Epistemic reasoning in AI

François Schwarzentruber

École Normale Supérieure Rennes

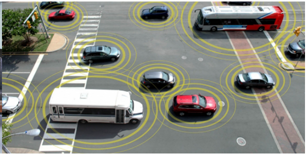
IJCAI-ECAI, Tutorial T27, 14 July 2018

Talks at IJCAI-ECAI 2018

- Game Description Language and Dynamic Epistemic Logic Compared. Thorsten Engesser, Robert Mattmüller, Bernhard Nebel, Michael Thielscher
- Single-Shot Epistemic Logic Program Solving. Manuel Bichler, Michael Morak, Stefan Woltran
- Model Checking Probabilistic Epistemic Logic for Probabilistic Multiagent Systems. Chen Fu, Andrea Turrini, Xiaowei Huang, Lei Song, Yuan Feng, Lijun Zhang
- The Complexity of Limited Belief Reasoning—The Quantifier-Free Case Yijia Chen, Abdallah Saffidine, Christoph Schwering
- Small Undecidable Problems in Epistemic Planning Sébastien Lê Cong, Sophie Pinchinat, _
- Multi-agent Epistemic Planning with Common Knowledge Qiang Liu, Yongmei Liu

Objective of this tutorial

1. Being able to understand these IJCAI-ECAI papers in the field
2. Being able to model epistemic multi-agent scenarios
3. Being able to contribute in the field
4. Promote automatic structures for proving decidability
   [Blumensath and Grädel 2000]
5. (if time) Advertise knowledge-base programs for writing policies



Many different settings

This tutorial is not a catalogue (although this slide is one):

- QdecPOMDP, decPOMDP 🗎 [Brafman, Shani, and Zilberstein 2013]
- Belief revision 🗎 [Alchourrón, Gärdenfors, and Makinson 1985]
- ATL with imperfect information 🗎 [Hoek and Wooldridge 2003]
- Epistemic situation calculus 🗎 [Scherl and Levesque 2003]
- Game Description Logic III 🗎 [Thielscher 2016]
- Dynamic epistemic logic 🗎 [Baltag, Moss, and Solecki 1998]
- Probabilistic Dynamic epistemic logic 🗎 [B. P. Kooi 2003]
- Interpreted systems 🗎 [Fagin et al. 1995]
- Explicit and implicit beliefs 🗎 [Lorini 2018]

Why we focus on Dynamic epistemic logic?

1. Action-oriented: it extends classical planning;
2. Has a nice classification of different decision problems.

Outline

Modeling using Dynamic Epistemic Logic (DEL)

Bounded epistemic planning

Unbounded epistemic planning

Automatic structures for decidability of unbounded epistemic planning when propositional pre/post

Knowledge-based programs

Conclusion

Outline

Modeling using Dynamic Epistemic Logic (DEL)

- Epistemic states
- Epistemic languages
- Actions
- Update product
- Dynamic language
- Succinct models

Bounded epistemic planning

Unbounded epistemic planning

Automatic structures for decidability of unbounded epistemic planning when propositional pre/post

Knowledge-based programs

Conclusion

Outline

Modeling using Dynamic Epistemic Logic (DEL)

- Epistemic states
- Epistemic languages
- Actions
- Update product
- Dynamic language
- Succinct models

Bounded epistemic planning

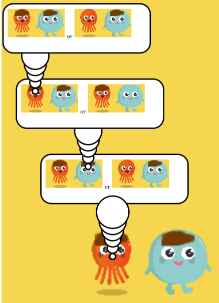
Unbounded epistemic planning

Automatic structures for decidability of unbounded epistemic planning when propositional pre/post

Knowledge-based programs

Conclusion

Examples of epistemic states



http://hintikkasworld.irisa.fr/

[demo IJCAI-ECAI 2018]

Epistemic states

[van Ditmarsch, van der Hoek, and B. Kooi 2008]

Let $AP = \{p, p_1, \ldots\}$ be a countable set of atomic propositions.
Let $AGT = \{a, b, c, \ldots\}$ be a finite set of agents.

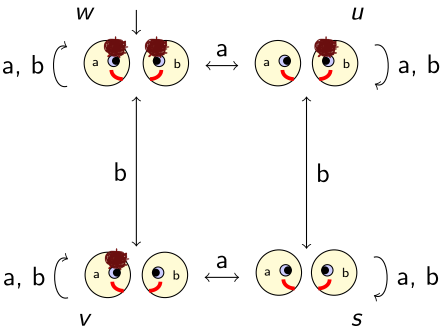
Definition

An epistemic model $\mathcal{M} = (W, (R_a)_{a \in AGT}, V)$ is a tuple where:

- $W = \{w, u, \ldots\}$ is a non-empty set of possible *worlds*;
- $R_a \subseteq W \times W$ is an *accessibility relation* for agent a;
- $V : W \to 2^{AP}$ is a *valuation function*.

A pair $(\mathcal{M}, w)$ is called a epistemic state, where w represents the actual world.

Example



- $W = \{w, u, v, s\}$;
- $R_a = \{(w, w), (w, u), (u, w), (u, u), (v, v), (v, s), (s, v), (s, s)\}$;
- $R_b = \{(w, w), (w, v), (v, w), (v, v), (u, u), (u, s), (s, u), (s, s)\}$;
- $V(w) = \{dirty_a, dirty_b\}$;
- $V(u) = \{dirty_b\}$;
- $V(v) = \{dirty_a\}$;
- $V(s) = \emptyset$.

Outline

Modeling using Dynamic Epistemic Logic (DEL)

- Epistemic states
- Epistemic languages
- Actions
- Update product
- Dynamic language
- Succinct models

Bounded epistemic planning

Unbounded epistemic planning

Automatic structures for decidability of unbounded epistemic planning when propositional pre/post

Knowledge-based programs

Conclusion

Syntax of $\mathcal{L}_{\mathbf{EL}}$

Definition

The syntax of $\mathcal{L}_{\mathbf{EL}}$ is given by the following grammar:

$$\varphi, \psi, \ldots \quad ::= \quad p \mid \neg\varphi \mid (\varphi \vee \psi) \mid K_a\varphi$$

where p ranges over AP and a ranges over AGT.

The size of φ is the number of symbols needed to write φ.

Notation (Dual operators)

$(\varphi \wedge \psi)$ *for* $\neg(\neg\varphi \vee \neg\psi)$;
$\hat{K}_a\varphi$ *for* $\neg K_a \neg\varphi$.

- $K_a\varphi$ is read 'agent a knows/believes that φ is true;
- $\hat{K}_a\varphi$ is read 'agent a considers φ as possible'.

Definition

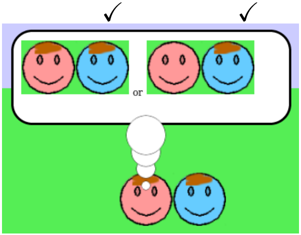
$\mathcal{L}_{Prop}$ is the set of propositional logic formulas.

Semantics of $\mathcal{L}_{\mathbf{EL}}$

Definition

The semantics of $\mathcal{L}_{\mathbf{EL}}$ is defined as follows:

$$\begin{array}{ll}
\mathcal{M}, w \models p & \text{if } p \in V(w); \\
\mathcal{M}, w \models \neg\varphi & \text{if it is not the case that } \mathcal{M}, w \models \varphi; \\
\mathcal{M}, w \models (\varphi \vee \psi) & \text{if } \mathcal{M}, w \models \varphi \text{ or } \mathcal{M}, w \models \psi; \\
\mathcal{M}, w \models K_a\varphi & \text{if for all } u \text{ s.t. } wR_au,\ \mathcal{M}, u \models \varphi
\end{array}$$

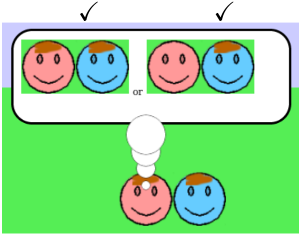


$\mathcal{M}, w \models K_a dirty_b$

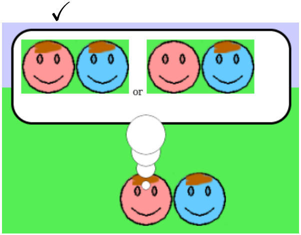
Dual operators

$$\mathcal{M}, w \models K_a\varphi \quad \text{if for all } u \text{ s.t. } wR_au, \mathcal{M}, u \models \varphi$$

$$\mathcal{M}, w \models \hat{K}_a\varphi \quad \text{if there exists } u \text{ s.t. } wR_au, \mathcal{M} \text{ and } u \models \varphi.$$



$\mathcal{M}, w \models K_a dirty_b$



$\mathcal{M}, w \models \hat{K}_a dirty_a$

Common knowledge

Common knowledge of φ among agents in group G

Definition

The syntax of $\mathcal{L}_{\mathbf{ELCK}}$ is given by the following grammar:

$$\varphi \ ::= \ p \ | \ \neg\varphi \ | \ (\varphi \vee \varphi) \ | \ K_a\varphi \ | \ C_G\varphi$$

where p ranges over AP, a ranges over AGT, and G ranges over 2^{AGT}.

Definition

The semantics of $\mathcal{L}_{\mathbf{ELCK}}$ extended by the following clause:

- $\mathcal{M}, w \models C_G\varphi$ if for all $u \in W, wR_G u$ implies $\mathcal{M}, u \models \varphi$ where R_G is the transitive closure of $\bigcup_{a\in G} R_a$.

Outline

Modeling using Dynamic Epistemic Logic (DEL)

- Epistemic states
- Epistemic languages
- Actions
- Update product
- Dynamic language
- Succinct models

Bounded epistemic planning

Unbounded epistemic planning

Automatic structures for decidability of unbounded epistemic planning when propositional pre/post

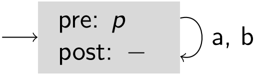
Knowledge-based programs

Conclusion

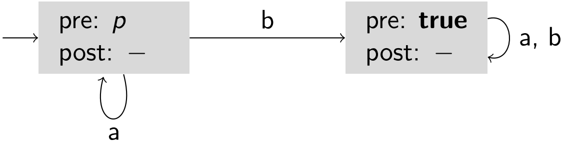
Examples of actions

[Baltag, Moss, and Solecki 1998]

Example (Public announcement of “*p*")

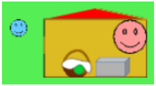


Example (Private announcement “*p*" to *a*)

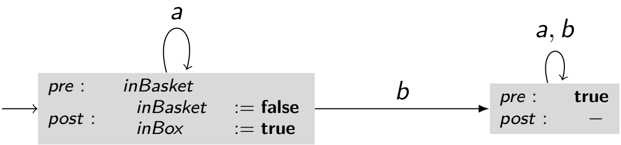


Modeling using Dynamic Epistemic Logic (DEL)
Epistemic states
Epistemic languages
Actions
Update product
Dynamic language
Succinct models
Bounded epistemic planning
Unbounded epistemic planning
Automatic structures for decidability of unbounded epistemic planning when propositional pre/post
Knowledge-based programs
Conclusion
References

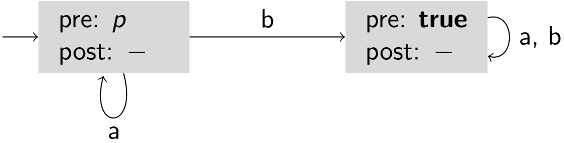
Examples of actions



Example (Transfer marble from basket to box)



Actions



Definition

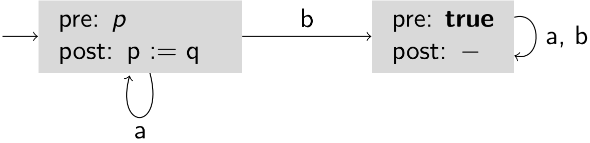
An event model $\mathcal{E} = (\mathrm{E}, (R^{\mathcal{E}}_a)_{a \in AGT}, pre, post)$ is a tuple where:

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

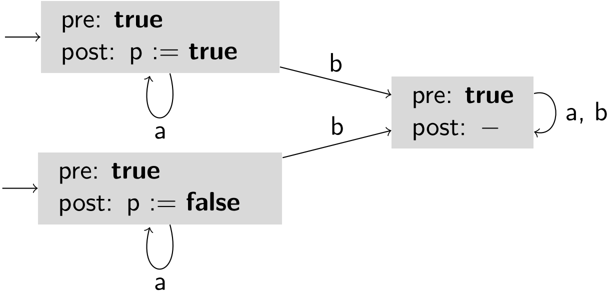
A pair $(\mathcal{E}, e)$ is called an action, where e represents the actual event of $(\mathcal{E}, e)$.
A pair $(\mathcal{E}, \mathrm{E}_0)$, for $\mathrm{E}_0 \subseteq \mathrm{E}$, is a non-deterministic action. The set E_0 is the set of triggerable events.

