Undecidability in Epistemic Planning

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

DTU Compute
Department of Applied Mathematics and Computer Science

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:

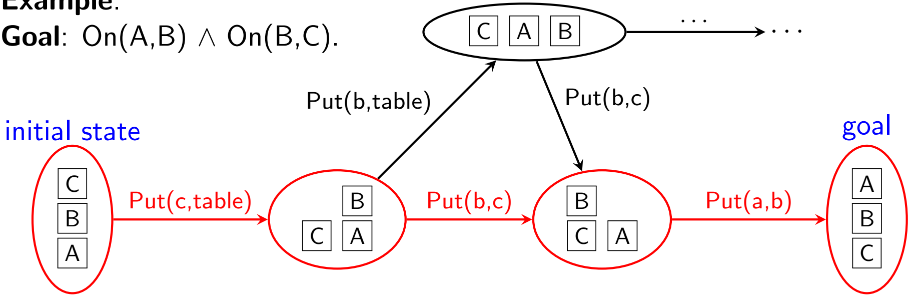
1. Motivation.
2. Introducing the basics: planning + logic + two-counter machines.
3. Sketching the proof: How to encode two-counter machines as epistemic planning problems.
4. Summary and related work.

Automated planning

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) $\wedge$ 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

?

goal

Tuesday, December 3rd
19.30 Workshop Dinner

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.

`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

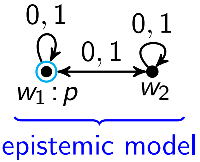
In **classical planning** states are models of propositional logic. Classical planning only deals with planning domains that are **deterministic**, **static**, **fully observable** and **single-agent**.

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

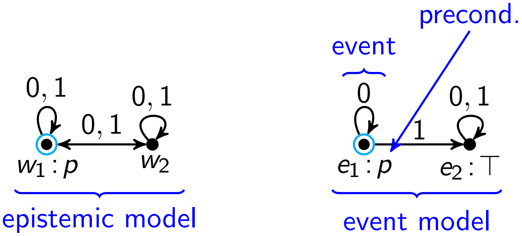
From **classical planning** to **epistemic planning**: Replace the propositional logic underlying classical planning by **Dynamic Epistemic Logic** (**DEL**).

	Classical	**DEL-based**
States	models of prop. logic	models of MA epist. logic
Goal formula	formula of prop. logic	formula of MA epist. logic
Actions	action schemas	event models of DEL

DEL by example: A private announcement

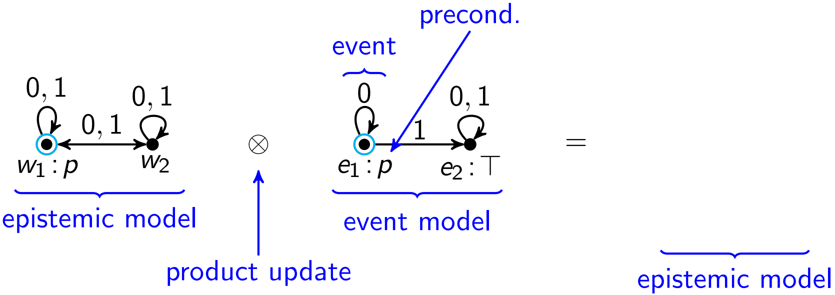


DEL by example: A private announcement



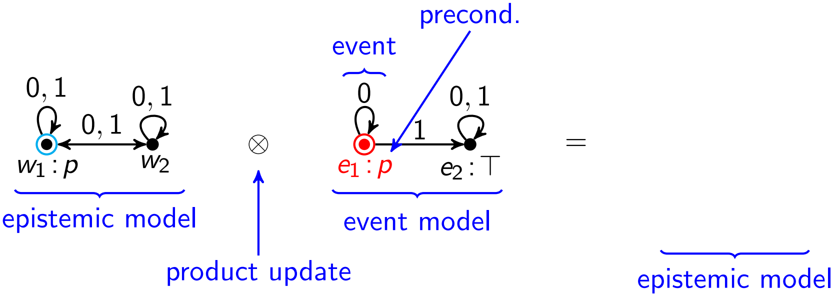
- **Event models**: Only preconditions, **no postconditions**. Means: Purely epistemic planning, no change of ontic facts.
- **Event model above**: Private announcement of p to agent 0.

