Dynamic epistemic logic

Tristan Charrier François Schwarzentruber

École Normale Supérieure Rennes

May 13, 2019

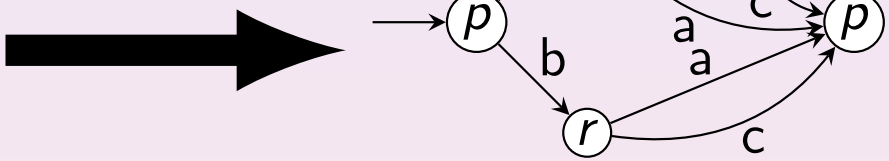
Outline

Outline

In the verification/model checking community

```
// HelloWin.cpp : Defines the entry point for the application.
//

#include "stdafx.h"
#include "resource.h"

#define MAX_LOADSTRING 100

// Global Variables:
HINSTANCE hInst;                                // current instance
TCHAR szTitle[MAX_LOADSTRING];                  // The title b
TCHAR szWindowClass[MAX_LOADSTRING];            // The

// Foward declarations of functions included in this code module:
ATOM                MyRegisterClass(HINSTANCE hInstance);
BOOL                InitInstance(HINSTANCE, int);
LRESULT CALLBACK    WndProc(HWND, UINT, WPARAM, LPARAM);
LRESULT CALLBACK    About(HWND, UINT, WPARAM, LPARAM);

int APIENTRY WinMain(HINSTANCE hInstance,
                     HINSTANCE hPrevInstance,
                     LPSTR     lpCmdLine,
                     int       nCmdShow)
{
    // TODO: Place code here.
    MSG msg;
    HACCEL hAccelTable;

    // Initialize global strings
    LoadString(hInstance, IDS_APP_TITLE, szTitle, MAX_LOADSTRING);
    LoadString(hInstance, IDC_HELLOWIN, szWindowClass, MAX_LOADSTRING);
    MyRegisterClass(hInstance);

    // Perform application initialization:
    if (!InitInstance (hInstance, nCmdShow))
    {
```

Program



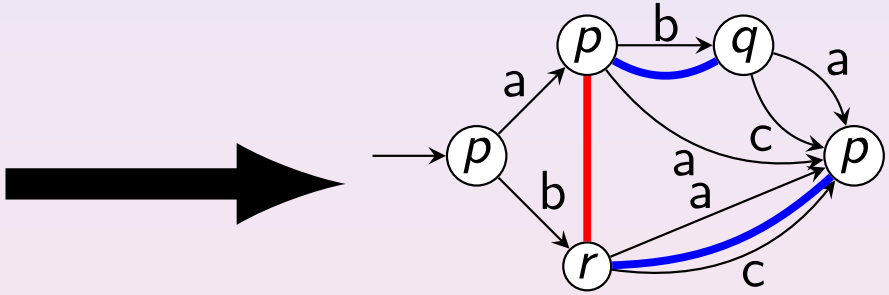
Transition system

Action = an edge $\longrightarrow$

In the verification/model checking community

```
// HelloWin.cpp : Defines the entry point for the application.
//

#include "stdafx.h"
#include "resource.h"

#define MAX_LOADSTRING 100

// Global Variables:
HINSTANCE hInst;                                // current instance
TCHAR szTitle[MAX_LOADSTRING];                  // The title b
TCHAR szWindowClass[MAX_LOADSTRING];            // The

// Foward declarations of functions included in this code module:
ATOM                MyRegisterClass(HINSTANCE hInstance);
BOOL                InitInstance(HINSTANCE, int);
LRESULT CALLBACK    WndProc(HWND, UINT, WPARAM, LPARAM);
LRESULT CALLBACK    About(HWND, UINT, WPARAM, LPARAM);

int APIENTRY WinMain(HINSTANCE hInstance,
                     HINSTANCE hPrevInstance,
                     LPSTR     lpCmdLine,
                     int       nCmdShow)
{
    // TODO: Place code here.
    MSG msg;
    HACCEL hAccelTable;

    // Initialize global strings
    LoadString(hInstance, IDS_APP_TITLE, szTitle, MAX_LOADSTRING);
    LoadString(hInstance, IDC_HELLOWIN, szWindowClass, MAX_LOADSTRING);
    MyRegisterClass(hInstance);

    // Perform application initialization:
    if (!InitInstance (hInstance, nCmdShow))
    {
```

Program



Transition system

Action = an edge $\longrightarrow$
Epistemic = edges

Outline

In philosophy / AI

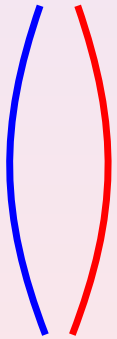
The mechanism of actions is important.

Public/private announcement	Announce 'She knows you hold $5\Diamond$'
Public action	play card $5\Diamond$
Private action	secretly remove card $5\Diamond$
Belief revision	learn p although believing $\neg p$

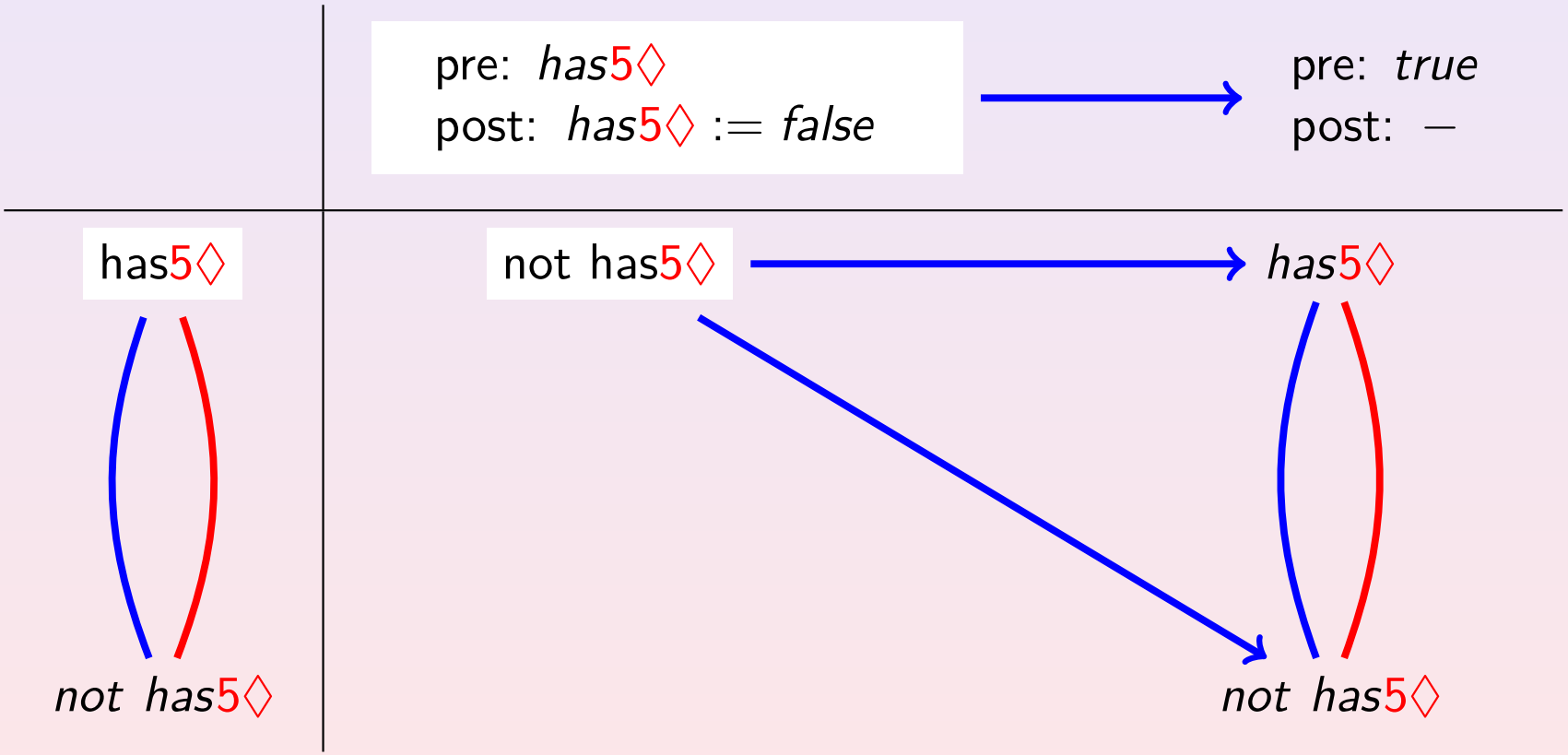
Solution: Dynamic epistemic logic

	State	Action
Classical planning	has5$\Diamond$	pre: *has*5$\Diamond$ post: *has*5$\Diamond$:= *false*

Solution: Dynamic epistemic logic

	State	Action
Classical planning	has5◊	pre: *has5◊* post: *has5◊* := *false*
DEL [Baltag et al. TARK 1998] [van Ditmarsch et al. 2007] = Kripkean models of classical planning	has5◊  *not has5◊*	pre: *has5◊* post: *has5◊* := *false*  pre: *true* post: –

Computing the next state: product update



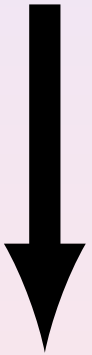
Outline

Syntactic specifications

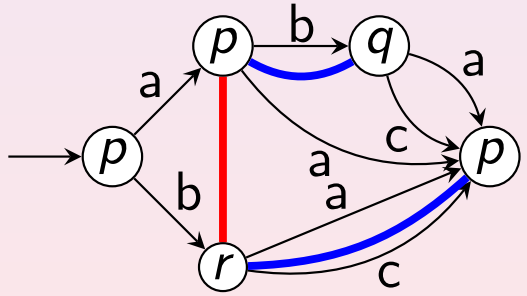
Game description language [Love et al. 2008] [Thielscher, IJCAI 2017]	agent a sees the game position
Flatland [Balbiani et al., IGPL 2014] [Gasquet, Goranko, _, AAMAS 2014] [Gasquet, Goranko, _, JAAMAS 2016]	agent a sees agent b
Visibility atoms [Charrier et al. KR 2016]	a sees the truth value of p
Paying attention to public announcements [Bolander et al. JoLLI 2016]	$B_a payAtt(b) \rightarrow [p!]B_a B_b p$
Asynchronous announcements [Knight et al. MS in CS 2019]	$[p!][read_a]K_a p$
Epistemic gossip [van Ditmarsch et al., JAL 2017]	$[call_{ab}]K_a secret_b$

From DEL to epistemic temporal logics

Syntactic specification



Models of
dynamic epistemic logic



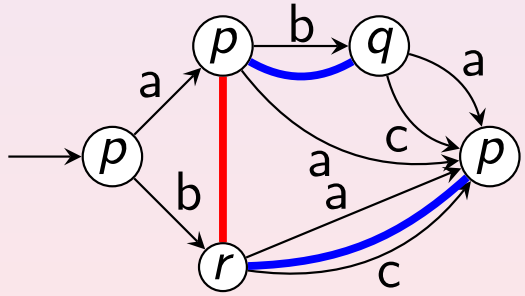
Epistemic temporal model

From DEL to epistemic temporal logics

Syntactic specification

- ✓ Easy to specify
- ✓ Succinct
- ✗ Ad-hoc languages
- ✗ Hand-crafted semantics

Models of
dynamic epistemic logic

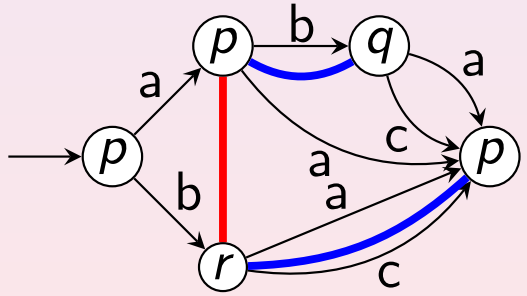


Epistemic temporal model

From DEL to epistemic temporal logics

Syntactic specification

Models of
dynamic epistemic logic



Epistemic temporal model

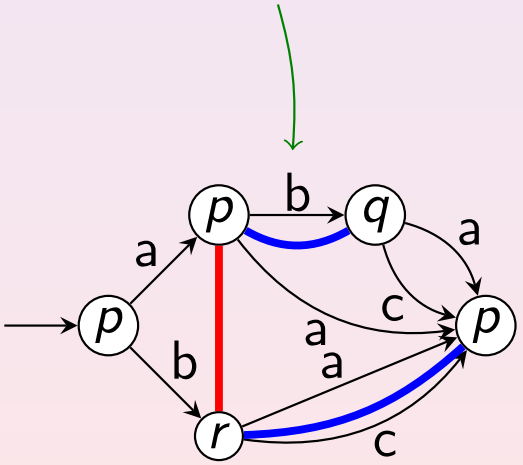
- ✓ Elegant Kripkean extension of classical planning
- ✓ Succinct
- ✓ Classification in terms of action types
- ✓ Has probabilistic extension
- ✓ Has extensions that encompass belief revision
- ✗ Perfect-recall only
- ✗ Synchronous only

From DEL to epistemic temporal logics

Syntactic specification

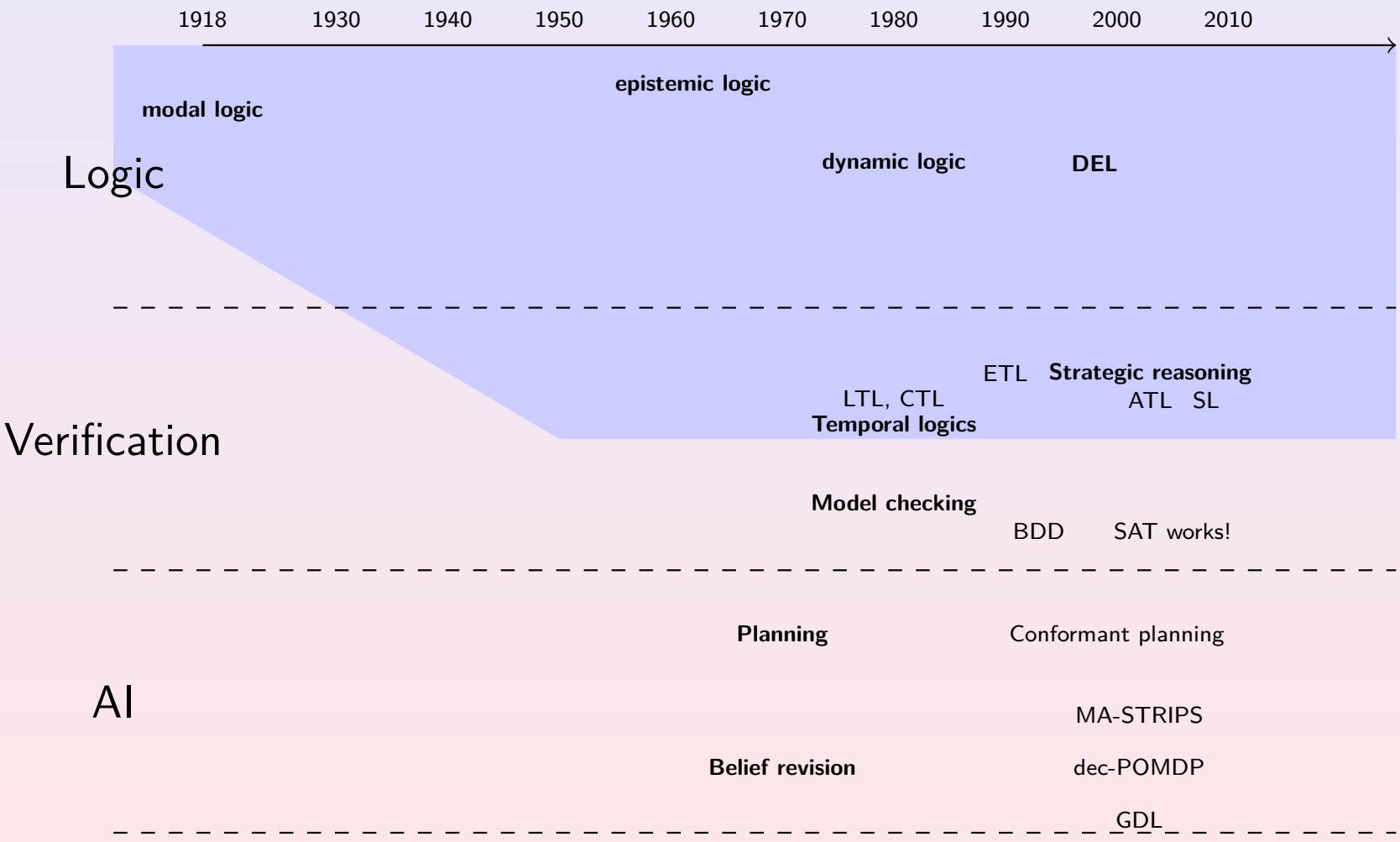
Models of
dynamic epistemic logic

- ✓ Elegant
- ✓ Allows for async/no perfect-recall semantics
- ✗ Type of actions lost
- ✗ Not Succinct (usually infinite)

