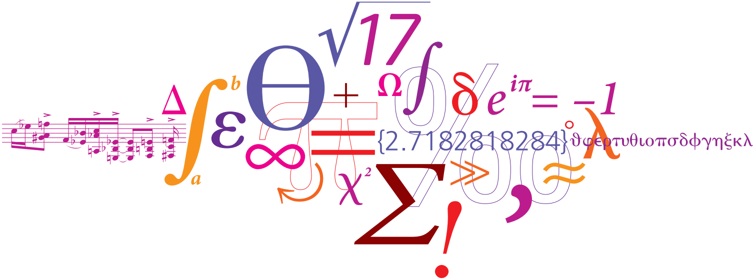
Undecidability in Epistemic Planning

Guillaume Aucher, IRISA, University of Rennes 1 – INRIA, France
Thomas Bolander, DTU Compute, Tech Univ of Denmark



DTU Compute
Department of Applied Mathematics and Computer Science

Introduction

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.

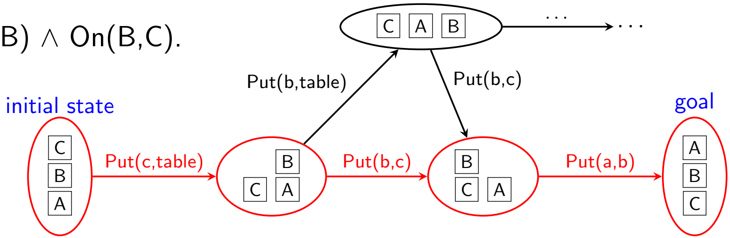
Structure of talk:

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 of results.

Planning and higher-order reasoning

Automated planning: Given a **goal formula**, an **initial state** and some **actions**, compute a sequence of actions that leads from the initial state to a state satisfying the goal formula.

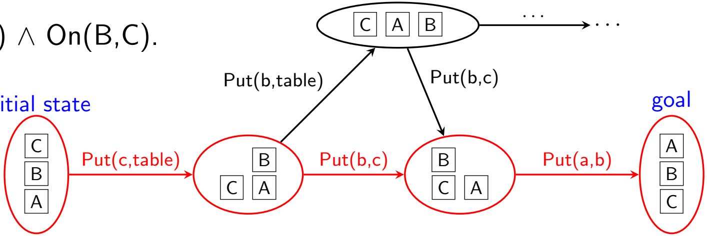
Example.
Goal: On(A,B) $\wedge$ On(B,C).



Planning and higher-order reasoning

Automated planning: Given a **goal formula**, an **initial state** and some **actions**, compute a sequence of actions that leads from the initial state to a state satisfying the goal formula.

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



Why higher-order reasoning in planning?

initial state

?

goal

IJCAI 2013 conference banquet

Wednesday, August 7
18:30

Our framework for planning with higher-order reasoning

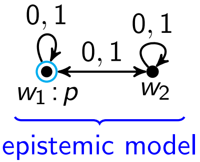
Epistemic planning: Our framework for planning with higher-order reasoning.

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

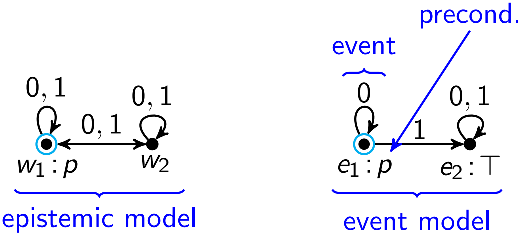
	Classical planning	**Epistemic planning**
States	models of prop. logic	models of MA epist. logic
Goal formula	formula of prop. logic	formula of MA epist. logic
Actions	induced by action schemas	event models of DEL

Epistemic planning can deal with: non-determinism, partial observability, sensing actions, multiple agents, higher-order reasoning.