Deterministic and non-deterministic actions

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



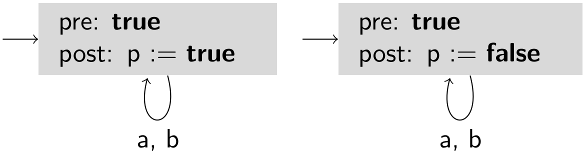
Non-deterministic action = multi-pointed event model



Public actions

Definition

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



pre: **true**
post: p := **true**

a, b

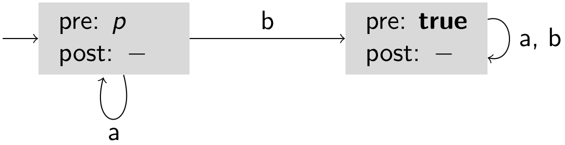
pre: **true**
post: p := **false**

a, b

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

Modeling using Dynamic Epistemic Logic (DEL)

- Epistemic states
- Epistemic languages
- Actions
- Update product
- Dynamic language
- Succinct models

Bounded epistemic planning

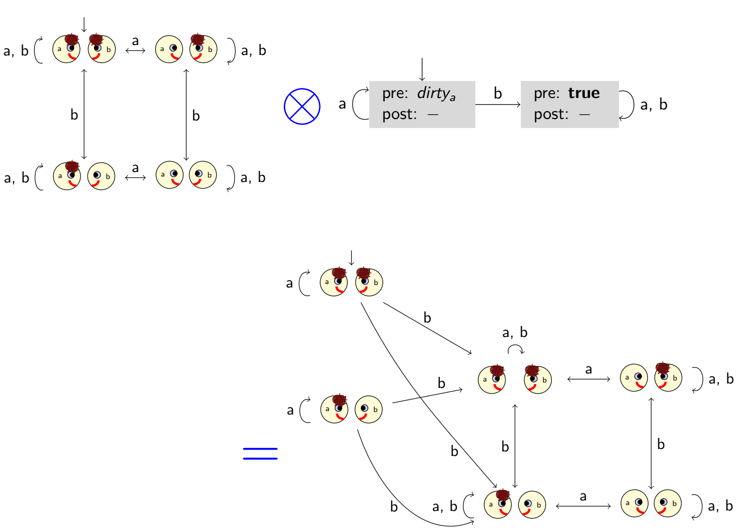
Unbounded epistemic planning

Automatic structures for decidability of unbounded epistemic planning when propositional pre/post

Knowledge-based programs

Conclusion

Example of an update product



pre: *dirty*$_a$
post: —

pre: **true**
post: —

Update product: formal definition

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

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

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

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

Pointed update products

Definition

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

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

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 $\mathcal{M} \otimes \mathcal{E} \otimes \dots \mathcal{E}$, n times.*
- *We write $we_1 \dots e_n \models \varphi$ instead of $\mathcal{M} \otimes \mathcal{E}^n, we_1 \dots e_n \models \varphi$.*

Outline

Modeling using Dynamic Epistemic Logic (DEL)

Epistemic states

Epistemic languages

Actions

Update product

Dynamic language

Succinct models

Bounded epistemic planning

Unbounded epistemic planning

Automatic structures for decidability of unbounded epistemic planning when propositional pre/post

Knowledge-based programs

Conclusion

Dynamic language

Definition

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

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

where $\mathcal{E}, \mathrm{E}_0$ 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:

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

Outline

Modeling using Dynamic Epistemic Logic (DEL)

- Epistemic states
- Epistemic languages
- Actions
- Update product
- Dynamic language
- Succinct models

Bounded epistemic planning

Unbounded epistemic planning

Automatic structures for decidability of unbounded epistemic planning when propositional pre/post

Knowledge-based programs

Conclusion

Possible world explosion

Example

Initially, number of possible worlds for Belote:

$$\binom{32}{8} \times \binom{24}{8} \times \binom{16}{8} \simeq 4 \times 10^{15}$$

Solution: succinct models

Represent succinctly epistemic and event models by:

- a Boolean formula to describe the valuations that correspond to the set of all worlds/events;
- programs (or Boolean formulas $R_a(\vec{x}, \vec{x}')$, or BDDs) for representing relations.

See [Benthem et al. 2015], [Benthem et al. 2018], [Charrier and Schwarzentruber 2017], [Charrier and Schwarzentruber 2018].

Modeling using Dynamic Epistemic Logic (DEL)
Epistemic states
Epistemic languages
Actions
Update product
Dynamic language
Succinct models
Bounded epistemic planning
Unbounded epistemic planning
Automatic structures for decidability of unbounded epistemic planning when propositional pre/post
Knowledge-based programs
Conclusion
References

Outline

Modeling using Dynamic Epistemic Logic (DEL)

Bounded epistemic planning
- Model checking problem
- Satisfiability problem

Unbounded epistemic planning

Automatic structures for decidability of unbounded epistemic planning when propositional pre/post

Knowledge-based programs

Conclusion

Outline

Modeling using Dynamic Epistemic Logic (DEL)

Bounded epistemic planning

- Model checking problem
  - A PSPACE procedure
  - PSPACE-hardness
- Satisfiability problem

Unbounded epistemic planning

Automatic structures for decidability of unbounded epistemic planning when propositional pre/post

Knowledge-based programs

Conclusion

Model checking problem

Definition

The *model checking problem* is defined as follows.

- Input:
  - An epistemic state $\mathcal{M}, w$;
  - A formula φ;
- Output: yes if $\mathcal{M}, w \models \varphi$; no otherwise.

Motivation: bounded epistemic planning

- Checking the existence of a bounded sequence of actions leading to a γ-state:

$$\mathcal{M}, w \models \langle \mathcal{E}, \mathsf{E}_0 \rangle \dots \langle \mathcal{E}, \mathsf{E}_0 \rangle \gamma$$

iff

there are actions $e_1, \dots e_n$ in E_0 such that $we_1, \dots, e_n \models \gamma$

- Checking the existence of a bounded strategy leading to a γ-state:

$$\mathcal{M}, w \models \langle \mathcal{E}, \mathsf{E}_0 \rangle [\mathcal{E}', \mathsf{E}'_0] \dots \langle \mathcal{E}, \mathsf{E}_0 \rangle [\mathcal{E}', \mathsf{E}'_0] \gamma$$

Dynamic-free language

Theorem

If φ is dynamic-free then the model checking problem is P-complete.

Proof.

- P-hardness: same lower bound proof as for temporal logic CTL 📄 [Schnoebelen 2002b]
- in P: next slide

□

Algorithm

```
function mc($\mathcal{M}, \varphi$)
    match $\varphi$ do
        case $p$ :
            return $\{w \mid p \text{ holds in } \mathcal{M}, w\}$
        case $\neg\psi$ :
            return $\overline{\text{mc}(\mathcal{M}, \psi)}$
        case $(\psi_1 \vee \psi_2)$ :
            return $\text{mc}(\mathcal{M}, \psi_1) \cup \text{mc}(\mathcal{M}, \psi_2)$
        case $K_a\psi$ :
            return $\{w \mid R_a(w) \subseteq \text{mc}(\mathcal{M}, \psi)\}$
check whether $w \in mc(\mathcal{M}, \varphi)$
```

Algorithm also for deterministic public actions

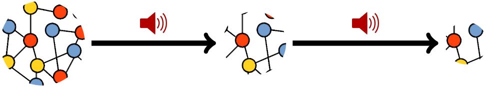
```
function mc($\mathcal{M}, \varphi$)
    match $\varphi$ do
        case $p$ :
            return $\{w \mid p$ holds in $\mathcal{M}, w\}$
        case $\neg\psi$ :
            return $\overline{\text{mc}(\mathcal{M}, \psi)}$
        case $(\psi_1 \vee \psi_2)$ :
            return $\text{mc}(\mathcal{M}, \psi_1) \cup \text{mc}(\mathcal{M}, \psi_2)$
        case $K_a\psi$ :
            return $\{w \mid R_a(w) \subseteq \text{mc}(\mathcal{M}, \psi)\}$
        case $\langle\mathcal{E}, e\rangle\psi$ :
            return $\text{mc}(\mathcal{M}, pre(e)) \cap \{w \mid (w, e) \in \text{mc}(\mathcal{M} \otimes \mathcal{E}, \psi)\}$
check whether $w \in mc(\mathcal{M}, \varphi)$
```

Main results

Theorem

Model checking with deterministic public actions is P-complete.

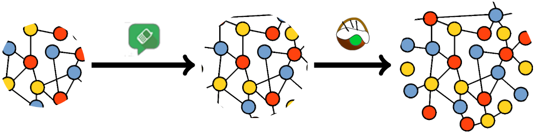
[van Benthem, 2011]



Theorem

Model checking is PSPACE-complete.

[Aucher et al, 2013]



Outline

Modeling using Dynamic Epistemic Logic (DEL)

Bounded epistemic planning

- Model checking problem
  - A PSPACE procedure
  - PSPACE-hardness
- Satisfiability problem

Unbounded epistemic planning

Automatic structures for decidability of unbounded epistemic planning when propositional pre/post

Knowledge-based programs

Conclusion

A PSPACE procedure for model checking

Specification

$$w\vec{e},\ \varphi \quad \longrightarrow \boxed{\quad \text{mc} \quad} \longrightarrow \quad \text{yes, if } w\vec{e} \models \varphi$$

(no otherwise)

such that $w\vec{e}$ is defined

A PSPACE procedure for model checking

```
function $mc(w\vec{e}, \varphi)$
    match $\varphi$ do
        case $p$ :
            return $inval(p, w\vec{e})$
        case $\neg\psi$ :
            return not $mc(w\vec{e}, \varphi)$
        case $(\psi_1 \vee \psi_2)$ :
            return $mc(\mathcal{M}, w, \psi_1)$ or $mc(\mathcal{M}, w, \psi_2)$
        case $K_a\psi$ :
            for $u\vec{f}$ such that $u \in R_a(w)$ and $\vec{e} \rightarrow_a \vec{f}$ do
                if $in(u\vec{f})$ and not $mc(u\vec{f}, \psi)$ then return false
            return true
        case $\langle \mathcal{E}, \mathrm{E}_0 \rangle \psi$ :
            for $e \in \mathrm{E}_0$ do
                if $mc(w\vec{e}, pre(e))$ and mc$(w\vec{e}::e, \psi)$ then return true
            return false
$mc(w, \varphi)$
```

Subroutines *inval* and *in*

```
function $inval(p, w\vec{e})$
    case $\vec{e} = \epsilon$: return ($p$ is true in $w$)
    case $\vec{e} = \vec{e}'::e$ and: $mc(w\vec{e}', post(e, p))$
```

```
function $in(w\vec{e})$
    case $\vec{e} = \epsilon$: return true
    case $\vec{e} = \vec{e}'::e$: return $mc(w\vec{e}', pre(e))$ and $in(w\vec{e}')$
```

Outline

Modeling using Dynamic Epistemic Logic (DEL)

Bounded epistemic planning

- Model checking problem
  - A PSPACE procedure
  - PSPACE-hardness
- Satisfiability problem

Unbounded epistemic planning

Automatic structures for decidability of unbounded epistemic planning when propositional pre/post

Knowledge-based programs

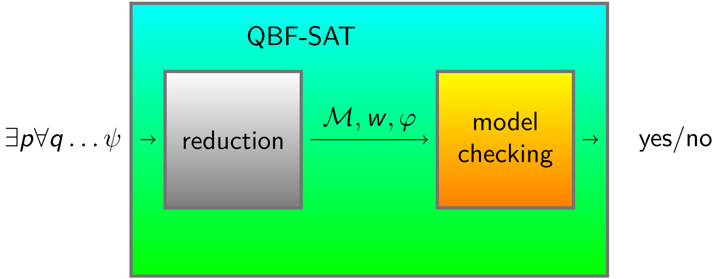
Conclusion

PSPACE-hardness

Theorem

Model checking is PSPACE-hard.