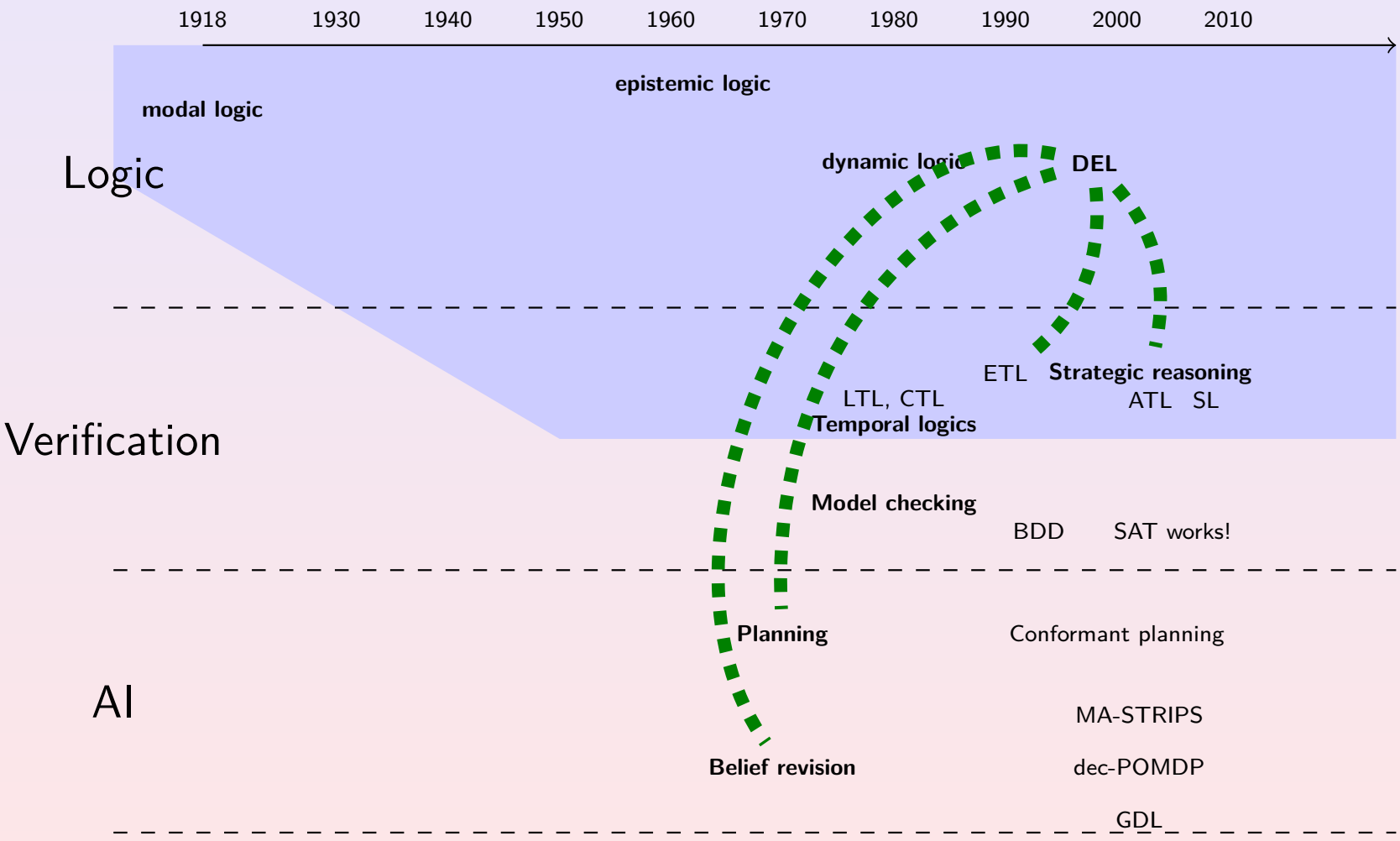


Epistemic temporal model

Timeline


1918
1930
1940
1950
1960
1970
1980
1990
2000
2010
epistemic logic
modal logic
Logic
dynamic logic
DEL
ETL
Strategic reasoning
LTL, CTL
ATL
SL
Temporal logics
Verification
Model checking
BDD
SAT works!
Planning
Conformant planning
AI
MA-STRIPS
Belief revision
dec-POMDP
GDL

Timeline


1918
1930
1940
1950
1960
1970
1980
1990
2000
2010
epistemic logic
modal logic
Logic
dynamic logic
DEL
ETL
Strategic reasoning
ATL
SL
LTL, CTL
Temporal logics
Verification
Model checking
BDD
SAT works!
Planning
Conformant planning
AI
MA-STRIPS
dec-POMDP
Belief revision
GDL

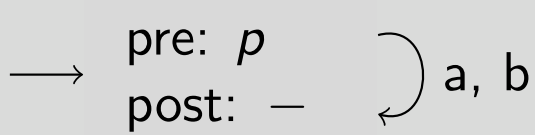
Outline

Outline

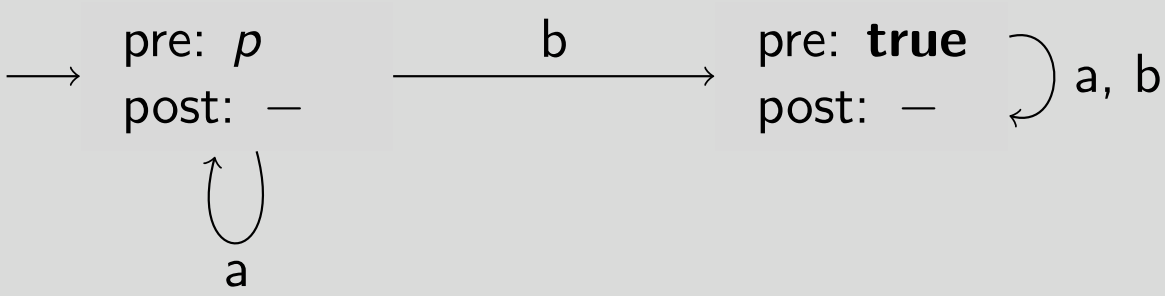
Examples of actions

[**baltag1998logic**]

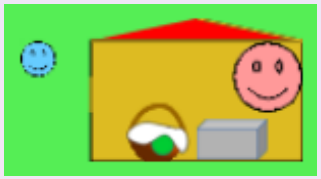
Example (Public announcement of "p")



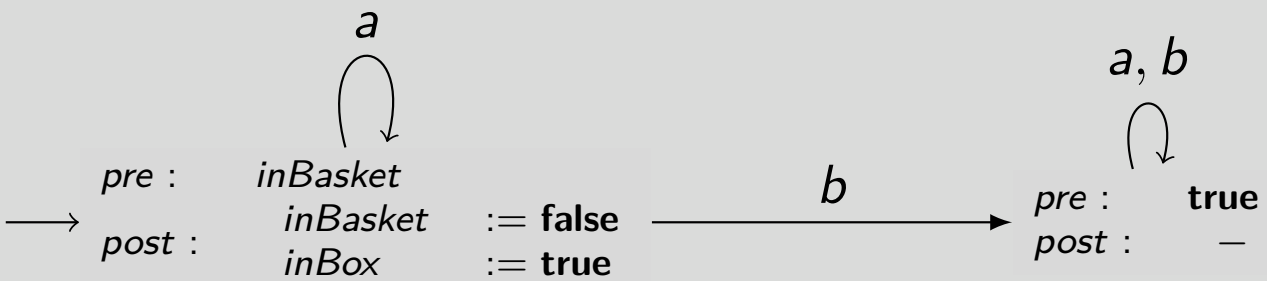
Example (Private announcement "p" to a)



Examples of actions

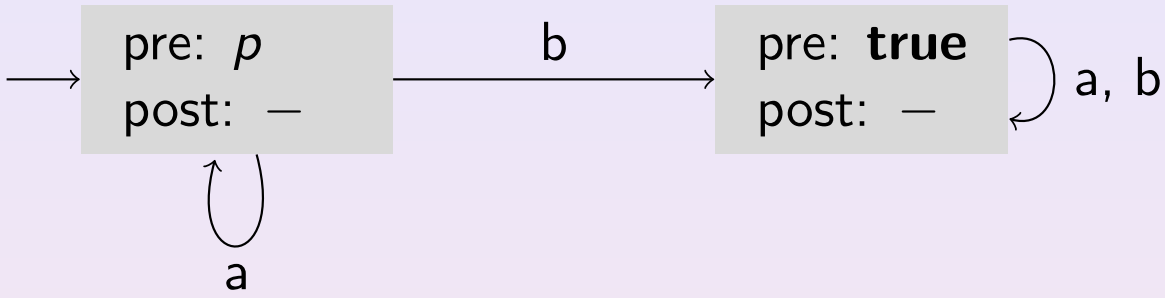


Example (Transfer marble from basket to box)



Outline

Actions



Definition

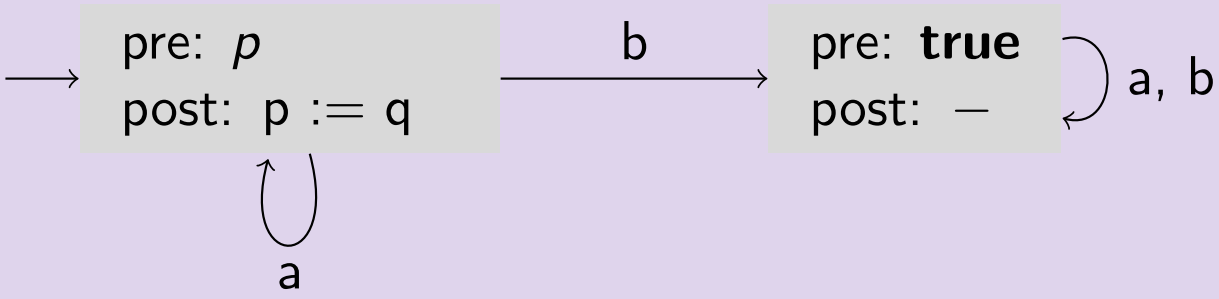
An event model $\mathcal{E} = (\mathsf{E}, (R_a^{\mathcal{E}})_{a \in AGT}, pre, post)$ is a tuple where:

- $\mathsf{E} = \{e, e', \dots\}$ is a non-empty finite set of possible events,
- $R_a^{\mathcal{E}} \subseteq \mathsf{E} \times \mathsf{E}$ is an accessibility relation on E for agent a,
- $pre : \mathsf{E} \to \mathcal{L}_{\mathbf{EL}}$ is a precondition function,
- $post : \mathsf{E} \times AP \to \mathcal{L}_{\mathbf{EL}}$ is a postcondition function.

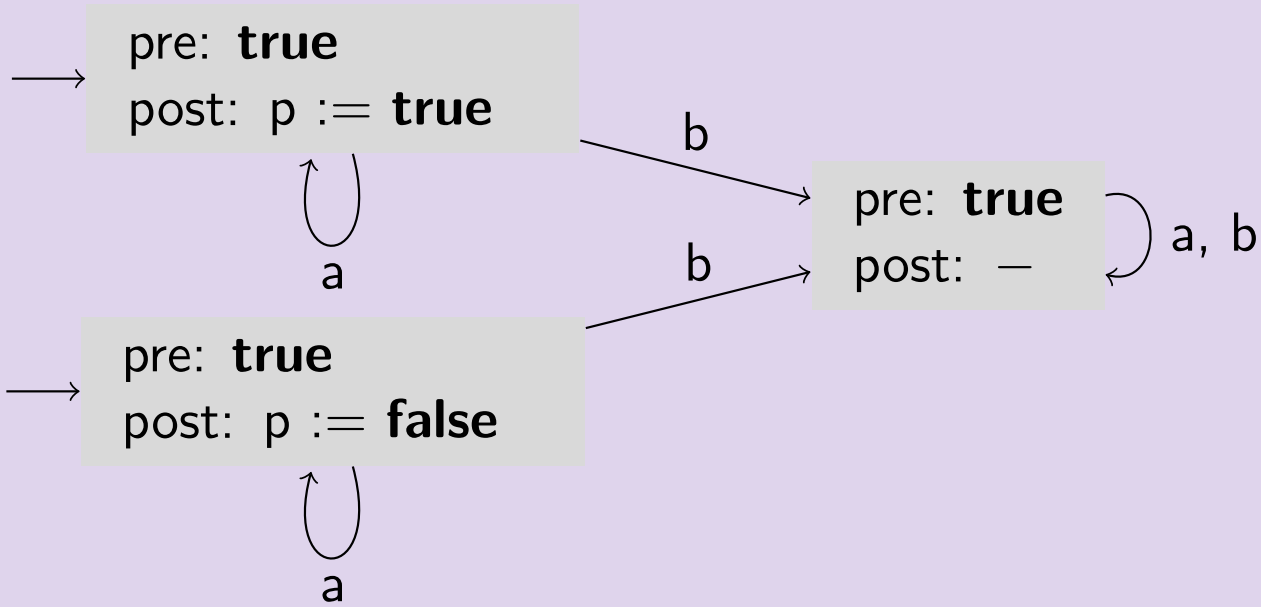
A pair $(\mathcal{E}, e)$ is called an action, where e represents the actual event of $(\mathcal{E}, e)$.
A pair $(\mathcal{E}, \mathsf{E}_0)$, for $\mathsf{E}_0 \subseteq \mathsf{E}$, is a non-deterministic action. The set E_0 is the set of triggerable events.