DEL by example: A private announcement

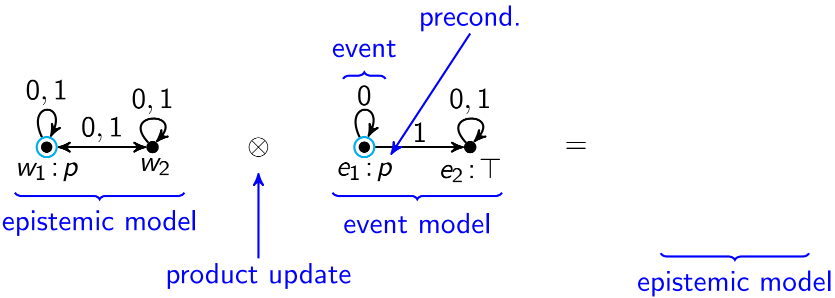


DEL by example: A private announcement



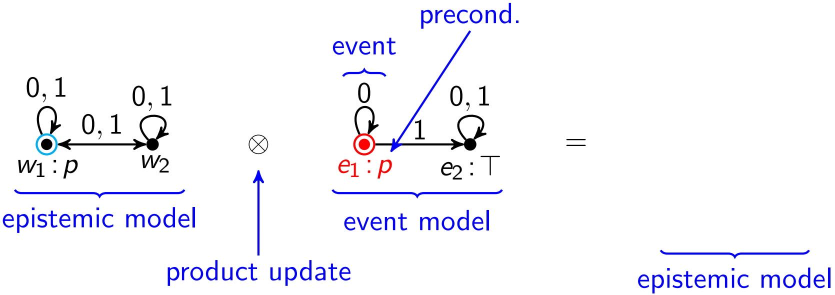
- **Event models**: Only preconditions, no postconditions. Means: Purely epistemic planning, no change of ontic facts.
- **Event model above**: Private announcement of p to agent 0.

DEL by example: A private announcement



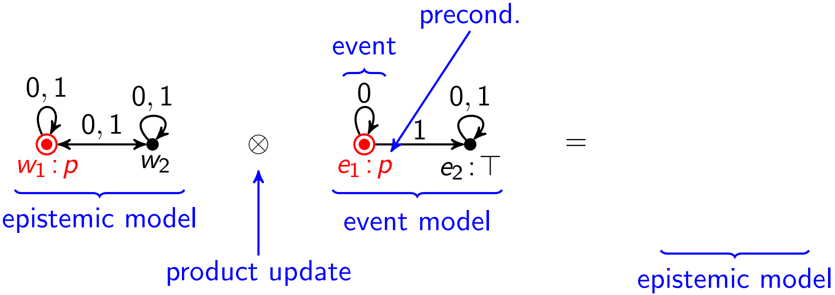
- **Event models**: Only preconditions, no postconditions. Means: Purely epistemic planning, no change of ontic facts.
- **Event model above**: Private announcement of p to agent 0.
- **Product update**: As in [Baltag et al., 1998].

DEL by example: A private announcement



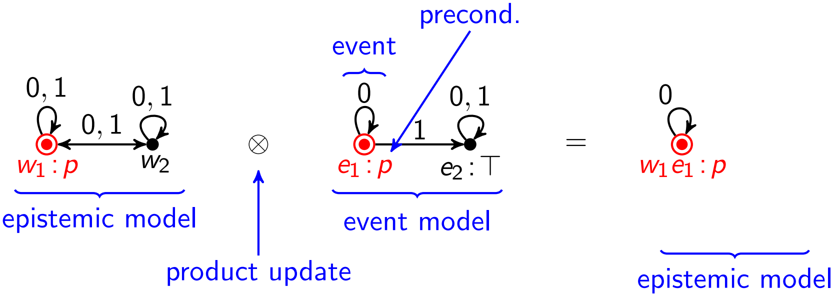
- **Event models**: Only preconditions, no postconditions. Means: Purely epistemic planning, no change of ontic facts.
- **Event model above**: Private announcement of p to agent 0.
- **Product update**: As in [Baltag et al., 1998].

