Learning Action Models: Qualitative Approach

Thomas Bolander, DTU Compute, Tech Univ of Denmark
Joint work with: Nina Gierasimczuk, ILLC, Amsterdam
(based on paper to appear at LORI 2015)

DTU Compute
Department of Applied Mathematics and Computer Science

Introduction

What our paper is about:

- Formal learning theory applied to dynamic epistemic logic (DEL).
- First paper to study the problem of learnability of action models in DEL.
- The goal is build agents that can **learn to plan**.

Our results are only the first few unsteady baby steps in action model learning. The **really** interesting stuff is all the future work...

DEL by example: A hidden coin toss

We use the **action models** of DEL [Baltag *et al.*, 1998] with added postconditions (ontic actions) as in [Ditmarsch *et al.*, 2008].

Let b mean "the coin faces the black side up".

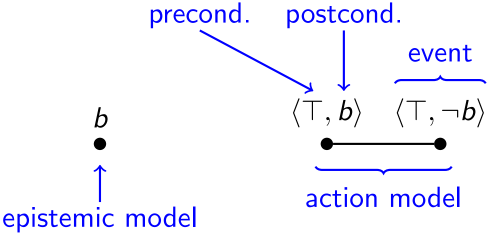
$$\overset{b}{\bullet}$$

epistemic model

DEL by example: A hidden coin toss

We use the **action models** of DEL [Baltag *et al.*, 1998] with added postconditions (ontic actions) as in [Ditmarsch *et al.*, 2008].

Let b mean "the coin faces the black side up".

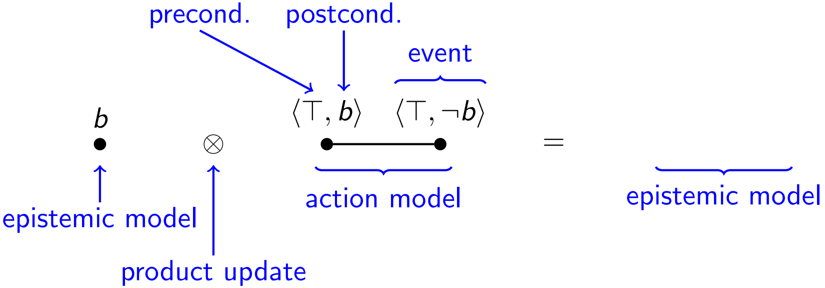


- **Action model**: Represent the action of the hidden coin toss.

DEL by example: A hidden coin toss

We use the **action models** of DEL [Baltag *et al.*, 1998] with added postconditions (ontic actions) as in [Ditmarsch *et al.*, 2008].

Let b mean "the coin faces the black side up".

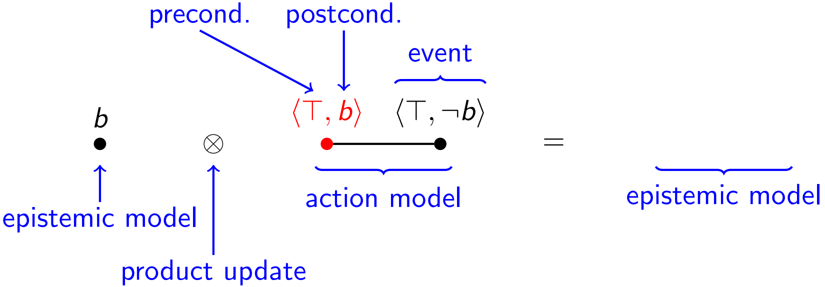


- **Action model**: Represent the action of the hidden coin toss.
- **Product update**: The updated model represents the situation after the action has taken place.

DEL by example: A hidden coin toss

We use the **action models** of DEL [Baltag *et al.*, 1998] with added postconditions (ontic actions) as in [Ditmarsch *et al.*, 2008].

Let b mean "the coin faces the black side up".