Deterministic and non-deterministic actions

Deterministic action = single-pointed event model $(\mathcal{E}, e)$



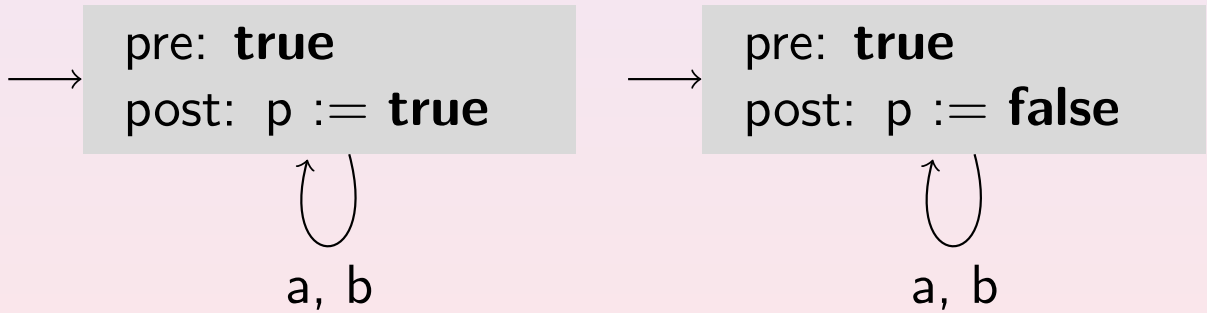
Non-deterministic action = multi-pointed event model



Public actions

Definition

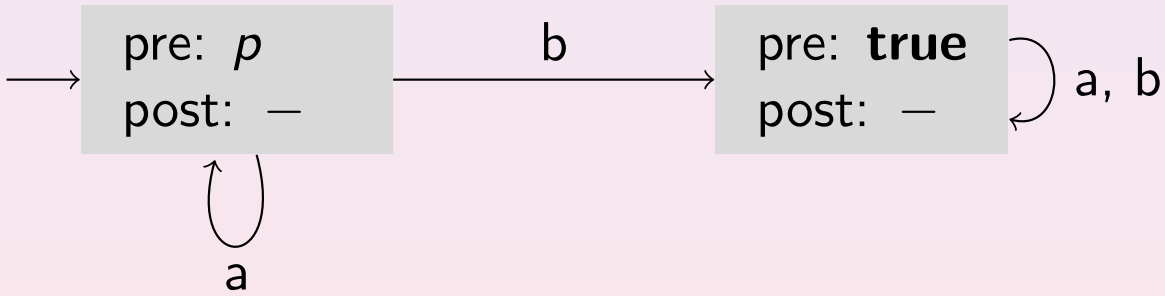
An action is said to be *public* if the accessibility relations in underlying event model are self-loops.



Non-ontic actions

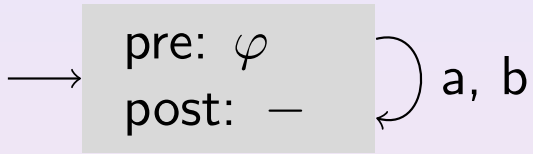
Definition

An action is said to be *non-ontic* if the postconditions are trivial: for all $e \in \mathrm{E}$, for all propositions $p \in AP$, $post(e, p) = p$.

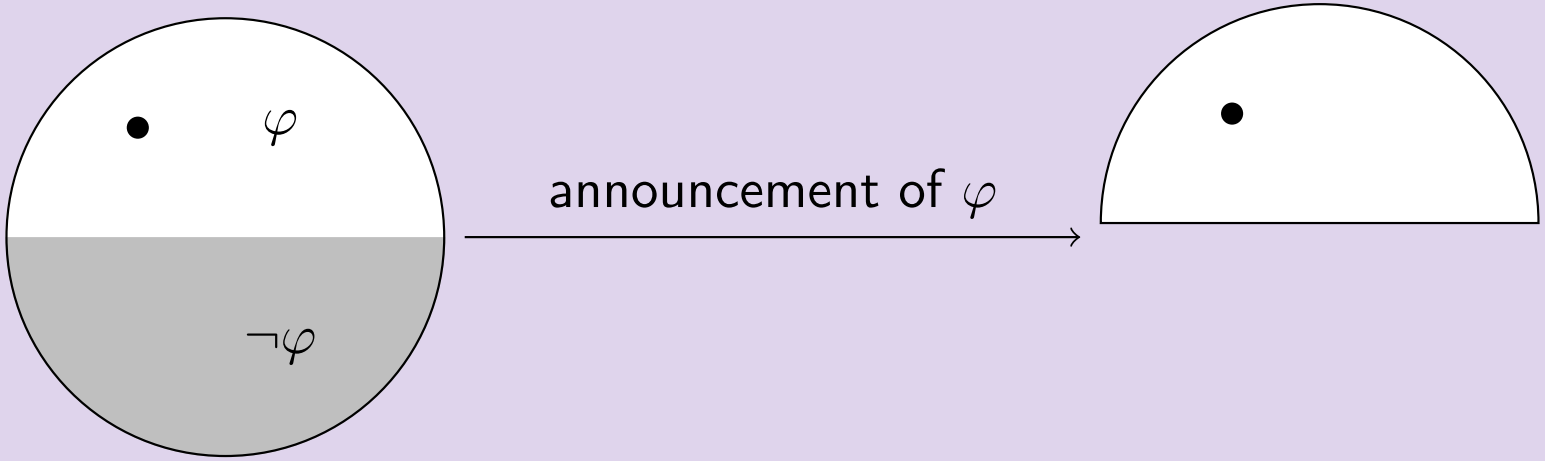


Outline

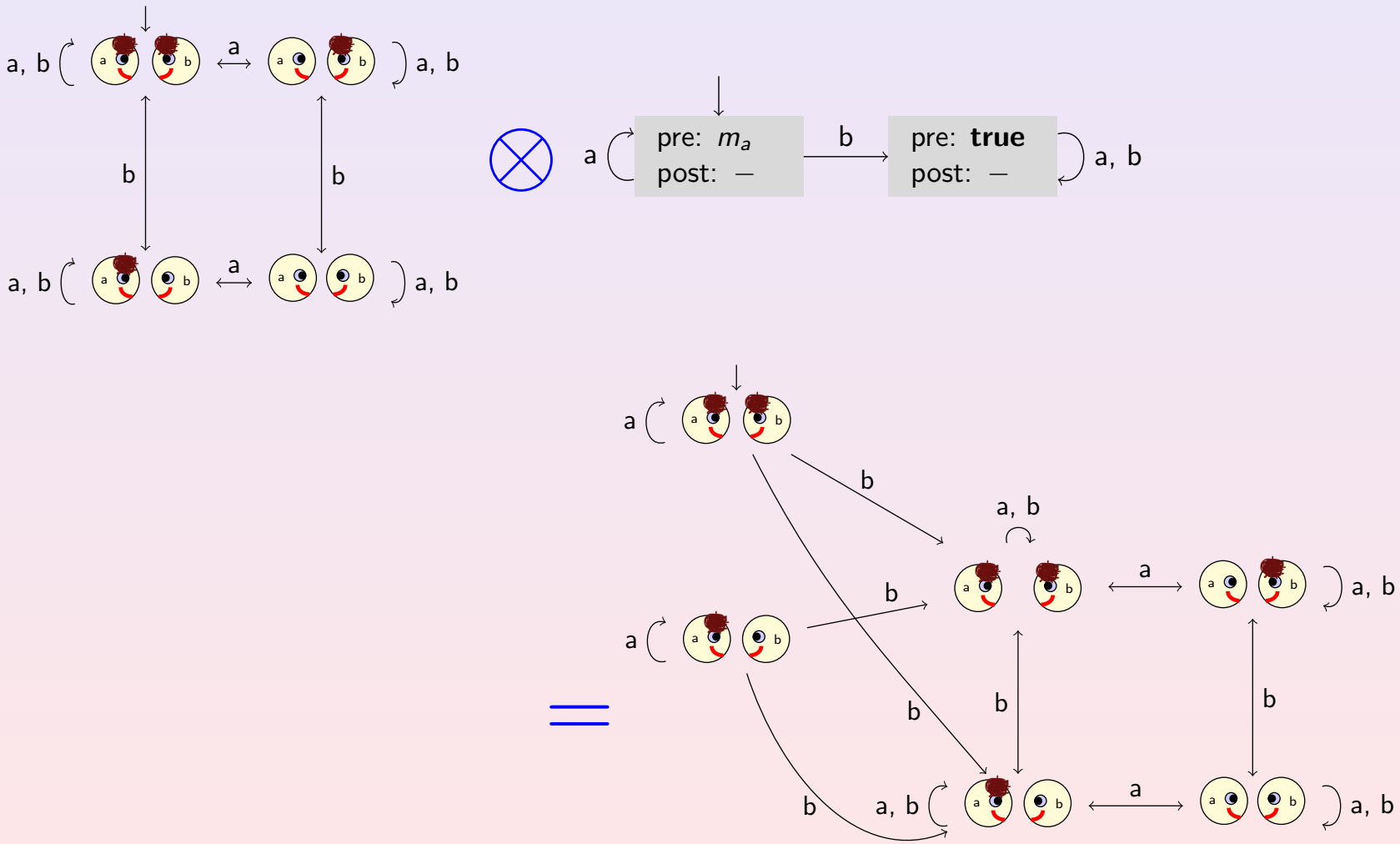
Effect of a public announcement of φ: only keep φ-worlds



In Hintikka's World: Try on several examples!



Example of an update product



Update product: formal definition

Let $\mathcal{M} = (W, \{R_a\}_{a \in AGT}, V)$ be an epistemic model and $\mathcal{E} = (\mathrm{E}, (R_a^{\mathcal{E}})_{a \in AGT}, pre, post)$ be an event model.

Definition

The update product of $\mathcal{M}$ and $\mathcal{E}$ is the epistemic model $\mathcal{M} \otimes \mathcal{E} = (W^{\otimes}, \{R_a^{\otimes}\}_{a \in AGT}, V^{\otimes})$ where:

$$W^{\otimes} = \{(w, e) \in W \times \mathrm{E} \mid \mathcal{M}, w \models pre(e)\},$$

$$R_a^{\otimes}(w, e) = \{(w', e') \in W^{\otimes} \mid w R_a w' \text{ and } e R_a^{\mathcal{E}} e'\},$$

$$V^{\otimes}(w, e) = \{p \in AP \mid \mathcal{M}, w \models post(e)(p)\}$$

Pointed update products

Definition

The successor state of an epistemic state $(\mathcal{M}, w)$ by action $(\mathcal{E}, e)$ is

$$(\mathcal{M}, w) \otimes (\mathcal{E}, e) =^{\text{def}} (\mathcal{M} \otimes \mathcal{E}, (w, e))$$

if $\mathcal{M}, w \models pre(e)$, otherwise it is undefined.

Notation

- *We write e instead of $(\mathcal{E}, e)$;*
- *We write the word 'we' instead of the pair (w, e);*
- *We write $\mathcal{M} \otimes \mathcal{E}^n$ for $\mathcal{M} \otimes \mathcal{E} \otimes \ldots \mathcal{E}$, n times.*
- *We write $we_1 \ldots e_n \models \varphi$ instead of $\mathcal{M} \otimes \mathcal{E}^n, we_1 \ldots e_n \models \varphi$.*

Outline

Dynamic language

Definition

The language $\mathcal{L}_{\mathbf{DELCK}}$ extends $\mathcal{L}_{\mathbf{ELCK}}$ with dynamic modalities and is defined by the following BNF:

$$\varphi \quad ::= \quad \top \mid p \mid \neg\varphi \mid (\varphi \vee \varphi) \mid K_a\varphi \mid C_G\varphi \mid \langle \mathcal{E}, \mathsf{E}_0 \rangle \varphi$$

where $\mathcal{E}, \mathsf{E}_0$ ranges over the set of non-deterministic actions.

Definition

We extend the definition $\mathcal{M}, w \models \varphi$ to $\mathcal{L}_{\mathbf{DELCK}}$ with the following clause:

- $\mathcal{M}, w \models \langle \mathcal{E}, \mathsf{E}_0 \rangle \varphi$ if there exists $e \in \mathsf{E}_0$ s.th.
$\mathcal{M}, w \models \mathit{pre}(e)$ and $\mathcal{M} \otimes \mathcal{E}, (w, e) \models \varphi$.

Dual operator

We define $[\mathcal{E}, \mathsf{E}_0]$ to be $\neg\langle\mathcal{E}, \mathsf{E}_0\rangle\neg$.

The semantics is:

- $\mathcal{M}, w \models [\mathcal{E}, \mathsf{E}_0]\varphi$ if for all $e \in \mathsf{E}_0$ we have $\mathcal{M}, w \models \mathit{pre}(e)$ implies $\mathcal{M} \otimes \mathcal{E}, (w, e) \models \varphi$;
- $\mathcal{M}, w \models \langle\mathcal{E}, \mathsf{E}_0\rangle\varphi$ if there exists $e \in \mathsf{E}_0$ s.th. $\mathcal{M}, w \models \mathit{pre}(e)$ and $\mathcal{M} \otimes \mathcal{E}, (w, e) \models \varphi$.

Outline

Expressivity

Theorem

DEL *and* **EL** *have the same expressivity.*

Idea: we remove the dynamic operators $[\mathcal{E}, E]$. Let us explain it just with public announcements:

$[\varphi!]\psi$: if φ holds then after having announced φ publicly, ψ holds.