DEL by example: A private announcement



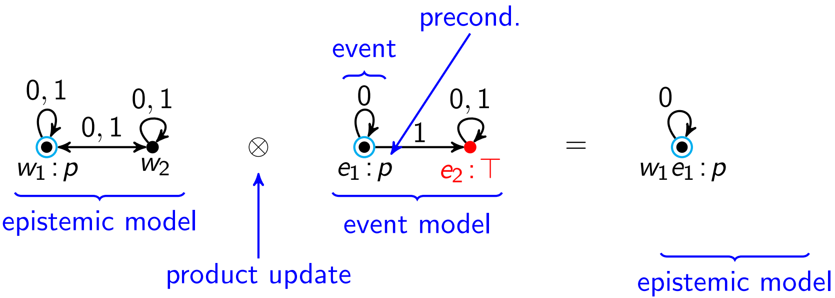
- **Event models**: Only preconditions, no postconditions. Means: Purely epistemic planning, no change of ontic facts.
- **Event model above**: Private announcement of p to agent 0.
- **Product update**: As in [Baltag et al., 1998].

DEL by example: A private announcement



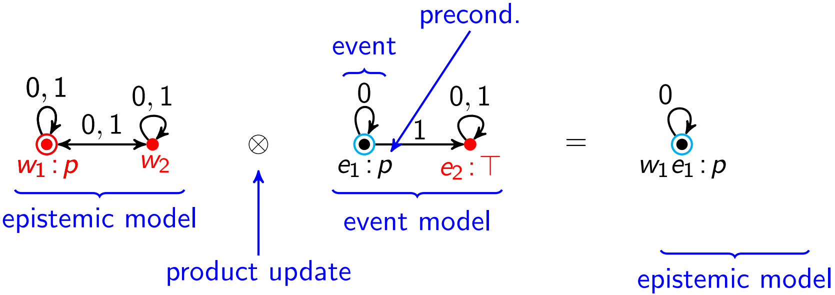
- **Event models**: Only preconditions, no postconditions. Means: Purely epistemic planning, no change of ontic facts.
- **Event model above**: Private announcement of p to agent 0.
- **Product update**: As in [Baltag et al., 1998].

DEL by example: A private announcement



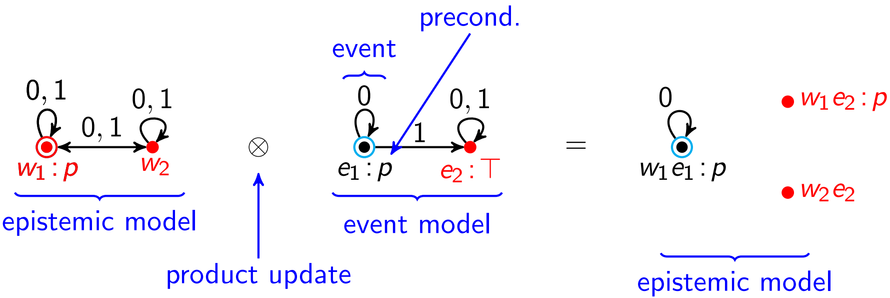
- **Event models**: Only preconditions, no postconditions. Means: Purely epistemic planning, no change of ontic facts.
- **Event model above**: Private announcement of p to agent 0.
- **Product update**: As in [Baltag et al., 1998].

