Dynamic Logics

Carlos Areces
carlos.areces@gmail.com

Spring Term 2018
Stanford

What we do in this lecture

- Present Epistemic Logics
- Discuss Dynamics. Dynamic Epistemic Logics
  - Public Announcements
  - Action Model Logics
- Relation Change Logics

Epistemic Logic

Epistemic logic is Modal Logic where we give some particular interpretation to its formulas.

Epistemic Logic

Epistemic logic is Modal Logic where we give some particular interpretation to its formulas.
For instance:

$\Box p$	I know that p.

Epistemic Logic

Epistemic logic is Modal Logic where we give some particular interpretation to its formulas.
For instance:

$\Box p$	I know that p.
$\neg\Box_{Bill} p$	Bill does not know that p.

Epistemic Logic

Epistemic logic is Modal Logic where we give some particular interpretation to its formulas.
For instance:

$\Box p$	I know that p.
$\neg\Box_{Bill}\, p$	Bill does not know that p.
$\Box_{Bill}\, p \vee \Box_{Bill} \neg p$	Bill knows whether p.

Epistemic Logic

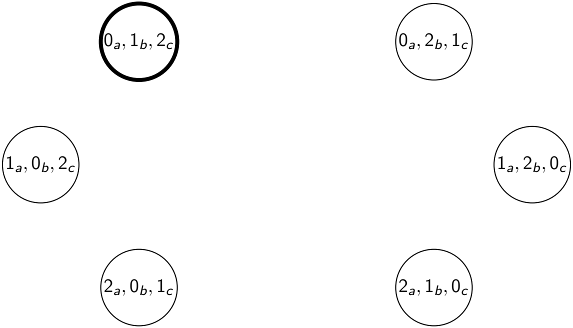
Epistemic logic is Modal Logic where we give some particular interpretation to its formulas.
For instance:

$\Box p$	I know that p.
$\neg\Box_{Bill} p$	Bill does not know that p.
$\Box_{Bill} p \vee \Box_{Bill}\neg p$	Bill knows whether p.
$\Box_{Bill}\Box_{Anne}\neg\Box_{Cath} p$	Bill knows Alice knows that Cath does not know that p.

A card game scenario: description of the problem

There are three agents: Anne (a), Bill (b) and Cath (c); each of them holds one of three possible cards: 0, 1 or 2. Propositional symbols such as 0_i, 1_i and 2_i state that the agent i is holding the card 0, 1 or 2, respectively.

A card game scenario: the modeling

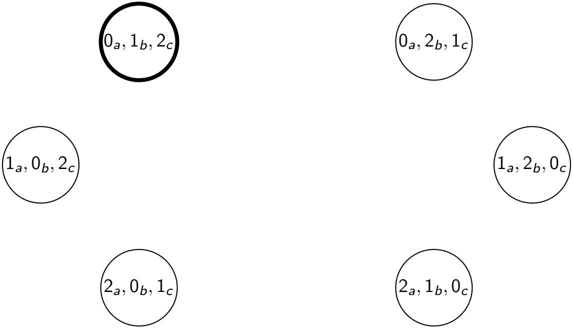


A card game scenario: the modeling

Each agent is uncertain about any other agents card.

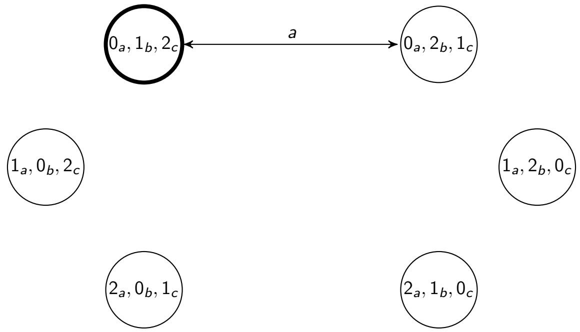
A card game scenario: the modeling

Each agent is uncertain about any other agents card.



