agent. Intuitively, $K_a\varphi$ reads as "agent a knows that φ holds". The syntax of $\mathcal{L}_{\mathbf{EL}}$ is given by the following grammar:

$$\varphi ::= p \mid \neg\varphi \mid (\varphi \vee \varphi) \mid K_a\varphi, \text{ where } p \in AP \text{ and } a \in Ag.$$

The semantics of $\mathcal{L}_{\mathbf{EL}}$ is given in terms of *epistemic models* that represent how the agents perceive the world.

Definition 1 *An* epistemic model $\mathcal{M} = (W, \{R_a\}_{a\in Ag}, V)$ *is a tuple where:*

- W *is a non-empty finite set of possible* worlds*,*
- $R_a \subseteq W \times W$ *is an* accessibility relation *for agent* a*,*
- $V : AP \to 2^W$ *is a* valuation function*.*

We write $w \in \mathcal{M}$ for $w \in W$, $|\mathcal{M}|$ for $|W|$, and $(\mathcal{M}, w)$ is called a *pointed epistemic model*. The intended meaning of wR_aw' is that in world w agent a considers that w' might be the actual world.

The semantics of $\mathcal{L}_{\mathbf{EL}}$ is defined as follows:

- $\mathcal{M}, w \models p$ if $w \in V(p)$;
- $\mathcal{M}, w \models \neg\varphi$ if it is not the case that $\mathcal{M}, w \models \varphi$;
- $\mathcal{M}, w \models (\varphi \vee \psi)$ if $\mathcal{M}, w \models \varphi$ or $\mathcal{M}, w \models \psi$;
- $\mathcal{M}, w \models K_a\varphi$ if for all w' s.t. wR_aw', $\mathcal{M}, w' \models \varphi$.

Dynamic epistemic logic (DEL) extends epistemic logic with modalities that represent the occurrence of events. In DEL events are represented by *event models*, defined below. In general DEL events can bring information and modify the world, and such events are called *ontic events* [van Ditmarsch and Kooi, 2006]; in this work however we focus on purely informative events, called *epistemic events* [Baltag *et al.*, 1998].

Definition 2 *An* event model $\mathcal{E} = (\mathsf{E}, \{\to_a\}_{a\in Ag}, pre)$ *is a tuple where:*

- E *is a non-empty finite set of possible* events*,*
- $\to_a \subseteq \mathsf{E} \times \mathsf{E}$ *is an* accessibility relation *on* E *for agent* a*,*
- $pre : \mathsf{E} \to \mathcal{L}_{\mathbf{EL}}$ *is a* precondition function*.*

We write $e \in \mathcal{E}$ for $e \in \mathsf{E}$, $|\mathcal{E}|$ for $|\mathsf{E}|$, and $(\mathcal{E}, e)$ is called a *pointed event model*, where e represents the actual event of $(\mathcal{E}, e)$. An event e can occur in a world w of an epistemic model $\mathcal{M}$ if, and only if, its precondition is verified, *i.e.* $\mathcal{M}, w \models pre(e)$, which leads to the following definition:

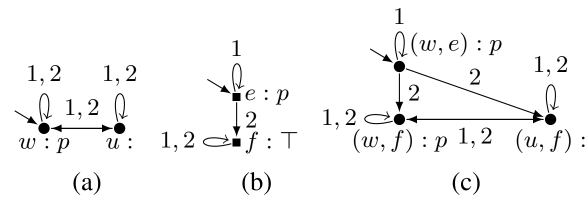

Figure 1: Example of a product.

Definition 3 *Given* $\mathcal{M} = (W, \{R_a\}_{a\in Ag}, V)$ *an epistemic model and* $\mathcal{E} = (\mathsf{E}, \{\to_a\}_{a\in Ag}, pre)$ *an event model, the* update product *of* $\mathcal{M}$ *and* $\mathcal{E}$ *is the epistemic model* $\mathcal{M} \otimes \mathcal{E} = (W^{\otimes}, \{R_a^{\otimes}\}_{a\in Ag}, V^{\otimes})$ *where:*

$$W^{\otimes} = \{(w, e) \in W \times \mathsf{E} \mid \mathcal{M}, w \models pre(e)\},$$
$$R_a^{\otimes}(w, e) = \{(w', e') \in W^{\otimes} \mid wR_aw' \text{ and } e \to_a e'\},$$
$$V^{\otimes}(p) = \{(w, e) \in W^{\otimes} \mid \mathcal{M}, w \models p\}$$

The product of a pointed epistemic model $(\mathcal{M}, w)$ with a pointed event model $(\mathcal{E}, e)$ is defined as $(\mathcal{M}, w) \otimes (\mathcal{E}, e) := (\mathcal{M} \otimes \mathcal{E}, (w, e))$ if $\mathcal{M}, w \models pre(e)$, otherwise it is undefined.

We can now define the syntax and semantics of DEL. The syntax is given by the following grammar:

$$\varphi ::= p \mid \neg\varphi \mid (\varphi \vee \varphi) \mid K_i\varphi \mid \langle\mathcal{E}, e\rangle\varphi,$$

where $p \in AP$, $i \in Ag$ and $(\mathcal{E}, e)$ is a pointed event model.

The semantics is the same as for $\mathcal{L}_{\mathbf{EL}}$, with the following additional case:

- $\mathcal{M}, w \models \langle\mathcal{E}, e\rangle\varphi$ if $\mathcal{M}, w \models pre(e)$, and $(\mathcal{M}, w) \otimes (\mathcal{E}, e) \models \varphi$.