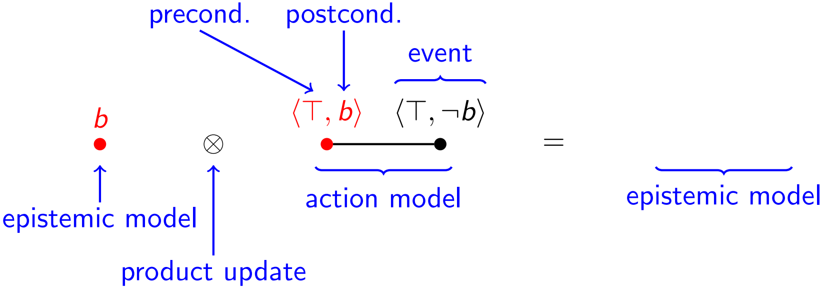


- **Action model**: Represent the action of the hidden coin toss.
- **Product update**: The updated model represents the situation after the action has taken place.

DEL by example: A hidden coin toss

We use the **action models** of DEL [Baltag *et al.*, 1998] with added postconditions (ontic actions) as in [Ditmarsch *et al.*, 2008].

Let b mean "the coin faces the black side up".

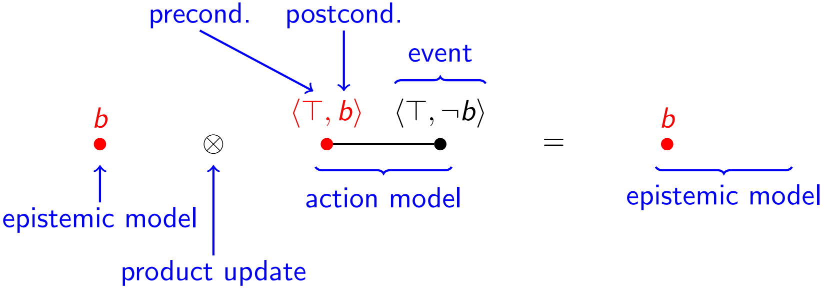


- **Action model**: Represent the action of the hidden coin toss.
- **Product update**: The updated model represents the situation after the action has taken place.

DEL by example: A hidden coin toss

We use the **action models** of DEL [Baltag *et al.*, 1998] with added postconditions (ontic actions) as in [Ditmarsch *et al.*, 2008].

Let b mean "the coin faces the black side up".

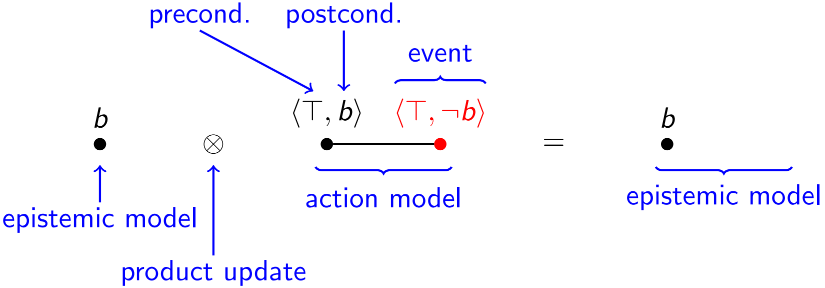


- **Action model**: Represent the action of the hidden coin toss.
- **Product update**: The updated model represents the situation after the action has taken place.

DEL by example: A hidden coin toss

We use the **action models** of DEL [Baltag *et al.*, 1998] with added postconditions (ontic actions) as in [Ditmarsch *et al.*, 2008].

Let b mean "the coin faces the black side up".