$[\varphi!]p$	says the same thing than	$(\varphi \to p)$
$[\varphi!](\psi \wedge \chi)$	says the same thing than	$([\varphi!]\psi \wedge [\varphi!]\chi)$
$[\varphi!]\neg\psi$	says the same thing than	$(\varphi \to \neg[\varphi!]\psi)$
$[\varphi!]K_a\psi$	says the same thing than	$(\varphi \to K_a[\varphi!]\psi)$
$[\varphi!][\psi!]\chi$	says the same thing than	$[\varphi \wedge [\varphi!]\psi!]\chi$

General proof in [Baltag, Moss and Solecki, 2003a]
DEL is more succinct: [Lutz, AAMAS 2006]

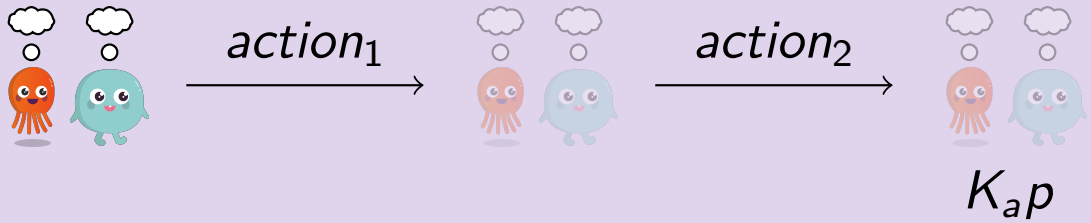
Outline

Outline

Model checking problem

Definition (model checking problem)

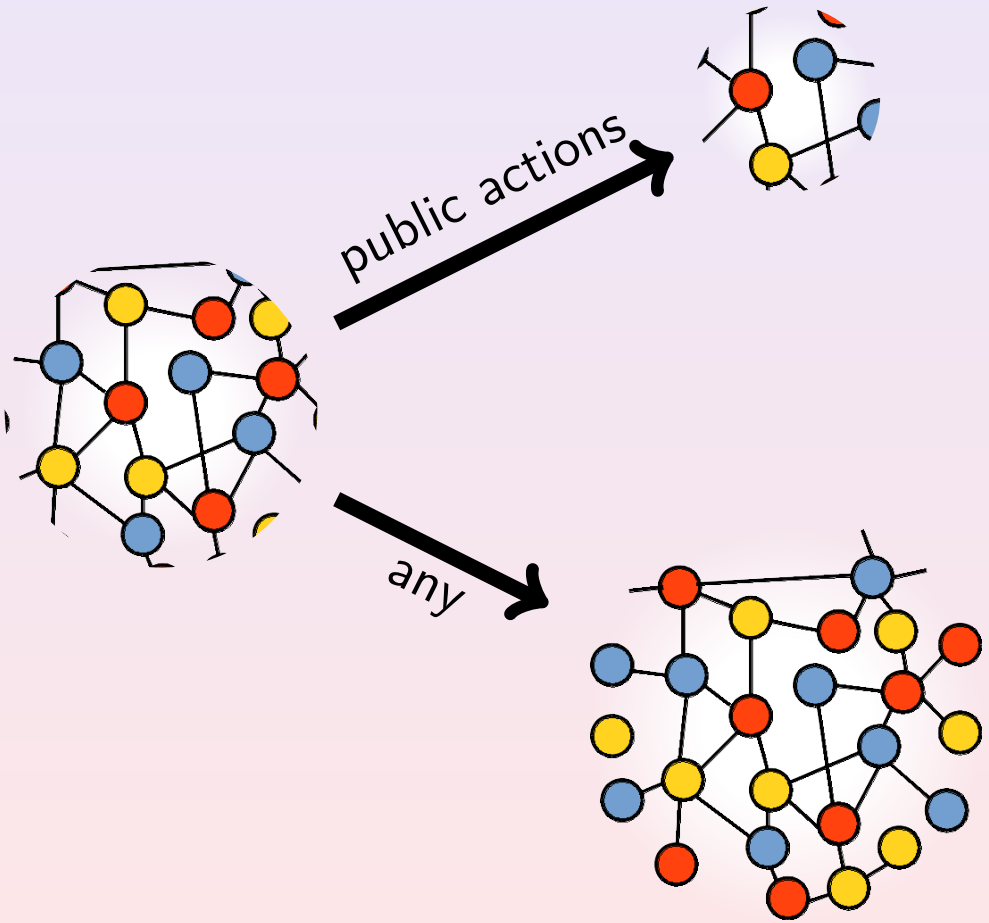
- Input:
  - 
  - An epistemic state
  - A formula, e.g. $\langle action_1; action_2 \rangle K_a p$;
- Output: yes if



no otherwise.

Outline

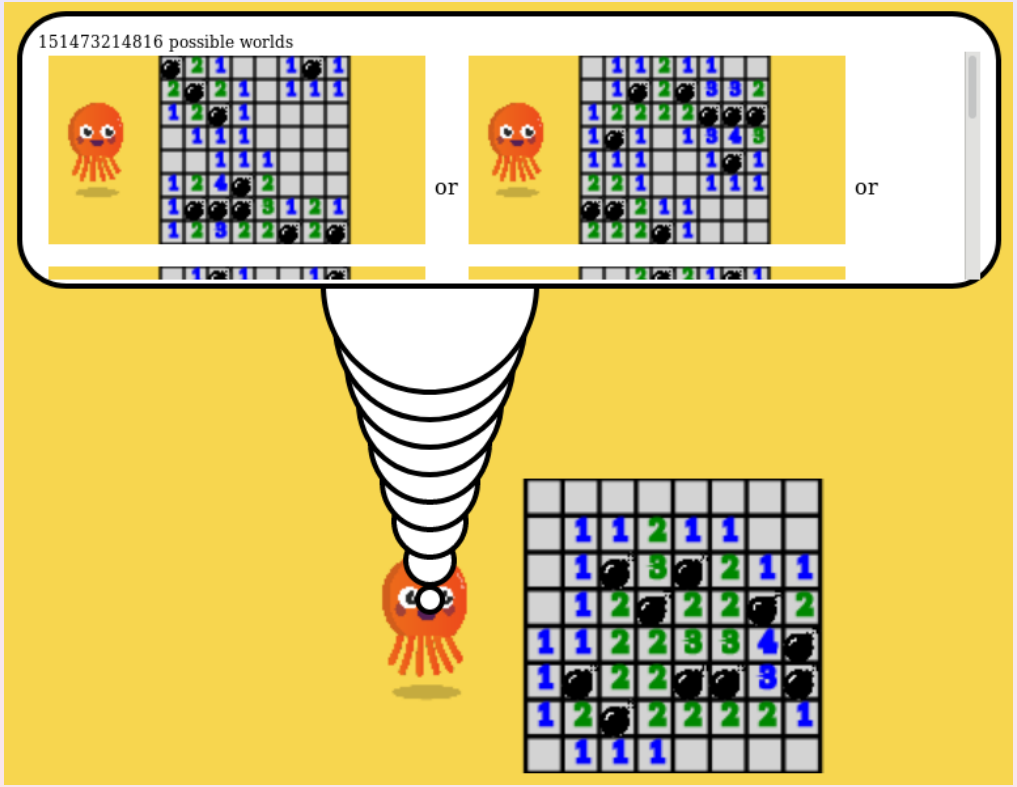
Model checking complexity



P-complete
[van Benthem, 2011]

PSPACE-complete
[Aucher, _, TARK 2013]
[Pol et al. 2016]

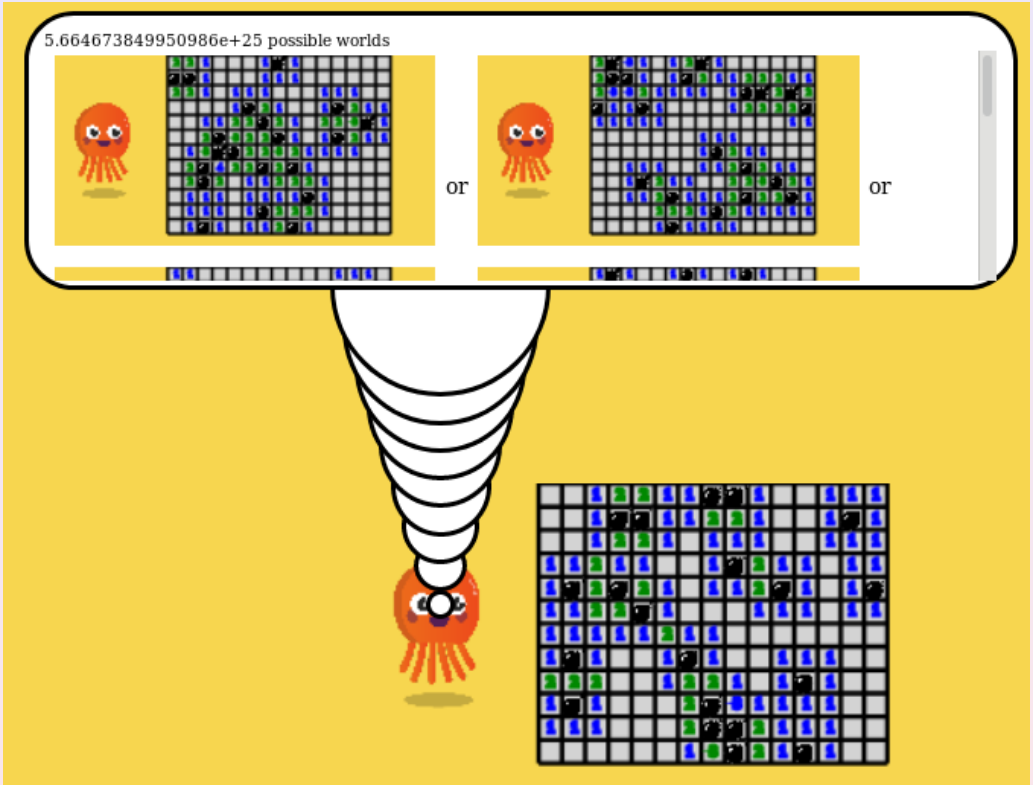
State explosion problem



Example

Minesweeper easy 8×8 with 10 bombs: $> 10^{12}$ possible worlds.

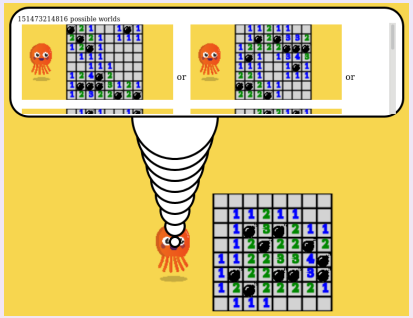
State explosion problem



Example

Minesweeper 10×12 with 20 bombs: $> 10^{25}$ possible worlds.

Solution to the state explosion problem

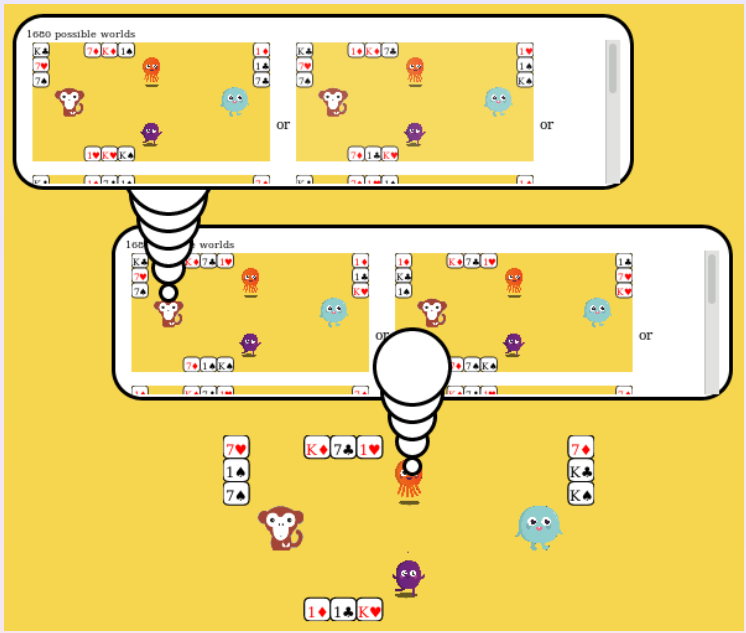


[DBLP:conf/lori/BenthemEGS15], [DBLP:journals/logcom/BenthemEGS18]

[Charrier _ AAMAS 2017], [Charrier _ AiML 2018]

- Succinct representations of epistemic states **and** actions;
- Easy to specify by means of accessibility programs;
- Succinct model checking still in PSPACE.

Impact



Theoretical

Existence of a (uniform) strategy in **bounded** imperfect info games is in PSPACE.

Implementation: PSPACE techniques

Symbolic Model checking implemented in Hintikka's World:

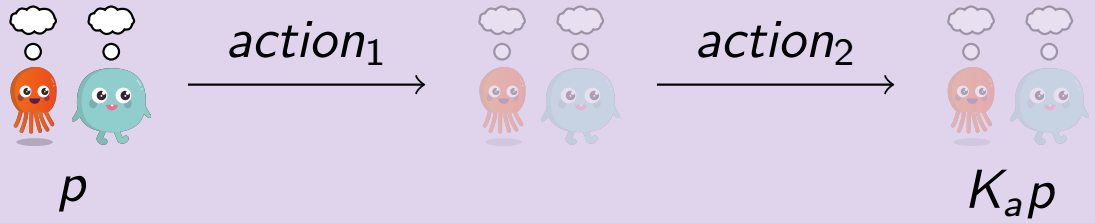
- by Sébastien Gamblin and Alexandre Niveau (univ. Caen)
- using BDDs (C wrapper of CUDD compiled in `wasm`).

Outline

Theorem proving

Motivation: parametrized verification

for all epistemic states in which p holds:

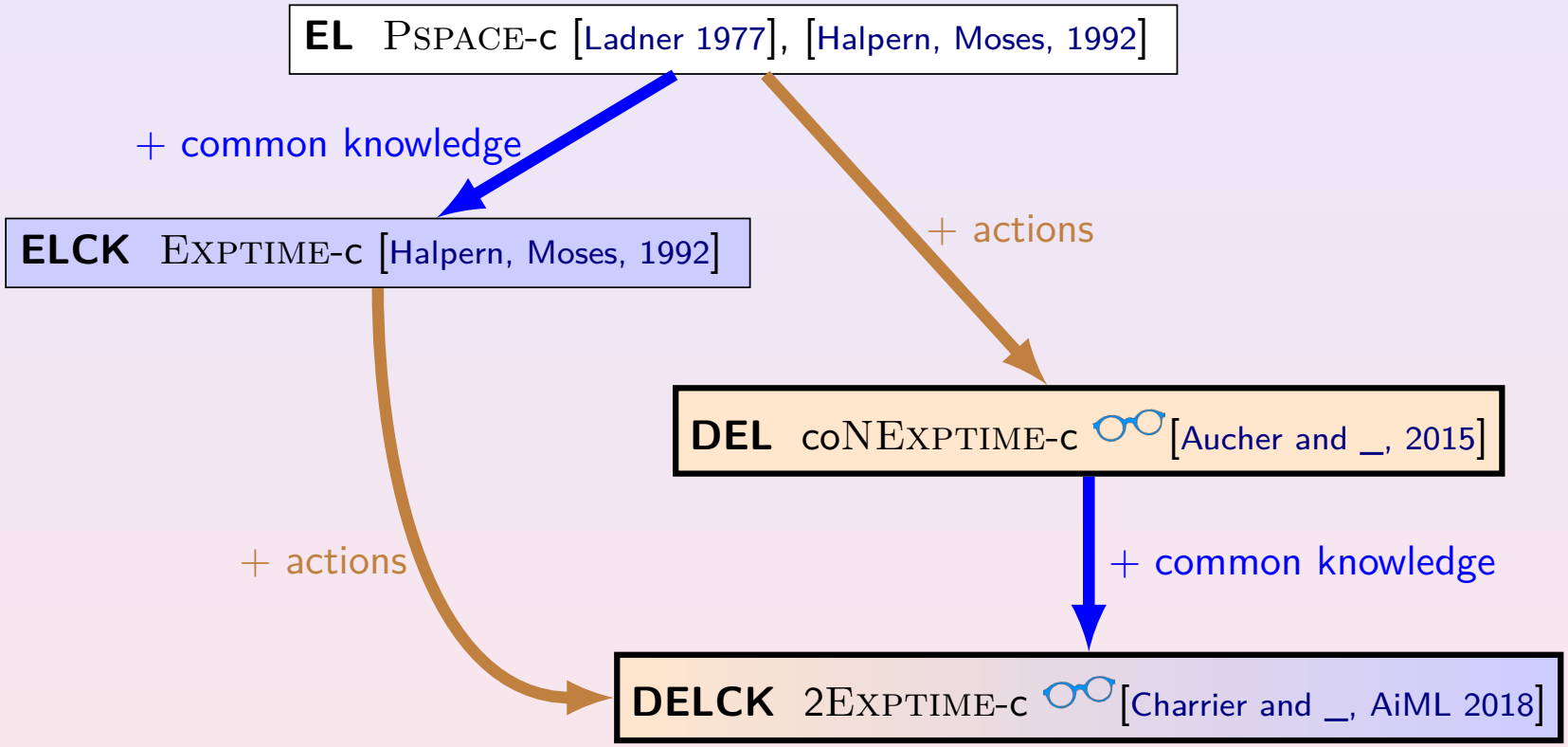


$p \to \langle action_1; action_2 \rangle K_a p$ is a *theorem*, i.e. true in all epistemic states.