DEL by example: A private announcement



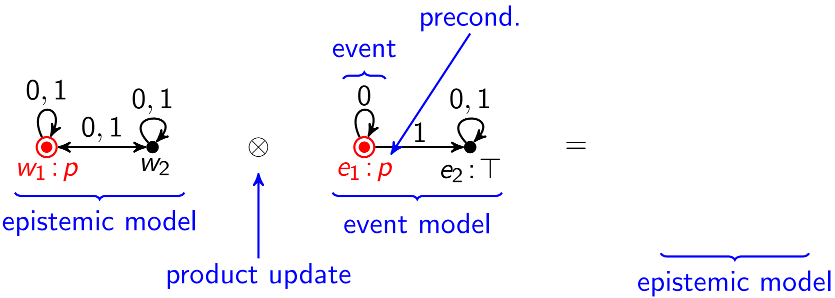
- **Event models**: Only preconditions, **no postconditions**. Means: Purely epistemic planning, no change of ontic facts.
- **Event model above**: Private announcement of p to agent 0.
- **Product update**: As in [Baltag et al., 1998].

DEL by example: A private announcement



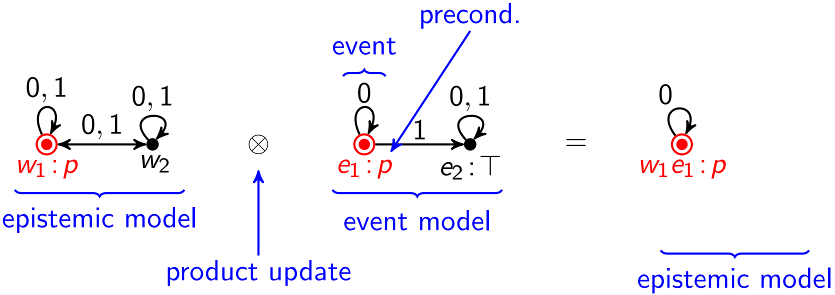
- **Event models**: Only preconditions, **no postconditions**. Means: Purely epistemic planning, no change of ontic facts.
- **Event model above**: Private announcement of p to agent 0.
- **Product update**: As in [Baltag et al., 1998].

DEL by example: A private announcement



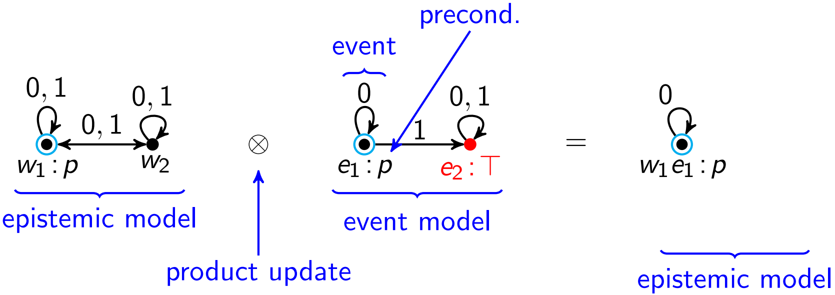
- **Event models**: Only preconditions, **no postconditions**. Means: Purely epistemic planning, no change of ontic facts.
- **Event model above**: Private announcement of p to agent 0.
- **Product update**: As in [Baltag et al., 1998].