Example 1 *Consider the pointed epistemic model* $(\mathcal{M}, w)$ *in Figure 1(a). Proposition* p *is true in the actual world* w *but both agents* a, b *do not know that* p *holds:* $\mathcal{M}, w \models \neg K_1p \wedge \neg K_2p$*. Figure 1(b) shows a pointed event model* $(\mathcal{E}, e)$ *where the precondition of the actual event* e *is* p*, and the one of event* f *is* $\top$*.* $(\mathcal{E}, e)$ *represents the event where agent* 1 *learns that* p *is true while agent 2 believes that nothing happens. Figure 1(c) shows the product* $(\mathcal{M}, w) \otimes (\mathcal{E}, e)$*, which represents the situation after event* $(\mathcal{E}, e)$*. Observe that agent 1 knows* p *and agent 2 does not.*

Remark 1 *We do not make any assumption on the nature of the accessibility relations in epistemic and event models.*

2.2 Epistemic planning

Let $\mathcal{C}$ be a class of pointed event models. The epistemic planning problem restricted to $\mathcal{C}$ is the following:

Definition 4 (Epistemic planning problem)

Input*: a pointed epistemic model* $(\mathcal{M}, w)$*, a finite set of pointed event models* $\mathbb{E} \subseteq \mathcal{C}$*, and an epistemic goal formula* φ_g*;*
Output*: yes if there exists a sequence of pointed event models* $(\mathcal{E}_1, e_1), \ldots, (\mathcal{E}_p, e_p) \in \mathbb{E}$ *(a* plan*) such that* $\mathcal{M}, w \models \langle\mathcal{E}_1, e_1\rangle \ldots \langle\mathcal{E}_p, e_p\rangle\varphi_g$*; no otherwise.*

We now establish the precise complexity of this problem for propositional event models.

3 Propositional preconditions

Let $\mathcal{C}_0$ be the class of (pointed) epistemic event models where preconditions are propositional formulas. For instance, the pointed event model depicted in Figure 1(c) is in $\mathcal{C}_0$ since both p and $\top$ are propositional formulas. The epistemic planning restricted to $\mathcal{C}_0$ is known to be PSPACE-hard [Bolander *et al.*, 2015]. We establish that it is actually PSPACE-complete.

As pointed out in [Löwe *et al.*, 2011], epistemic event models with propositional preconditions commute. Formally:

Lemma 1 *For all pointed epistemic models* $(\mathcal{M}, w)$*, for all pointed event models* $(\mathcal{E}_1, e_1)$ *and* $(\mathcal{E}_2, e_2)$ *in* $\mathcal{C}_0$*,* $\mathcal{M} \otimes \mathcal{E}_1 \otimes \mathcal{E}_2, (w, e_1, e_2)$ *exists iff* $\mathcal{M} \otimes \mathcal{E}_2 \otimes \mathcal{E}_1, (w, e_2, e_1)$ *exists, and in that case they are bisimilar.*

As a consequence, in the rest of the section, the order in which events are applied in an initial world is indifferent. Only the number of times each event occurs is relevant, and the proof of our result heavily relies on this property.

We first establish a preliminary result on the model checking problem for a dedicated language: we extend the dynamic epistemic language with iterations of event models in $\mathcal{C}_0$, that is, constructions of the form $\langle(\mathcal{E}, e)^\ell\rangle\psi$ where $(\mathcal{E}, e)$ is a pointed event model in $\mathcal{C}_0$ and ℓ is a positive integer. We suppose here that ℓ is written *in binary* so that this language, called $\mathcal{L}^{it}_{\mathcal{C}_0}$, is exponentially more succinct than DEL. Classically, the model checking problem for $\mathcal{L}^{it}_{\mathcal{C}_0}$ is, given a pointed epistemic model $(\mathcal{M}, w)$ and a formula $\Phi \in \mathcal{L}^{it}_{\mathcal{C}_0}$, to decide whether $\mathcal{M}, w \models \Phi$.

Proposition 1 *Model checking* $\mathcal{L}^{it}_{\mathcal{C}_0}$ *is in* PSPACE.

Proof We design a deterministic algorithm that takes as an input a pointed epistemic model $(\mathcal{M}, w_0)$ and a formula $\Phi \in \mathcal{L}^{it}_{\mathcal{C}_0}$, and decides whether $\mathcal{M}, w_0 \models \Phi$. Without loss of generality, we suppose that all event models appearing in the formula are the same, noted $\mathcal{E} = (\mathsf{E}, \rightarrow, pre)$ (if not, we replace each one by their disjoint union). Let $e_1, \ldots, e_{|\mathcal{E}|}$ be an enumeration of the possible events in $\mathcal{E}$. By Lemma 1, all permutations of events in a tuple $(w, e_{i_1}, \ldots, e_{i_p})$ are equivalent in the sense that either they all are worlds in $\mathcal{M} \otimes \mathcal{E}^p$ and they all are bisimilar, or none of them exists: only the number of times each event occurs is relevant. For a world w and a vector $\vec{n} = (n_1, \ldots, n_{|\mathcal{E}|})$, we thus let $w \bullet \vec{n}$ denote the representative permutation $(w, \underbrace{e_1, \ldots, e_1}_{n_1 \text{ times}}, \ \ldots \ , \underbrace{e_{|\mathcal{E}|}, \ldots, e_{|\mathcal{E}|}}_{n_{|\mathcal{E}|} \text{ times}})$.