Proof.



$\varphi := \langle p := \mathit{false} \cup p := \mathit{true} \rangle [q := \mathit{false} \cup q := \mathit{true}] \ldots \psi$ □

PSPACE-hardness

Theorem

Model checking is PSPACE-hard already for:

- *Non-deterministic public actions (previous slide);*
- *Deterministic epistemic actions Pol, Rooij, and Szymanik 2015, Bolander, Jensen, and Schwarzentruber 2015a.*

Further reading: parameterized complexity for DEL model checking: Pol, Rooij, and Szymanik 2015

	Explicit models	Succinct models
Deterministic public actions	P-c	PSPACE-c
All	PSPACE-c	PSPACE-c

Outline

Modeling using Dynamic Epistemic Logic (DEL)

Bounded epistemic planning

- Model checking problem
- Satisfiability problem

Unbounded epistemic planning

Automatic structures for decidability of unbounded epistemic planning when propositional pre/post

Knowledge-based programs

Conclusion

Satisfiability problem definition

Definition

The *satisfiability problem in DEL* is the following decision problem.

- Input: a formula φ;
- Output: yes if there is an epistemic state $\mathcal{M}, w$ such that $\mathcal{M}, w \models \varphi$; no otherwise.

Motivation: parameterized bounded epistemic planning

- there exists a bounded sequence of actions leading to a γ-state from any ψ-epistemic state

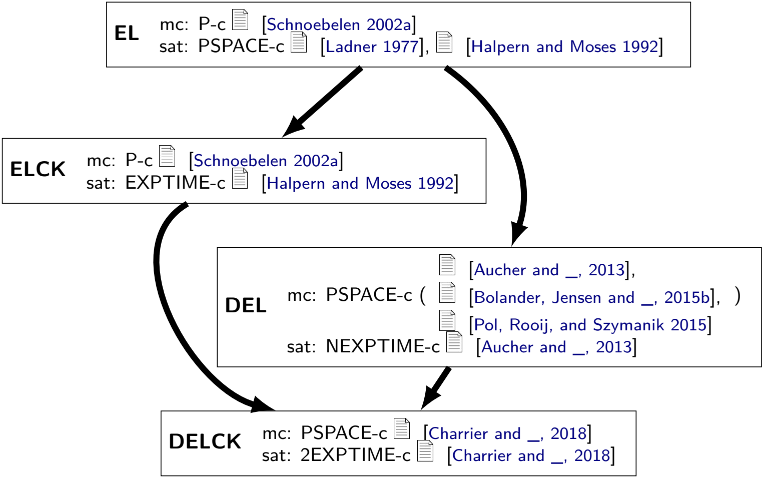
  iff $\psi \rightarrow \langle \mathcal{E}, \mathsf{E}_0 \rangle \dots \langle \mathcal{E}, \mathsf{E}_0 \rangle \gamma$ is satisfiable

- There is a bounded strategy leading to a γ-state from any ψ-epistemic state:

  iff

  $\psi \rightarrow \langle \mathcal{E}, \mathsf{E}_0 \rangle [\mathcal{E}', \mathsf{E}'_0] \dots \langle \mathcal{E}, \mathsf{E}_0 \rangle [\mathcal{E}', \mathsf{E}'_0] \gamma$ is satisfiable

Complexity results



EL mc: P-c [Schnoebelen 2002a]
sat: PSPACE-c [Ladner 1977], [Halpern and Moses 1992]

ELCK mc: P-c [Schnoebelen 2002a]
sat: EXPTIME-c [Halpern and Moses 1992]

DEL mc: PSPACE-c ([Aucher and _, 2013], [Bolander, Jensen and _, 2015b], [Pol, Rooij, and Szymanik 2015])
sat: NEXPTIME-c [Aucher and _, 2013]

DELCK mc: PSPACE-c [Charrier and _, 2018]
sat: 2EXPTIME-c [Charrier and _, 2018]

All complexities remain the same for succinct event models in the language, except P-c becomes PSPACE-c (see [Charrier and _, 2018]).

Outline

Modeling using Dynamic Epistemic Logic (DEL)

Bounded epistemic planning

Unbounded epistemic planning

- Epistemic planning problem
- Planning as a first-order query in DEL structures
- Undecidability
- Event model restrictions

Automatic structures for decidability of unbounded epistemic planning when propositional pre/post

Knowledge-based programs

Conclusion

Outline

Modeling using Dynamic Epistemic Logic (DEL)

Bounded epistemic planning

Unbounded epistemic planning

- Epistemic planning problem
- Planning as a first-order query in DEL structures
- Undecidability
- Event model restrictions

Automatic structures for decidability of unbounded epistemic planning when propositional pre/post

Knowledge-based programs

Conclusion

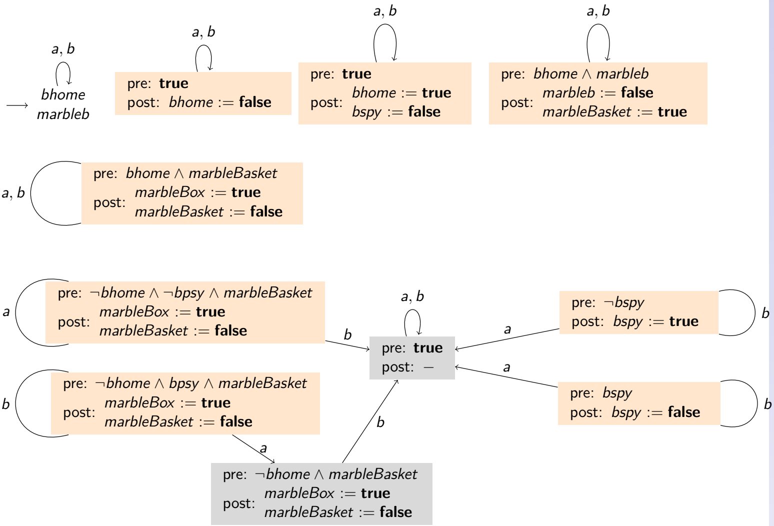
Epistemic planning instance

Definition

An epistemic planning instance is a tuple $\mathcal{M}, w, \mathcal{E}, \mathsf{E}_0, \gamma$ where:

- $\mathcal{M}, w$ is a pointed epistemic model; (initial situation)
- $\mathcal{E}$ is an event model;
- E_0 is a subset of events in $\mathcal{E}$; (repertoire of events)
- γ an epistemic formula. (goal)

Example of planning instance $(\mathcal{M}, w, \mathcal{E}, \boxed{\mathsf{E}_0}, \gamma)$:



a, b

$\longrightarrow$ *bhome* *marbleb*

a, b

pre: **true**
post: *bhome* := **false**

a, b

pre: **true**
post: *bhome* := **true**
bspy := **false**

a, b

pre: *bhome* $\wedge$ *marbleb*
post: *marbleb* := **false**
marbleBasket := **true**

a, b

pre: *bhome* $\wedge$ *marbleBasket*
post: *marbleBox* := **true**
marbleBasket := **false**

a

pre: $\neg$*bhome* $\wedge$ $\neg$*bpsy* $\wedge$ *marbleBasket*
post: *marbleBox* := **true**
marbleBasket := **false**

b

pre: $\neg$*bhome* $\wedge$ *bpsy* $\wedge$ *marbleBasket*
post: *marbleBox* := **true**
marbleBasket := **false**

b

a, b

pre: **true**
post: $-$

a

pre: $\neg$*bspy*
post: *bspy* := **true**

b

a

pre: *bspy*
post: *bspy* := **false**

b

a

b

pre: $\neg$*bhome* $\wedge$ *marbleBasket*
post: *marbleBox* := **true**
marbleBasket := **false**

$$\gamma := K_b marbleBox \wedge K_a K_b marbleBasket$$

Epistemic planning problem

Definition

The epistemic planning problem is defined as follows:

- Input: an epistemic planning instance $(\mathcal{M}, w, \mathcal{E}, \mathsf{E}_0, \gamma)$;
- Output: yes if there exists a sequence $e_1, \ldots, e_\ell \in \mathsf{E}_0$ such that $we_1 \ldots e_\ell \models \gamma$; no otherwise.

Outline

Modeling using Dynamic Epistemic Logic (DEL)

Bounded epistemic planning

Unbounded epistemic planning

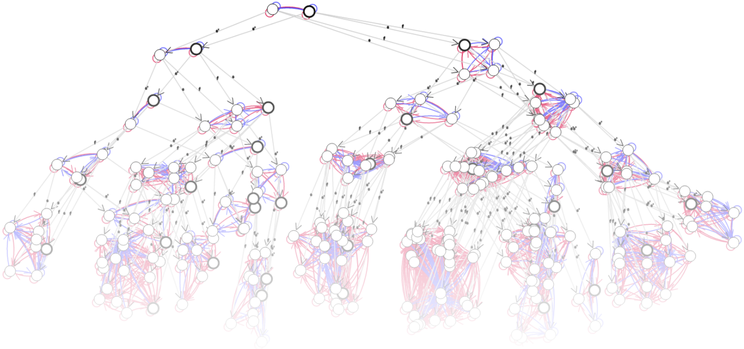
- Epistemic planning problem
- Planning as a first-order query in DEL structures
- Undecidability
- Event model restrictions

Automatic structures for decidability of unbounded epistemic planning when propositional pre/post

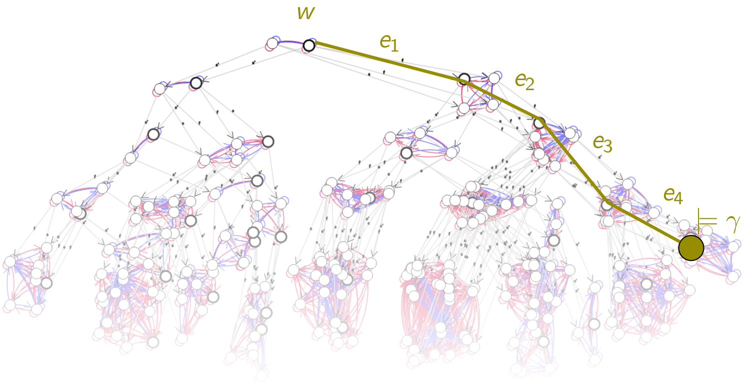
Knowledge-based programs

Conclusion

Planning as a first-order query in DEL structures



Planning as a first-order query in DEL structures



DEL presentation: formal definition

Definition

A DEL presentation is a pair $(\mathcal{M}, \mathcal{E})$ where $\mathcal{M}$ is an epistemic model and $\mathcal{E}$ is an event model.

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

Notation

- *$\mathcal{H}_n$ is the set of worlds of $\mathcal{M} \otimes \mathcal{E}^n$.*
- *Worlds of $\mathcal{M} \otimes \mathcal{E}^n$ are written $h = we_1 \dots e_n$.*

DEL structure: formal definition

Let $(\mathcal{M}, \mathcal{E})$ be a DEL presentation. A DEL structure is the unraveling of some DEL presentation $(\mathcal{M}, \mathcal{E})$.