Definition

- Input: a formula φ;
- Output: yes if φ is a theorem, no otherwise.

Theorem proving is highly intractable

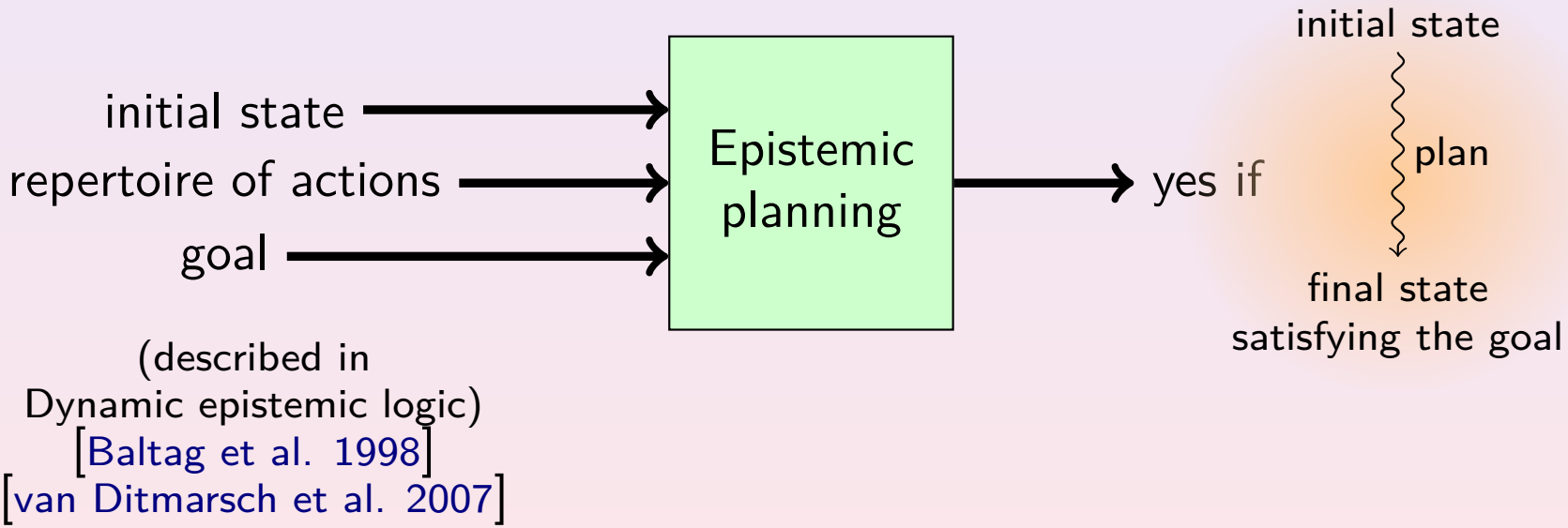


- Semi-product modal logics have high complexities;
  [Gabbay et al. Many-Dimensional Modal Logics: Theory and Applications, 2003]
- Model checking more practical than theorem proving
  [Halper, Vardi, KR 1991].

Outline

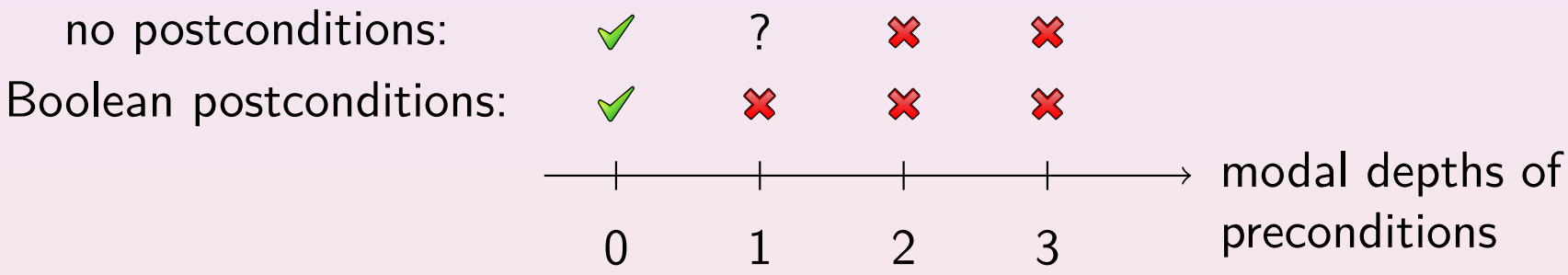
Epistemic planning

[Andersen, Bolander, 2011]

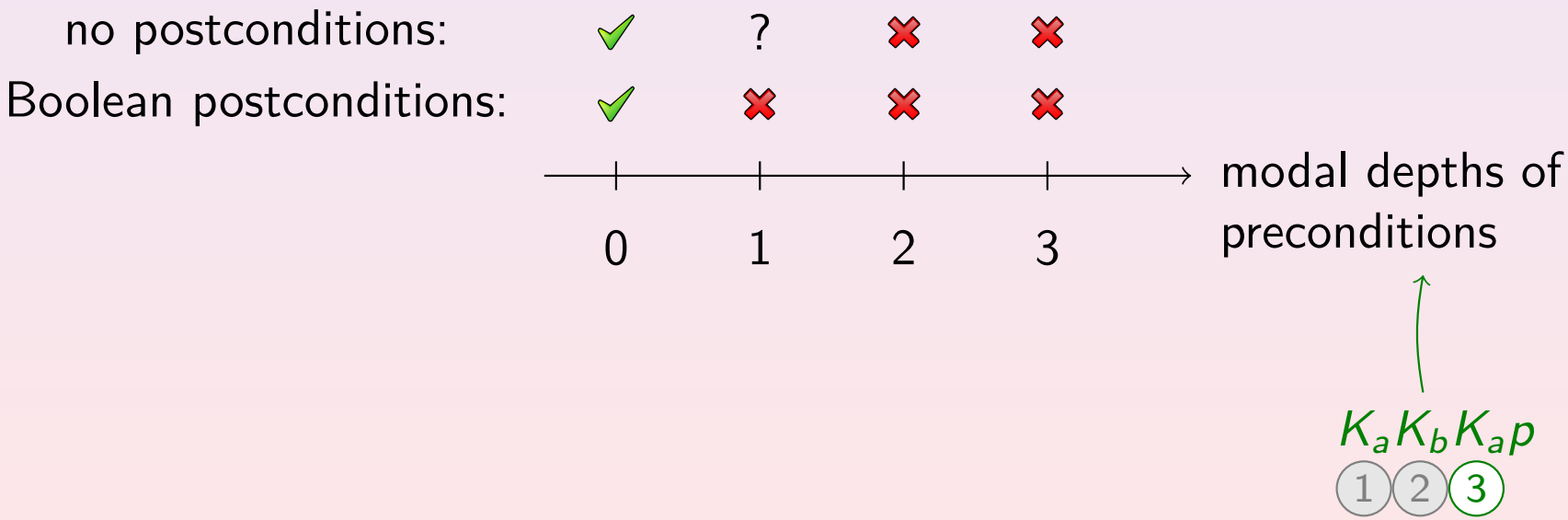


(described in
Dynamic epistemic logic)
[Baltag et al. 1998]
[van Ditmarsch et al. 2007]

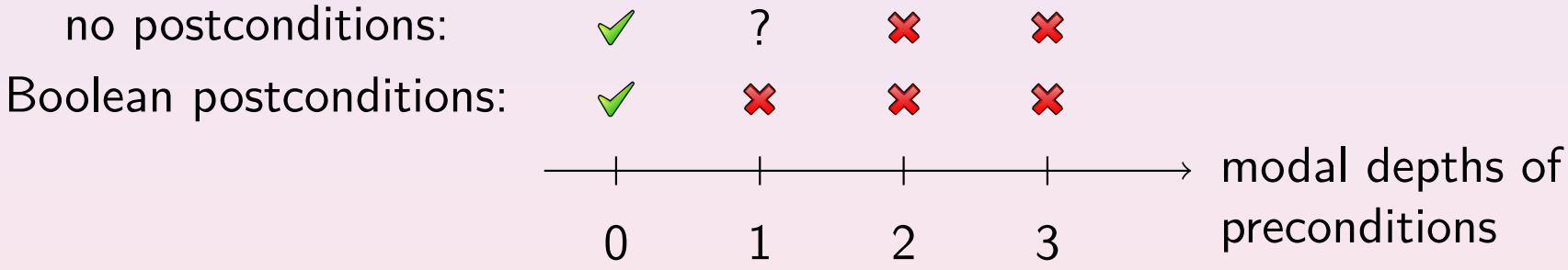
Decidability and undecidability of epistemic planning



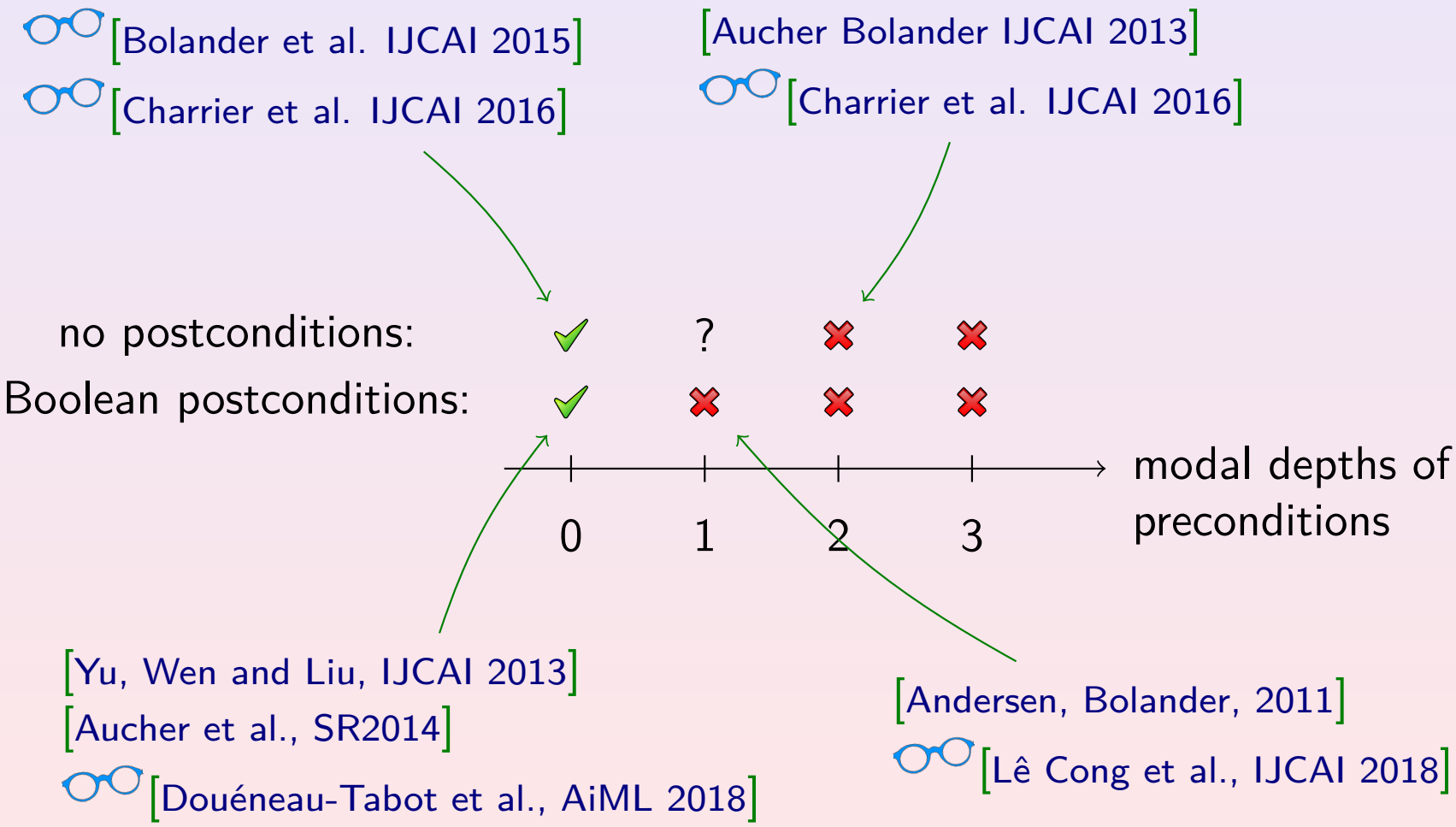
Decidability and undecidability of epistemic planning



Decidability and undecidability of epistemic planning



Decidability and undecidability of epistemic planning



Outline

Epistemic planning is undecidable

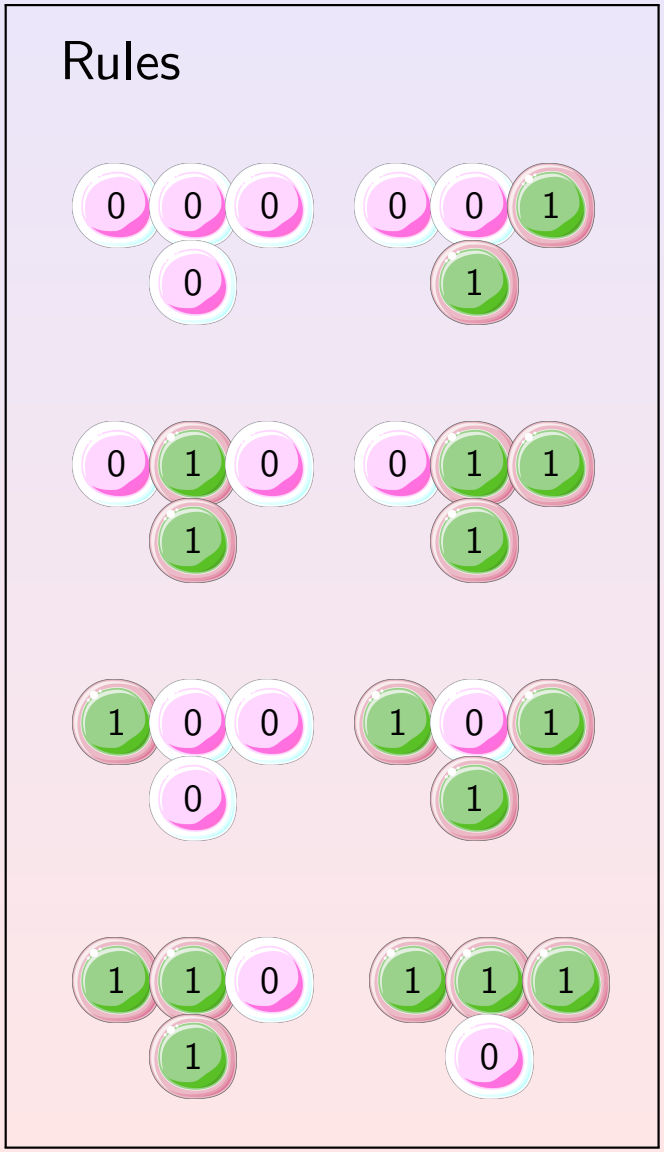
Theorem

Epistemic planning is undecidable for:

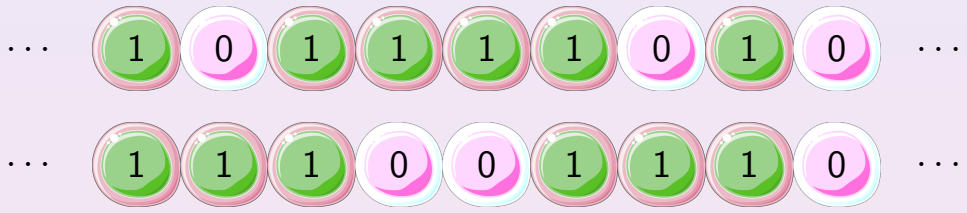
$$\overbrace{\underbrace{\textit{two agents} + \textit{Boolean post} + \textit{md(pre)} \leq 1}_{\textit{[Andersen, Bolander, JANCL 2011]}} + \textit{fixed repertoire of one action} + \textit{6 atomic propositions}}^{\textit{[Lê Cong, Pinchinat, \_, IJCAI-ECAI 2018]}}$$

Proof: reduction from halting problem of a small universal cellular automaton.