Let $\mathtt{mc}$ be the algorithm given in Figure 2, and let $0^{|\mathcal{E}|}$ denote the null $|\mathcal{E}|$-vector. We claim that $\mathtt{mc}(\mathcal{M}, w_0, \mathcal{E}, 0^{|\mathcal{E}|}, \Phi)$ returns true iff $\mathcal{M}, w_0 \models \Phi$. To prove this claim we establish that for all $w \in \mathcal{M}$, all integers $n_1, \ldots, n_{|\mathcal{E}|}$ and all subformula φ of Φ, the following property $\mathcal{P}$ holds:

If $\mathcal{M} \otimes \mathcal{E}^{\sum_{i=1}^{|\mathcal{E}|} n_i}, w \bullet \vec{n}$ exists then
$\mathtt{mc}(\mathcal{M}, w, \mathcal{E}, (n_1, \ldots, n_{|\mathcal{E}|}), \varphi)$ returns true iff
$\mathcal{M} \otimes \mathcal{E}^{\sum_{i=1}^{|\mathcal{E}|} n_i}, w \bullet \vec{n} \models \varphi$.

Property $\mathcal{P}$ is proven by induction on φ. We omit the boolean cases and case $\langle(\mathcal{E}, e_i)^\ell\rangle\psi$ which are trivial.

Case $K_a\psi$: the algorithm has to check that ψ holds in all a-successors of $w \bullet \vec{n}$ in $\mathcal{M} \otimes \mathcal{E}^{\sum_{i=1}^{|\mathcal{E}|} n_i}$. Every a-successor of $w \bullet \vec{n}$ is a permutation of some $u \bullet \vec{\ell}$ and is bisimilar to it. We thus need to enumerate all worlds u and vectors $\vec{\ell}$ that represent some a-successor, and verify that ψ holds in $u \bullet \vec{\ell}$. Given a tuple $u \bullet \vec{\ell}$, to check whether it is a permutation of some a-successor of $w \bullet \vec{n}$, we first check that it is an existing world in $\mathcal{M} \otimes \mathcal{E}^{\sum_{i=1}^{|\mathcal{E}|} n_i}$. Since events are purely epistemic and propositional, preconditions of successive events can all be checked in the initial world u. This is done by calling function $\mathtt{preok}(\mathcal{M}, u, \mathcal{E}, \vec{\ell})$, which checks that for all $i \in \{1, \ldots, |\mathcal{E}|\}$, if $\ell_i > 0$ then $pre(e_i)$ is true in u. Next, we check that some permutation of $u \bullet \vec{\ell}$ is indeed a-related to $w \bullet \vec{n}$: we should first have $u \in R_a(w)$; then, it should be possible to map each occurrence of an event e_i in $w \bullet \vec{n}$ to some occurrence of some a-related event e_j in $u \bullet \vec{\ell}$ so as to form a bijection. Deciding whether such a bijection exists amounts to solving the following integer linear program: checking whether there exist positive integers $(x_{i,j})_{(i,j) \in \{1,\ldots,|\mathcal{E}|\}^2 | e_j \in R_a(e_i)}$, where $x_{i,j}$ is the number of times e_j is chosen as a-successor for e_i, such that:

$$(S) \begin{cases} n_i = \sum_{j | e_j \in R_a(e_i)} x_{i,j} & \text{for all } i \in \{1, \ldots, |\mathcal{E}|\}, \\ \ell_j = \sum_{i | e_i \in R_a(e_j)} x_{i,j} & \text{for all } j \in \{1, \ldots, |\mathcal{E}|\}. \end{cases}$$

This is done by calling $\mathtt{succ}(\mathcal{E}, a, \vec{n}, \vec{\ell})$.

Spatial complexity. The maximal number of nested calls is bounded by $|\Phi|$, so that the number of local variables to be stored is polynomial in $|\Phi|$. Next, the space used to store vector $\vec{n}$ in each call is in $O(|\Phi|^2)$ (see [Charrier *et al.*, 2016] for details). Finally, checking consistency of a system (S) can be done in non-deterministic time polynomial in the number of bits needed to encode $\vec{n}$ and $\vec{\ell}$ [Papadimitriou, 1981], and therefore in deterministic space polynomial in $|\Phi|$. ■

Theorem 1 *The epistemic planning problem restricted to* $\mathcal{C}_0$ *is in* PSPACE.

Proof We adapt the algorithm given in [Bolander *et al.*, 2015] (Theorem 5.8). First it is proved in [Sadzik, 2006] that, noting $\simeq_d$ the d-bisimulation[1] for event models (see [Bolander *et al.*, 2015; Sadzik, 2006; van Ditmarsch *et al.*, 2007]), for every $d \geq 0$, every pointed event model $(\mathcal{E}, e)$ is $\simeq_d$-stabilizing at iteration $|\mathcal{E}|^d$; formally, $(\mathcal{E}, e_i)^k \simeq_d (\mathcal{E}, e_i)^{k+1}$ for all $k \geq |\mathcal{E}|^d$.[2] Secondly, by Lemma 1, event models with propositional preconditions commute. Therefore, the following algorithm correctly solves the epistemic planning problem for event models with propositional preconditions:

Given input $\langle(\mathcal{M}, w), \{(\mathcal{E}_1, e_1), \ldots, (\mathcal{E}_m, e_m)\}, \varphi_g\rangle$:

1. Compute d, the modal depth of the goal formula φ_g;
2. For each $i \in \{1, \ldots, m\}$, non-deterministically guess $n_i \in \{0, \ldots, |\mathcal{E}_i|^d\}$;
3. Accept if $\mathcal{M}, w \models \langle(\mathcal{E}_1, e_1)^{n_1}\rangle \ldots \langle(\mathcal{E}_m, e_m)^{n_m}\rangle\varphi_g$.