DEL by example: A private announcement



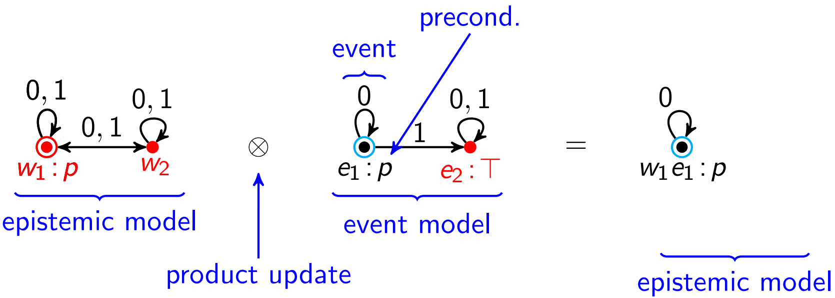
- **Event models**: Only preconditions, **no postconditions**. Means: Purely epistemic planning, no change of ontic facts.
- **Event model above**: Private announcement of p to agent 0.
- **Product update**: As in [Baltag et al., 1998].

DEL by example: A private announcement



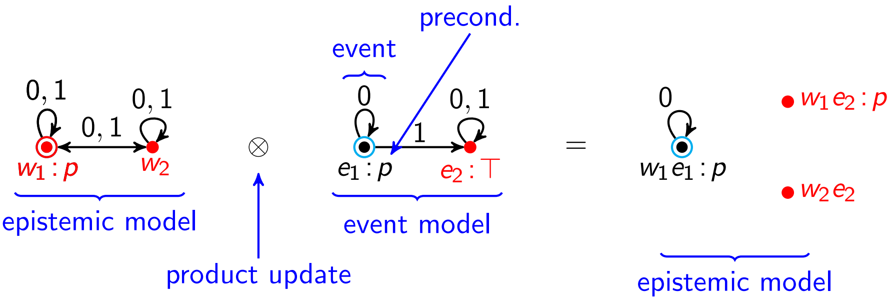
- **Event models**: Only preconditions, **no postconditions**. Means: Purely epistemic planning, no change of ontic facts.
- **Event model above**: Private announcement of p to agent 0.
- **Product update**: As in [Baltag et al., 1998].

DEL by example: A private announcement



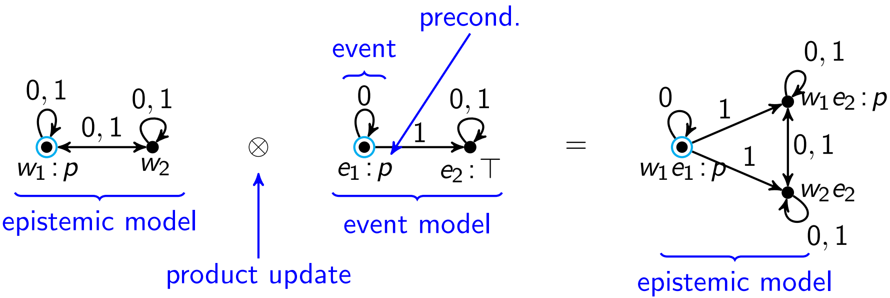
- **Event models**: Only preconditions, **no postconditions**. Means: Purely epistemic planning, no change of ontic facts.
- **Event model above**: Private announcement of p to agent 0.
- **Product update**: As in [Baltag et al., 1998].

DEL by example: A private announcement



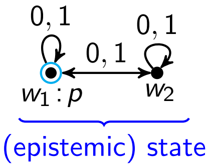
- **Event models**: Only preconditions, **no postconditions**. Means: Purely epistemic planning, no change of ontic facts.
- **Event model above**: Private announcement of p to agent 0.
- **Product update**: As in [Baltag et al., 1998].

DEL by example: A private announcement



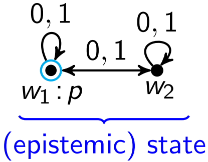
- **Event models**: Only preconditions, **no postconditions**. Means: Purely epistemic planning, no change of ontic facts.
- **Event model above**: Private announcement of p to agent 0.
- **Product update**: As in [Baltag et al., 1998].
- **In resulting model**: Agent 0 knows p ($\Box_0 p$ holds), but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