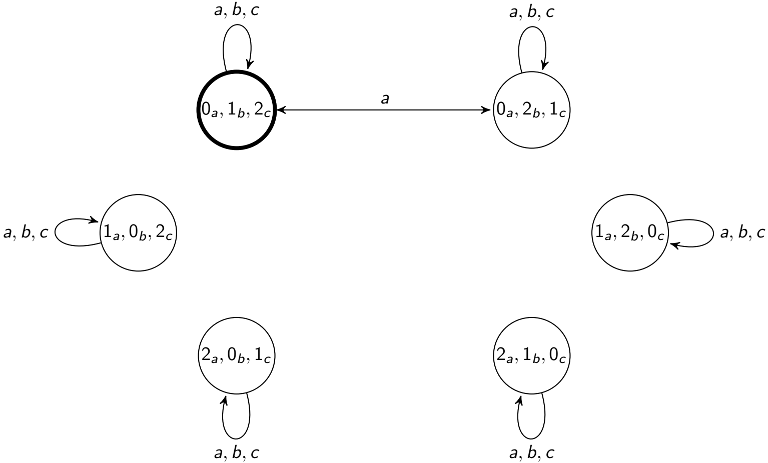
A card game scenario: the modeling

Each agent is uncertain about any other agents card.



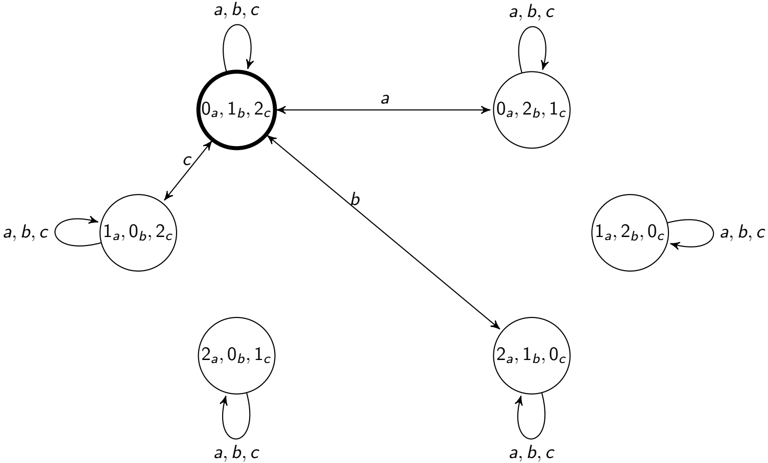
A card game scenario: the modeling

Each agent is uncertain about any other agents card.



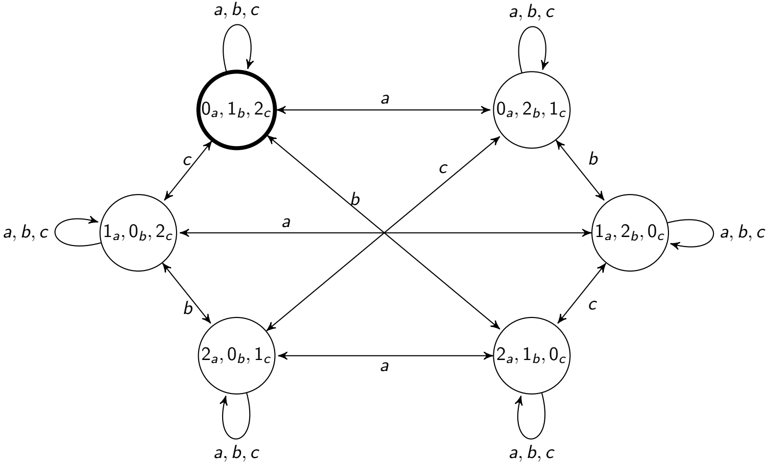
A card game scenario: the modeling

Each agent is uncertain about any other agents card.



A card game scenario: the modeling

Each agent is uncertain about any other agents card.