DEL by example: A private announcement



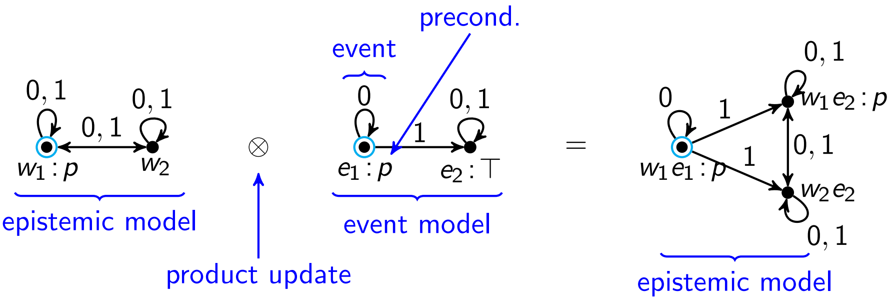
- **Event models**: Only preconditions, no postconditions. Means: Purely epistemic planning, no change of ontic facts.
- **Event model above**: Private announcement of p to agent 0.
- **Product update**: As in [Baltag et al., 1998].

DEL by example: A private announcement



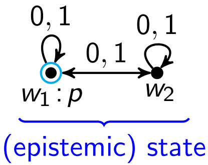
- **Event models**: Only preconditions, no postconditions. Means: Purely epistemic planning, no change of ontic facts.
- **Event model above**: Private announcement of p to agent 0.
- **Product update**: As in [Baltag et al., 1998].

DEL by example: A private announcement



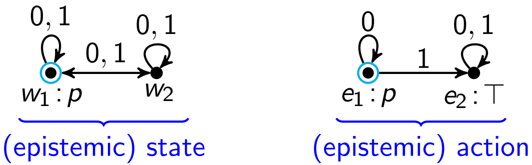
- **Event models**: Only preconditions, no postconditions. Means: Purely epistemic planning, no change of ontic facts.
- **Event model above**: Private announcement of p to agent 0.
- **Product update**: As in [Baltag et al., 1998].
- **In resulting model**: Agent 0 knows p ($\Box_0 p$ holds), but agent 1 didn't learn anything.

Planning interpretation of DEL



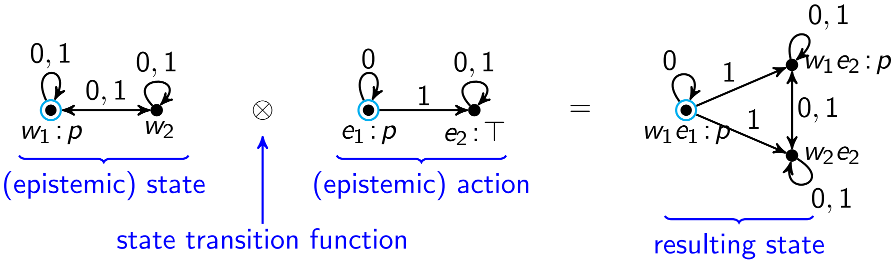
- **Epistemic states**: Pointed, finite epistemic models.

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, ϕ_g), where

- s_0 is the **initial state**: an epistemic state.
- A is a finite set of epistemic actions.
- ϕ_g is the **goal formula**: a formula of epistemic logic.

Epistemic planning tasks and plan existence problem

Definition

An **epistemic planning task** is (s_0, A, ϕ_g), where

- s_0 is the **initial state**: an epistemic state.
- A is a finite set of epistemic actions.
- ϕ_g is the **goal formula**: a formula of epistemic logic.

Definition

A **solution** to a planning task (s_0, A, ϕ_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, ϕ_g), where

- s_0 is the **initial state**: an epistemic state.
- A is a finite set of epistemic actions.
- ϕ_g is the **goal formula**: a formula of epistemic logic.

Definition

A **solution** to a planning task (s_0, A, ϕ_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, ϕ_g), does it have a solution?"

We will now show undecidability of the plan existence problem...

Two-counter machines

Configurations: | k | l | m |, where $k, l, m \in \mathbb{N}$.

R_0 R_1

register 0 register 1

Instruction set: $\mathrm{inc}(0), \mathrm{inc}(1), \mathrm{jump}(j), \mathrm{jzdec}(0, j), \mathrm{jzdec}(1, j), \mathrm{halt}$.