Definition

The DEL structure denoted by $(\mathcal{M}, \mathcal{E})$ is the structure

$$\mathcal{M}\mathcal{E}^* = (\mathcal{H}, \rightarrow, (R_a)_{a \in AGT}, (p)_{p \in AP}),$$

where

- $\mathcal{H} = \bigcup_{n \in \mathbb{N}} \mathcal{H}_n$; (histories)
- $h \rightarrow h'$ whenever $h' = he$ for some event e;
- hR_ah' whenever hR_ah' in $\mathcal{M} \otimes \mathcal{E}^n$, for some n;
- $p(h)$ holds if p holds in h in $\mathcal{M} \otimes \mathcal{E}^n$.

Epistemic logic embedded in First-order logic

Theorem

Given an epistemic formula γ, one can effectively compute a first-order formula $tr(\gamma)(x)$ such that

$$\mathcal{ME}^*, h \models \gamma \text{ iff } \mathcal{ME}^*, [x := h] \models tr(\gamma)(x).$$

Example

γ	$tr(\gamma)(x)$
$K_a p$	$\forall y R_a(x,y) \rightarrow p(y)$
$q \wedge \hat{K}_a q$	$q(x) \wedge \exists y R_a(x,y) \wedge q(y)$

Planning as a first-order query

Proposition

A planning instance $\mathcal{M}, w, \mathcal{E}, \mathsf{E}_0, \gamma$ is positive

iff there exists a history $we_1 \ldots e_\ell$ of $\mathcal{M}\mathcal{E}^*$ such that:

- $e_1, \ldots, e_\ell \in \mathsf{E}_0$;
- $we_1 \ldots e_\ell \models \gamma$;

iff $\mathcal{M}\mathcal{E}^* \models \exists x(\mathit{historyE0}(x) \wedge tr(\gamma)(x))$

PS: handling `historyE0`(x) is small technical detail...

Outline

Modeling using Dynamic Epistemic Logic (DEL)

Bounded epistemic planning

Unbounded epistemic planning

- Epistemic planning problem
- Planning as a first-order query in DEL structures
- Undecidability
- Event model restrictions

Automatic structures for decidability of unbounded epistemic planning when propositional pre/post

Knowledge-based programs

Conclusion

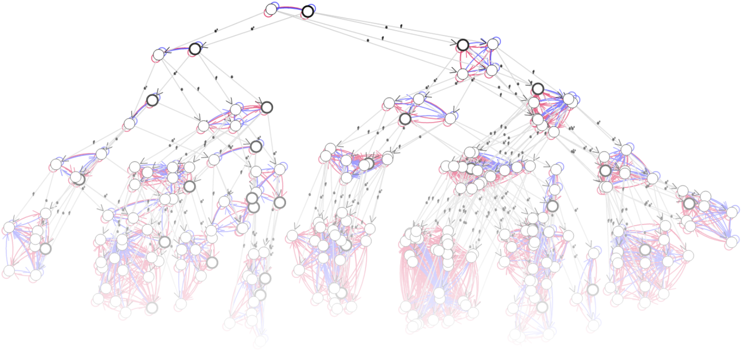
Undecidability of epistemic planning

Theorem

Epistemic planning problem is undecidable.

Proof.

DEL structures are Turing-complete! ([Bolander and Andersen 2011], [Cong, Pinchinat, and Schwarzentruber 2018])



□

Outline

Modeling using Dynamic Epistemic Logic (DEL)

Bounded epistemic planning

Unbounded epistemic planning

- Epistemic planning problem
- Planning as a first-order query in DEL structures
- Undecidability
- Event model restrictions

Automatic structures for decidability of unbounded epistemic planning when propositional pre/post

Knowledge-based programs

Conclusion

Event model restrictions

Modal depth

K_aK_bp: $md = 2$

$K_a\hat{K}_b\hat{K}_cp$: md $= 3$

pre post	$md = 0$	$md = 1$	$md \geq 2$
non-ontic	dec	?	undec
ontic	dec	undec	

🟥 What we just seen

🟧 Similar proof (see [Aucher and Bolander 2013], [Charrier, Maubert, and Schwarzentruber 2016])

⬜ Open problem

🟩 Next section!

Outline

Modeling using Dynamic Epistemic Logic (DEL)

Bounded epistemic planning

Unbounded epistemic planning

Automatic structures for decidability of unbounded epistemic planning when propositional pre/post

- PDEL Planning
- Automatic structures
- PDEL structures are automatic
- Wrap up

Knowledge-based programs

Conclusion

Outline

Modeling using Dynamic Epistemic Logic (DEL)

Bounded epistemic planning

Unbounded epistemic planning

Automatic structures for decidability of unbounded epistemic planning when propositional pre/post

- PDEL Planning
- Automatic structures
- PDEL structures are automatic
- Wrap up

Knowledge-based programs

Conclusion

PDEL Planning

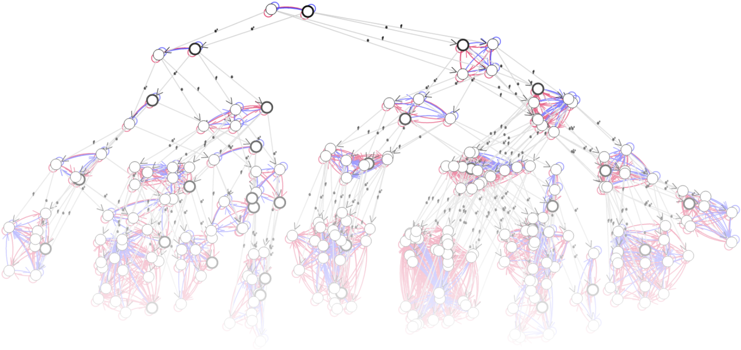
Call PDEL presentation a DEL presentation where every precondition is propositional, and call PDEL structure a DEL structure arising from a PDEL presentation.

Definition (PDEL planning)

- Input: an epistemic planning instance $(\mathcal{M}, w, \mathcal{E}, \mathsf{E}_0, \varphi)$ where $(\mathcal{M}, \mathcal{E})$ is a PDEL presentation;
- Output: yes if there exists a history $we_1 \ldots e_\ell$ in $\mathcal{M}\mathcal{E}^*$ such that $we_1 \ldots e_\ell \models \varphi$ and $e_1, \ldots, e_\ell \in \mathsf{E}_0$.

Is PDEL planning decidable?

Issue: the DEL structure is infinite...



Two possible attitudes towards infinite objects

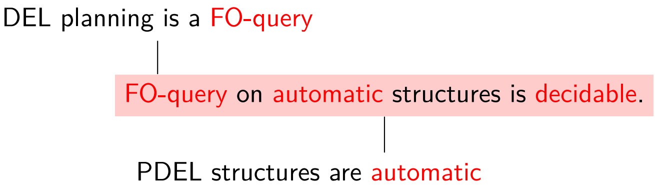
- Try to prove Turing-completeness hence undecidability;
- Try to prove regularity of the structure hence decidability.

PDEL planning is decidable.

Theorem

PDEL planning is decidable ([Yu, Wen, and Liu 2013], [Aucher, Maubert, and Pinchinat 2014]).

Proof.



DEL planning is a FO-query

FO-query on automatic structures is decidable.

PDEL structures are automatic

□

It is even decidable for epistemic linear μ-calculus!
[Douéneau-Tabot, Pinchinat, and Schwarzentruber 2018]

Outline

Modeling using Dynamic Epistemic Logic (DEL)

Bounded epistemic planning

Unbounded epistemic planning

Automatic structures for decidability of unbounded epistemic planning when propositional pre/post

- PDEL Planning
- Automatic structures
  - Finite automata
  - Automatic presentations
  - First-order logic on automatic structures
- PDEL structures are automatic
- Wrap up

Knowledge-based programs

Conclusion

Outline

Modeling using Dynamic Epistemic Logic (DEL)

Bounded epistemic planning

Unbounded epistemic planning

Automatic structures for decidability of unbounded epistemic planning when propositional pre/post

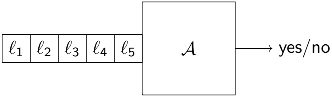
- PDEL Planning
- Automatic structures
  - Finite automata
  - Automatic presentations
  - First-order logic on automatic structures
- PDEL structures are automatic
- Wrap up

Knowledge-based programs

Conclusion

Finite automata

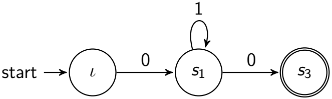
Let Σ be an alphabet. Σ^* is the set of all finite words over Σ.



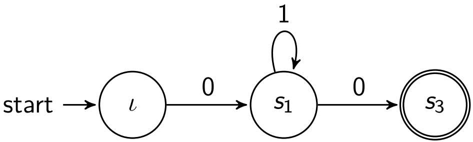
Definition

A word automaton $\mathcal{A}$ is a tuple $\mathcal{A} = (S, \iota, \Delta, F)$ where

- S is a finite set of states, $\iota \in S$ is the initial state;
- $\Delta \subseteq S \times \Sigma \times S$ is the transition relation;
- $F \subseteq S$ is the set of accepting states.



Regular languages



- An execution of $\mathcal{A}$ on $\alpha = \ell_1 \dots \ell_n \in \Sigma^*$...
- A word is accepted by $\mathcal{A}$ if there exists an accepting execution of $\mathcal{A}$ on it.
- The language accepted by $\mathcal{A}$ is the set $L(\mathcal{A}) \subseteq \Sigma^*$ of all words accepted by $\mathcal{A}$.

Definition

A language $L \subseteq \Sigma^*$ is regular if there exists a finite automaton $\mathcal{A}$ such that $L = L(\mathcal{A})$.

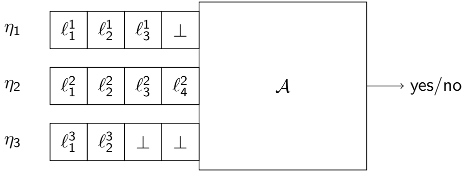
The language accepted by the automaton drawn above is the set of words of the form $01 \dots 10$, and is often written 01^*0.

Theorem

The emptiness problem for word automata is decidable in NLOGSPACE.

Regular relations

Let $\Sigma_\perp = \Sigma \cup \{\perp\}$, where $\perp$ is a fresh symbol.



η_1
ℓ^1_1 ℓ^1_2 ℓ^1_3 $\perp$
η_2
ℓ^2_1 ℓ^2_2 ℓ^2_3 ℓ^2_4
η_3
ℓ^3_1 ℓ^3_2 $\perp$ $\perp$
$\mathcal{A}$
yes/no

Definition

The convolution of $\eta_1, \ldots, \eta_n \in \Sigma^*$, written $\odot(\eta_1, \ldots, \eta_n)$, is the word over alphabet $(\Sigma_\perp)^n$ obtained by left-aligning $\eta_1, \ldots, \eta_n$ while completing with $\perp$.

Definition

The convolution of a relation $R \subseteq (\Sigma^*)^n$ is the language

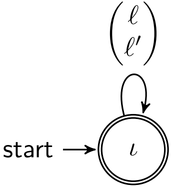
$$\odot R = \{\odot(\eta_1, \ldots, \eta_n) \mid (\eta_1, \ldots, \eta_n) \in R\} \subseteq ((\Sigma_\perp)^n)^*$$

Definition

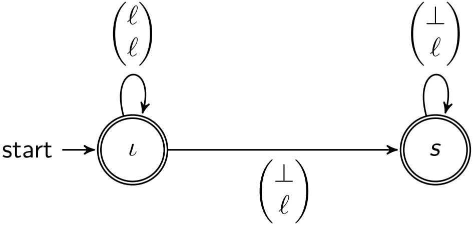
$R \subseteq (\Sigma^*)^n$ is regular whenever there is a finite automaton over alphabet $(\Sigma_\perp)^n$ that accepts $\odot R$.

Examples of regular relations

- The binary equal-length relation *el*, *i.e.*, pairs (η, η') with $|\eta| = |\eta'|$.