Epistemic Modeling - summing up

Given a description of a situation, the modeler determines:

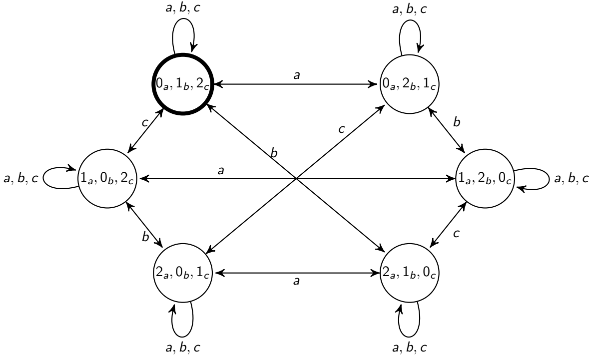
- The set of relevant propositions.
- The set of relevant agents.
- The set of states.
- For each agent, an indistinguishability relation over the states (am equivalence relation).

Making announcements...

Let us suppose that Anne tells the other agents she does not have card 2. It can be modeled as "it is publicly announced that Anne does not have card 2".

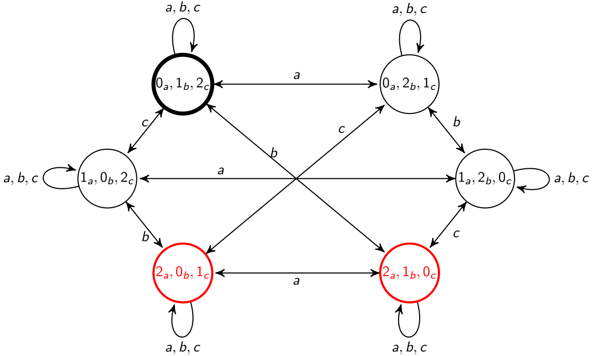
Making announcements...

Let us suppose that Anne tells the other agents she does not have card 2. It can be modeled as "it is publicly announced that Anne does not have card 2".



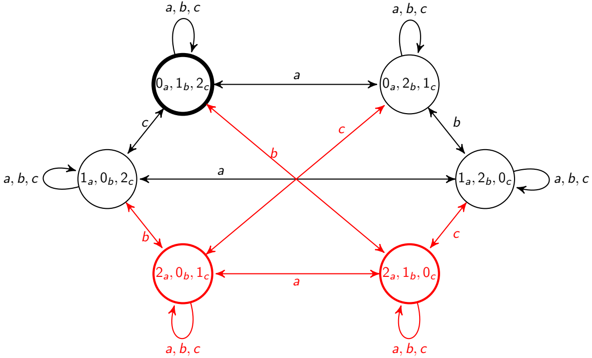
Making announcements...

Let us suppose that Anne tells the other agents she does not have card 2. It can be modeled as “it is publicly announced that Anne does not have card 2”.



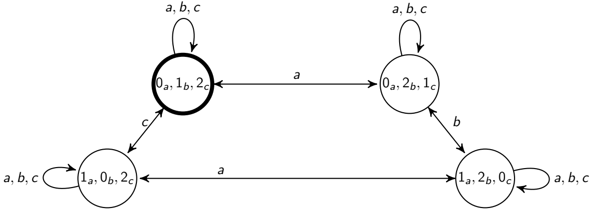
Making announcements...

Let us suppose that Anne tells the other agents she does not have card 2. It can be modeled as "it is publicly announced that Anne does not have card 2".



Making announcements...

Let us suppose that Anne tells the other agents she does not have card 2. It can be modeled as "it is publicly announced that Anne does not have card 2".



Public Announcements

Is a very simple extension of epistemic logic, with the public announcement operator

$$[!\psi]\varphi$$

It says "after that ψ is announced, φ holds".
This is semantically modeled with a restriction on the original model:

$$\mathcal{M}, w \models [!\psi]\varphi \text{ iff } \mathcal{M}, w \models \psi \text{ implies } \mathcal{M}_{|\psi}, w \models \varphi.$$

Public Announcements

Is a very simple extension of epistemic logic, with the public announcement operator

$$[!\psi]\varphi$$

It says "after that ψ is announced, φ holds".
This is semantically modeled with a restriction on the original model:

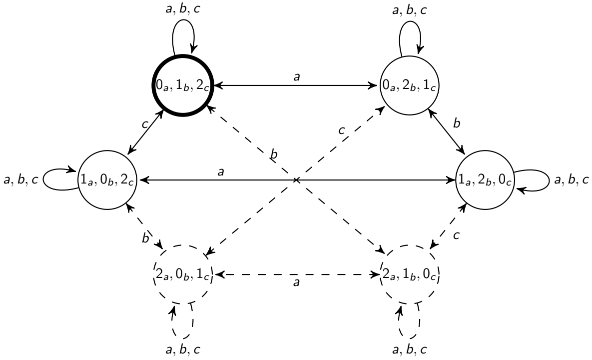
$$\mathcal{M}, w \models [!\psi]\varphi \;\text{ iff }\; \mathcal{M}, w \models \psi \text{ implies } \mathcal{M}_{|\psi}, w \models \varphi.$$

$\mathcal{M}_{|\psi} = \langle W_{|\psi}, R_{|\psi}, V_{|\psi} \rangle$ where

$$W_{|\psi} = \{w \in W \mid \mathcal{M}, w \models \psi\} \qquad \begin{array}{l} R_{|\psi} = R \cap (W_{|\psi} \times W_{|\psi}) \\ V_{|\psi} = V \cap W_{|\psi} \end{array}$$

Using the Logic to Make announcements

Before the announcement, it's not true that Bill knows Anne has card 0. After the announcement of Anne has not card 2, Bill knows Anne has card 0: $\neg\Box_b 0_a \wedge [!\neg 2_a]\Box_b 0_a$.



Working with the new operators

The operator of public announcement is quite different of other modal operators we already knew: we evaluate a piece of the formula after the model has been transformed.

Working with the new operators

The operator of public announcement is quite different of other modal operators we already knew: we evaluate a piece of the formula after the model has been transformed.

$$\mathcal{M}, w \models [!\psi]\varphi \text{ iff } \mathcal{M}, w \models \psi \text{ implies } \mathcal{M}_{|\psi}, w \models \varphi.$$

Working with the new operators

The operator of public announcement is quite different of other modal operators we already knew: we evaluate a piece of the formula after the model has been transformed.

$$\mathcal{M}, w \models [!\psi]\varphi \text{ iff } \mathcal{M}, w \models \psi \text{ implies } \mathcal{M}_{|\psi}, w \models \varphi.$$

However, all the well-known tools to work with modal logics can be used to work with PAL.

Working with the new operators

The operator of public announcement is quite different of other modal operators we already knew: we evaluate a piece of the formula after the model has been transformed.

$$\mathcal{M}, w \models [!\psi]\varphi \text{ iff } \mathcal{M}, w \models \psi \text{ implies } \mathcal{M}_{|\psi}, w \models \varphi.$$

However, all the well-known tools to work with modal logics can be used to work with PAL.

For instance, we would like to know how much can we say with this logic, i.e., we are interested in its expressive power.

Comparing Languages

We say that a language $\mathcal{L}'$ is more or equally expressive than $\mathcal{L}$ (we write $\mathcal{L} \leq \mathcal{L}'$) iff there is a translation $\mathsf{Tr} : \mathcal{L} \to \mathcal{L}'$ such that, for all $\mathcal{L}$-formula φ we have

$$\mathcal{M}, w \models \varphi \;\text{ iff }\; \mathcal{M}, w \models \mathsf{Tr}(\varphi).$$

Comparing Languages

We say that a language $\mathcal{L}'$ is more or equally expressive than $\mathcal{L}$ (we write $\mathcal{L} \leq \mathcal{L}'$) iff there is a translation $\mathsf{Tr} : \mathcal{L} \to \mathcal{L}'$ such that, for all $\mathcal{L}$-formula φ we have

$$\mathcal{M}, w \models \varphi \ \text{ iff } \ \mathcal{M}, w \models \mathsf{Tr}(\varphi).$$

$\mathcal{L} = \mathcal{L}'$ (both have the same expressive power) iff $\mathcal{L} \leq \mathcal{L}'$ and $\mathcal{L}' \leq \mathcal{L}$.