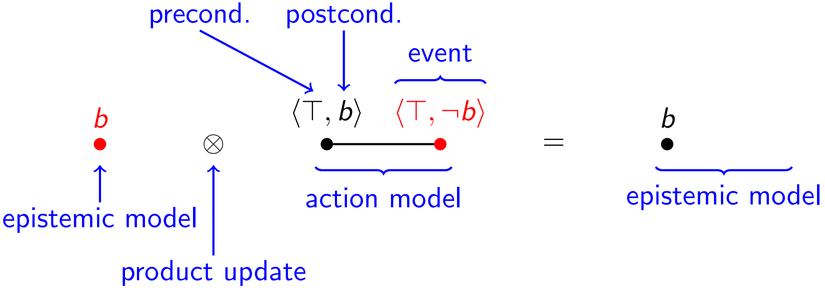


- **Action model**: Represent the action of the hidden coin toss.
- **Product update**: The updated model represents the situation after the action has taken place.

DEL by example: A hidden coin toss

We use the **action models** of DEL [Baltag *et al.*, 1998] with added postconditions (ontic actions) as in [Ditmarsch *et al.*, 2008].

Let b mean "the coin faces the black side up".

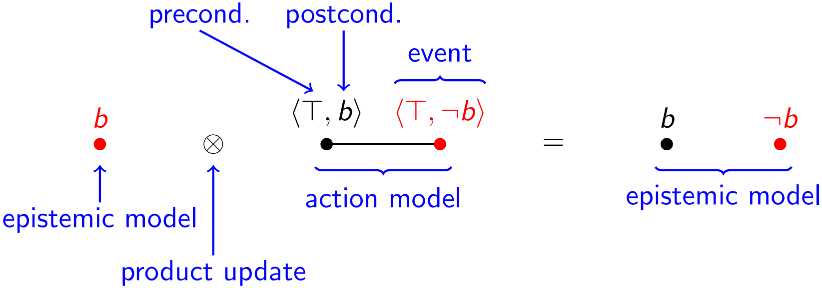


- **Action model**: Represent the action of the hidden coin toss.
- **Product update**: The updated model represents the situation after the action has taken place.

DEL by example: A hidden coin toss

We use the **action models** of DEL [Baltag *et al.*, 1998] with added postconditions (ontic actions) as in [Ditmarsch *et al.*, 2008].

Let b mean "the coin faces the black side up".

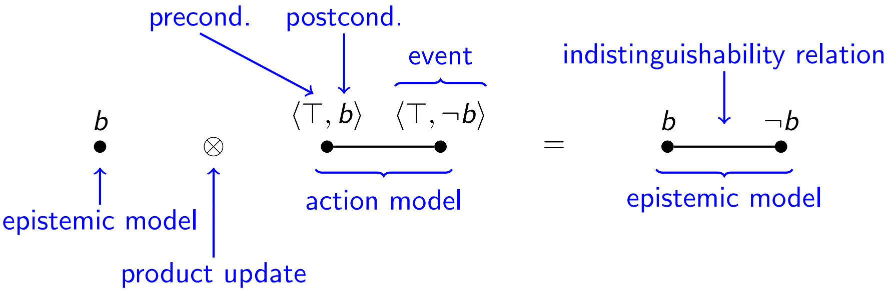


- **Action model**: Represent the action of the hidden coin toss.
- **Product update**: The updated model represents the situation after the action has taken place.

DEL by example: A hidden coin toss

We use the **action models** of DEL [Baltag *et al.*, 1998] with added postconditions (ontic actions) as in [Ditmarsch *et al.*, 2008].

Let b mean "the coin faces the black side up".



- **Action model**: Represent the action of the hidden coin toss.
- **Product update**: The updated model represents the situation after the action has taken place.

Learning facts vs. learning actions

Learning facts by eliminating nodes in epistemic models:

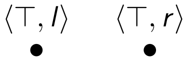
Learning facts vs. learning actions

Learning facts by eliminating nodes in epistemic models:

Learning facts vs. learning actions

Learning facts by eliminating nodes in epistemic models:

Learning actions by eliminating nodes in action models:



Learning facts vs. learning actions

Learning facts by eliminating nodes in epistemic models:

Learning actions by eliminating nodes in action models:

Observations, streams and identifiability

- Agents learn actions (action models) by a **stream** (infinite sequence) of **observations** (s, s') for that action: when executing the action in state s, state s' will result.
- **Finite identifiability**: after a finite sequence of observations, the agent says "stop" and identifies the correct action model.
- **Identifiability in the limit**: after a finite sequence of observations, the agent settles on a particular action model and never changes her mind (but is never able to say "stop").