This algorithm is non-deterministic. The first step is clearly performed in space polynomial in the size of the input. Concerning the second point, each n_i can be exponential in d

[1] Bisimulation up to modal depth d.

[2] Actually a better bound is proved in [Bolander *et al.*, 2015].

function $\texttt{mc}(\mathcal{M}, w, \mathcal{E}, \vec{n}, \varphi)$
  match φ
    case p: **return** (p is true in w)
    case $\neg\psi$: **return** not $\texttt{mc}(\mathcal{M}, w, \mathcal{E}, \vec{n}, \psi)$
    case $(\psi_1 \vee \psi_2)$:
      return $\texttt{mc}(\mathcal{M}, w, \mathcal{E}, \vec{n}, \psi_1)$ or $\texttt{mc}(\mathcal{M}, w, \mathcal{E}, \vec{n}, \psi_2)$
    case $K_a\psi$:
      for $u \in R_a(w), \vec{\ell} \in \mathbb{N}^{|\mathcal{E}|}$ s.t. $\sum_{i=1}^{|\mathcal{E}|} \ell_i = \sum_{i=1}^{|\mathcal{E}|} n_i$
        if $\left(\begin{array}{l} \texttt{preok}(\mathcal{M}, u, \mathcal{E}, \vec{\ell}) \\ \text{and } \texttt{succ}(\mathcal{E}, a, \vec{n}, \vec{\ell}) \\ \text{and not } \texttt{mc}(\mathcal{M}, u, \mathcal{E}, \vec{\ell}, \psi) \end{array} \right)$ **then**
          return false
      return true
    case $\langle(\mathcal{E}, e_i)^\ell\rangle\psi$:
      if $pre(e_i)$ is false in w **then**
        return false
      return $\texttt{mc}(\mathcal{M}, w, \mathcal{E}, (n_1, .., n_{i-1}, n_i + \ell, n_{i+1}, .., n_k), \psi)$

Figure 2: Algorithm $\texttt{mc}$ for model checking $\mathcal{L}^{it}_{\mathcal{C}_0}$.

and thus in $|\varphi_g|$, but its binary representation uses polynomial space. Since $\langle(\mathcal{E}_1, e_1)^{n_1}\rangle \dots \langle(\mathcal{E}_m, e_m)^{n_m}\rangle\varphi_g$ is an $\mathcal{L}^{it}_{\mathcal{C}_0}$ formula, it follows from Proposition 1 that the last step can also be performed in polynomial space. The epistemic planning problem restricted to $\mathcal{C}_0$ is therefore in NPSPACE and thus in PSPACE by Savitch's theorem [Savitch, 1970]. ■

We now turn to the case of modal preconditions with bounded modal depth.

4 Preconditions of bounded modal depth

Let $\mathcal{C}_2$ be the class of event models with preconditions of modal depth at most two. In this section, we prove the following theorem by refining the reduction given in [Aucher and Bolander, 2013].

Theorem 2 *The epistemic planning problem restricted to $\mathcal{C}_2$ is undecidable.*

We first recall the halting problem for two-counter machines, known to be undecidable [Minsky, 1967], and then we reduce it to the epistemic planning problem restricted to $\mathcal{C}_2$.

4.1 Two-counter machines

We present two-counter machines as introduced in [Minsky, 1967].

Definition 5 *A two-counter machine M is a sequence of instructions $(I_0, \dots, I_N)$ where*

- *For each $\ell < N$, I_ℓ is either $inc(i)$, $goto(\ell')$ or $gotocond(i, \ell')$, with $i \in \{1, 2\}$, $\ell' \le N$ and $\ell \neq \ell'$;*
- *$I_N = halt$.*

We call program line *a pair $k{:}I_k$.*

Example 2 *The four program lines shown on the right define a two-counter machine M_{ex}.*

0:inc(1)
1:gotocond(1, 3)
2:goto(0)
3:halt

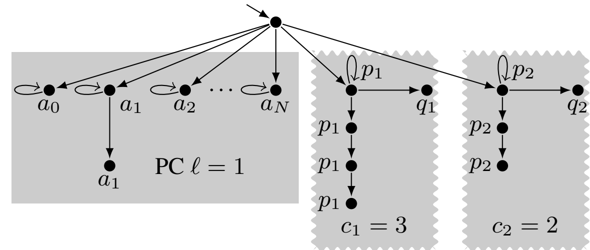

Figure 3: Pointed epistemic model $(\mathcal{M}, w)_{(1,3,2)}$.

A configuration of a two-counter machine M is a triple (ℓ, c_1, c_2) where $\ell \in \{0, \dots, N\}$ is the program counter and $c_1, c_2 \in \mathbb{N}$ are the two data counters.
Let $C_M = \{0, \dots, N\} \times \mathbb{N} \times \mathbb{N}$ be the set of all possible configurations.