- The binary prefix relation $\preceq$.



Closure properties of regular relations

Theorem

Let R, R' be regular relations over Σ^*. Then the following relations are also regular:

- *Union* $R \cup R'$;
- *Intersection* $R \cap R'$;
- *Relative complementation* $R \setminus R'$;

Moreover there is an effective procedure that, given automata for $\odot R$ and $\odot R'$, computes an automaton for the convolution of each of the resulting relations.

Proof.

Use standard automata constructions, *e.g.*, synchronous product for intersection. □

Remark

Computing the automaton for $\odot R \setminus R'$ requires to complement $\mathcal{A}$ for $\odot R'$, that relies on the determinization of $\mathcal{A}$. (an exponential cost in general; it is a powerset construction).

The projection of a regular relation is regular

Theorem

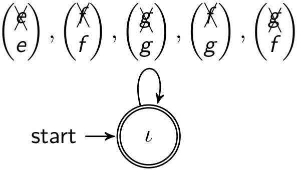
Let $R \subseteq (\Sigma^*)^r$ be regular relation.
Then one can effectively compute an automaton $\mathcal{B}$ s.t.

$$L(\mathcal{B}) = \odot(\{(\eta_2, \ldots, \eta_r) | \text{ there exists } \eta_1, (\eta_1, \eta_2, \ldots, \eta_r) \in R\}).$$

Proof.

Forget the first coordinate. □

Example


$\binom{\cancel{e}}{e}, \binom{\cancel{f}}{f}, \binom{\cancel{g}}{g}, \binom{\cancel{f}}{g}, \binom{\cancel{g}}{f}$
start
ι

Remark

The projected automaton is non-deterministic in general.

Outline

Modeling using Dynamic Epistemic Logic (DEL)

Bounded epistemic planning

Unbounded epistemic planning

Automatic structures for decidability of unbounded epistemic planning when propositional pre/post

- PDEL Planning
- Automatic structures
  - Finite automata
  - Automatic presentations
  - First-order logic on automatic structures
- PDEL structures are automatic
- Wrap up

Knowledge-based programs

Conclusion

Automatic presentations

Let $\mathcal{S} = \langle S, (R_i)_{i \in I} \rangle$ be a structure.

Definition

An automatic presentation of $\mathcal{S}$ consists of a pair $(\bar{\mathcal{A}}, \nu)$ s.t.

- $\bar{\mathcal{A}}$ is a tuple of automata $\langle \mathcal{A}_S, (\mathcal{A}_{R_i})_{i \in I} \rangle$;
- $\nu : L(\mathcal{A}_S) \to S$ is a bijective mapping, and we let

$$\nu^{-1}(R_i) := \{(\eta_1, \ldots, \eta_{r_i}) \in (\Sigma^*)^{r_i} \mid R_i(\nu(\eta_1), \ldots, \nu(\eta_{r_i}))\}.$$

s.t. $L(\mathcal{A}_{R_i}) = \odot \nu^{-1}(R_i)$.

Intuitively, words from $L(\mathcal{A}_S)$ encode elements of S (via mapping ν) in such a way that the induced relations $\nu^{-1}(R_i)$ are regular.
An automatic structure is a structure that has an automatic presentation.

Example

$(\mathbb{N}, succ)$ with $succ = \{(n, n+1) \mid n \in \mathbb{N}\}$ is an automatic structure: take alphabet $\Sigma = \{\ell\}$ and $\nu : \ell^* \to \mathbb{N}$, and automaton for relation $\odot succ$ is the one for words of the form $\binom{\ell}{\ell} \ldots \binom{\ell}{\ell} \binom{\perp}{\ell}$.

Other examples of automatic structures

- Every finite structure is automatic.
- Given a DEL presentation where pre/post are propositional, the associated DEL structure is automatic.
  (next section)

Outline

Modeling using Dynamic Epistemic Logic (DEL)

Bounded epistemic planning

Unbounded epistemic planning

Automatic structures for decidability of unbounded epistemic planning when propositional pre/post

- PDEL Planning
- Automatic structures
  - Finite automata
  - Automatic presentations
  - First-order logic on automatic structures
- PDEL structures are automatic
- Wrap up

Knowledge-based programs

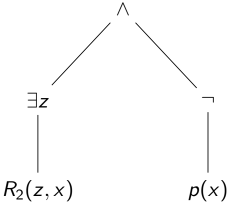
Conclusion

First-order logic on automatic structures

Theorem

For every automatic presentation $(\bar{A}, \nu)$ of structure,every first-order formula $\Phi(x_1, \ldots, x_n)$ induces a relation R of arity n with $\nu^{-1}(R)$ regular. Moreover, the automaton for $\odot\nu^{-1}(R)$ can be effectively computed.

Take $\exists z R_2(z, x) \wedge \neg p(x)$.



Bottom-up construction:

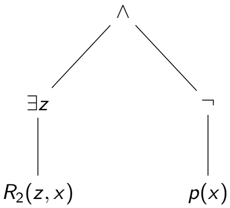
1. Project $\mathcal{A}_{R_2(z,x)}$ on first component and get $\mathcal{A}_{\exists z R_2(x,z)}$;
2. Complement $\mathcal{A}_{p(x)}$, get $\mathcal{A}^c_{p(x)}$, compute $\mathcal{A}_S \cap \mathcal{A}^c_{p(x)}$ and get $\mathcal{A}_{\neg p(x)}$;
3. Compute $\mathcal{A}_{\exists z R_2(x,z)} \cap \mathcal{A}_{\neg p(x)}$ to get $\mathcal{A}_{\exists z R_2(z,x) \wedge \neg p(x)}$.

First-order logic on automatic structures

Theorem

For every automatic presentation $(\bar{\mathcal{A}}, \nu)$ of structure,every first-order formula $\Phi(x_1, \ldots, x_n)$ induces a relation R of arity n with $\nu^{-1}(R)$ regular. Moreover, the automaton for $\odot\nu^{-1}(R)$ can be effectively computed.

Take $\exists z R_2(z, x) \wedge \neg p(x)$.



Bottom-up construction:

1. Project $\mathcal{A}_{R_2(z,x)}$ on first component and get $\mathcal{A}_{\exists z R_2(x,z)}$;
2. Complement $\mathcal{A}_{p(x)}$, get $\mathcal{A}^c_{p(x)}$, compute $\mathcal{A}_S \cap \mathcal{A}^c_{p(x)}$ and get $\mathcal{A}_{\neg p(x)}$;
3. Compute $\mathcal{A}_{\exists z R_2(x,z)} \cap \mathcal{A}_{\neg p(x)}$ to get $\mathcal{A}_{\exists z R_2(z,x) \wedge \neg p(x)}$.

Corollary

The first-order theory of each automatically presentable structure is decidable.

Outline

Modeling using Dynamic Epistemic Logic (DEL)

Bounded epistemic planning

Unbounded epistemic planning

Automatic structures for decidability of unbounded epistemic planning when propositional pre/post

- PDEL Planning
- Automatic structures
- PDEL structures are automatic
- Wrap up

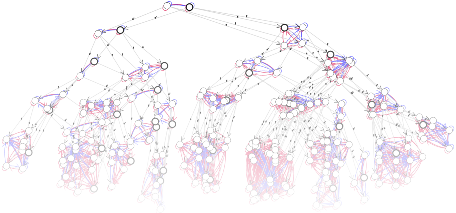
Knowledge-based programs

Conclusion

PDEL structures are automatic

Theorem

Given a PDEL presentation $(\mathcal{M}, \mathcal{E})$, the structure $\mathcal{M}\mathcal{E}^* = (\mathcal{H}, \to, (R_a)_{a \in AGT}, (p)_{p \in AP})$ is automatic.



Proof: We exhibit an automatic presentation $(\bar{\mathcal{A}}, \nu)$.

First, $\nu := id$, that is, every history $we_1 \dots e_n \in \mathcal{H}$ is encoded as the word $we_1 \dots e_n \in (W \cup \mathsf{E})^*$.

Now we define $\bar{\mathcal{A}} = \langle \mathcal{A}_{\mathcal{H}}, \mathcal{A}_{\to}, (\mathcal{A}_{R_a})_{a \in AGT}, (\mathcal{A}_p)_{p \in AP} \rangle$.

Some ideas for $\mathcal{A}_{\mathcal{H}}$

Notation

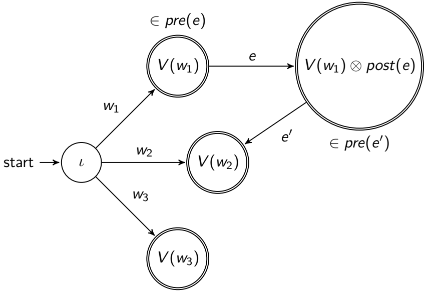
- *Given an event e, view $pre(e)$ as a subset of valuations.*

  $$e.g., \text{ view } p \vee q \text{ as } \{\{p\}, \{q\}, \{p, q\}\}.$$

- *For all valuations P, let $P \otimes post(e)$ be the valuation P updated by $post(e)$*

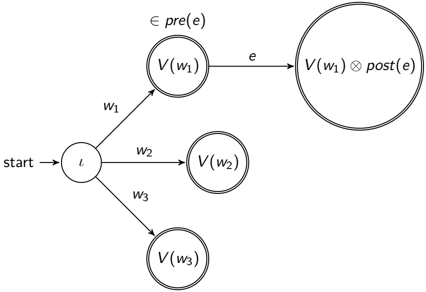
  $$e.g., \{p, q\} \otimes [p := \bot, r := \top] = \{q, r\}.$$

Idea for $\mathcal{A}_{\mathcal{H}}$:



$$L(\mathcal{A}_{\mathcal{H}}) = \{w_1, w_2, w_3, \ldots, w_1e, \ldots, w_1ee', \ldots\}$$

Definition of $\mathcal{A}_{\mathcal{H}}$, and of $\mathcal{A}_p$ ($p \in AP$)



Let $\mathcal{A}_{\mathcal{H}} = (S, \iota, \Delta, S \setminus \{\iota\})$ where

- $S = \{\iota\} \cup 2^{AP}$;
- $(\iota, w, V(w)) \in \Delta$, for every $w \in W$;
- $(P, e, P \otimes post(e)) \in \delta$ whenever $P \in pre(e)$.

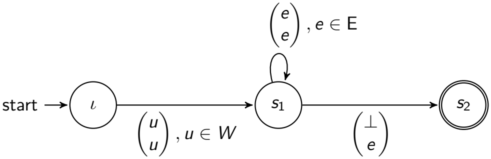
Incidentally, we take $\mathcal{A}_p = (S, \iota, \Delta, \{P \mid p \in P\})$.

Definition of $\mathcal{A}_\rightarrow$

We want an automaton for

$$\odot(\rightarrow) = \{\begin{pmatrix} u \\ u \end{pmatrix} \dots \begin{pmatrix} e_n \\ e_n \end{pmatrix} \begin{pmatrix} \perp \\ e_{n+1} \end{pmatrix} \mid ue_1 \dots e_n e_{n+1} \in \mathcal{H}\}$$

- First, consider $\mathcal{A}$:



start
ι
s_1
s_2
$\begin{pmatrix} u \\ u \end{pmatrix}, u \in W$
$\begin{pmatrix} e \\ e \end{pmatrix}, e \in \mathsf{E}$
$\begin{pmatrix} \perp \\ e \end{pmatrix}$

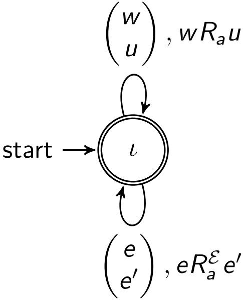
- Second, we make sure that accepted pairs are histories. Build automaton $\mathcal{B}$ for the binary relation $\mathcal{H} \times \mathcal{H}$ and define:

$$\mathcal{A}_\rightarrow = \mathcal{A} \cap \mathcal{B}$$

Definition of $\mathcal{A}_{R_a}$

$$\mathcal{A}_{R_a} = \mathcal{A} \cap \mathcal{B}$$

- where $\mathcal{A}$ is:



- and automaton $\mathcal{B}$ is as previous slide before for $\mathcal{H} \times \mathcal{H}$.

