Undecidability in Epistemic Planning

Thomas Bolander, DTU Compute, Tech Univ of Denmark
Joint work with: Guillaume Aucher, Univ Rennes 1
Introduction

This talk is based on [Aucher & Bolander, IJCAI 2013]. Our paper in a nutshell:

**What we have shown:** Undecidability of planning when allowing (arbitrary levels of) higher-order reasoning (*epistemic planning*). Higher-order reasoning here means reasoning about the beliefs of yourself and other agents (and nesting of such).

**How we have shown it:** Reduction of the halting problem for two-counter machines.

**Structure of talk:**
1. Motivation.
2. Introducing the basics: planning + logic + two-counter machines.
3. Sketching the proof: How to encode two-counter machines as epistemic planning problems.
4. Summary and related work.
Automated planning (or, simply, planning):

- Given is a planning task: initial state + goal formula + finite set of actions.
- Aim is to compute a solution: sequence of actions that leads from the initial state to a state satisfying the goal formula.

Example.
Goal: On(A,B) \land On(B,C).

In automated planning, such a graph is called a state space (induced by a planning task).
Why higher-order reasoning in planning?

initial state

? 

Tuesday, December 3rd
19.30 Workshop Dinner

goal

For more motivation for higher-order reasoning in planning, see my talk at the workshop on False-belief tasks and logic at ILLC on Thursday. 
http://jakubszymanik.com/false-belief/
Why higher-order reasoning in planning?

initial state

? goal

Tuesday, December 3rd
19.30 Workshop Dinner

For more motivation for higher-order reasoning in planning, see my talk at the workshop on False-belief tasks and logic at ILLC on Thursday.

Null

http://jakubszymanik.com/false-belief/
Our framework for planning with higher-order reasoning

In classical planning states are models of propositional logic. Classical planning only deals with planning domains that are deterministic, static, fully observable and single-agent.
Our framework for planning with higher-order reasoning

In classical planning states are models of propositional logic. Classical planning only deals with planning domains that are deterministic, static, fully observable and single-agent.

Our planning framework, epistemic planning, does away with all of these limiting assumptions on planning domains.
Our framework for planning with higher-order reasoning

In classical planning states are models of propositional logic. Classical planning only deals with planning domains that are deterministic, static, fully observable and single-agent.

Our planning framework, epistemic planning, does away with all of these limiting assumptions on planning domains.

From classical planning to epistemic planning: Replace the propositional logic underlying classical planning by Dynamic Epistemic Logic (DEL).

<table>
<thead>
<tr>
<th>States</th>
<th>Classical</th>
<th>DEL-based</th>
</tr>
</thead>
<tbody>
<tr>
<td>Goal formula</td>
<td>models of prop. logic</td>
<td>models of MA epist. logic</td>
</tr>
<tr>
<td>Actions</td>
<td>formula of prop. logic</td>
<td>formula of MA epist. logic</td>
</tr>
<tr>
<td></td>
<td>action schemas</td>
<td>event models of DEL</td>
</tr>
</tbody>
</table>

Bolander: Undecidability in Epistemic Planning – p. 5/17
DEL by example: A private announcement

Event models: Only preconditions, no postconditions. Means:

• Event model above: Private announcement of $p$ to agent 0.
• Product update: As in [Baltag et al., 1998].
• In resulting model: Agent 0 knows $\square 0 p$ holds, but agent 1 didn’t learn anything.
**DEL by example: A private announcement**

- **Event models**: Only preconditions, **no postconditions**. Means: Purely epistemic planning, no change of ontic facts.
- **Event model above**: Private announcement of \( p \) to agent 0.
**Event models**: Only preconditions, **no postconditions**. Means: Purely epistemic planning, no change of ontic facts.

**Event model above**: Private announcement of $p$ to agent 0.

**Product update**: As in [Baltag et al., 1998].
DEL by example: A private announcement

- **Event models**: Only preconditions, **no postconditions**. Means: Purely epistemic planning, no change of ontic facts.
- **Event model above**: Private announcement of $p$ to agent 0.
- **Product update**: As in [Baltag et al., 1998].
DEL by example: A private announcement

- **Event models**: Only preconditions, no postconditions. Means: Purely epistemic planning, no change of ontic facts.

- **Event model above**: Private announcement of $p$ to agent 0.

- **Product update**: As in [Baltag et al., 1998].
DEL by example: A private announcement

- **Event models**: Only preconditions, **no postconditions**. Means: Purely epistemic planning, no change of ontic facts.

- **Event model above**: Private announcement of $p$ to agent 0.