The transition function $\to_M$ on C_M is defined as follows. For all $(\ell, c_1, c_2) \in C_M$:

- If $I_\ell = inc(1)$, $(\ell, c_1, c_2) \to_M (\ell + 1, c_1 + 1, c_2)$;
- If $I_\ell = inc(2)$, $(\ell, c_1, c_2) \to_M (\ell + 1, c_1, c_2 + 1)$;
- If $I_\ell = goto(\ell')$, $(\ell, c_1, c_2) \to_M (\ell', c_1, c_2)$;
- If $I_\ell = gotocond(1, \ell')$,
$$(\ell, c_1, c_2) \to_M \begin{cases} (\ell', 0, c_2) & \text{if } c_1 = 0; \\ (\ell + 1, c_1 - 1, c_2) & \text{otherwise;} \end{cases}$$
- If $I_\ell = gotocond(2, \ell')$,
$$(\ell, c_1, c_2) \to_M \begin{cases} (\ell', c_1, 0) & \text{if } c_2 = 0; \\ (\ell + 1, c_1, c_2 - 1) & \text{otherwise.} \end{cases}$$

A two-counter machine M *halts* if there exist c_1, c_2 such that $(0, 0, 0) \to^*_M (N, c_1, c_2)$, where $\to^*_M$ denotes the reflexive transitive closure of $\to_M$. For instance, the machine M_{ex} given in Example 2 above does not halt. The halting problem for two-counter machines consists in deciding, given a two-counter machine, whether it halts or not. This problem is well known to be undecidable [Minsky, 1967].

4.2 The reduction

We define an effective reduction tr that, given a two-counter machine M, computes an instance $tr(M)$ of the epistemic planning problem restricted to $\mathcal{C}_2$. We fix M and the rest of the section is devoted to defining $tr(M) = \langle(\mathcal{M}_0, w_0); \mathbb{E}; \varphi_g\rangle$ and justifying its correctness (Proposition 2). As in [Aucher and Bolander, 2013], we only use one agent a ($\Box$ stands for K_a and $\Diamond$ stands for $\neg K_a \neg$), configurations of M are represented by pointed epistemic models, and the initial pointed epistemic model represents the initial configuration $(0, 0, 0)$. Each program line $\ell{:}I_\ell$ is represented by one or two pointed event model(s), such that a plan corresponds to a sequence of program lines. The goal formula expresses that the final pointed epistemic model represents a halting configuration.

Pointed epistemic models

Let (ℓ, c_1, c_2) be a configuration of M. We describe the pointed epistemic model $(\mathcal{M}_{(\ell,c_1,c_2)}, w_{(\ell,c_1,c_2)})$ (shortened

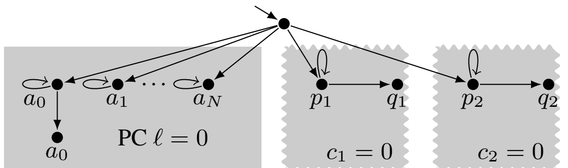

Figure 4: Pointed epistemic model representing the initial configuration $(0, 0, 0)$.

as $(\mathcal{M}, w)_{(\ell,c_1,c_2)}$) that represents (ℓ, c_1, c_2). For instance, Figure 3 shows $(\mathcal{M}, w)_{(1,3,2)}$. It is a tree-like structure rooted at $w_{(\ell,c_1,c_2)}$. In each world except the root, there is exactly one true atomic proposition, and we call p-world any world where p holds. The root $w_{(\ell,c_1,c_2)}$ verifies no atomic proposition, and it has three groups of children, one for each counter:

Program counter. For each program line $\ell' : I_{\ell'}$, $w_{(\ell,c_1,c_2)}$ has one reflexive child labeled by proposition $a_{\ell'}$. The a_ℓ-child of $w_{(\ell,c_1,c_2)}$ has a child also labeled by a_ℓ, without any outgoing edge: we say that there is an *a_ℓ-strip*.

Data counter** c_i**. For each $i \in \{1, 2\}$, $w_{(\ell,c_1,c_2)}$ has a reflexive p_i-child that has an irreflexive q_i-child, and is followed by a chain of irreflexive p_i-worlds of length c_i.

We now define the first component of $tr(M)$: $(\mathcal{M}_0, w_0) := (\mathcal{M}, w)_{(0,0,0)}$, depicted in Figure 4. We call *configuration model* a pointed epistemic model of the form $(\mathcal{M}, w)_{(\ell,c_1,c_2)}$.