Example. Possible stream on language with a single proposition p:

$$(\emptyset, \{p\}), (\{p\}, \emptyset), (\emptyset, \{p\}), (\{p\}, \emptyset), (\emptyset, \{p\}), (\{p\}, \emptyset), \ldots$$

Basic results on learnability

Restrictions on action models (actions) imposed in **all** of the following (including all results):

- Only **fully observable** actions: partially observable are not learnable in the strict sense.
- Only **propositional** actions: all preconditions of all events are formulas of propositional logic (not epistemic formulas).

Basic results on learnability

Restrictions on action models (actions) imposed in **all** of the following (including all results):

- Only **fully observable** actions: partially observable are not learnable in the strict sense.
- Only **propositional** actions: all preconditions of all events are formulas of propositional logic (not epistemic formulas).

Theorem 1. The set of deterministic actions is finitely identifiable.

Basic results on learnability

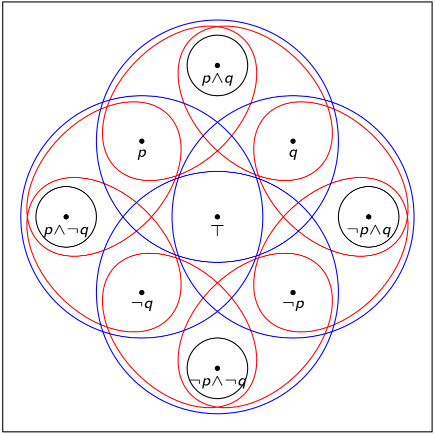
Restrictions on action models (actions) imposed in **all** of the following (including all results):

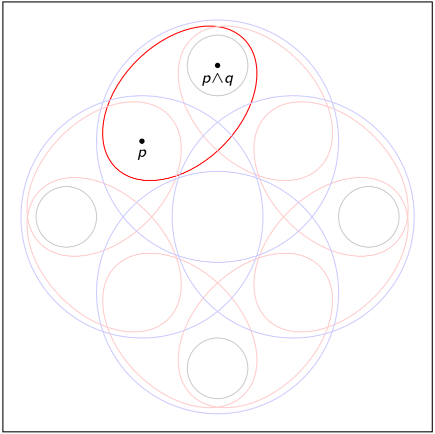
- Only **fully observable** actions: partially observable are not learnable in the strict sense.
- Only **propositional** actions: all preconditions of all events are formulas of propositional logic (not epistemic formulas).

Theorem 1. The set of deterministic actions is finitely identifiable.

Theorem 2. The set of (possibly non-deterministic) actions is not finitely identifiable, only identifiable in the limit.

Learning actions via update: precondition-free atomic actions





Left: Initial action model containing all possible postconditions. The blue and red sets correspond to possible observations.

Right: The action model after receiving the observation $(\{q\}, \{p, q\})$.

Learning actions via update: (non-atomic) deterministic actions with maximal preconditions

Maximal preconditions: all preconditions are maximally consistent conjunctions of propositional literals (e.g. $p \wedge \neg q$ in the language over $\{p, q\}$).

Examples in the language over a single proposition $\{p\}$.

$\langle p, \top \rangle \quad \langle \neg p, \top \rangle$
$\langle p, \neg p \rangle \quad \langle \neg p, p \rangle$

Learning actions via update: (non-atomic) deterministic actions with maximal preconditions

Maximal preconditions: all preconditions are maximally consistent conjunctions of propositional literals (e.g. $p \land \neg q$ in the language over $\{p, q\}$).