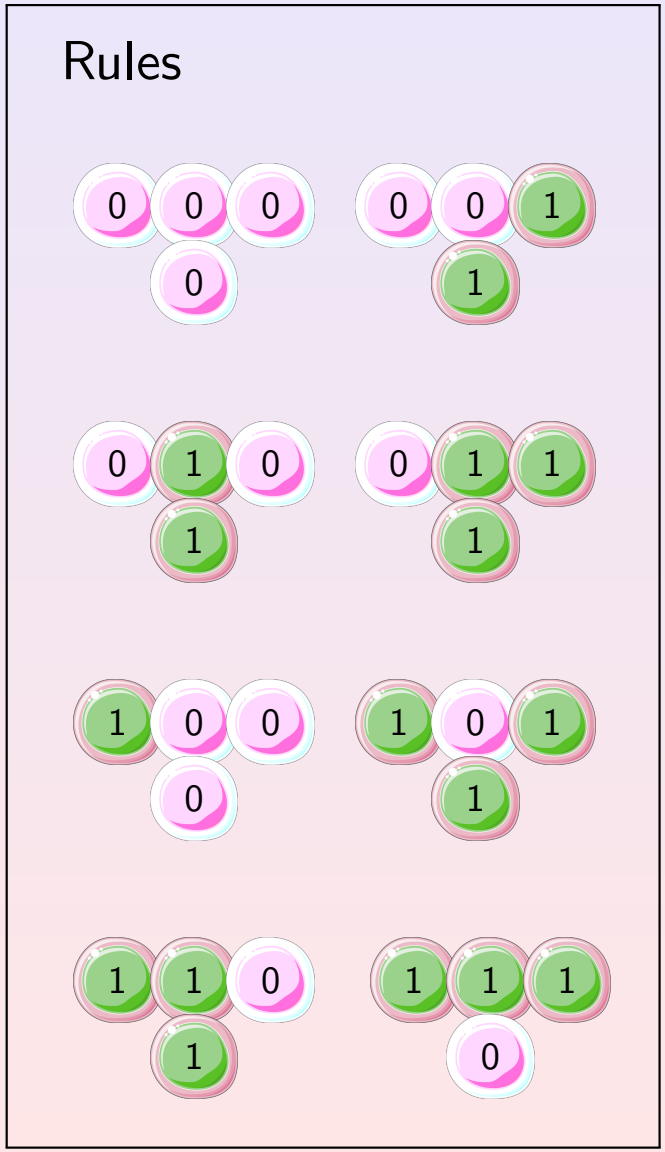
Example: the 110 Rule cellular automaton



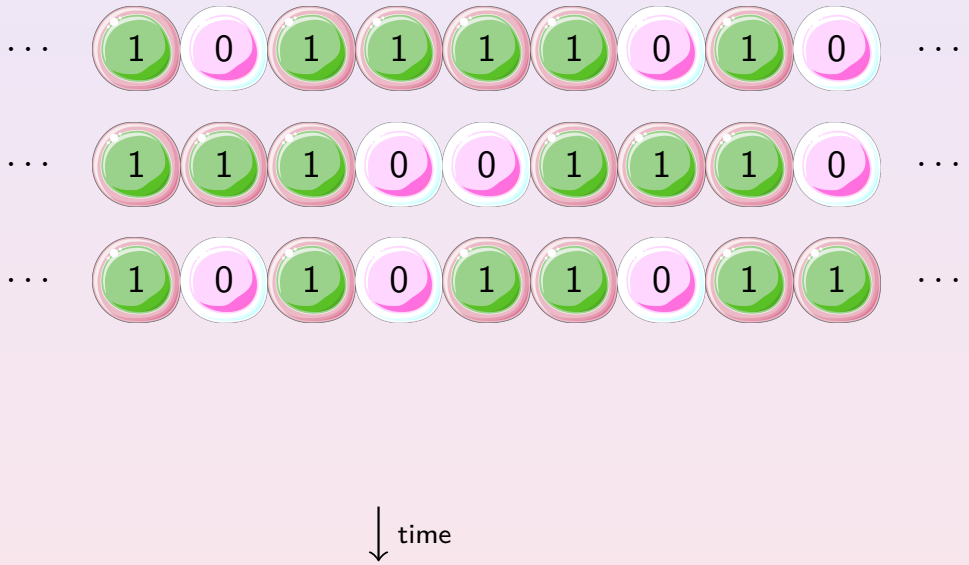
Example: the 110 Rule cellular automaton

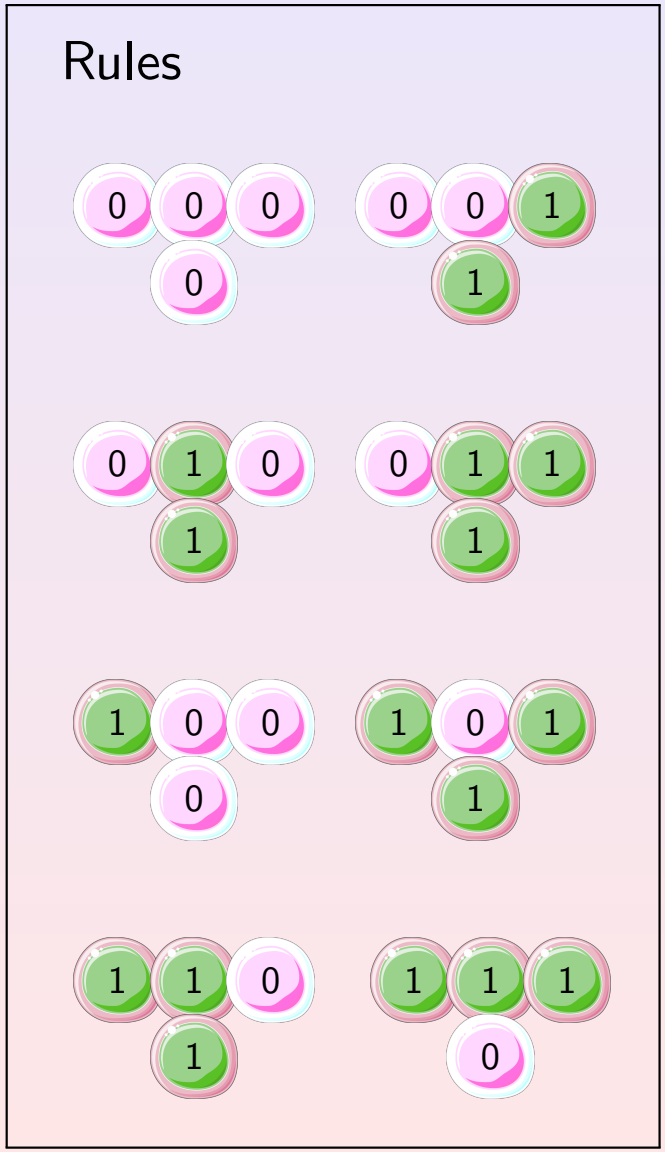


time

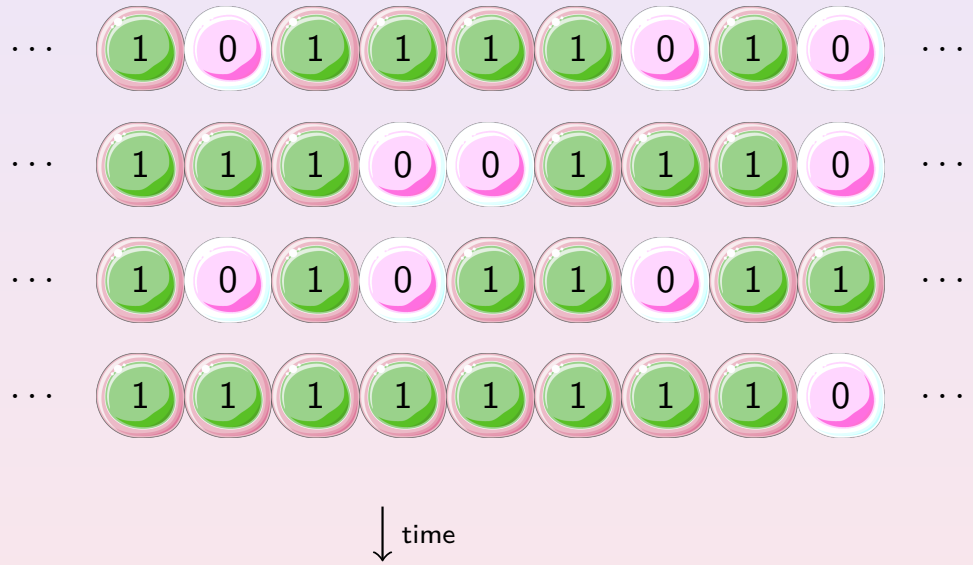


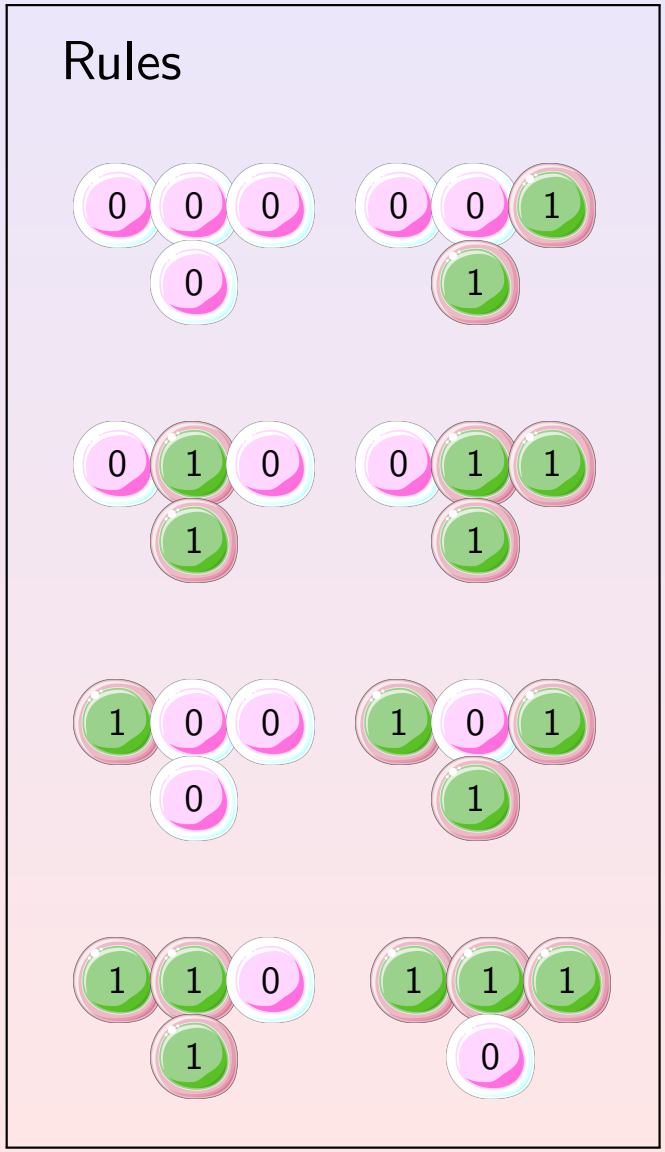
Example: the 110 Rule cellular automaton