But.. what can we say with public announcements?

The answer can be a bit surprising: We can say exactly the same as with BML

In fact, we have reduction axioms which translate announcements in basic formulas:

1. $[!\psi]p \leftrightarrow (\psi \rightarrow p)$
2. $[!\psi]\neg\varphi \leftrightarrow (\psi \rightarrow \neg[!\psi]\varphi)$
3. $[!\psi](\varphi \wedge \chi) \leftrightarrow ([!\psi]\varphi \wedge [!\psi]\varphi)$
4. $[!\psi]\Diamond_a\varphi \leftrightarrow (\psi \rightarrow \Diamond_a[!\psi]\varphi)$
5. $[!\psi][!\chi]\varphi \leftrightarrow [!\psi \wedge [!\psi]\chi]\varphi$

But.. what can we say with public announcements?

The answer can be a bit surprising: We can say exactly the same as with BML

In fact, we have reduction axioms which translate announcements in basic formulas:

1. $[!\psi]p \leftrightarrow (\psi \rightarrow p)$
2. $[!\psi]\neg\varphi \leftrightarrow (\psi \rightarrow \neg[!\psi]\varphi)$
3. $[!\psi](\varphi \wedge \chi) \leftrightarrow ([!\psi]\varphi \wedge [!\psi]\varphi)$
4. $[!\psi]\Diamond_a\varphi \leftrightarrow (\psi \rightarrow \Diamond_a[!\psi]\varphi)$
5. $[!\psi][!\chi]\varphi \leftrightarrow [!\psi \wedge [!\psi]\chi]\varphi$

But the formula can be considerably larger. (How much?)

The Translation to BML

Theorem

Public Announcement Logic and the Basic Modal Logic are equally expressive.

Proof.

The successive application of the reduction axioms leads us to a formula without announcements, i.e., a formula in BML. We need to prove valid each of the five equivalences mentioned. $\square$

The Proof

1. Let us suppose that $\mathcal{M}, w \models [!\psi]p$. Then, by definition of $[!\psi]$ we have $\mathcal{M}, w \models \psi$ implies $\mathcal{M}_{|\psi}, w \models p$. But $\mathcal{M}_{|\psi}, w \models p$ iff $\mathcal{M}, w \models p$, then we have $\mathcal{M}, w \models \psi \to p$.

4. Suppose $\mathcal{M}, w \models [!\psi]\Diamond_a\varphi$. We have by definition of $[!\psi]$ that $\mathcal{M}, w \models \psi$ implies $\mathcal{M}_{|\psi}, w \models \Diamond_a\varphi$. By definition of $\Diamond_a$, we have $\mathcal{M}, w \models \psi$ implies there is a $v \in W_{|\psi}$ s.t. $(w, v) \in R_{|\psi}$ and $\mathcal{M}_{|\psi}, v \models \varphi$. By definition of $[!\psi]$, $\mathcal{M}, w \models \psi$ implies there is a $v \in W_{|\psi}$ s.t. $(w, v) \in R_{|\psi}$ and $\mathcal{M}, v \models [!\psi]\varphi$, and by $\Diamond_a$, we have $\mathcal{M}, w \models \psi$ implies $\mathcal{M}, w \models \Diamond_a[!\psi]\varphi$. Then, $\mathcal{M}, w \models \psi \to \Diamond_a[!\psi]\varphi$.

What about the size?

We said: Any formula with public announcements can be translated into one without public announcements.

What about the size?

We said: Any formula with public announcements can be translated into one without public announcements.

However, this is not for free: the translation introduces a blow up in the size of the formula.

What about the size?

We said: Any formula with public announcements can be translated into one without public announcements.

However, this is not for free: the translation introduces a blow up in the size of the formula.

Let us analyze for instance, the following axiom:

$$[!\psi][!\chi]\varphi \leftrightarrow [!\psi \wedge [!\psi]\chi]\varphi$$

Succinctness and Complexity

Observation

There are formulas in Public Announcement Logic which are exponentially more succinct than their correspondent translation.