Pointed event models

For each program line $\ell{:}I_\ell$ of M where I_ℓ is of the form $\mathit{goto}(\ell')$ or $\mathit{inc}(i)$, we define a pointed event model $(\mathcal{E}_{\ell:I_\ell}, e_{\ell:I_\ell})$ (shortened as $(\mathcal{E}, e)_{\ell:I_\ell}$) that mimics the semantics of $\ell{:}I_\ell$ (Figures 6 and 8). For each program line $\ell{:}\mathit{gotocond}(i, \ell')$ of M, we define *two* pointed event models $(\mathcal{E}, e)_{\ell:\mathit{gotocond}(i,\ell')}$ and $(\mathcal{E}^{>0}, e^{>0})_{\ell:\mathit{gotocond}(i,\ell')}$, respectively for the case $c_i = 0$ and $c_i > 0$ (Figure 9). These pointed event models form the second component of $tr(M)$:

$$\mathbb{E} := \{(\mathcal{E}, e)_{\ell:I_\ell} \mid \ell < N\} \cup \{(\mathcal{E}^{>0}, e^{>0})_{\ell:I_\ell} \mid \ell < N \text{ and } I_\ell = \mathit{gotocond}(i, \ell')\}.$$

In model $(\mathcal{M}, w)_{(\ell,c_1,c_2)}$ where $\ell < N$, the only pointed event model of $\mathbb{E}$ that should be applied is the one representing the behavior of program line $\ell{:}I_\ell$ in configuration (ℓ, c_1, c_2). This event model is defined as follows:

$$\mathbb{E}(\ell, c_1, c_2) := \begin{cases} (\mathcal{E}^{>0}, e^{>0})_{\ell:I_\ell} & \text{if } I_\ell = \mathit{gotocond}(i, \ell') \\ & \text{and } c_i > 0, \\ (\mathcal{E}, e)_{\ell:I_\ell} & \text{otherwise.} \end{cases}$$

The product with any other event model from $\mathbb{E}$ results in a model that is not *valid* according to the following definition:

Definition 6 *A pointed epistemic model* $(\mathcal{M}, w)$ *is* valid *if* w *has an* a_ℓ*-child for each* $\ell \in \{0, \dots, N\}$ *and a* p_i*-child for each* $i \in \{1, 2\}$.

Note that by definition, every configuration model is valid. Further down, we will define event models of $\mathbb{E}$ such that:

Lemma 2 *For every configuration* (ℓ, c_1, c_2)*, it holds that*

1. $(\mathcal{M}, w)_{(\ell,c_1,c_2)} \otimes \mathbb{E}(\ell, c_1, c_2)$ *is isomorphic*[3] *to* $(\mathcal{M}, w)_{(\ell',c_1',c_2')}$*, where* $(\ell, c_1, c_2) \to_M (\ell', c_1', c_2')$.
2. *The product of* $(\mathcal{M}, w)_{(\ell,c_1,c_2)}$ *with any other event model from* $\mathbb{E}$ *is defined but not valid.*
3. *For any set of event models* $(\mathbb{E}_1, \dots, \mathbb{E}_n)$ *and any event model* $\mathbb{E}_{n+1}$*, if* $(\mathcal{M}, w)_{(\ell,c_1,c_2)} \otimes \mathbb{E}_1 \otimes \cdots \otimes \mathbb{E}_n$ *is not valid,* $(\mathcal{M}, w)_{(\ell,c_1,c_2)} \otimes \mathbb{E}_1 \otimes \cdots \otimes \mathbb{E}_n \otimes \mathbb{E}_{n+1}$ *is also not valid.*

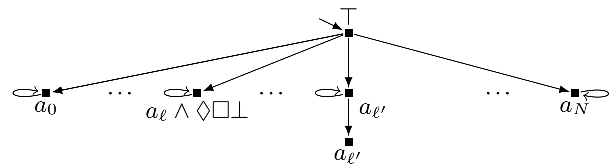

Figure 5: Event model portion $\mathit{repl}(\ell, \ell')$ for $\ell \neq \ell'$.

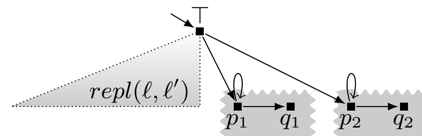

Figure 6: Event model for $\ell{:}\mathit{goto}(\ell')$.

We now describe the event models in $\mathbb{E}$ and at the same time we prove Lemma 2. Each of these models has three groups (from left to right on Figures 6, 8, 9), that update respectively the program counter group, the data counter c_1 group and the data counter c_2 group of configuration models.

Event model for $\ell : \mathit{goto}(\ell')$**.** The pointed event model $(\mathcal{E}, e)_{\ell:\mathit{goto}(\ell')}$, that mimics the effect of $\ell{:}\mathit{goto}(\ell')$, is depicted in Figure 6. Portion $\mathit{repl}(\ell, \ell')$ concerns the program counter group and is described in Figure 5. The two other groups leave the data counter groups c_1 and c_2 unchanged.

- The product $(\mathcal{M}, w)_{(\ell,c_1,c_2)} \otimes (\mathcal{E}, e)_{\ell:\mathit{goto}(\ell')}$ is isomorphic to $(\mathcal{M}, w)_{(\ell',c_1,c_2)}$: indeed, portion $\mathit{repl}(\ell, \ell')$ removes the a_ℓ-strip and adds an $a_{\ell'}$-strip in the program counter group (recall that $\ell \neq \ell'$).
- The product $(\mathcal{M}, w)_{(\ell'',c_1,c_2)} \otimes (\mathcal{E}, e)_{\ell:\mathit{goto}(\ell')}$ with $\ell'' \neq \ell$ is not valid. Indeed, as $(\mathcal{M}, w)_{(\ell'',c_1,c_2)}$ does not have an a_ℓ-strip in its program counter group, its a_ℓ-world violates precondition $a_\ell \wedge \Diamond\Box\bot$ in portion $\mathit{repl}(\ell, \ell')$. As a consequence, $(\mathcal{M}, w)_{(\ell'',c_1,c_2)} \otimes (\mathcal{E}, e)_{\ell:\mathit{goto}(\ell')}$ has no a_ℓ-child at its root and is thus not valid.