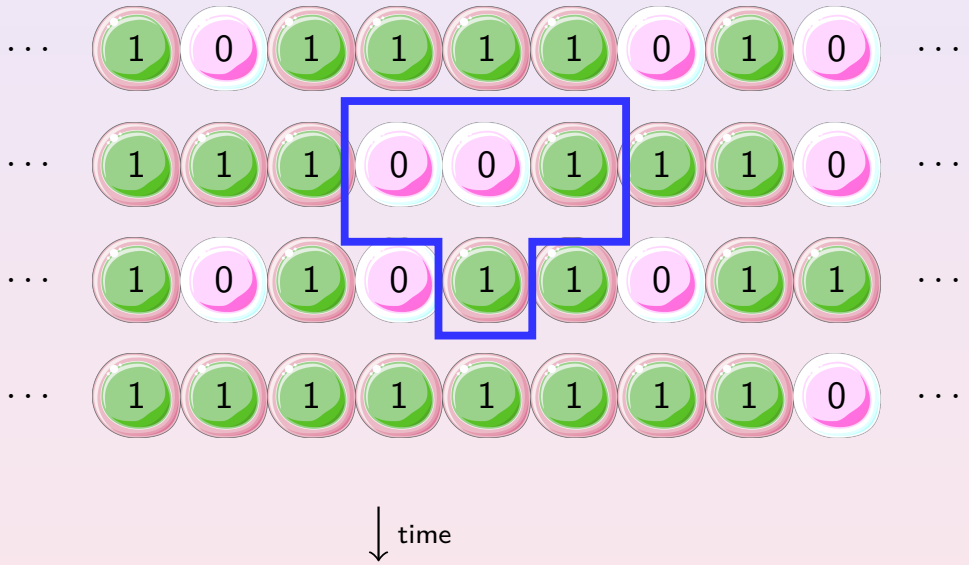


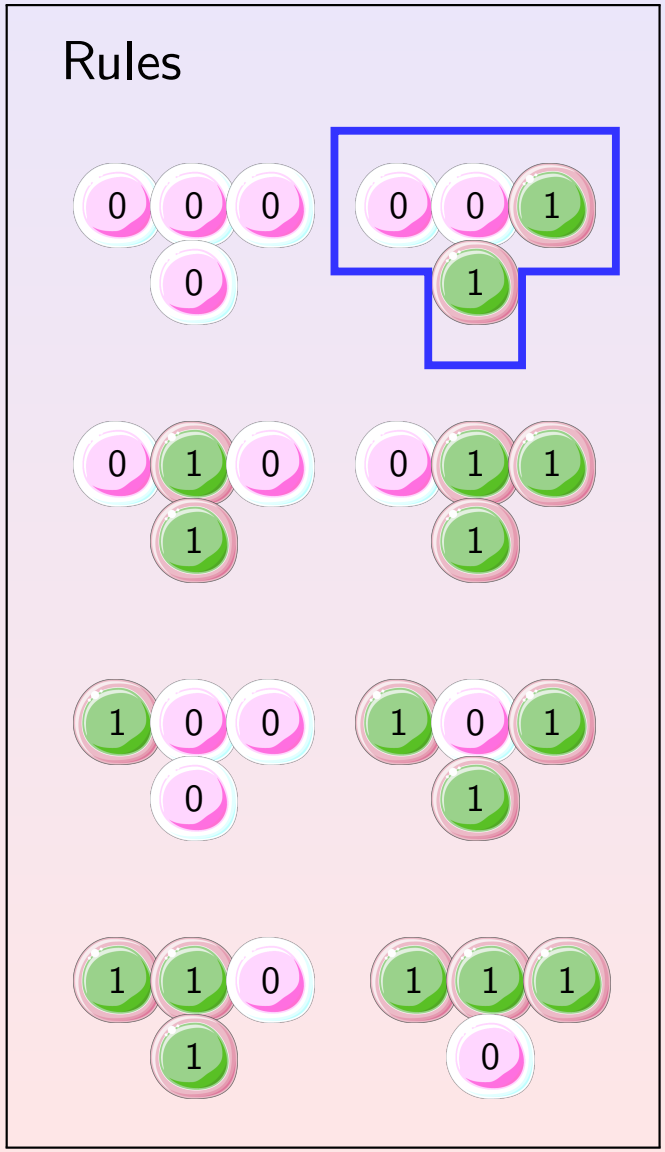
Example: the 110 Rule cellular automaton



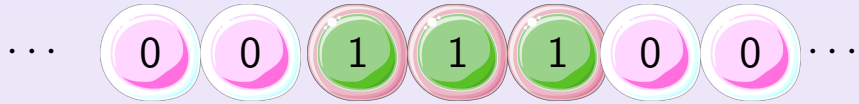


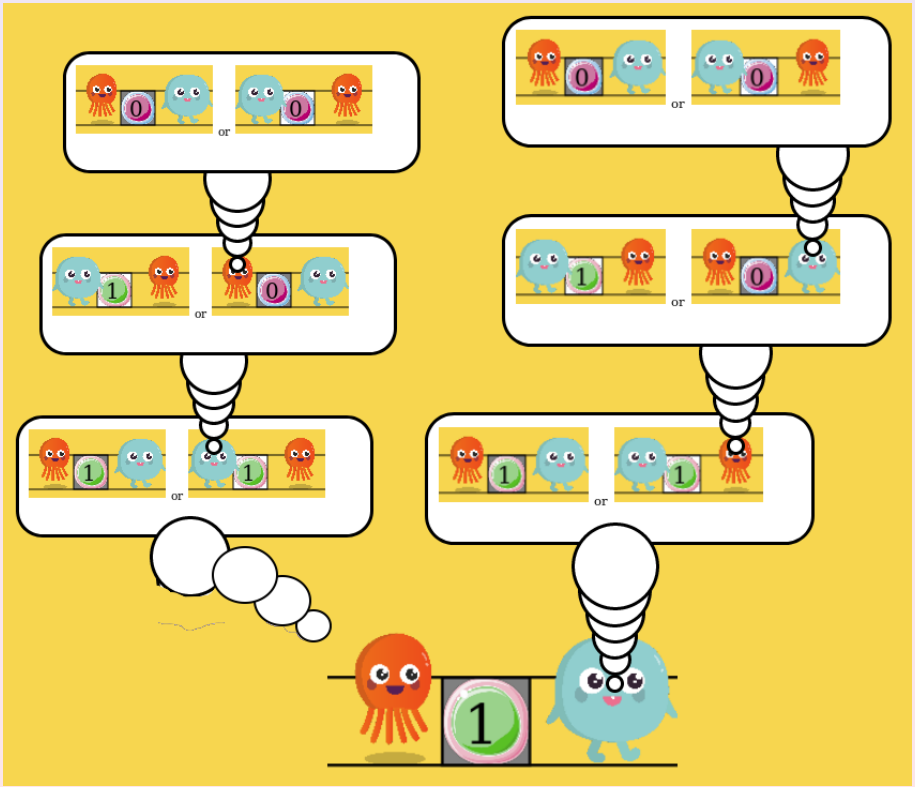
Example: the 110 Rule cellular automaton





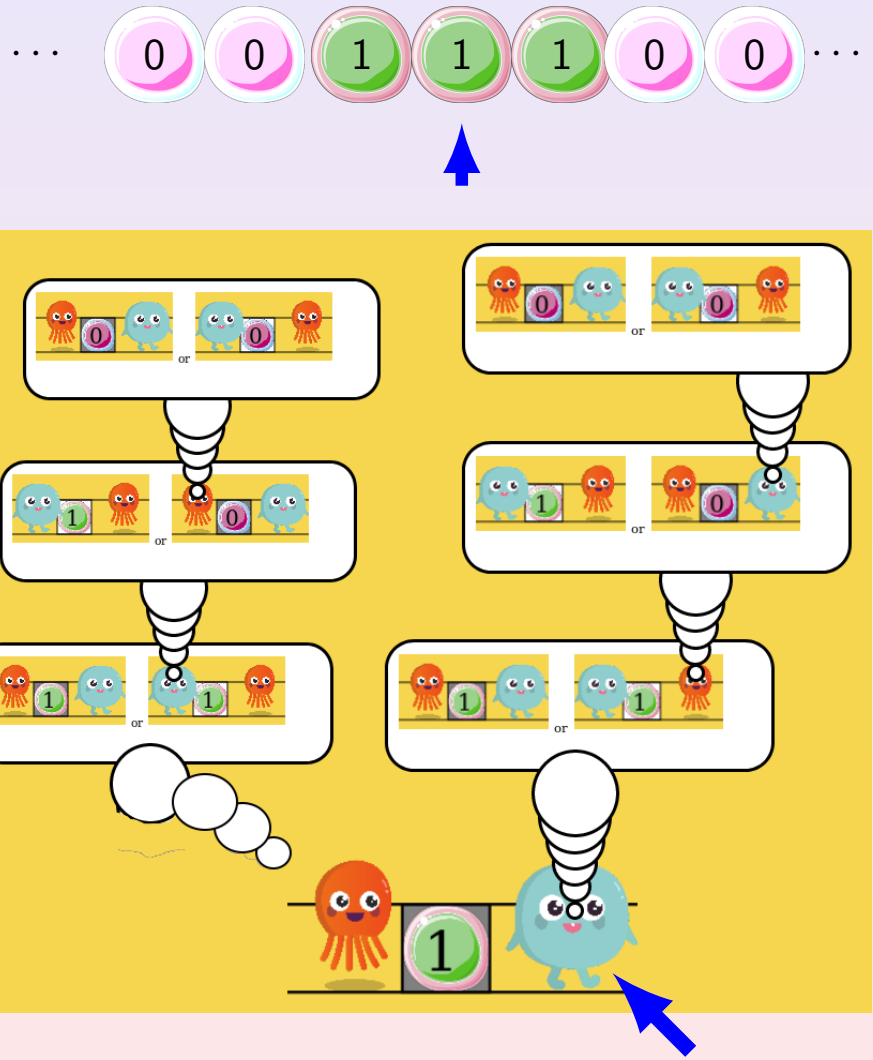
Encoding an automaton configuration in a state





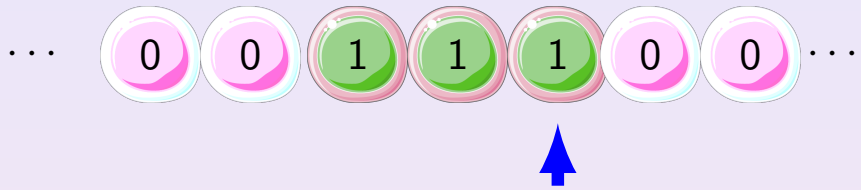
In Hintikka's World: Cellular automaton

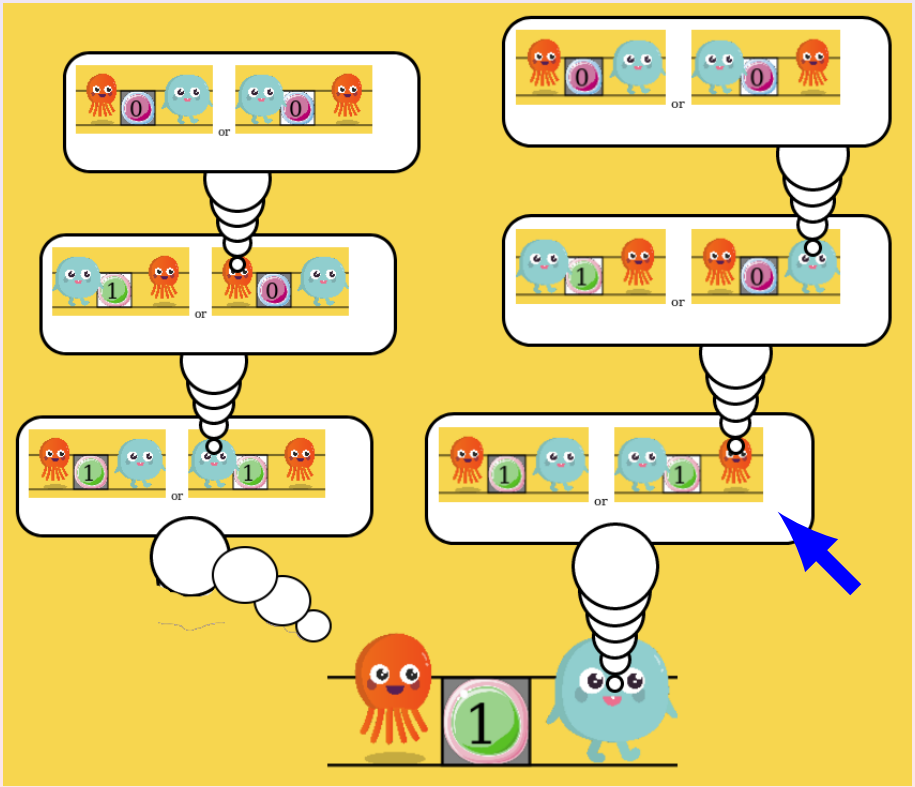
Encoding an automaton configuration in a state



In Hintikka's World: Cellular automaton

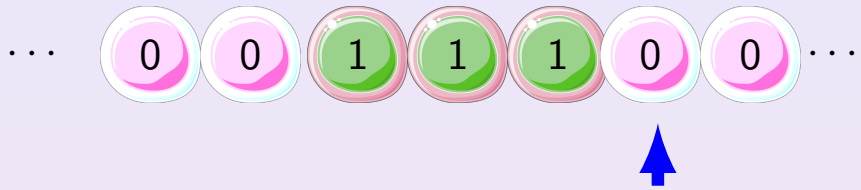
Encoding an automaton configuration in a state

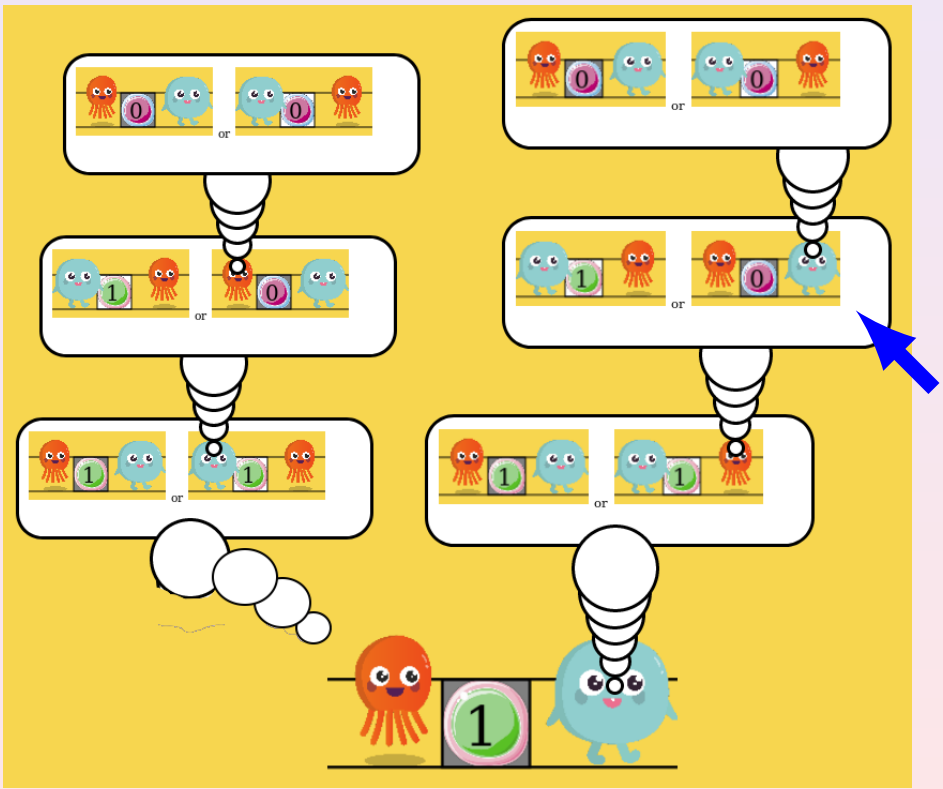




In Hintikka's World: Cellular automaton

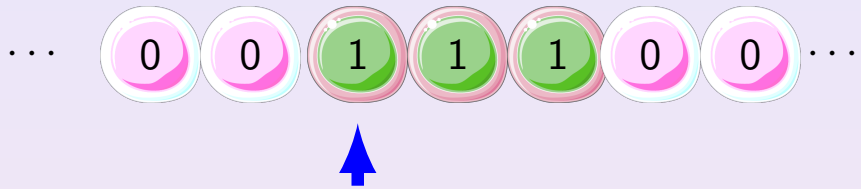
Encoding an automaton configuration in a state