- **Product update**: As in [Baltag et al., 1998].
**Event models**: Only preconditions, **no postconditions**. Means: Purely epistemic planning, no change of ontic facts.

**Event model above**: Private announcement of $p$ to agent 0.

**Product update**: As in [Baltag et al., 1998].
**Event models**: Only preconditions, **no postconditions**. Means: Purely epistemic planning, no change of ontic facts.

**Event model above**: Private announcement of $p$ to agent 0.

**Product update**: As in [Baltag et al., 1998].
Event models: Only preconditions, no postconditions. Means: Purely epistemic planning, no change of ontic facts.

Event model above: Private announcement of $p$ to agent 0.

Product update: As in [Baltag et al., 1998].
**Event models**: Only preconditions, **no postconditions**. Means: Purely epistemic planning, no change of ontic facts.

**Event model above**: Private announcement of $p$ to agent 0.

**Product update**: As in [Baltag et al., 1998].

**In resulting model**: Agent 0 knows $p$ ($\Box_0 p$ holds), but agent 1 didn’t learn anything.
Planning interpretation of DEL

- **Epistemic states**: Pointed, finite epistemic models.
Planning interpretation of DEL

- **Epistemic states**: Pointed, finite epistemic models.
- **Epistemic actions**: Pointed, finite event models.
Planning interpretation of DEL

- **Epistemic states**: Pointed, finite epistemic models.
- **Epistemic actions**: Pointed, finite event models.
- **Result of applying an action in a state**: Product update of state with action.
Epistemic planning tasks and plan existence problem

Definition
An **epistemic planning task** is \((s_0, A, \phi_g)\), where

- \(s_0\) is the **initial state**: an epistemic state.
- \(A\) is a finite set of epistemic actions.
- \(\phi_g\) is the **goal formula**: a formula of epistemic logic.
Definition
An **epistemic planning task** is \((s_0, A, \phi_g)\), where
- \(s_0\) is the **initial state**: an epistemic state.
- \(A\) is a finite set of epistemic actions.
- \(\phi_g\) is the **goal formula**: a formula of epistemic logic.

Definition
A **solution** to a planning task \((s_0, A, \phi_g)\) is a sequence of actions \(a_1, \ldots, a_n \in A\) such that \(s_0 \otimes a_1 \otimes \cdots \otimes a_n \models \phi_g\).
Epistemic planning tasks and plan existence problem

Definition
An epistemic planning task is \((s_0, A, \phi_g)\), where

- \(s_0\) is the initial state: an epistemic state.
- \(A\) is a finite set of epistemic actions.
- \(\phi_g\) is the goal formula: a formula of epistemic logic.

Definition
A solution to a planning task \((s_0, A, \phi_g)\) is a sequence of actions \(a_1, \ldots, a_n \in A\) such that \(s_0 \otimes a_1 \otimes \cdots \otimes a_n \models \phi_g\).

Definition
The plan existence problem in epistemic planning is the following decision problem “Given an epistemic planning task \((s_0, A, \phi_g)\), does it have a solution?”

We will now show undecidability of the plan existence problem...
Two-counter machines

Configurations: $\begin{bmatrix} k & l & m \end{bmatrix}$, where $k, l, m \in \mathbb{N}$.

Instruction set: inc(0), inc(1), jump(j), jzdec(0, j), jzdec(1, j), halt.

Computation step example:

The halting problem for two-counter machines is undecidable [Minsky, 1967].
Proof idea for undecidability of epistemic planning

Our proof idea is this. For each two-register machine, construct a corresponding planning task where:

- The **initial state** encodes the initial configuration of the machine.
- The **actions** encode the instructions of the machine.
- The **goal formula** is true of all epistemic states representing halting configurations of the machine.

Then show that the two-register machine halts iff the corresponding planning task has a solution. (Execution paths of the planning task encodes computations of the machine).
Encodings

Encoding configurations as epistemic states:

$$\begin{array}{c|c|c}
& k & l \\
k & p_1 & l + 1 \\
l & p_2 & m + 1 \\
m & p_3 & \\
\end{array}$$

$$\Rightarrow$$

$$\begin{array}{c}
\text{k + 1 worlds} \\
\text{l + 1 worlds} \\
\text{m + 1 worlds} \\
\end{array}$$
**Encodings**

Encoding configurations as epistemic states:

<table>
<thead>
<tr>
<th>k</th>
<th>l</th>
<th>m</th>
</tr>
</thead>
</table>

\[ k + 1 \] worlds

\[ \nabla \]

\[ \neg (p_1 \lor p_2 \lor p_3) \]