Succinctness and Complexity

Observation
There are formulas in Public Announcement Logic which are exponentially more succinct than their correspondent translation.

Theorem
The satisfiability problem of Public Announcement Logic is $\textsc{PSpace}$*-complete.*

Succinctness and Complexity

Observation

There are formulas in Public Announcement Logic which are exponentially more succinct than their correspondent translation.

Theorem

The satisfiability problem of Public Announcement Logic is PSPACE*-complete.*

For PAL, succinctness does not affect complexity.

Static vs. Dynamic

- We moved to the dynamic setting, but using the same tools as in static modal logics.

Static vs. Dynamic

- We moved to the dynamic setting, but using the same tools as in static modal logics.
- But we saw that a dynamic logic (PAL), is equally expressive as a static logic (BML).

Static vs. Dynamic

- We moved to the dynamic setting, but using the same tools as in static modal logics.
- But we saw that a dynamic logic (PAL), is equally expressive as a static logic (BML).
- Disappointing?

Static vs. Dynamic

- We moved to the dynamic setting, but using the same tools as in static modal logics.
- But we saw that a dynamic logic (PAL), is equally expressive as a static logic (BML).
- Disappointing? Sometimes dynamic operators can be represented by a static operator, but with consequences.

Static vs. Dynamic

- We moved to the dynamic setting, but using the same tools as in static modal logics.
- But we saw that a dynamic logic (PAL), is equally expressive as a static logic (BML).
- Disappointing? Sometimes dynamic operators can be represented by a static operator, but with consequences.
- We mentioned a potential blow up in the size of the formula. Sometimes it can be worse.

Relevant Bibliography

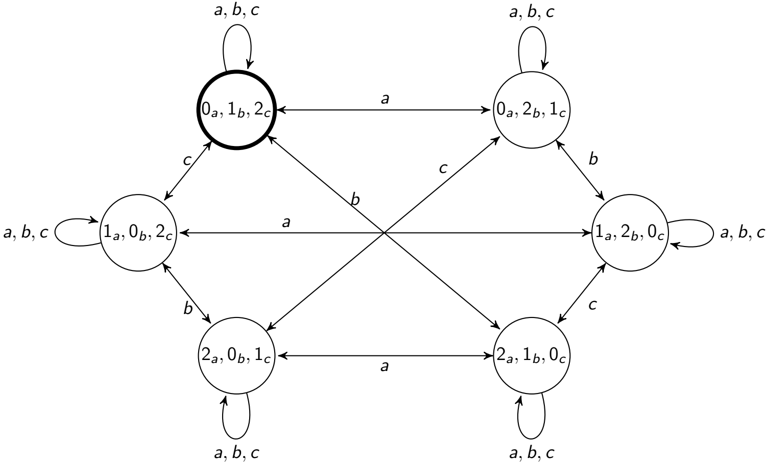
Jan Plaza was one of the pioneers in the investigation of logics to describe changes of knowledge as a result of communication among agents. In his article in 1989 "Logics of Public Communications" he introduced public announcements (as a binary operator $\varphi + \psi$, back then).

`http://faculty.plattsburgh.edu/jan.plaza/`

Plaza, J.; (2007). *Logics of public communications.* Synthese 158(2): 165–179.

A card game scenario: the modeling

Each agent is uncertain about any other agents card.

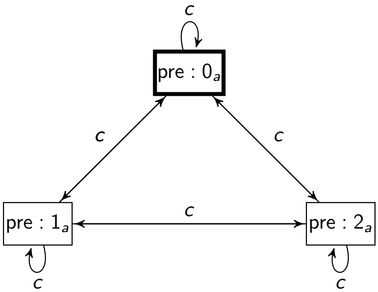


The action of showing a card

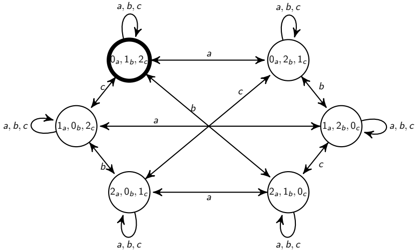
Anne shows card 0 to Bill (i.e., Cath is still uncertain about Anne's card)