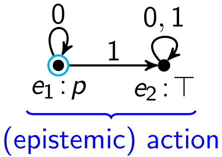
Planning interpretation of DEL



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

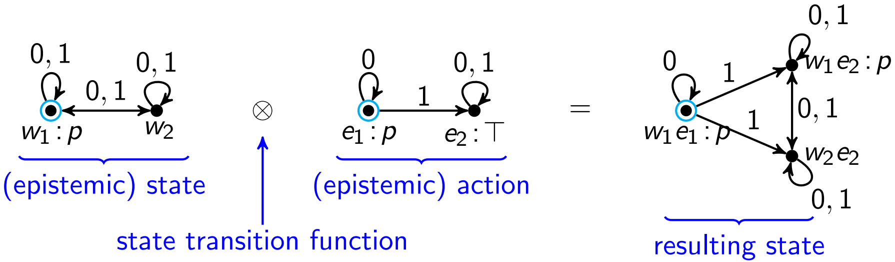
Planning interpretation of DEL





- **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

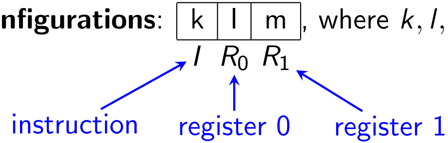
A **solution** to a planning task (s_0, A, ϕ_g) is a sequence of actions $a_1, \dots, a_n \in A$ such that $s_0 \otimes a_1 \otimes \cdots \otimes a_n \models \phi_g$.

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...

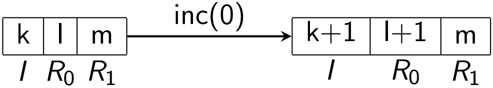
Two-counter machines

Configurations: 

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

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

Computation step example:



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

Proof idea for undecidability of epistemic planning

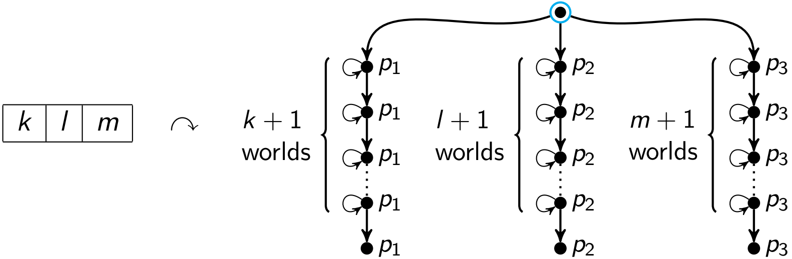
Our proof idea is this. For each two-register machine, construct a corresponding planning task where:

- 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).

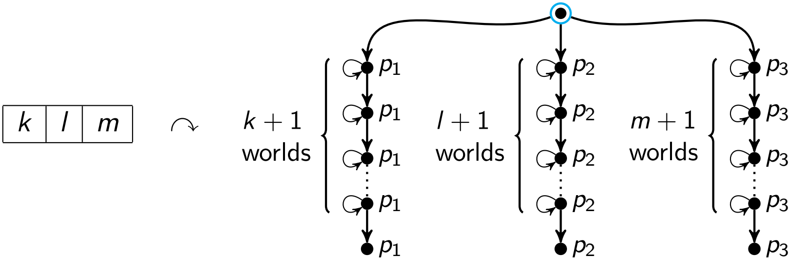
Encodings

Encoding configurations as epistemic states:

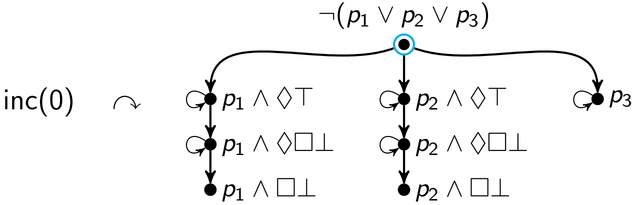


Encodings

Encoding configurations as epistemic states:

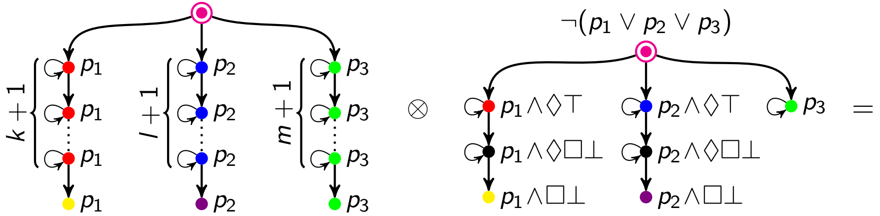


Encoding instructions as epistemic actions:



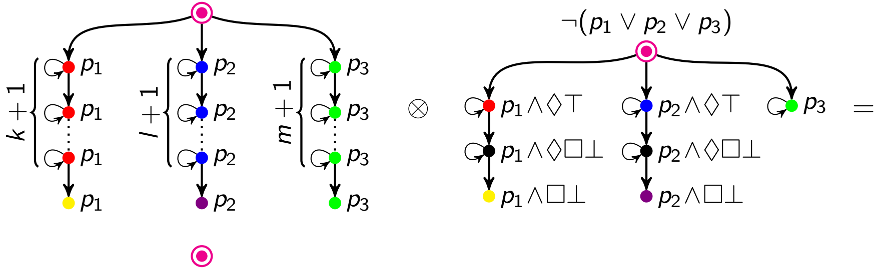
The computation step $\boxed{k \mid l \mid m} \xrightarrow{\text{inc}(0)} \boxed{k+1 \mid l+1 \mid m}$ is mimicked by:

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



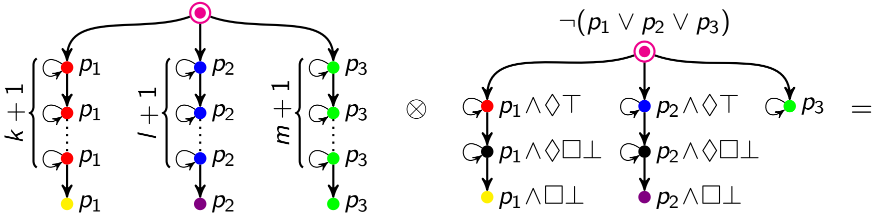
The computation step $\boxed{k \mid l \mid m} \xrightarrow{\text{inc}(0)} \boxed{k+1 \mid l+1 \mid m}$ is mimicked by:

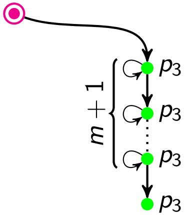
$$encoding(\boxed{k \mid l \mid m}) \otimes encoding(\text{inc}(0)) =$$



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

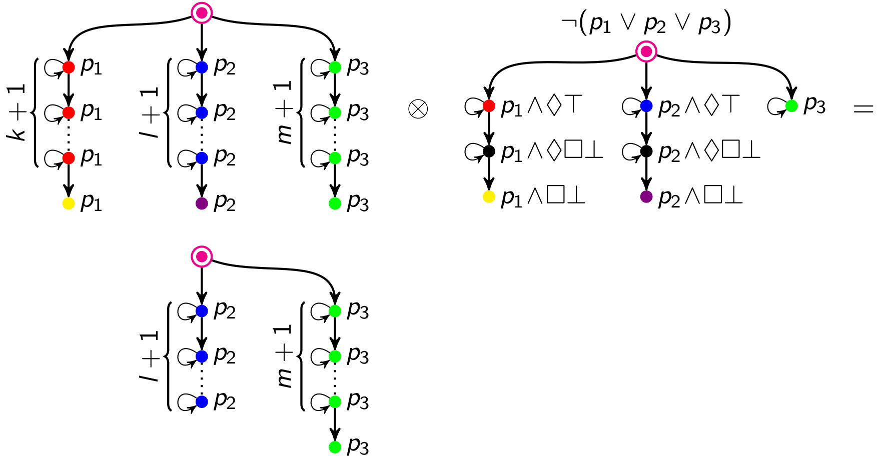
$$\textit{encoding}(\boxed{k \mid l \mid m}) \otimes \textit{encoding}(\text{inc}(0)) =$$