Encoding instructions as epistemic actions:

\[ \text{inc}(0) \]

\[ p_1 \land \Diamond \top \]
\[ p_1 \land \Diamond \Box \bot \]
\[ p_1 \land \Box \bot \]

\[ p_2 \land \Diamond \top \]
\[ p_2 \land \Diamond \Box \bot \]
\[ p_2 \land \Box \bot \]

\[ p_3 \land \Diamond \top \]
\[ p_3 \land \Diamond \Box \bot \]
\[ p_3 \land \Box \bot \]
The computation step \(\begin{array}{c}k \\ l \\ m \end{array}\) \(\rightarrow\) \(\begin{array}{c}k+1 \\ l+1 \\ m \end{array}\) is mimicked by:

\[
\text{encoding}\left(\begin{array}{c}k \\ l \\ m \end{array}\right) \otimes \text{encoding}(\text{inc}(0)) =
\]

\[
\neg (p_1 \lor p_2 \lor p_3)
\]

\[
\begin{array}{c}
p_1 \\ p_2 \\ p_3 
\end{array}
\]

\[
\begin{array}{c}
p_1 \lor 1 \\ p_2 \lor 1 \\ p_3 \lor 1 
\end{array}
\]

\[
\begin{array}{c}
p_1 \lor □⊥ \\ p_2 \lor □⊥ \\ p_3 \lor □⊥ 
\end{array}
\]
The computation step $k \quad l \quad m \quad \text{inc}(0) \quad k + 1 \quad l + 1 \quad m$ is mimicked by:

$$
\text{encoding}(k \quad l \quad m) \otimes \text{encoding}(\text{inc}(0)) =
$$

\begin{align*}
&\neg (p_1 \lor p_2 \lor p_3) \\
&= p_1 \land \lozenge \top \\
&\quad \land p_2 \land \lozenge \top \\
&\quad \land p_3
\end{align*}
The computation step is mimicked by:

\[
\text{encoding}(\begin{array}{c}
k \\
l \\
m
\end{array}) \otimes \text{encoding}(\text{inc}(0)) =
\]

\[
\begin{array}{c}
p_1 \\
p_2 \\
p_3
\end{array}
\]

\[
\begin{array}{c}
k + 1 \\
l + 1 \\
m + 1
\end{array}
\]

\[
\neg(p_1 \lor p_2 \lor p_3)
\]

\[
\begin{array}{c}
p_1 \land \Diamond T \\
p_2 \land \Diamond T \\
p_3
\end{array}
\]

\[
\begin{array}{c}
p_1 \land \Box \bot \\
p_2 \land \Box \bot \\
p_3
\end{array}
\]
The computation step is mimicked by:

$$\text{encoding}(\begin{bmatrix} k & l & m \end{bmatrix}) \otimes \text{encoding}(\text{inc}(0)) =$$

$$\neg(p_1 \lor p_2 \lor p_3)$$
The computation step\(\begin{array}{c} k \\
\uparrow \cr l \\
\uparrow \cr m \end{array} \xrightarrow{\text{inc}(0)} \begin{array}{c} k + 1 \\
\uparrow \cr l + 1 \\
\uparrow \cr m \end{array}\) is mimicked by:

\[
\text{encoding}(\begin{array}{c} k \\
\uparrow \cr l \\
\uparrow \cr m \end{array}) \otimes \text{encoding}(\text{inc}(0)) =
\]

\[
\neg (p_1 \lor p_2 \lor p_3) \otimes p_1 \land \lozenge \top \quad p_2 \land \lozenge \top \quad p_3 =
\]

\[
\neg (p_1 \land \lozenge \perp) \quad p_2 \land \lozenge \perp \quad p_3 \land \lozenge \perp
\]
The computation step is mimicked by:

\[
\text{encoding}(\begin{array}{c}
k \\
l \\
m
\end{array}) \otimes \text{encoding}(\text{inc}(0)) =
\]

\[
\neg (p_1 \lor p_2 \lor p_3)
\]

\[
\begin{array}{c}
\neg (p_1 \lor p_2 \lor p_3) \\
\otimes
\end{array}
\]

\[
\begin{array}{c}
p_1 \land \Diamond T \\
p_2 \land \Diamond T \\
p_3
\end{array}
\]

\[
\begin{array}{c}
p_1 \land \Box \bot \\
p_2 \land \Box \bot
\end{array}
\]

\[
\begin{array}{c}
p_1 \\
p_2
\end{array}
\]
The computation step $k \mid l \mid m$ is mimicked by:

$$\text{encoding}(k \mid l \mid m) \otimes \text{encoding}(\text{inc}(0)) =$$

$$\neg (p_1 \lor p_2 \lor p_3)$$
The computation step $\begin{array}{c} k \\ l \\ m \end{array} \xrightarrow{\text{inc}(0)} \begin{array}{c} k + 1 \\ l + 1 \\ m \end{array}$ is mimicked by:

\[
\text{encoding}(\begin{array}{c} k \\ l \\ m \end{array}) \otimes \text{encoding}(\text{inc}(0)) = \]

\[
\neg(p_1 \lor p_2 \lor p_3) = \]

\[
\text{encoding}(\begin{array}{c} k + 1 \\ l + 1 \\ m \end{array})
\]
Summary of results on (un)decidability of plan existence in epistemic planning