Examples in the language over a single proposition $\{p\}$.

observation:
$(\emptyset, \{p\})$

$$\begin{array}{ll} \langle p, \top \rangle & \langle \neg p, \top \rangle \\ \langle p, \neg p \rangle & \langle \neg p, p \rangle \end{array} \longrightarrow$$

Learning actions via update: (non-atomic) deterministic actions with maximal preconditions

Maximal preconditions: all preconditions are maximally consistent conjunctions of propositional literals (e.g. $p \wedge \neg q$ in the language over $\{p, q\}$).

Examples in the language over a single proposition $\{p\}$.

$$\begin{array}{ll} \langle p, \top \rangle & \langle \neg p, \top \rangle \\ \langle p, \neg p \rangle & \langle \neg p, p \rangle \end{array} \xrightarrow{\substack{\text{observation:} \\ (\emptyset, \{p\})}} \begin{array}{ll} \langle p, \top \rangle & \cancel{\langle \neg p, \top \rangle} \\ \langle p, \neg p \rangle & \langle \neg p, p \rangle \end{array}$$

Learning actions via update: (non-atomic) deterministic actions with maximal preconditions

Maximal preconditions: all preconditions are maximally consistent conjunctions of propositional literals (e.g. $p \wedge \neg q$ in the language over $\{p, q\}$).

Examples in the language over a single proposition $\{p\}$.

$$\boxed{\begin{array}{ll} \langle p, \top \rangle & \langle \neg p, \top \rangle \\ \langle p, \neg p \rangle & \langle \neg p, p \rangle \end{array}} \xrightarrow{\text{observation: } (\emptyset, \{p\})} \boxed{\begin{array}{ll} \langle p, \top \rangle & \cancel{\langle \neg p, \top \rangle} \\ \langle p, \neg p \rangle & \langle \neg p, p \rangle \end{array}} \xrightarrow{\text{observation: } (\{p\}, \emptyset)}$$

Learning actions via update: (non-atomic) deterministic actions with maximal preconditions

Maximal preconditions: all preconditions are maximally consistent conjunctions of propositional literals (e.g. $p \wedge \neg q$ in the language over $\{p, q\}$).

Examples in the language over a single proposition $\{p\}$.

$$\begin{array}{cc} \langle p, \top\rangle & \langle \neg p, \top\rangle \\ \langle p, \neg p\rangle & \langle \neg p, p\rangle \end{array} \xrightarrow{\text{observation: } (\emptyset, \{p\})} \begin{array}{cc} \langle p, \top\rangle & \cancel{\langle \neg p, \top\rangle} \\ \langle p, \neg p\rangle & \langle \neg p, p\rangle \end{array} \xrightarrow{\text{observation: } (\{p\}, \emptyset)} \begin{array}{cc} \cancel{\langle p, \top\rangle} & \cancel{\langle \neg p, \top\rangle} \\ \langle p, \neg p\rangle & \langle \neg p, p\rangle \end{array}$$

Learning actions via update: (non-atomic) deterministic actions with maximal preconditions

Maximal preconditions: all preconditions are maximally consistent conjunctions of propositional literals (e.g. $p \wedge \neg q$ in the language over $\{p, q\}$).

Examples in the language over a single proposition $\{p\}$.

$$\begin{array}{cc} \langle p, \top \rangle & \langle \neg p, \top \rangle \\ \langle p, \neg p \rangle & \langle \neg p, p \rangle \end{array} \xrightarrow[(\emptyset, \{p\})]{\text{observation:}} \begin{array}{cc} \langle p, \top \rangle & \cancel{\langle \neg p, \top \rangle} \\ \langle p, \neg p \rangle & \langle \neg p, p \rangle \end{array} \xrightarrow[(\{p\}, \emptyset)]{\text{observation:}} \begin{array}{cc} \cancel{\langle p, \top \rangle} & \cancel{\langle \neg p, \top \rangle} \\ \langle p, \neg p \rangle & \langle \neg p, p \rangle \end{array}$$

$$\begin{array}{cc} \langle p, \top \rangle & \langle \neg p, \top \rangle \\ \langle p, \neg p \rangle & \langle \neg p, p \rangle \end{array}$$

Learning actions via update: (non-atomic) deterministic actions with maximal preconditions

Maximal preconditions: all preconditions are maximally consistent conjunctions of propositional literals (e.g. $p \wedge \neg q$ in the language over $\{p, q\}$).