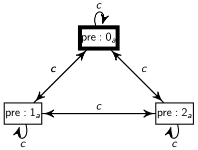
The action of showing a card

Anne shows card 0 to Bill (i.e., Cath is still uncertain about Anne's card)

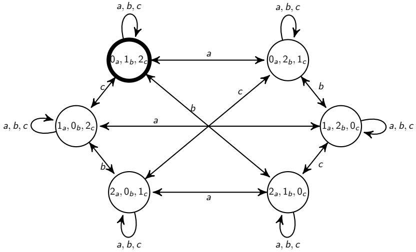


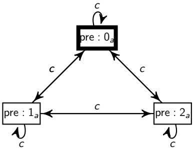
Anne shows card 0 to Bill



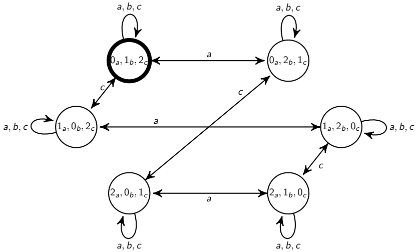


Anne shows card 0 to Bill



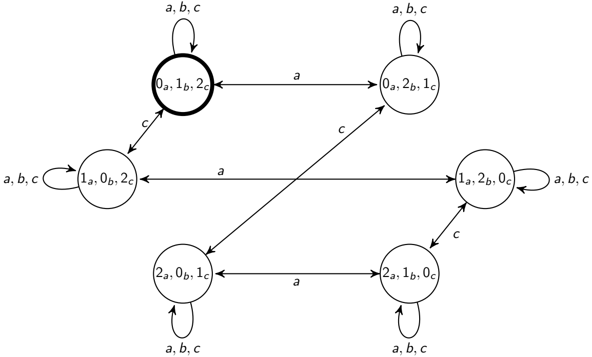


$\Longrightarrow$



Anne shows card 0 to Bill

Now, the formula $\Box_b 0_a \wedge \neg\Box_c 0_a$ holds at the evaluation point.



Action Models

An action model $\mathcal{E}$ is a structure $\langle E, \rightarrow, \text{pre} \rangle$ where:

- E is a set of actions.
- $\rightarrow_a$ is an equivalence relation over E.
- $\text{pre} : E \rightarrow \mathcal{L}$ is a function which assigns a pre-condition (a formula in $\mathcal{L}$) to each $e \in E$.

Action Model Logic - Syntax

The set FORM of formulas of action model logic over PROP and AGT is defined as:

$$\text{FORM} ::= \top \mid p \mid \neg\varphi \mid \varphi \wedge \varphi' \mid \Box_a\varphi \mid [\mathcal{E}, e]\varphi,$$

where $p \in$ PROP, $a \in$ AGT, $\varphi, \varphi' \in$ FORM, and $\mathcal{E}, e$ is a pointed action model such that pre(e) is a formula built in a previous step of the induction.

Action Model Logic - Semantics

Action Model Logic (a.k.a. DEL in the community) is an extension of basic epistemic logic with an operator $[\mathcal{E}, e]$, which has an action model as argument. The semantics is

$$\mathcal{M}, w \models [\mathcal{E}, e]\varphi \text{ iff } \mathcal{M}, w \models \mathrm{pre}(e) \text{ implies } (\mathcal{M} \otimes \mathcal{E})(w, e) \models \varphi,$$

Action Model Logic - Semantics

Action Model Logic (a.k.a. DEL in the community) is an extension of basic epistemic logic with an operator $[\mathcal{E}, e]$, which has an action model as argument. The semantics is

$$\mathcal{M}, w \models [\mathcal{E}, e]\varphi \ \text{ iff } \ \mathcal{M}, w \models \mathsf{pre}(e) \text{ implies } (\mathcal{M} \otimes \mathcal{E})(w, e) \models \varphi,$$

$(\mathcal{M} \otimes \mathcal{E})(w, e)$ is the result of executing $\mathcal{E}, e$ in $\mathcal{M}, w$.