<table>
<thead>
<tr>
<th>L</th>
<th>transitive</th>
<th>Euclidean</th>
<th>reflexive</th>
</tr>
</thead>
<tbody>
<tr>
<td>K</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>KT</td>
<td>✓</td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>K4</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>K45</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>S4</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>S5</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

← belief
← knowledge

---

Theorem

The figure to the right summarises our results on decidability (D) and undecidability (UD) of the plan existence problem in epistemic planning.

<table>
<thead>
<tr>
<th></th>
<th>Single-agent planning</th>
<th>Multi-agent planning</th>
</tr>
</thead>
<tbody>
<tr>
<td>L</td>
<td></td>
<td></td>
</tr>
<tr>
<td>K</td>
<td>UD</td>
<td>UD</td>
</tr>
<tr>
<td>KT</td>
<td>UD</td>
<td>UD</td>
</tr>
<tr>
<td>K4</td>
<td>UD</td>
<td>UD</td>
</tr>
<tr>
<td>K45</td>
<td>D</td>
<td>UD</td>
</tr>
<tr>
<td>S4</td>
<td>UD</td>
<td>UD</td>
</tr>
<tr>
<td>S5</td>
<td>D</td>
<td>UD</td>
</tr>
</tbody>
</table>
Corollary: Undecidability of model checking in $\mathcal{L}_{DEL}^*$

The $DEL$ language $\mathcal{L}_{DEL}^*$ is defined by the following BNF:

$$
\phi ::= p \mid \neg \phi \mid (\phi \land \phi) \mid \Box i \phi \mid [\pi] \phi
$$

$$
\pi ::= (E, e) \mid (\pi \cup \pi) \mid (\pi; \pi) \mid \pi^*
$$

where $p \in P$, $i \in A$ and $(E, e)$ is any pointed event model [van Ditmarsch et al., 2007]. Define $\langle \pi \rangle \phi := \neg[\pi] \neg \phi$.

Semantics:

$$
\mathcal{M}, w \models [(E, e)] \phi \iff \mathcal{M}, w \models pre(e) \text{ implies } (\mathcal{M}, w) \otimes (E, e) \models \phi
$$

$$
\mathcal{M}, w \models [\pi \cup \gamma] \phi \iff \mathcal{M}, w \models [\pi] \phi \text{ and } \mathcal{M}, w \models [\gamma] \phi
$$

$$
\mathcal{M}, w \models [\pi; \gamma] \phi \iff \mathcal{M}, w \models [\pi][\gamma] \phi
$$

$$
\mathcal{M}, w \models [\pi^*] \phi \iff \mathcal{M}, w \models [\pi]^n \phi, \text{ for all } n
$$
Corollary: Undecidability of model checking in $\mathcal{L}^*_\text{DEL}$

[Miller & Moss, 2005] shows that the **satisfiability** problem of $\mathcal{L}^*_\text{DEL}$ is undecidable. Our results above immediately gives us that even the **model checking** problem is undecidable.

**Theorem**

*The model checking problem of the language $\mathcal{L}^*_\text{DEL}$ is undecidable.*

**Proof.**

The plan existence problem considered above is reducible to the model checking problem of $\mathcal{L}^*_\text{DEL}$: Consider an epistemic planning task $\mathcal{T} = (s_0, \{a_1, \ldots, a_m\}, \phi_g)$. $\mathcal{T}$ has a solution iff the following holds:

$$s_0 \models \langle (a_1 \cup \cdots \cup a_m)^* \rangle \phi_g.$$
Summary and related work

- Previously known undecidability results for DEL-based epistemic planning: S5, with postconditions, $\geq 3$ agents [Bolander & Andersen, JANCL 2011].

- New results presented here: S5, without postconditions, $\geq 2$ agents [Aucher & Bolander, IJCAI 2013].

- In essence: allowing arbitrary levels of higher-order reasoning leads to undecidability of planning. Reason: no bound on level of higher-order reasoning $\Rightarrow$ no bound on depth of epistemic state $\Rightarrow$ no bound of size of epistemic states $\Rightarrow$ state space can become infinite.