This ends the proof of Theorem on Slide 85.

Outline

Modeling using Dynamic Epistemic Logic (DEL)

Bounded epistemic planning

Unbounded epistemic planning

Automatic structures for decidability of unbounded epistemic planning when propositional pre/post

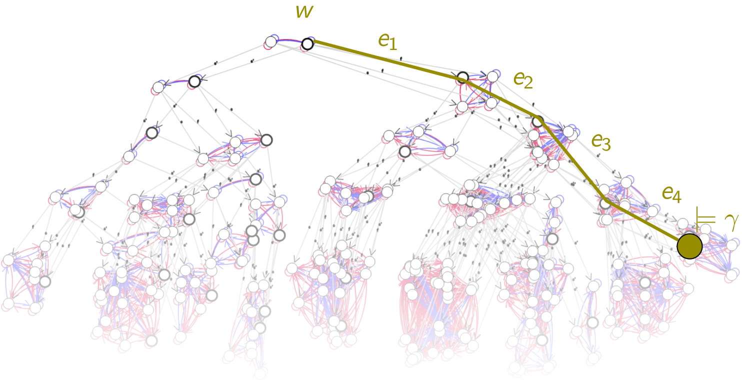
- PDEL Planning
- Automatic structures
- PDEL structures are automatic
- Wrap up

Knowledge-based programs

Conclusion

Epistemic planning: a view on the DEL structure

- Input: an epistemic planning instance $(\mathcal{M}, w, \mathcal{E}, \mathsf{E}_0, \varphi)$ where $(\mathcal{M}, \mathcal{E})$ is a PDEL presentation;
- Output: yes if <u>there exists</u> a history $we_1 \ldots e_\ell$ in $\mathcal{M}\mathcal{E}^*$ such that $we_1 \ldots e_\ell \models \varphi$ and $e_1, \ldots, e_\ell \in \mathsf{E}_0$.



Amounts to query $\mathcal{M}\mathcal{E}^* \models \exists x(\texttt{historyE0}(x) \wedge tr(\gamma)(x))$

Decidability of propositional epistemic planning

- Input: an epistemic planning instance $(\mathcal{M}, w, \mathcal{E}, \mathsf{E}_0, \varphi)$ where $(\mathcal{M}, \mathcal{E})$ is a PDEL presentation;
- Output: yes if there exists a history $we_1 \dots e_\ell$ in $\mathcal{ME}^*$ such that $we_1 \dots e_\ell \models \varphi$ and $e_1, \dots, e_\ell \in \mathsf{E}_0$.
  Ex: $\gamma = K_a \hat{K}_b p$.

Amounts to query $\mathcal{ME}^* \models \exists x(\texttt{historyE0}(x) \wedge tr(\gamma)(x))$.
Sketch of an algorithm:

1. (For predicate `historyE0`) Take $\mathcal{A}_{\texttt{historyE0}}$ that accepts all words $we_1 \dots e_n$ with $e_1, \dots, e_n \in \mathsf{E}_0$;
2. Compute $\mathcal{A}_{tr(\gamma)}$;
   Ex: $tr(\gamma)(x) = \forall y[R_a(x,y) \rightarrow \exists z(R_b(y,z) \wedge p(z))]$.
   $L(\mathcal{A}_{tr(\gamma)}) = \{h \mid \mathcal{ME}^*, [x := h] \models tr(\gamma)(x)\}$.
3. Compute $\mathcal{A}$ s.t. $L(\mathcal{A}) = L(\mathcal{A}_{\texttt{historyE0}}) \cap L(\mathcal{A}_{tr(\gamma)})$
4. Return “yes” if $L(\mathcal{A}) \neq \emptyset$, “no” otherwise.

Propositional epistemic plan synthesis

Since $\nu : L(\mathcal{A}_{\mathcal{H}}) \to \mathcal{H}$ is the identity mapping, *i.e.*, $\nu^{-1}(h) = h$, we can synthesize the set of successful plans for γ.

Theorem

Let $\mathcal{A}$ be the automaton for $\mathtt{historyE0}(x) \wedge tr(\gamma)(x)$. Then $L(\mathcal{A})$ contains exactely all words/histories $we_1 \ldots e_\ell$ s.t.

- $e_1, \ldots, e_\ell \in \mathsf{E}_0$;
- $\mathcal{M}\mathcal{E}^*, we_1 \ldots e_\ell \models \gamma$.

Corollary

Let $(\mathcal{M}, w, \mathcal{E}, \mathsf{E}_0, \varphi)$ be an instance of PDEL planning problem. We can effectively construct an automaton accepting the set of successful plans, i.e., sequences $e_1 \ldots e_\ell \in \mathsf{E}_0^*$ such that

$$\mathcal{M}\mathcal{E}^*, we_1 \ldots e_\ell \models \gamma$$

Complexity of PDEL planning

That is of the query $\mathcal{ME}^* \models \exists x(\mathtt{historyE0}(x) \wedge tr(\gamma)(x))$.

- The complexity is at most d-EXPTIME where d is the alternation depth of $\exists x(\mathtt{historyE0}(x) \wedge tr(\gamma)(x))$.

  E.g. take $\exists x \forall y \exists z R(x, y, z)$, which is $\exists x \neg \exists y \neg \exists z R(x, y, z)$

  To build the automaton for $\neg\psi$, one needs to complement $\mathcal{A}_\psi$. Since $\mathcal{A}_\psi$ may result from projection operations, it may involve a determinization, hence an exponential blow up.

- The lower bound complexity of the PDEL planning is unknown.

Outline

Modeling using Dynamic Epistemic Logic (DEL)

Bounded epistemic planning

Unbounded epistemic planning

Automatic structures for decidability of unbounded epistemic planning when propositional pre/post

Knowledge-based programs

- Motivation
- Syntax of Knowledge-based programs
- Semantics
- Mathematical Properties
- Succinctness
- Conclusion

Conclusion

Outline

Modeling using Dynamic Epistemic Logic (DEL)

Bounded epistemic planning

Unbounded epistemic planning

Automatic structures for decidability of unbounded epistemic planning when propositional pre/post

Knowledge-based programs

- Motivation
- Syntax of Knowledge-based programs
- Semantics
- Mathematical Properties
- Succinctness
- Conclusion

Conclusion

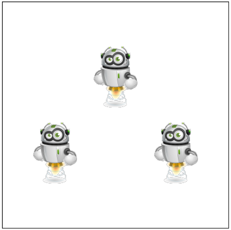
Automation of complex tasks

Building surveillance

Nuclear decommissioning

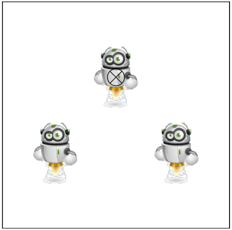
Intelligent farming

Multiple robots



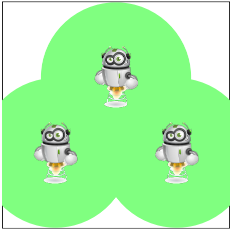
more robust/efficient than

Multiple robots

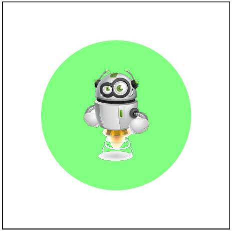


more **robust**/efficient than

Multiple robots



more robust/**efficient** than



Modeling using Dynamic Epistemic Logic (DEL)

Bounded epistemic planning

Unbounded epistemic planning

Automatic structures for decidability of unbounded epistemic planning when propositional pre/post

Knowledge-based programs

Motivation

Syntax of Knowledge-based programs

Semantics

Mathematical Properties

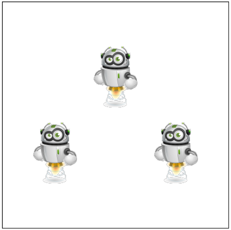
Succinctness

Conclusion

Conclusion

References

Multiple robots

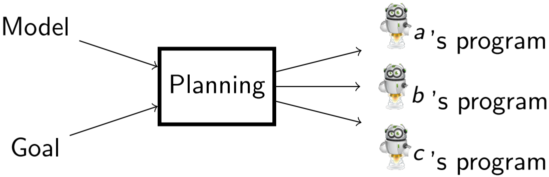


more robust/efficient than

Settings

- Cooperative agents;
- Common goal;
- Imperfect information;

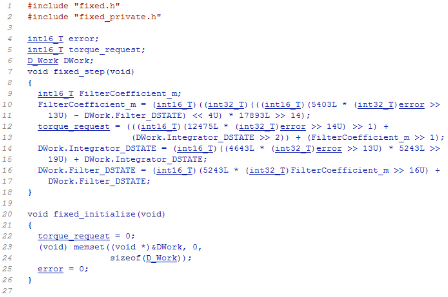
Methodology



Need: understandable system

Motivation

- Legal issues in case of failure
- Interaction with humans



```
#include "fixed.h"
#include "fixed_private.h"

int16_T error;
int16_T torque_request;
D_Work DWork;
void fixed_step(void)
{
  int16_T FilterCoefficient_m;
  FilterCoefficient_m = (int16_T)((int32_T)(((int16_T)(5403L * (int32_T)error >>
    13U) - DWork.Filter_DSTATE) << 4U) * 17893L >> 14);
  torque_request = (((int16_T)(12475L * (int32_T)error >> 14U) >> 1) +
                    (DWork.Integrator_DSTATE >> 2)) + (FilterCoefficient_m >> 1);
  DWork.Integrator_DSTATE = (int16_T)((4643L * (int32_T)error >> 13U) * 5243L >>
    19U) + DWork.Integrator_DSTATE;
  DWork.Filter_DSTATE = (int16_T)(5243L * (int32_T)FilterCoefficient_m >> 16U) +
    DWork.Filter_DSTATE;
}

void fixed_initialize(void)
{
  torque_request = 0;
  (void) memset((void *)&DWork, 0,
                sizeof(D_Work));
  error = 0;
}

```

⊗

Advertising: use of knowledge-based programs

[Fagin et al. 1995]

KBP for agent *a*

```
listenRadio
if $K_a strike$
    toStation
else
    toAirport
```

KBP for agent *b*

```
readNewsPaper
if $K_b strike$
    toStation
else
    toAirport
```

✓

- Understand coordination of agents in QdecPOMDP;
- Succinctness;
- (-) (Un)decidability/complexity issues.

Recent work [Saffidine, Schwarzentruber, and Zanuttini 2018] that extends the mono-agent case in [Lang and Zanuttini 2012], [Lang and Zanuttini 2013].

Outline

Modeling using Dynamic Epistemic Logic (DEL)

Bounded epistemic planning

Unbounded epistemic planning

Automatic structures for decidability of unbounded epistemic planning when propositional pre/post

Knowledge-based programs

- Motivation
- Syntax of Knowledge-based programs
- Semantics
- Mathematical Properties
- Succinctness
- Conclusion

Conclusion

Properties expressed in epistemic logic

Language constructions

room 43 is safe   door 12 is locked   *justobserved*(🔥)   . . .

$\neg\ldots$

$(\ldots \vee \ldots)$

$(\ldots \wedge \ldots)$

$(\ldots \rightarrow \ldots)$

$(K_{\ldots}\ldots)$

Example

$(K_a \text{ door 12 is locked }) \wedge \neg(K_c \text{ door 12 is locked })$

$K_a(K_c \text{ door 12 is locked }) \vee K_a\neg(K_c \text{ door 12 is locked })$

Program constructions

Language constructions

turn left   stay   broadcast temperature

...; ...

if φ **then** ...**else** ...

while φ **do** ...