Examples in the language over a single proposition $\{p\}$.

$$\begin{array}{cc} \langle p, \top \rangle & \langle \neg p, \top \rangle \\ \langle p, \neg p \rangle & \langle \neg p, p \rangle \end{array} \xrightarrow{\text{observation: } (\emptyset, \{p\})} \begin{array}{cc} \langle p, \top \rangle & \cancel{\langle \neg p, \top \rangle} \\ \langle p, \neg p \rangle & \langle \neg p, p \rangle \end{array} \xrightarrow{\text{observation: } (\{p\}, \emptyset)} \begin{array}{cc} \cancel{\langle p, \top \rangle} & \cancel{\langle \neg p, \top \rangle} \\ \langle p, \neg p \rangle & \langle \neg p, p \rangle \end{array}$$

$$\begin{array}{cc} \langle p, \top \rangle & \langle \neg p, \top \rangle \\ \langle p, \neg p \rangle & \langle \neg p, p \rangle \end{array} \xrightarrow{\text{observation: } (\emptyset, \{p\})}$$

Learning actions via update: (non-atomic) deterministic actions with maximal preconditions

Maximal preconditions: all preconditions are maximally consistent conjunctions of propositional literals (e.g. $p \wedge \neg q$ in the language over $\{p, q\}$).

Examples in the language over a single proposition $\{p\}$.

$$\begin{array}{cc}\langle p,\top\rangle & \langle\neg p,\top\rangle\\ \langle p,\neg p\rangle & \langle\neg p,p\rangle\end{array} \xrightarrow[(\emptyset,\{p\})]{\text{observation:}} \begin{array}{cc}\langle p,\top\rangle & \cancel{\langle\neg p,\top\rangle}\\ \langle p,\neg p\rangle & \langle\neg p,p\rangle\end{array} \xrightarrow[(\{p\},\emptyset)]{\text{observation:}} \begin{array}{cc}\cancel{\langle p,\top\rangle} & \cancel{\langle\neg p,\top\rangle}\\ \langle p,\neg p\rangle & \langle\neg p,p\rangle\end{array}$$

$$\begin{array}{cc}\langle p,\top\rangle & \langle\neg p,\top\rangle\\ \langle p,\neg p\rangle & \langle\neg p,p\rangle\end{array} \xrightarrow[(\emptyset,\{p\})]{\text{observation:}} \begin{array}{cc}\langle p,\top\rangle & \cancel{\langle\neg p,\top\rangle}\\ \langle p,\neg p\rangle & \langle\neg p,p\rangle\end{array}$$

Learning actions via update: (non-atomic) deterministic actions with maximal preconditions

Maximal preconditions: all preconditions are maximally consistent conjunctions of propositional literals (e.g. $p \wedge \neg q$ in the language over $\{p, q\}$).

Examples in the language over a single proposition $\{p\}$.

$$\begin{array}{cc} \langle p, \top \rangle & \langle \neg p, \top \rangle \\ \langle p, \neg p \rangle & \langle \neg p, p \rangle \end{array} \xrightarrow{\text{observation: } (\emptyset, \{p\})} \begin{array}{cc} \langle p, \top \rangle & \cancel{\langle \neg p, \top \rangle} \\ \langle p, \neg p \rangle & \langle \neg p, p \rangle \end{array} \xrightarrow{\text{observation: } (\{p\}, \emptyset)} \begin{array}{cc} \cancel{\langle p, \top \rangle} & \cancel{\langle \neg p, \top \rangle} \\ \langle p, \neg p \rangle & \langle \neg p, p \rangle \end{array}$$

$$\begin{array}{cc} \langle p, \top \rangle & \langle \neg p, \top \rangle \\ \langle p, \neg p \rangle & \langle \neg p, p \rangle \end{array} \xrightarrow{\text{observation: } (\emptyset, \{p\})} \begin{array}{cc} \langle p, \top \rangle & \cancel{\langle \neg p, \top \rangle} \\ \langle p, \neg p \rangle & \langle \neg p, p \rangle \end{array} \xrightarrow{\text{observation: } (\{p\}, \{p\})}$$