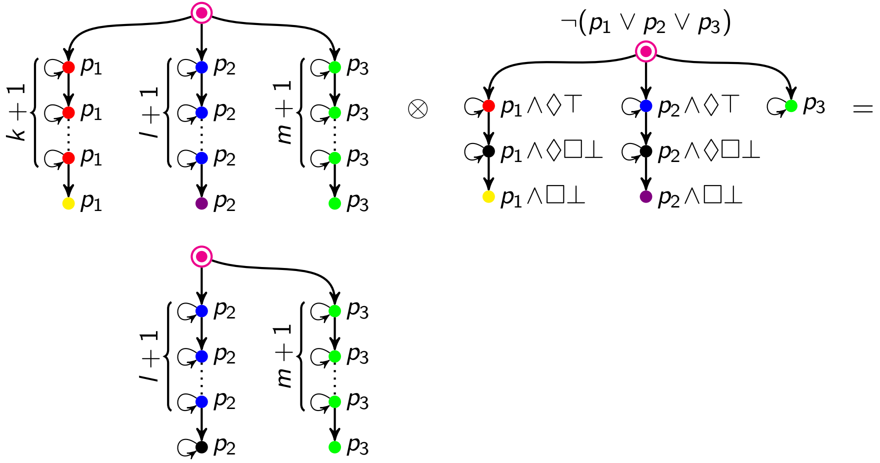
The computation step $\boxed{k \mid l \mid m} \xrightarrow{\text{inc}(0)} \boxed{k+1 \mid l+1 \mid m}$ is mimicked by:

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



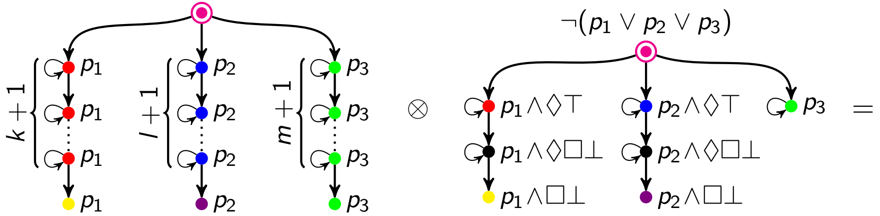
The computation step $\boxed{k \mid l \mid m} \xrightarrow{\text{inc}(0)} \boxed{k+1 \mid l+1 \mid m}$ is mimicked by:

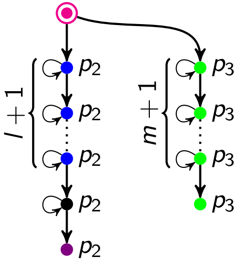
$encoding(\boxed{k \mid l \mid m}) \otimes encoding(\text{inc}(0)) =$



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

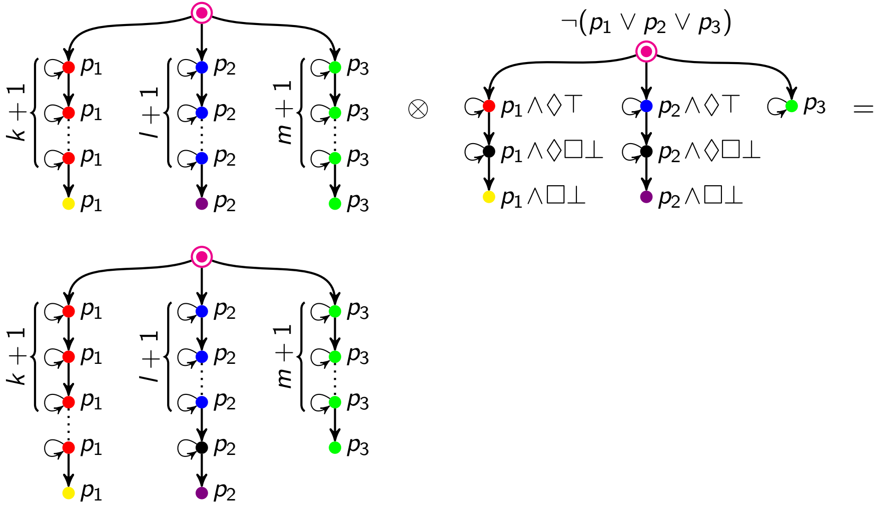
$$encoding(\boxed{k \mid l \mid m}) \otimes encoding(\text{inc}(0)) =$$