- Decidable fragments of epistemic planning:
  - Single-agent K45 and S5: Replace epistemic states by their bisimulation contractions. These have bounded depth.
  - Multi-agent planning with propositional preconditions [Yu, Wen & Liu, 2013]: Replace epistemic states by their $k$-bisimulation contractions, where $k$ is the modal depth of the goal formula. These have bounded depth.
Summary and related work

- Previously known undecidability results for DEL-based epistemic planning: S5, with postconditions, $\geq 3$ agents [Bolander & Andersen, JANCL 2011].

- New results presented here: S5, without postconditions, $\geq 2$ agents [Aucher & Bolander, IJCAI 2013].
Summary and related work

• Previously known undecidability results for DEL-based epistemic planning: S5, with postconditions, $\geq 3$ agents [Bolander & Andersen, JANCL 2011].

• New results presented here: S5, without postconditions, $\geq 2$ agents [Aucher & Bolander, IJCAI 2013].

• In essence: allowing arbitrary levels of higher-order reasoning leads to undecidability of planning. Reason: no bound on level of higher-order reasoning $\Rightarrow$ no bound on depth of epistemic state $\Rightarrow$ no bound of size of epistemic states $\Rightarrow$ state space can become infinite.

Decidable fragments of epistemic planning:

• Single-agent K45 and S5: Replace epistemic states by their bisimulation contractions. These have bounded depth.

• Multi-agent planning with propositional preconditions [Yu, Wen & Liu, 2013]: Replace epistemic states by their $k$-bisimulation contractions, where $k$ is the modal depth of the goal formula. These have bounded depth.
Summary and related work

• Previously known undecidability results for DEL-based epistemic planning: S5, with postconditions, $\geq 3$ agents [Bolander & Andersen, JANCL 2011].

• New results presented here: S5, without postconditions, $\geq 2$ agents [Aucher & Bolander, IJCAI 2013].

• In essence: allowing arbitrary levels of higher-order reasoning leads to undecidability of planning. Reason: no bound on level of higher-order reasoning $\Rightarrow$ no bound on depth of epistemic state $\Rightarrow$ no bound of size of epistemic states $\Rightarrow$ state space can become infinite.

• Decidable fragments of epistemic planning:
Summary and related work

• Previously known undecidability results for DEL-based epistemic planning: S5, with postconditions, $\geq 3$ agents [Bolander & Andersen, JANCL 2011].

• New results presented here: S5, without postconditions, $\geq 2$ agents [Aucher & Bolander, IJCAI 2013].

• In essence: allowing arbitrary levels of higher-order reasoning leads to undecidability of planning. Reason: no bound on level of higher-order reasoning $\Rightarrow$ no bound on depth of epistemic state $\Rightarrow$ no bound of size of epistemic states $\Rightarrow$ state space can become infinite.

• Decidable fragments of epistemic planning:
  • Single-agent K45 and S5: Replace epistemic states by their bisimulation contractions. These have bounded depth.
Summary and related work

- Previously known undecidability results for DEL-based epistemic planning: S5, \textit{with postconditions}, $\geq 3$ agents [Bolander & Andersen, JANCL 2011].
- New results presented here: S5, \textit{without postconditions}, $\geq 2$ agents [Aucher & Bolander, IJCAI 2013].
- In essence: \textit{allowing arbitrary levels of higher-order reasoning leads to undecidability of planning}. Reason: no bound on level of higher-order reasoning $\Rightarrow$ no bound on depth of epistemic state $\Rightarrow$ no bound of size of epistemic states $\Rightarrow$ state space can become infinite.
- \textbf{Decidable} fragments of epistemic planning:
  - Single-agent K45 and S5: Replace epistemic states by their bisimulation contractions. These have bounded depth.
  - Multi-agent planning with propositional preconditions [Yu, Wen & Liu, 2013]: Replace epistemic states by their $k$-bisimulation contractions, where $k$ is the modal depth of the goal formula. These have bounded depth.
Summary and related work

- Other formalisms for epistemic planning:
  - **Decentralised POMDPs**: Finite state space explicitly given. Planning complexities are wrt. this state space.
  - **Formalisms based on concurrent epistemic game structures** (ATEL [Hoek & Wooldridge, 2002], ATOL [Jamroga et al., 2004], CSL [Jamroga & Aagotnes, 2007], etc.): Finite state space explicitly given. Planning complexities are wrt. this state space.

So in these formalisms you cannot model e.g. the message sending actions in the coordinated attack problem.