Learning actions via update: (non-atomic) deterministic actions with maximal preconditions

Maximal preconditions: all preconditions are maximally consistent conjunctions of propositional literals (e.g. $p \land \neg q$ in the language over $\{p, q\}$).

Examples in the language over a single proposition $\{p\}$.

$$\begin{array}{cc} \langle p, \top\rangle & \langle \neg p, \top\rangle \\ \langle p, \neg p\rangle & \langle \neg p, p\rangle \end{array} \xrightarrow{\text{observation: } (\emptyset, \{p\})} \begin{array}{cc} \langle p, \top\rangle & \cancel{\langle \neg p, \top\rangle} \\ \langle p, \neg p\rangle & \langle \neg p, p\rangle \end{array} \xrightarrow{\text{observation: } (\{p\}, \emptyset)} \begin{array}{cc} \cancel{\langle p, \top\rangle} & \cancel{\langle \neg p, \top\rangle} \\ \langle p, \neg p\rangle & \langle \neg p, p\rangle \end{array}$$

$$\begin{array}{cc} \langle p, \top\rangle & \langle \neg p, \top\rangle \\ \langle p, \neg p\rangle & \langle \neg p, p\rangle \end{array} \xrightarrow{\text{observation: } (\emptyset, \{p\})} \begin{array}{cc} \langle p, \top\rangle & \cancel{\langle \neg p, \top\rangle} \\ \langle p, \neg p\rangle & \langle \neg p, p\rangle \end{array} \xrightarrow{\text{observation: } (\{p\}, \{p\})} \begin{array}{cc} \langle p, \top\rangle & \cancel{\langle \neg p, \top\rangle} \\ \cancel{\langle p, \neg p\rangle} & \langle \neg p, p\rangle \end{array}$$

Learning actions via update: deterministic actions with minimal preconditions

A simple update is no longer sufficient. But sufficient to always conjecture the set of minimal events using the following order:

$$e \leq e' \quad := \quad \mathit{pre}(e') \models \mathit{pre}(e) \text{ and } \mathit{post}(e') \models \mathit{post}(e)$$

Example. $\langle p, r\rangle \leq \langle p \wedge q, r \wedge s\rangle$. (Ockham's razor, cf. Kevin's talk!)

Important: All non-minimal events are preserved "in the background".

Example. Learning the functioning of an n-bit counter. Case $n = 2$:

Current action model:

$\langle \top, \top \rangle$

Current state of counter:

b_1	b_2
0	0

Learning actions via update: deterministic actions with minimal preconditions

A simple update is no longer sufficient. But sufficient to always conjecture the set of minimal events using the following order:

$$e \leq e' \quad := \quad pre(e') \models pre(e) \text{ and } post(e') \models post(e)$$

Example. $\langle p, r \rangle \leq \langle p \wedge q, r \wedge s \rangle$. (Ockham's razor, cf. Kevin's talk!)

Important: All non-minimal events are preserved "in the background".

Example. Learning the functioning of an n-bit counter. Case $n = 2$:

Current action model:

$\cancel{\langle \top, \top \rangle}$,
$\langle \neg b_1, b_2 \rangle, \langle b_1, \top \rangle, \langle \neg b_2, b_2 \rangle, \langle b_2, \top \rangle$

Current state of counter:

b_1	b_2
0	1

Learning actions via update: deterministic actions with minimal preconditions

A simple update is no longer sufficient. But sufficient to always conjecture the set of minimal events using the following order:

$$e \leq e' \quad := \quad \mathit{pre}(e') \models \mathit{pre}(e) \ \text{ and } \ \mathit{post}(e') \models \mathit{post}(e)$$

Example. $\langle p, r \rangle \leq \langle p \wedge q, r \wedge s \rangle$. (Ockham's razor, cf. Kevin's talk!)