Example (knowledge-based program for agent *a*)

```
if $K_a($door 12 is locked$\wedge justobserved($🔥$))$ then
    turn left
    broadcast temperature
else
    stay
```

Outline

Modeling using Dynamic Epistemic Logic (DEL)

Bounded epistemic planning

Unbounded epistemic planning

Automatic structures for decidability of unbounded epistemic planning when propositional pre/post

Knowledge-based programs

- Motivation
- Syntax of Knowledge-based programs
- Semantics
  - Models: QdecPOMDP
  - Operational semantics of KBPs
- Mathematical Properties
- Succinctness
- Conclusion

Outline

Modeling using Dynamic Epistemic Logic (DEL)

Bounded epistemic planning

Unbounded epistemic planning

Automatic structures for decidability of unbounded epistemic planning when propositional pre/post

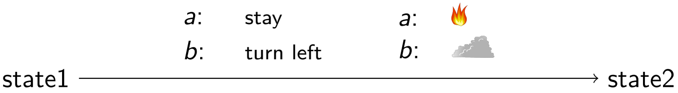
Knowledge-based programs

- Motivation
- Syntax of Knowledge-based programs
- Semantics
  - Models: QdecPOMDP
  - Operational semantics of KBPs
- Mathematical Properties
- Succinctness
- Conclusion

QdecPOMDP

Qualitative decentralized Partially Observable Markov Decision Processes
= Concurrent game structures with observations.

Transitions of the form:



a: stay *a*: 🔥

b: turn left *b*: ☁

state1 ⟶ state2

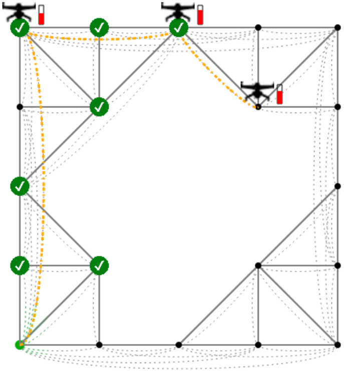
A non-empty set of possible initial states;

A set of goal states.

States

Typically, a state describes:

- positions of agents;
- battery levels;
- etc.



Outline

Modeling using Dynamic Epistemic Logic (DEL)

Bounded epistemic planning

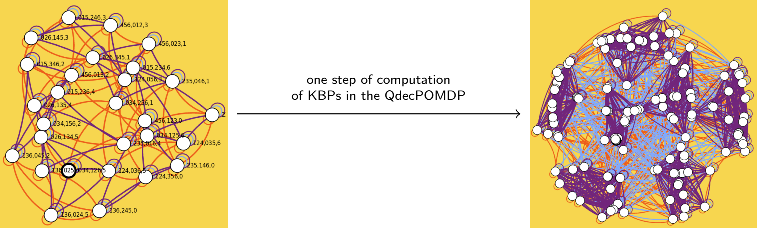
Unbounded epistemic planning

Automatic structures for decidability of unbounded epistemic planning when propositional pre/post

Knowledge-based programs

- Motivation
- Syntax of Knowledge-based programs
- Semantics
  - Models: QdecPOMDP
  - Operational semantics of KBPs
- Mathematical Properties
- Succinctness
- Conclusion

Operational semantics



one step of computation of KBPs in the QdecPOMDP

Epistemic structure

Higher-order knowledge about:

- the current state of the QdecPOMDP;
- the current program counters in KBPs.

Assumptions

Common knowledge of:

- the QdecPOMDP;
- the KBPs;
- synchronicity of the system;
  - tests last 0 unit of time;
  - actions last 1 unit of time.

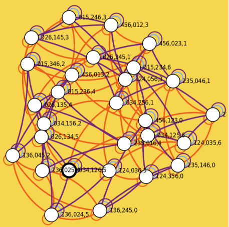
KBP for agent *a*

```
listenRadio
if $K_a strike$
    toStation
else
    toAirport
```

KBP for agent *b*

```
readNewsPaper
if $K_b strike$
    toStation
else
    toAirport
```

Epistemic structures at time T: worlds



Worlds $=$ $\underset{\text{(wait few slides)}}{\text{consistent}}$ histories of the form

$$s^0 \overrightarrow{pc}^0 \boxed{\overrightarrow{obs}^1 s^1 \overrightarrow{pc}^1} \quad \dots \quad \boxed{\overrightarrow{obs}^T s^T \overrightarrow{pc}^T}$$

where

$\overrightarrow{obs}^t$	vector of observations at time t
s^t	state at time t
$\overrightarrow{pc}^t$	vector of program counters at time t

```
● listenRadio
■if $K_a strike$ then
    toStation
else
    toAirport
▲
```

Epistemic structures at time t: indistinguishability relations

Agent a confuses two histories iff she has received the same observations.

$$\begin{array}{c} s^0\overrightarrow{pc}^0\boxed{\overrightarrow{obs}^1 s^1\overrightarrow{pc}^1} \quad \dots \quad \boxed{\overrightarrow{obs}^T s^T\overrightarrow{pc}^T} \\ R_a \\ s'^0\overrightarrow{pc}'^0\boxed{\overrightarrow{obs}'^1 s'^1\overrightarrow{pc}'^1} \dots \boxed{\overrightarrow{obs}'^T s'^T\overrightarrow{pc}'^T} \end{array} \quad \text{iff} \quad \begin{array}{c} \text{for all } t \in \{1, \dots, T\}, \\ \overrightarrow{obs}_a^t = \overrightarrow{obs}_a'^t \end{array}$$

Program counters

Definition (Program counter)

(guard, action just executed, continuation)

```
● listenRadio
■if $K_a strike$ then
    toStation
else
    toAirport
▲
```

($\top$, start , ●)

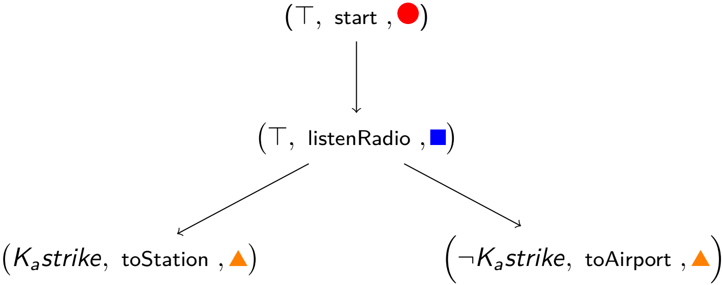
($\top$, listenRadio , ■)

($K_a strike$, toStation , ▲)

($\neg K_a strike$, toAirport , ▲)

Control-flow graph

```
● listenRadio
■ if $K_a strike$ then
    toStation
else
    toAirport
▲
```



(⊤, start, ●)

(⊤, listenRadio, ■)

($K_a strike$, toStation, ▲)

($\neg K_a strike$, toAirport, ▲)

Consistent histories (explained with one agent)

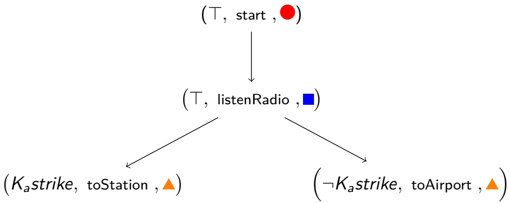
In the QdecPOMDP:

$$s^0 \xrightarrow{\text{listenRadio }, 🔊} s^1$$

$$s^1 \xrightarrow{\text{toStation }, ☀} s^2$$

KBP control-flow graph

```
● listenRadio
■if $K_a strike$ then
    toStation
else
    toAirport
▲
```



$(\top,$ start $, ●)$
$(\top,$ listenRadio $, ■)$
$(K_a strike,$ toStation $, ▲)$
$(\neg K_a strike,$ toAirport $, ▲)$

$$\underbrace{s^0\,(\top,\ \text{start}\ , ●)\ \boxed{🔊 s^1\,(\top,\ \text{listenRadio}\ , ■)}}_{\models K_a strike}\ \boxed{☀ s^2\,(K_a strike,\ \text{toStation}\ , ▲)}$$

Outline

Modeling using Dynamic Epistemic Logic (DEL)

Bounded epistemic planning

Unbounded epistemic planning

Automatic structures for decidability of unbounded epistemic planning when propositional pre/post

Knowledge-based programs

- Motivation
- Syntax of Knowledge-based programs
- Semantics
- Mathematical Properties
  - Verification
  - Execution Problem
- Succinctness
- Conclusion

Outline

Modeling using Dynamic Epistemic Logic (DEL)

Bounded epistemic planning

Unbounded epistemic planning

Automatic structures for decidability of unbounded epistemic planning when propositional pre/post

Knowledge-based programs

- Motivation
- Syntax of Knowledge-based programs
- Semantics
- Mathematical Properties
  - Verification
  - Execution Problem
- Succinctness
- Conclusion

Verification problem

Definition

Input:

- A QdecPOMDP model (given in STRIPS-like symbolic form);
- Knowledge-based programs for each agent;

Output: yes if all executions of the KBPs lead to a goal state.

Theorem

The verification problem for while-free KBPs is PSPACE-complete, and is undecidable for general KBPs.

Outline

Modeling using Dynamic Epistemic Logic (DEL)

Bounded epistemic planning

Unbounded epistemic planning

Automatic structures for decidability of unbounded epistemic planning when propositional pre/post

Knowledge-based programs

- Motivation
- Syntax of Knowledge-based programs
- Semantics
- Mathematical Properties
  - Verification
  - Execution Problem
- Succinctness
- Conclusion

Execution Problem

Input:

- an agent *a*;
- a QdecPOMDP model;
- policies (e.g. KBPs), one for each agent;
- a local view of the history for agent *a*.

Output: the action *act* agent *a* should take.

Execution Problem (decision problem)

Input:

- an agent *a*;
- a QdecPOMDP model;
- policies (e.g. KBPs), one for each agent;
- a local view of the history for agent *a*;
- an action *act*.

Output: yes, if the next action of agent *a* is *act*; no otherwise.

Reactive policy representation

Definition (reactive policy representation)

A class of policy representations is *reactive*
iff its corresponding execution problem is in P.

Example (Tree policies are reactive policy representation)

if justobserved(🔥) **then** turn left **else** stay

Unless P = PSPACE, KBPs are not reactive. Indeed:

Proposition

The execution problem for KBPs is PSPACE-complete.

Outline

Modeling using Dynamic Epistemic Logic (DEL)

Bounded epistemic planning

Unbounded epistemic planning

Automatic structures for decidability of unbounded epistemic planning when propositional pre/post

Knowledge-based programs

- Motivation
- Syntax of Knowledge-based programs
- Semantics
- Mathematical Properties
- Succinctness
- Conclusion

Conclusion

Modal depth

Modal depth = number of nested '$K_{\dots}$' operators.

Formulas	Modal depths
$\mathit{justobserved}$(🔥)	0
$K_a p$	1
$K_a(K_b p)$	2

Succinctness

Theorem ([Lang, Zanuttini, 2012] for $d = 1$; [AAAI2018], for $d > 1$)

Let $d \geq 1$.
There is a poly(n)-size QdecPOMDP family $(\mathcal{M}_{n,d})_{n \in \mathbb{N}}$ for which:

1. *there is a d-modal depth poly(n)-size valid KBP family;*
2. *no $(d-1)$-modal depth valid KBP family;*
3. *assuming NP $\not\subseteq$ P/poly, for any reactive policy representations, no poly(n)-size valid policy family.*

Succinctness

Theorem ([Lang, Zanuttini, 2012] for $d = 1$; [AAAI2018], for $d > 1$)

Let $d \geq 1$.
There is a $poly(n)$-size QdecPOMDP family $(\mathcal{M}_{n,d})_{n \in \mathbb{N}}$ for which:

1. *there is a d-modal depth $poly(n)$-size valid KBP family;*
2. *no $(d-1)$-modal depth valid KBP family;*
3. *assuming NP $\not\subseteq$ P/poly, for any reactive policy representations, no $poly(n)$-size valid policy family.*

Proof idea. $\mathcal{M}_{n,d}$:

- run a $poly(n)$-time protocol revealing a $poly(n)$-size 3-CNF β;
- β satisfiable iff a d-md$_{\text{non } d-1\text{-md expressible}}$ epistemic property holds.