Event model for $\ell : \mathit{inc}(i)$**.** Figure 8 shows $(\mathcal{E}, e)_{\ell:\mathit{inc}(1)}$ that mimics the effect of $\ell : \mathit{inc}(1)$ (for $\mathit{inc}(2)$, the construction is symmetric). Portion $\mathit{repl}(\ell, \ell + 1)$ is meant to increment the program counter. Portion $\mathit{lengthen}(1)$ (described

[3] More precisely, the reachable parts of the pointed epistemic models are isomorphic.

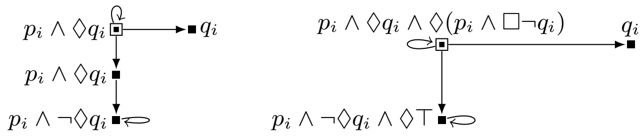

Figure 7: Event model portions $lengthen(i)$ and $shorten(i)$.

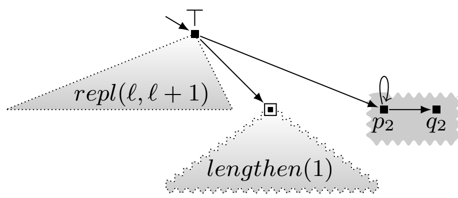

Figure 8: Event model for $\ell{:}inc(1)$.

in Figure 7) is meant to increment the data counter c_1. The intermediate event of precondition $p_1 \wedge \Diamond q_1$ duplicates once the p_1-child of the root: it adds one p_1-world at the start of the p_1-chain. The last group leaves data counter c_2 unchanged.

- The product $(\mathcal{M}, w)_{(\ell,c_1,c_2)} \otimes (\mathcal{E}, e)_{\ell:inc(1)}$ is isomorphic to $(\mathcal{M}, w)_{(\ell+1,c_1+1,c_2)}$.
- For the same reason as for $(\mathcal{E}, e)_{\ell:goto(\ell')}$, the product $(\mathcal{M}, w)_{(\ell'',c_1,c_2)} \otimes (\mathcal{E}, e)_{\ell:inc(1)}$ with $\ell'' \neq \ell$ is not valid.

Event models for ℓ : *gotocond*(i, ℓ')**.** Figure 9 describes models $(\mathcal{E}, e)_{\ell:gotocond(1,\ell')}$ and $(\mathcal{E}^{>0}, e^{>0})_{\ell:gotocond(1,\ell')}$. They mimic the effect of ℓ : *gotocond*$(1, \ell')$ in case $c_1 = 0$ and case $c_1 > 0$, respectively (for ℓ : *gotocond*$(2, \ell')$, constructions are symmetric).

- $(\mathcal{M}, w)_{(\ell,0,c_2)} \otimes (\mathcal{E}, e)_{\ell:gotocond(1,\ell')}$ is isomorphic to $(\mathcal{M}, w)_{(\ell',0,c_2)}$: indeed, precondition $\neg\Diamond(p_1 \wedge \neg\Diamond q_1)$ checks that the p_1-chain in the data counter c_1 group is of length 0. Here it is the case, so that the data counter group c_1 remains unchanged. However, when $c_1 > 0$, the p_1-child of the root of $(\mathcal{M}, w)_{(\ell,c_1,c_2)}$ violates this precondition. It is thus removed, so that the product $(\mathcal{M}, w)_{(\ell,c_1,c_2)} \otimes (\mathcal{E}, e)_{\ell:gotocond(1,\ell')}$ is not valid.
- $(\mathcal{M}, w)_{(\ell,c_1,c_2)} \otimes (\mathcal{E}^{>0}, e^{>0})_{\ell:3gotocond(1,\ell')}$ with $c_1 > 0$ is isomorphic to $(\mathcal{M}, w)_{(\ell+1,c_1-1,c_2)}$. Indeed, portion $shorten(1)$ (Figure 7) is meant to decrement data counter c_1 by one: precondition $p_1 \wedge \neg\Diamond q_1 \wedge \Diamond\top$ checks that we are in the p_1-chain (p_1), but not at the start ($\neg\Diamond q_1$) nor the end ($\Diamond\top$) of the chain. The last world of the p_1-chain is thus removed when $c_1 > 0$. When $c_1 = 0$, precondition $\Diamond(p_i \wedge \Box\neg q_i)$ is violated by the p_1-child of the root of $(\mathcal{M}, w)_{(\ell,0,c_2)}$: indeed, this precondition checks that the length of the p_1-chain is at least 1. The product $(\mathcal{M}, w)_{(\ell,0,c_2)} \otimes (\mathcal{E}^{>0}, e^{>0})_{\ell:gotocond(1,\ell')}$ is thus not valid.
- For $\ell'' \neq \ell$, $(\mathcal{M}, w)_{(\ell'',c_1,c_2)} \otimes (\mathcal{E}, e)_{\ell:gotocond(1,\ell')}$ and $(\mathcal{M}, w)_{(\ell'',c_1,c_2)} \otimes (\mathcal{E}^{>0}, e^{>0})_{\ell:gotocond(1,\ell')}$ are not valid.