Important: All non-minimal events are preserved "in the background".

Example. Learning the functioning of an n-bit counter. Case $n = 2$:

Current action model:

$\cancel{\langle \top, \top \rangle}$,
$\cancel{\langle \neg b_1, b_2 \rangle}, \langle b_1, \top \rangle, \langle \neg b_2, b_2 \rangle, \cancel{\langle b_2, \top \rangle}$,
$\langle \neg b_1 \wedge b_2, b_1 \wedge \neg b_2 \rangle$

Current state of counter:

b_1	b_2
1	0

Learning actions via update: deterministic actions with minimal preconditions

A simple update is no longer sufficient. But sufficient to always conjecture the set of minimal events using the following order:

$$e \leq e' \quad := \quad pre(e') \models pre(e) \text{ and } post(e') \models post(e)$$

Example. $\langle p, r \rangle \leq \langle p \wedge q, r \wedge s \rangle$. (Ockham's razor, cf. Kevin's talk!)

Important: All non-minimal events are preserved "in the background".

Example. Learning the functioning of an n-bit counter. Case $n = 2$:

Current action model:

$\cancel{\langle \top, \top \rangle}$,
$\cancel{\langle \neg b_1, b_2 \rangle}, \langle b_1, \top \rangle, \langle \neg b_2, b_2 \rangle, \cancel{\langle b_2, \top \rangle}$,
$\langle \neg b_1 \wedge b_2, b_1 \wedge \neg b_2 \rangle$

Current state of counter:

b_1	b_2
1	0

Learning actions via update: deterministic actions with minimal preconditions

A simple update is no longer sufficient. But sufficient to always conjecture the set of minimal events using the following order:

$$e \leq e' \quad := \quad pre(e') \models pre(e) \text{ and } post(e') \models post(e)$$

Example. $\langle p, r \rangle \leq \langle p \wedge q, r \wedge s \rangle$. (Ockham's razor, cf. Kevin's talk!)

Important: All non-minimal events are preserved "in the background".

Example. Learning the functioning of an n-bit counter. Case $n = 2$:

Current action model:

$\cancel{\langle \top, \top \rangle}$,
$\cancel{\langle \neg b_1, b_2 \rangle}, \langle b_1, \top \rangle, \langle \neg b_2, b_2 \rangle, \cancel{\langle b_2, \top \rangle}$,
$\langle \neg b_1 \wedge b_2, b_1 \wedge \neg b_2 \rangle$

Current state of counter:

b_1	b_2
1	0

Resulting action model: $n + 1$ events (instead of 2^n as in the case of maximal preconditions).

Also in the paper:

- **Action library learning**: Simultaneous learning of several different actions. Most relevant case for planning.

Also in the paper:

- **Action library learning**: Simultaneous learning of several different actions. Most relevant case for planning.

Related work in the automated planning literature:

- Walsh and Littman [2008] study qualitative learning of STRIPS action schemas. We are more general in successfully treating also: negative preconditions, negative postconditions, conditional effects.

Also in the paper:

- **Action library learning**: Simultaneous learning of several different actions. Most relevant case for planning.

Related work in the automated planning literature:

- Walsh and Littman [2008] study qualitative learning of STRIPS action schemas. We are more general in successfully treating also: negative preconditions, negative postconditions, conditional effects.

Future work:

- Extended classes of actions: arbitrary pre- and post-conditions, partial observability, multiple agents (joint learning).
- Computational complexity.
- Proactive learning (using consecutive streams).
- Ultimate goal: general learning-and-planning agents.