Outline

Modeling using Dynamic Epistemic Logic (DEL)

Bounded epistemic planning

Unbounded epistemic planning

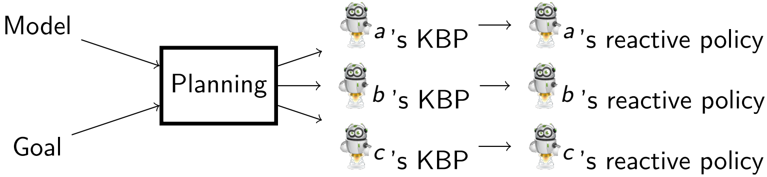
Automatic structures for decidability of unbounded epistemic planning when propositional pre/post

Knowledge-based programs

- Motivation
- Syntax of Knowledge-based programs
- Semantics
- Mathematical Properties
- Succinctness
- Conclusion

Conclusion

Conclusion



Model

Goal

Planning

a's KBP ⟶ *a*'s reactive policy

b's KBP ⟶ *b*'s reactive policy

c's KBP ⟶ *c*'s reactive policy

Higher-order knowledge...

- for get explainable policies (e.g. making cooperation visible)
- for concise programs

Outline

Modeling using Dynamic Epistemic Logic (DEL)

Bounded epistemic planning

Unbounded epistemic planning

Automatic structures for decidability of unbounded epistemic planning when propositional pre/post

Knowledge-based programs

Conclusion

Perspectives

- Design efficient implementation for PSPACE problems;
- Extend algorithms with probabilities;
- Learn policies that are knowledge-based policies;
- Limited beliefs: more efficient and natural behaviors.

Acknowledgment

- Sophie Pinchinat: the slides about automatic structures
- Tristan Charrier, PhD Student, many complexity results in succinct models, model checking, SAT
- Gaëtan Douenot-Tabot: his work in automata theory
- Hans van Ditmarsch, Valentin Goranko, Andreas Herzig, Emiliano Lorini, Thomas Bolander, Abdallah Saffidine, Bruno Zanuttini, etc.
- The School of Art LISAA, Rennes, for the design of Hintikka's world
- Anass Lakhar, Eva Soulier: students contributing to Hintikka's world

Guillaume Aucher and Thomas Bolander. "Undecidability in Epistemic Planning". In: *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013*. 2013. URL: http://www.aaai.org/ocs/index.php/IJCAI/IJCAI13/paper/view/6903.

C.E. Alchourrón, P. Gärdenfors, and D. Makinson. "On The Logic of Theory Change: Partial Meet Functions for Contraction and Revision". In: *Journal of Symbolic Logic* 50.2 (1985), pp. 510–530.

Guillaume Aucher, Bastien Maubert, and Sophie Pinchinat. "Automata Techniques for Epistemic Protocol Synthesis". In: *Proceedings 2nd International Workshop on Strategic Reasoning, Grenoble, France, April 5-6, 2014*. 2014, pp. 97–103. DOI: 10.4204/EPTCS.146.13.

Guillaume Aucher and François Schwarzentruber. "On the Complexity of Dynamic Epistemic Logic". In: *Proceedings of the 14th Conference on Theoretical Aspects of Rationality and Knowledge (TARK 2013), Chennai, India, January 7-9, 2013*. 2013. URL: http://www.tark.org/proceedings/tark%5C_jan7%5C_13/p19-aucher.pdf.

Thomas Bolander and Mikkel Birkegaard Andersen. "Epistemic planning for single and multi-agent systems". In: *Journal of Applied Non-Classical Logics* 21.1 (2011), pp. 9–34. DOI: 10.3166/jancl.21.9-34. URL: http://dx.doi.org/10.3166/jancl.21.9-34.

Johan van Benthem et al. "Symbolic Model Checking for Dynamic Epistemic Logic". In: *Logic, Rationality, and Interaction - 5th International Workshop, LORI 2015 Taipei, Taiwan, October 28-31, 2015, Proceedings*. 2015, pp. 366–378. DOI: 10.1007/978-3-662-48561-3_30. URL: http://dx.doi.org/10.1007/978-3-662-48561-3_30.

Johan van Benthem et al. “Symbolic model checking for Dynamic Epistemic Logic - S5 and beyond”. In: *J. Log. Comput.* 28.2 (2018), pp. 367–402. DOI: 10.1093/logcom/exx038. URL: https://doi.org/10.1093/logcom/exx038.

Achim Blumensath and Erich Grädel. “Automatic structures”. In: *Logic in Computer Science, 2000. Proceedings. 15th Annual IEEE Symposium on*. IEEE. 2000, pp. 51–62.

Thomas Bolander, Martin Holm Jensen, and François Schwarzentruber. “Complexity Results in Epistemic Planning”. In: *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*. 2015, pp. 2791–2797. URL: http://ijcai.org/papers15/Abstracts/IJCAI15-395.html.

Thomas Bolander, Martin Holm Jensen, and François Schwarzentruber. “Complexity Results in Epistemic Planning”. In: *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*. 2015, pp. 2791–2797. URL: `http://ijcai.org/Abstract/15/395`.

Alexandru Baltag, Lawrence S Moss, and Slawomir Solecki. “The logic of public announcements, common knowledge, and private suspicions”. In: *Proceedings of the 7th conference on Theoretical aspects of rationality and knowledge*. Morgan Kaufmann Publishers Inc. 1998, pp. 43–56.

Ronen I. Brafman, Guy Shani, and Shlomo Zilberstein. “Qualitative Planning under Partial Observability in Multi-Agent Domains”. In: *Proc. 27th AAAI Conference on Artificial Intelligence (AAAI 2013)*. AAAI Press, 2013, pp. 130–137.

Tristan Charrier, Bastien Maubert, and François Schwarzentruber. "On the Impact of Modal Depth in Epistemic Planning". In: *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, USA, July 12-15, 2016*. 2016.

Sébastien Lê Cong, Sophie Pinchinat, and François Schwarzentruber. "Small Undecidable Problems in Epistemic Planning". In: *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*. 2018, pp. 4780–4786. DOI: `10.24963/ijcai.2018/664`. URL: `https://doi.org/10.24963/ijcai.2018/664`.

Tristan Charrier and François Schwarzentruber. "A Succinct Language for Dynamic Epistemic Logic". In: *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems, AAMAS 2017, São Paulo, Brazil, May 8-12, 2017*. 2017, pp. 123–131. URL: http://dl.acm.org/citation.cfm?id=3091148.

Tristan Charrier and François Schwarzentruber. "Complexity of Dynamic Epistemic Logic with Common Knowledge". In: *Advances in Modal Logic 7, papers from the 12th conference on "Advances in Modal Logic," held in Bern, Switzerland, 27-31 August 2018*. 2018.

Gaëtan Douéneau-Tabot, Sophie Pinchinat, and François Schwarzentruber. "Chain-Monadic Second Order Logic over Regular Automatic Trees and Epistemic Planning Synthesis". In: *Advances in Modal Logic 7, papers from the 12th conference on "Advances in Modal Logic," held in Bern, Switzerland, 27-31 August 2018*. 2018.

R. Fagin et al. *Reasoning about Knowledge*. MIT Press, 1995.

Joseph Y. Halpern and Yoram Moses. “A Guide to Completeness and Complexity for Modal Logics of Knowledge and Belief”. In: *Artif. Intell.* 54.2 (1992), pp. 319–379. DOI: 10.1016/0004-3702(92)90049-4. URL: http://dx.doi.org/10.1016/0004-3702(92)90049-4.

Wiebe van der Hoek and Michael Wooldridge. “Cooperation, Knowledge, and Time: Alternating-time Temporal Epistemic Logic and its Applications”. In: *Studia Logica* 75.1 (2003), pp. 125–157. DOI: 10.1023/A:1026185103185. URL: https://doi.org/10.1023/A:1026185103185.

Barteld P. Kooi. “Probabilistic Dynamic Epistemic Logic”. In: *Journal of Logic, Language and Information* 12.4 (2003), pp. 381–408. DOI: 10.1023/A:1025050800836. URL: https://doi.org/10.1023/A:1025050800836.

Richard E. Ladner. “The Computational Complexity of Provability in Systems of Modal Propositional Logic”. In: *SIAM J. Comput.* 6.3 (1977), pp. 467–480. DOI: 10.1137/0206033. URL: https://doi.org/10.1137/0206033.

Emiliano Lorini. “In Praise of Belief Bases: Doing Epistemic Logic Without Possible Worlds”. In: *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, Louisiana, USA, February 2-7, 2018*. 2018. URL: https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16867.

Jérôme Lang and Bruno Zanuttini. "Knowledge-Based Programs as Plans - The Complexity of Plan Verification". In: *ECAI 2012 - 20th European Conference on Artificial Intelligence. Including Prestigious Applications of Artificial Intelligence (PAIS-2012) System Demonstrations Track, Montpellier, France, August 27-31 , 2012*. 2012, pp. 504–509. DOI: 10.3233/978-1-61499-098-7-504. URL: http://dx.doi.org/10.3233/978-1-61499-098-7-504.

Jérôme Lang and Bruno Zanuttini. "Knowledge-Based Programs as Plans: Succinctness and the Complexity of Plan Existence". In: *Proceedings of the 14th Conference on Theoretical Aspects of Rationality and Knowledge (TARK 2013), Chennai, India, January 7-9, 2013*. 2013. URL: http://www.tark.org/proceedings/tark_jan7_13/p138-lang.pdf.

Iris van de Pol, Iris van Rooij, and Jakub Szymanik. “Parameterized Complexity Results for a Model of Theory of Mind Based on Dynamic Epistemic Logic”. In: *Proceedings Fifteenth Conference on Theoretical Aspects of Rationality and Knowledge, TARK 2015, Carnegie Mellon University, Pittsburgh, USA, June 4-6, 2015*. 2015, pp. 246–263. DOI: 10.4204/EPTCS.215.18. URL: https://doi.org/10.4204/EPTCS.215.18.

Philippe Schnoebelen. “The Complexity of Temporal Logic Model Checking”. In: *Advances in Modal Logic 4, papers from the fourth conference on "Advances in Modal logic," held in Toulouse (France) in October 2002*. 2002, pp. 393–436.

Philippe Schnoebelen. “The Complexity of Temporal Logic Model Checking.” In: *Advances in modal logic* 4.393-436 (2002), p. 35.

Richard B. Scherl and Hector J. Levesque. “Knowledge, action, and the frame problem”. In: *Artif. Intell.* 144.1-2 (2003), pp. 1–39. DOI: 10.1016/S0004-3702(02)00365-X. URL: https://doi.org/10.1016/S0004-3702(02)00365-X.

Abdallah Saffidine, François Schwarzentruber, and Bruno Zanuttini. “Knowledge-Based Policies for Qualitative Decentralized POMDPs”. In: *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, Louisiana, USA, February 2-7, 2018*. 2018. URL: https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/17029.

Michael Thielscher. “GDL-III: A Proposal to Extend the Game Description Language to General Epistemic Games”. In: *ECAI 2016 - 22nd European Conference on Artificial Intelligence, 29 August-2 September 2016, The Hague, The Netherlands - Including Prestigious Applications of Artificial Intelligence (PAIS 2016)*. 2016, pp. 1630–1631. DOI: 10.3233/978-1-61499-672-9-1630. URL: http://dx.doi.org/10.3233/978-1-61499-672-9-1630.

Hans van Ditmarsch, Wiebe van der Hoek, and Barteld Kooi. *Dynamic Epistemic Logic*. Dordecht: Springer, 2008.

Quan Yu, Ximing Wen, and Yongmei Liu. “Multi-Agent Epistemic Explanatory Diagnosis via Reasoning about Actions”. In: *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013*. 2013, pp. 1183–1190.