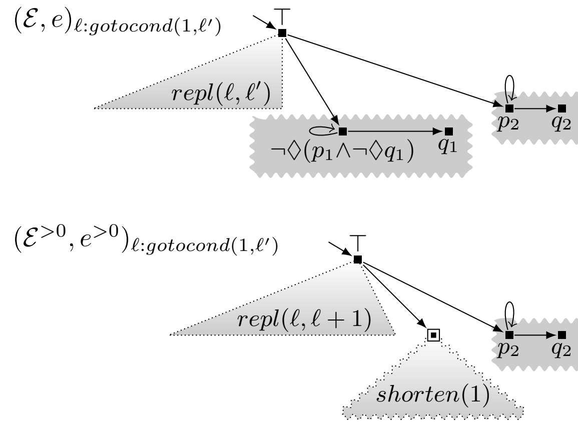

Figure 9: Event models for $\ell{:}gotocond(1, \ell')$.

Note that the product of a non-valid pointed epistemic model with any pointed event model is not valid since no event model can create the missing children of the root without using postconditions.

Goal formula
The goal formula φ_g in $tr(M)$ is $\varphi_{valid} \wedge \varphi_{halt}$, where:

- $\varphi_{valid} := \bigwedge_{\ell=0}^{N} \Diamond a_\ell \wedge \Diamond p_1 \wedge \Diamond p_2$, and
- $\varphi_{halt} := \Diamond(a_N \wedge \Diamond\Box\bot)$.

Proposition 2 *M halts iff there is a plan for $tr(M)$.*

The proof can be found in [Charrier *et al.*, 2016].

4.3 Comparison

In [Aucher and Bolander, 2013] the program counter as well as the data counters are represented with chains of worlds, and incrementation, decrementation and replacement of a value by another one are implemented on such chains. While the first two operations can be performed with preconditions of modal depth two, $repl(\ell, \ell')$ requires unbounded nesting in general to be implemented on chains. We observed that unlike data counters, the program counter is *bounded* so that we can avoid chains for its representation, and provide an alternative gadget for $repl(\ell, \ell')$ that only uses preconditions of modal depth two.

5 Future work

The natural continuation is to complete Table 1. First, is the epistemic planning problem decidable for preconditions of modal depth one and no postconditions, or do modalities in preconditions immediately bring about undecidability? Second, what is the exact complexity of the problem with propositional pre- and postconditions? It is known to be decidable [Yu *et al.*, 2013], with a non-elementary upper bound [Aucher *et al.*, 2014] and a PSPACE lower bound [Bolander *et al.*, 2015]; this a big gap that should be bridged.

References

[Aucher and Bolander, 2013] Guillaume Aucher and Thomas Bolander. Undecidability in epistemic planning.

In *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013*, 2013.

[Aucher and Schwarzentruber, 2013] Guillaume Aucher and François Schwarzentruber. On the complexity of dynamic epistemic logic. *TARK 2013*, abs/1310.6406, 2013.

[Aucher *et al.*, 2014] Guillaume Aucher, Bastien Maubert, and Sophie Pinchinat. Automata techniques for epistemic protocol synthesis. In *SR*, pages 97–103, 2014.

[Baltag *et al.*, 1998] Alexandru Baltag, Lawrence S Moss, and Slawomir Solecki. The logic of public announcements, common knowledge, and private suspicions. In *Proceedings of the 7th conference on Theoretical aspects of rationality and knowledge*, pages 43–56. Morgan Kaufmann Publishers Inc., 1998.

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

[Bolander *et al.*, 2015] Thomas Bolander, Martin Holm Jensen, and François Schwarzentruber. Complexity results in epistemic planning. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, pages 2791–2797, 2015.

[Charrier *et al.*, 2016] Tristan Charrier, Bastien Maubert, and François Schwarzentruber. On the Impact of Modal Depth in Epistemic Planning (Extended version). To appear, IRISA, 2016.

[Löwe *et al.*, 2011] Benedikt Löwe, Eric Pacuit, and Andreas Witzel. DEL planning and some tractable cases. In *Logic, Rationality, and Interaction - Third International Workshop, LORI 2011, Guangzhou, China, October 10-13, 2011. Proceedings*, pages 179–192, 2011.

[Minsky, 1967] Marvin L Minsky. *Computation: finite and infinite machines*. Prentice-Hall, Inc., 1967.

[Papadimitriou, 1981] Christos H. Papadimitriou. On the complexity of integer programming. *J. ACM*, 28(4):765–768, 1981.

[Sadzik, 2006] Tomasz Sadzik. Exploring the Iterated Update Universe. ILLC Publications PP-2006-26, 2006.

[Savitch, 1970] Walter J. Savitch. Relationships between nondeterministic and deterministic tape complexities. *J. Comput. Syst. Sci.*, 4(2):177–192, 1970.

[van Ditmarsch and Kooi, 2006] Hans van Ditmarsch and Barteld Kooi. Semantic results for ontic and epistemic change. *Logic and the Foundations of Game and Decision Theory (LOFT 7)*, page 87, 2006.

[van Ditmarsch *et al.*, 2007] Hans van Ditmarsch, Wiebe van der Hoek, and Barteld Pieter Kooi. *Dynamic epistemic logic*, volume 337. Springer Science & Business Media, 2007.

[Yu *et al.*, 2013] Quan Yu, Ximing Wen, and Yongmei Liu. Multi-agent epistemic explanatory diagnosis via reasoning about actions. In Francesca Rossi, editor, *IJCAI*. IJCAI/AAAI, 2013.