Computation step example:

$$\boxed{k \mid l \mid m} \xrightarrow{\mathrm{inc}(0)} \boxed{k+1 \mid l+1 \mid m}$$

R_0 R_1 R_0 R_1

The halting problem for two-counter machines is undecidable [Minsky, 1967].

Proof idea for undecidability of epistemic planning

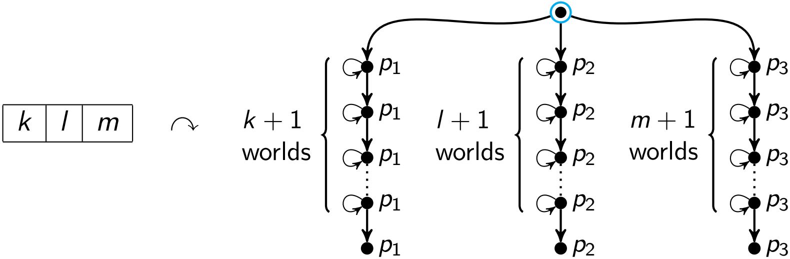
Our proof idea is this. For each two-register machine, construct a corresponding planning task where:

- 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).

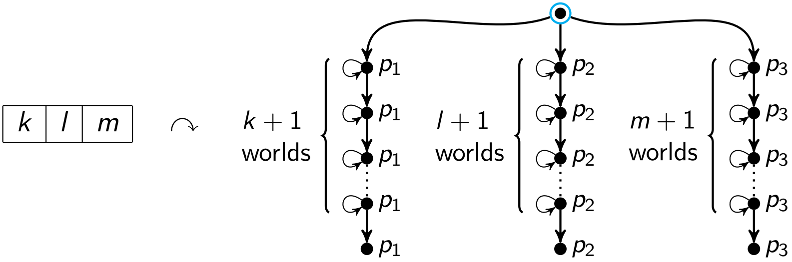
Encodings

Encoding configurations as epistemic states:

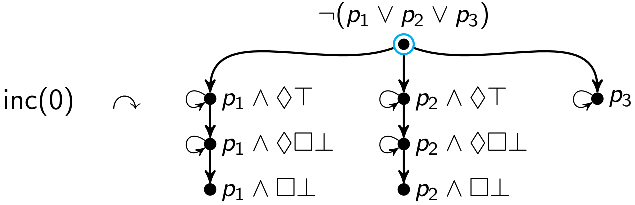


Encodings

Encoding configurations as epistemic states:

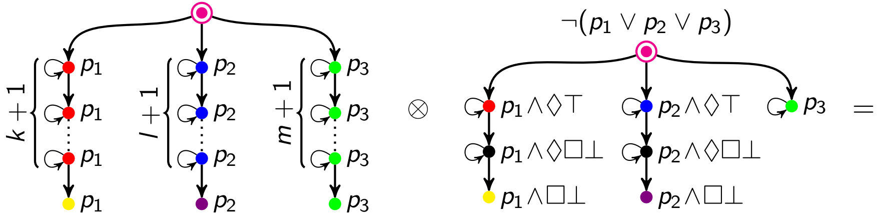


Encoding instructions as epistemic actions:



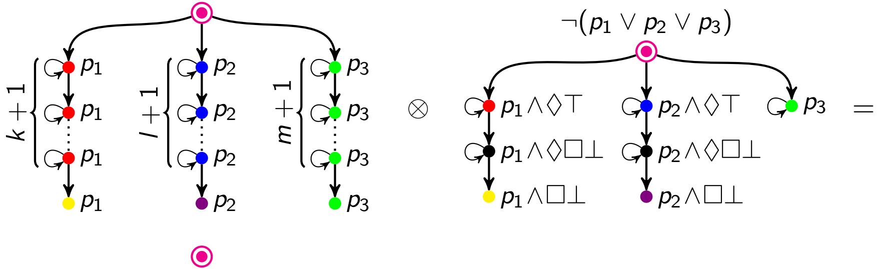
The computation step $\boxed{k \mid l \mid m} \xrightarrow{\text{inc}(0)} \boxed{k+1 \mid l+1 \mid m}$ is mimicked by:

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



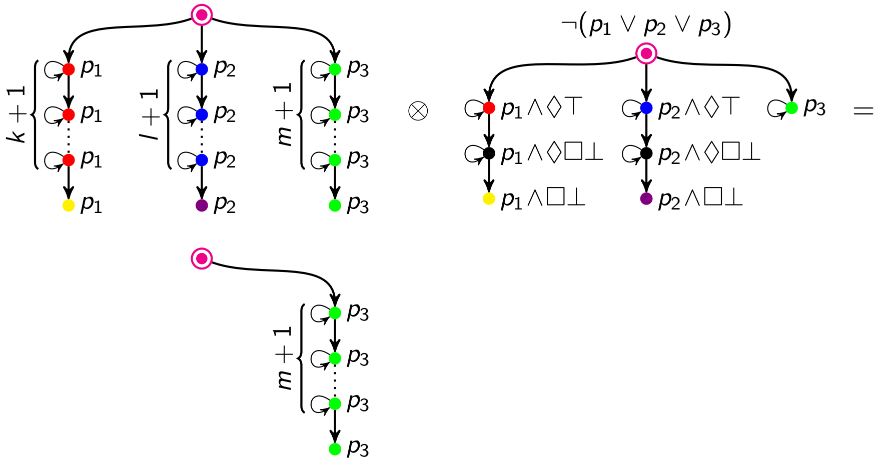
The computation step $\boxed{k \mid l \mid m} \xrightarrow{\text{inc}(0)} \boxed{k+1 \mid l+1 \mid m}$ is mimicked by:

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



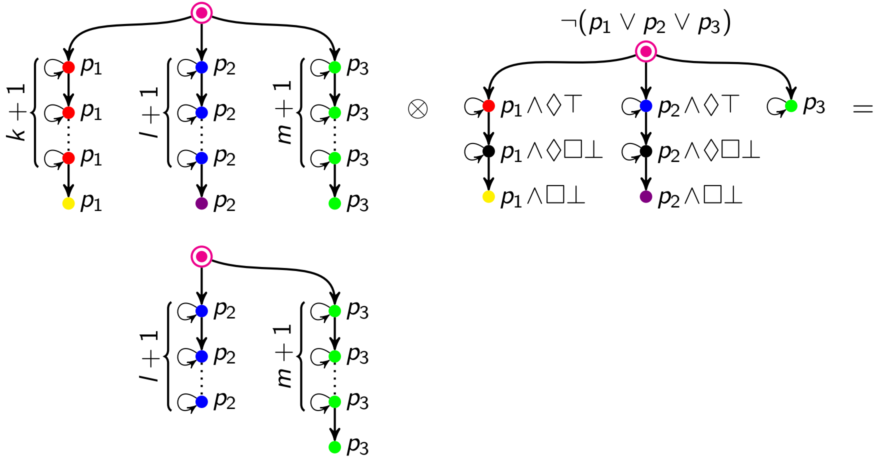
The computation step $\boxed{k \mid l \mid m} \xrightarrow{\text{inc}(0)} \boxed{k+1 \mid l+1 \mid m}$ is mimicked by:

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



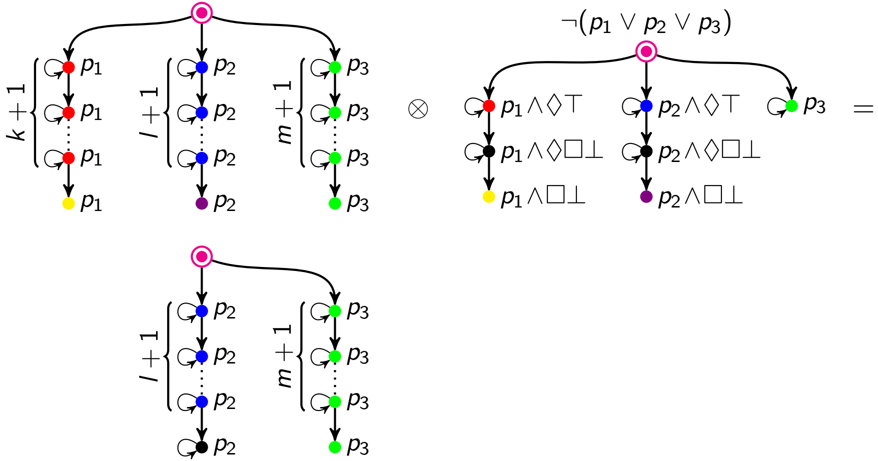
The computation step $\boxed{k \mid l \mid m} \xrightarrow{\text{inc}(0)} \boxed{k+1 \mid l+1 \mid m}$ is mimicked by:

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



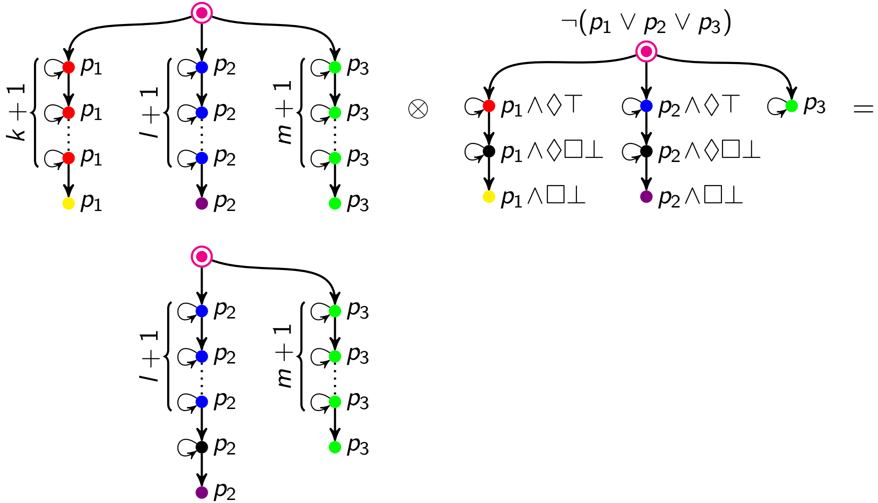
The computation step $\boxed{k \mid l \mid m} \xrightarrow{\text{inc}(0)} \boxed{k+1 \mid l+1 \mid m}$ is mimicked by:

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



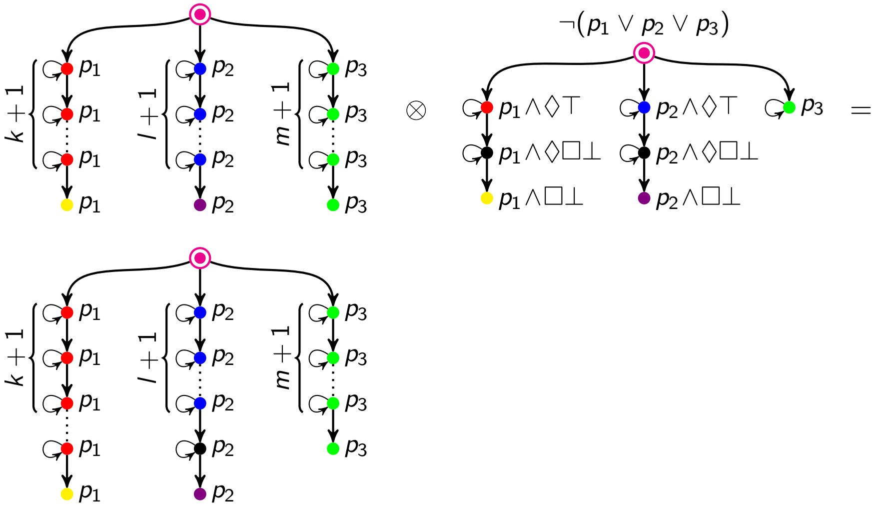
The computation step $\boxed{k \mid l \mid m} \xrightarrow{\text{inc}(0)} \boxed{k+1 \mid l+1 \mid m}$ is mimicked by:

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



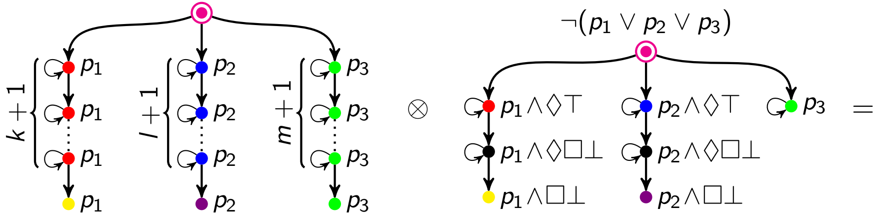
The computation step $\boxed{k \mid l \mid m} \xrightarrow{\text{inc}(0)} \boxed{k+1 \mid l+1 \mid m}$ is mimicked by:

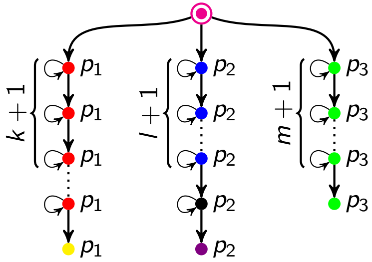
$$\mathit{encoding}(\boxed{k \mid l \mid m}) \otimes \mathit{encoding}(\text{inc}(0)) =$$



The computation step $\boxed{k \mid l \mid m} \xrightarrow{\text{inc}(0)} \boxed{k+1 \mid l+1 \mid m}$ is mimicked by:

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





$= \mathit{encoding}(\boxed{k+1 \mid l+1 \mid m})$

Summary of results on (un)decidability of plan existence in epistemic planning

L	transitive	Euclidean	reflexive	
K				
KT			✓	
K4	✓			
K45	✓	✓		← belief
S4	✓		✓	
S5	✓	✓	✓	← knowledge

Theorem

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

	Single-agent planning	Multi-agent planning
K	**UD**	UD
KT	UD	UD
K4	UD	UD
K45	D	UD
S4	UD	UD
S5	D	UD

Decidable fragments: Quan Yu, Ximing Wen and Yongmei Liu: Multi-Agent Epistemic Explanatory Diagnosis via Reasoning about Actions, IJCAI 2013.

Summary

- We prove that **allowing arbitrary levels of higher-order reasoning leads to undecidability of planning**. Even in the propositional and purely epistemic case.
- Essence of the problem: Even if your beliefs are strengthened through your actions, it might just mean that you ignorance is pushed to deeper and deeper levels. And we can put no bound on this depth of ignorance, hence no bound on depth (size) of epistemic states.

Corollary: Undecidability of model checking in $\mathcal{L}^*_{DEL}$

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

$$\begin{aligned}\phi &::= p \mid \neg\phi \mid (\phi \wedge \phi) \mid \Box_i \phi \mid [\pi]\phi \\ \pi &::= (\mathcal{E}, e) \mid (\pi \cup \pi) \mid (\pi ; \pi) \mid \pi^*\end{aligned}$$

where $p \in P$, $i \in \mathcal{A}$ and $(\mathcal{E}, e)$ is any epistemic action [van Ditmarsch *et al.*, 2007].

Theorem

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

Proof.

The plan existence problem considered above is reducible to the model checking problem of $\mathcal{L}^*_{DEL}$. $\square$