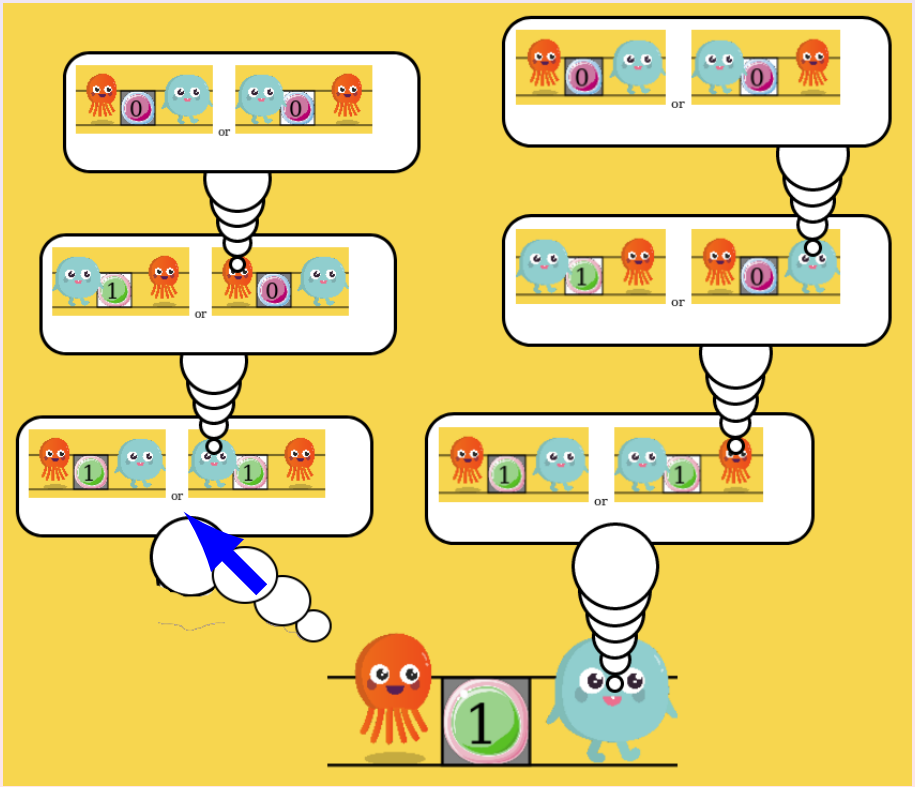




In Hintikka's World: Cellular automaton

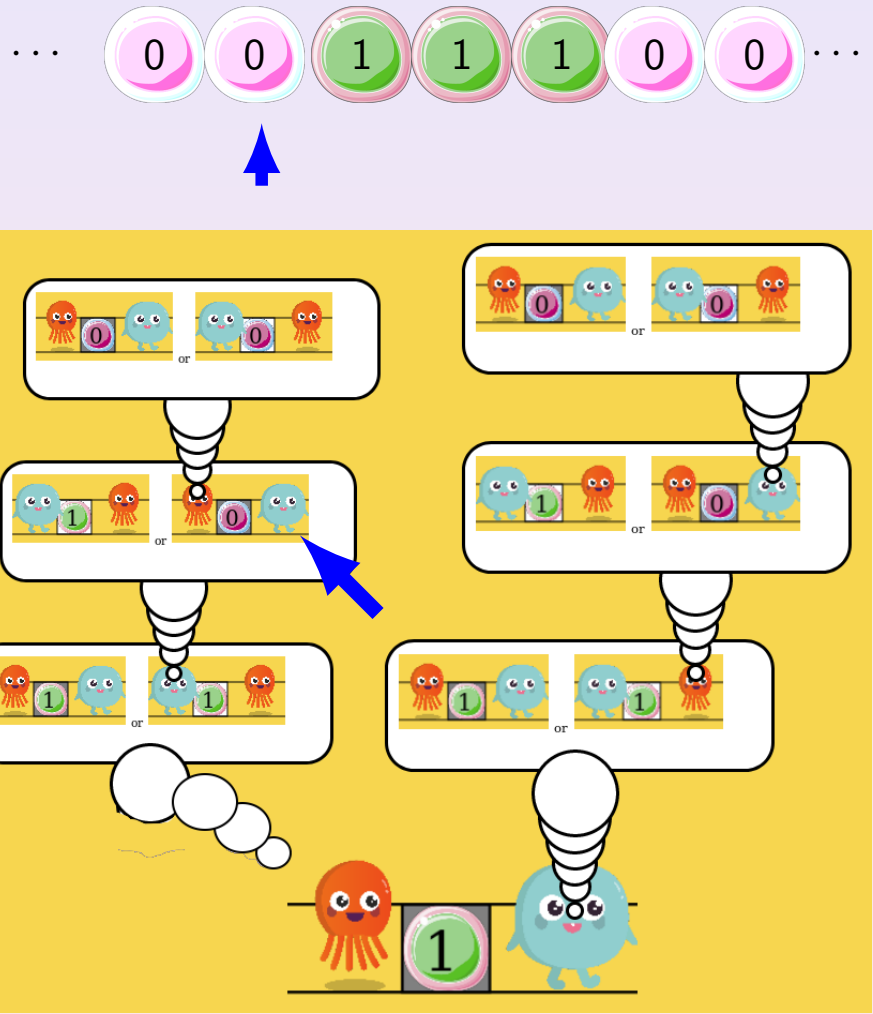
Encoding an automaton configuration in a state





In Hintikka's World: Cellular automaton

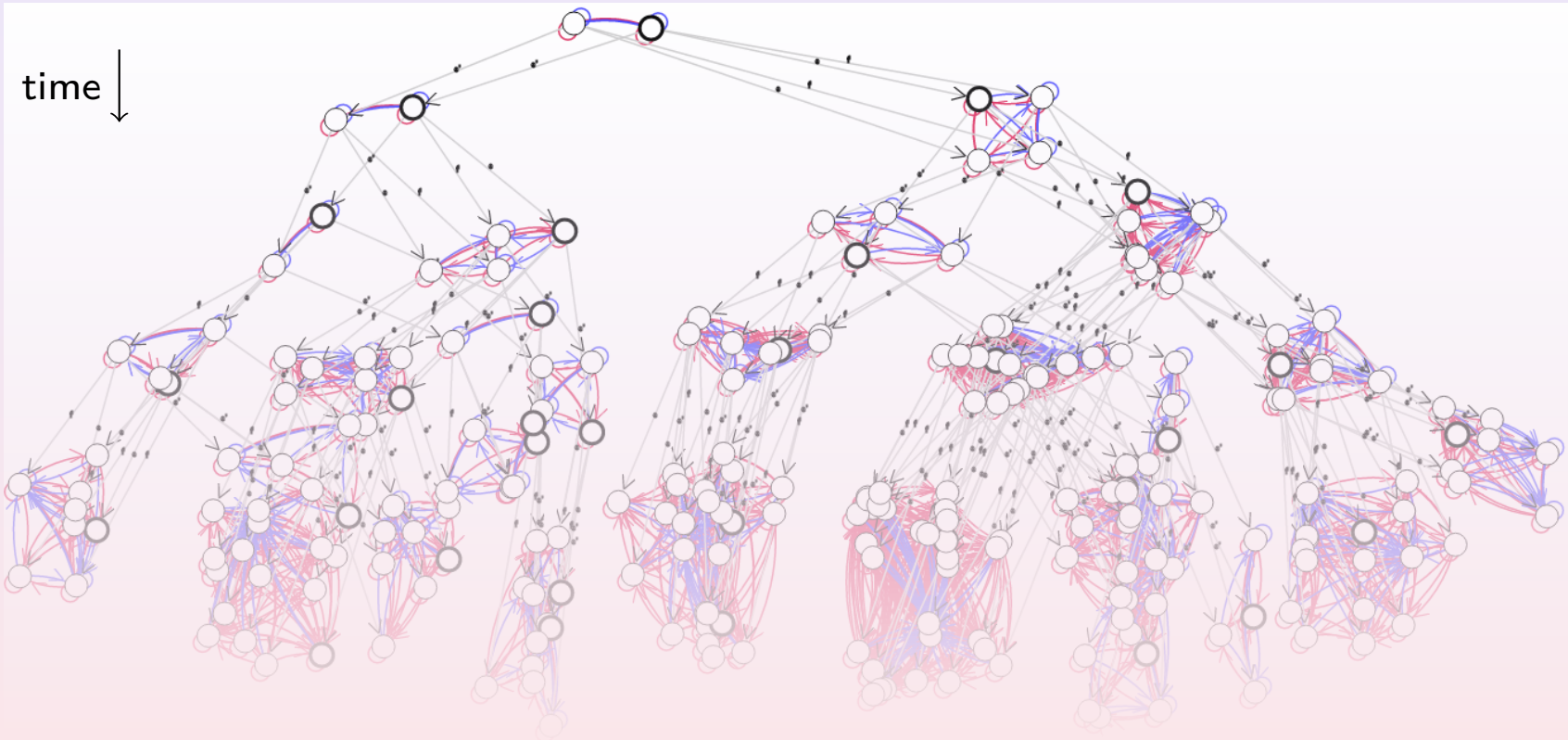
Encoding an automaton configuration in a state



In Hintikka's World: Cellular automaton

Outline

(Infinite) epistemic temporal structures



Epistemic planning: first-order query $\exists x, goal(x)$

Decidability when pre/post are Boolean

Theorem ([DBLP:conf/ijcai/YuWL13], [DBLP:journals/corr/AucherMP14])

When pre/post are Boolean, epistemic planning is decidable.

Epistemic planning is a first-order-query

first-order-query on automatic structures is decidable.

Epistemic temporal structures are automatic

Theorem (👓 [Douéneau-Tabot, Pinchinat and _, 2018])

Even decidable for goals in epistemic linear μ-calculus.

Automatic structure = defined by automatas

$$\langle \mathbb{N} \quad , \quad \mathit{iseven}? \quad , \quad \leqslant \rangle$$

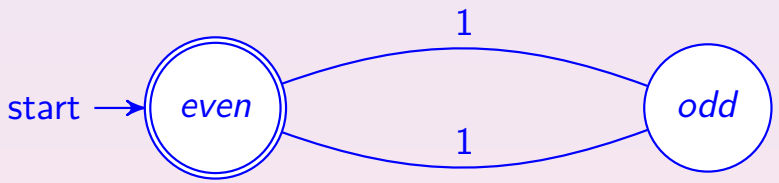
$$\mathcal{A}_{\mathbb{N}} \qquad \mathcal{A}_{\mathit{iseven}?} \qquad \mathcal{A}_{\leqslant}$$

$$\mathit{enc} : \quad \mathbb{N} \to \{1\}^* \qquad n \mapsto 1^n$$

$\mathcal{A}_{\mathbb{N}}$

Example of an automatic structure

$$\langle \mathbb{N}, \mathit{iseven?}, \leqslant \rangle$$

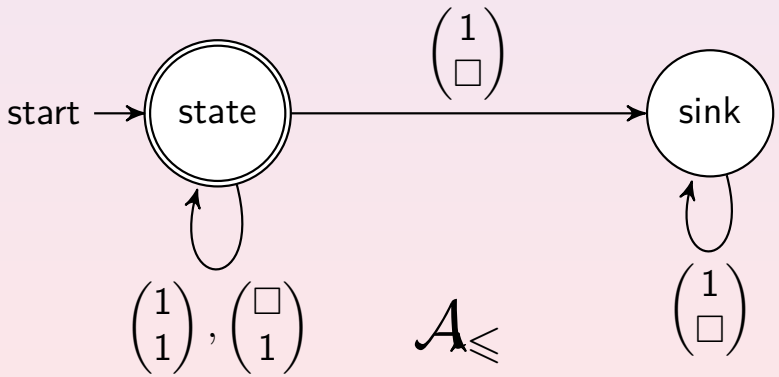


$$\mathcal{A}_{\mathit{iseven?}}$$

Example of an automatic structure

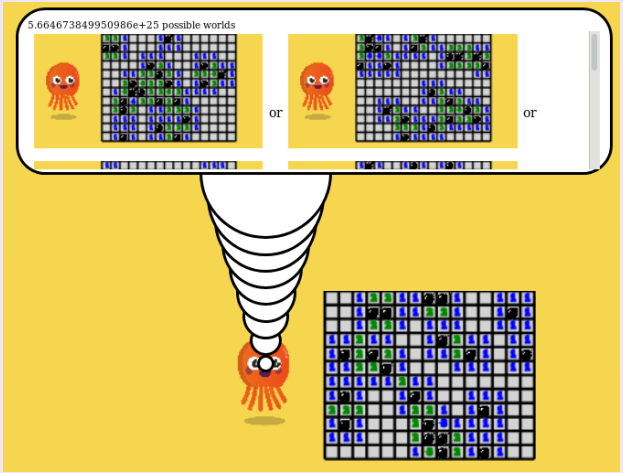
$$\mathcal{S} = \langle \mathbb{N}, \mathit{iseven}?, \leqslant \rangle$$

- $2 \leqslant 5$ iff "$11 \leqslant 11111$"
- $2 \leqslant 5$ iff word $\binom{1}{1}\binom{1}{1}\binom{\square}{1}\binom{\square}{1}\binom{\square}{1}$ is accepted by $\mathcal{A}_{\leqslant}$



Outline

Strategies



Definition

A *strategy* for player a is a function σ that maps any history $we_1...e_n$ to a deterministic epistemic action in the repertoire of a.

Definition

A *uniform strategy* for player a is a strategy σ such that

if $we_1...e_n \sim_a ue'_1...e'_n$ then

$$\sigma(we_1...e_n) = \sigma(ue'_1..e'_n)$$

Undecidability even for Boolean pre/post

Theorem

[Reif, Peterson, 1979] [Coulombe and Lynch, Def. 1, p. 14:7, FUN 2018] [Maubert et al., IJCAI 2019]
The existence of uniform strategies for two players against an environment for achieving a goal φ is undecidable.

Decidability cases

public actions
[Belardinelli et al., 2017]
[Maubert et al., IJCAI 2019]

hierarchical information
[Maubert et al., 2018]
[Maubert et al., IJCAI 2019]

(picture idea from Raphael Berthon)

Complexity results on epistemic planning

	one centralized planner [Bolander et al. IJCAI 2015]	many players [Maubert et al., 2019]
public announcements	NP-c	PSPACE-c
public actions	PSPACE-c	EXPTIME-c
Boolean pre/post	decidable	undecidable [Reif, Peterson, 1979]
all	undecidable	

Uninformed semantics.

Perspectives: DEL and formal language theory

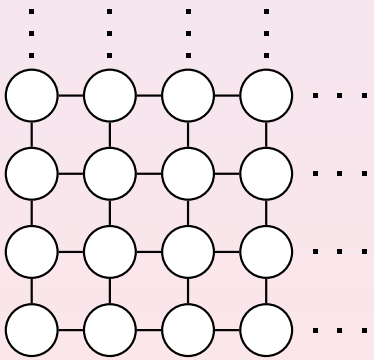
Question

Is epistemic planning one agent (pre md 1, ~~post~~) decidable?

First-order query is decidable

First-order query is undecidable

Automatic structures



Turing-complete structures

Pushdown automata?
Caucal hierarchy?

Perspectives

- Connection with logics for reasoning about strategies such as Alternating temporal-time logic, Strategy Logic, etc.


[Maubert et al., 2019]

- Describing protocols/policies