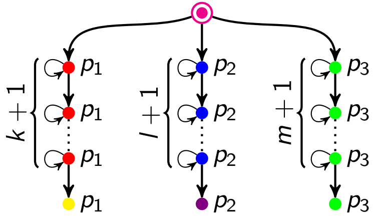
The computation step $\boxed{k \mid l \mid m} \xrightarrow{\text{inc}(0)} \boxed{k+1 \mid l+1 \mid m}$ is mimicked by:

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

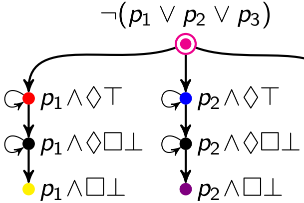


The computation step $\boxed{k \mid l \mid m} \xrightarrow{\text{inc}(0)} \boxed{k+1 \mid l+1 \mid m}$ is mimicked by:

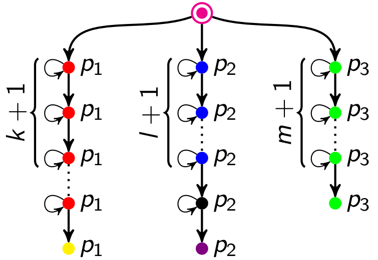
$$\textit{encoding}(\boxed{k \mid l \mid m}) \otimes \textit{encoding}(\text{inc}(0)) =$$



$\otimes$



$=$



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

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

L	transitive	Euclidean	reflexive
K			
KT			✓
K4	✓		
K45	✓	✓	
S4	✓		✓
S5	✓	✓	✓

← belief (K45)

← knowledge (S5)

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

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

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

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

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

Semantics:

$$\begin{array}{lll} \mathcal{M}, w \models [(\mathcal{E}, e)]\phi & \text{iff} & \mathcal{M}, w \models pre(e) \text{ implies } (\mathcal{M}, w) \otimes (\mathcal{E}, e) \models \phi \\ \mathcal{M}, w \models [\pi \cup \gamma]\phi & \text{iff} & \mathcal{M}, w \models [\pi]\phi \text{ and } \mathcal{M}, w \models [\gamma]\phi \\ \mathcal{M}, w \models [\pi ; \gamma]\phi & \text{iff} & \mathcal{M}, w \models [\pi][\gamma]\phi \\ \mathcal{M}, w \models [\pi^*]\phi & \text{iff} & \mathcal{M}, w \models [\pi]^n\phi, \text{ for all } n \end{array}$$

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

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

Theorem

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

Proof.

The plan existence problem considered above is reducible to the model checking problem of $\mathcal{L}^*_{DEL}$: Consider an epistemic planning task $\mathcal{T} = (s_0, \{a_1, \dots, a_m\}, \phi_g)$. $\mathcal{T}$ 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**, ≥ 3 **agents** [Bolander & Andersen, JANCL 2011].

Summary and related work

- Previously known undecidability results for DEL-based epistemic planning: S5, **with postconditions**, ≥ 3 **agents** [Bolander & Andersen, JANCL 2011].
- New results presented here: S5, **without postconditions**, ≥ 2 **agents** [Aucher & Bolander, IJCAI 2013].

Summary and related work

- Previously known undecidability results for DEL-based epistemic planning: S5, **with postconditions**, ≥ 3 **agents** [Bolander & Andersen, JANCL 2011].
- New results presented here: S5, **without postconditions**, ≥ 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.

Summary and related work

- Previously known undecidability results for DEL-based epistemic planning: S5, **with postconditions**, ≥ 3 **agents** [Bolander & Andersen, JANCL 2011].
- New results presented here: S5, **without postconditions**, ≥ 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**, ≥ 3 **agents** [Bolander & Andersen, JANCL 2011].
- New results presented here: S5, **without postconditions**, ≥ 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, **with postconditions**, ≥ 3 **agents** [Bolander & Andersen, JANCL 2011].
- New results presented here: S5, **without postconditions**, ≥ 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

- 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.