Action Model Logic - Semantics

Action Model Logic (a.k.a. DEL in the community) is an extension of basic epistemic logic with an operator $[\mathcal{E}, e]$, which has an action model as argument. The semantics is

$$\mathcal{M}, w \models [\mathcal{E}, e]\varphi \ \text{ iff } \ \mathcal{M}, w \models \mathrm{pre}(e) \text{ implies } (\mathcal{M} \otimes \mathcal{E})(w, e) \models \varphi,$$

$(\mathcal{M} \otimes \mathcal{E})(w, e)$ is the result of executing $\mathcal{E}, e$ in $\mathcal{M}, w$.

Remember:

$$\mathcal{M}, w \models [!\psi]\varphi \ \text{ iff } \ \mathcal{M}, w \models \psi \text{ implies } \mathcal{M}_{|\psi}, w \models \varphi.$$

Action Model Logic - Semantics

Action Model Logic (a.k.a. DEL in the community) is an extension of basic epistemic logic with an operator $[\mathcal{E}, e]$, which has an action model as argument. The semantics is

$$\mathcal{M}, w \models [\mathcal{E}, e]\varphi \;\text{ iff }\; \mathcal{M}, w \models \mathsf{pre}(e) \text{ implies } (\mathcal{M} \otimes \mathcal{E})(w, e) \models \varphi,$$

$(\mathcal{M} \otimes \mathcal{E})(w, e)$ is the result of executing $\mathcal{E}, e$ in $\mathcal{M}, w$.

Remember:

$$\mathcal{M}, w \models [!\psi]\varphi \;\text{ iff }\; \mathcal{M}, w \models \psi \text{ implies } \mathcal{M}_{|\psi}, w \models \varphi.$$

Again we are talking about truthful announcements.

Product Updates

Given a relational model $\mathcal{M} = \langle W, R, V \rangle$ and an action model $\mathcal{E} = \langle E, \rightarrow, \text{pre} \rangle$, $w \in W$ and $e \in E$, the product $(\mathcal{M} \otimes \mathcal{E})(w, e)$ is a new relational model $\langle W', R', V' \rangle$ where:

- $W' = \{(w, e) \mid w \in W, e \in E, \text{ and } \mathcal{M}, w \models \text{pre}(e)\}$.

Product Updates

Given a relational model $\mathcal{M} = \langle W, R, V \rangle$ and an action model $\mathcal{E} = \langle E, \rightarrow, \mathrm{pre} \rangle$, $w \in W$ and $e \in E$, the product $(\mathcal{M} \otimes \mathcal{E})(w, e)$ is a new relational model $\langle W', R', V' \rangle$ where:

- $W' = \{(w, e) \mid w \in W, e \in E, \text{ and } \mathcal{M}, w \models \mathrm{pre}(e)\}$.
- $(w, e) \in V'(p)$ iff $w \in V(p)$.

Product Updates

Given a relational model $\mathcal{M} = \langle W, R, V \rangle$ and an action model $\mathcal{E} = \langle E, \rightarrow, \mathrm{pre} \rangle$, $w \in W$ and $e \in E$, the product $(\mathcal{M} \otimes \mathcal{E})(w, e)$ is a new relational model $\langle W', R', V' \rangle$ where:

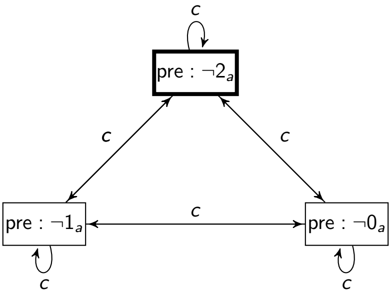
- $W' = \{(w, e) \mid w \in W, e \in E, \text{ and } \mathcal{M}, w \models \mathrm{pre}(e)\}$.
- $(w, e) \in V'(p)$ iff $w \in V(p)$.
- $((w, e), (v, f)) \in R'_a$ iff $(w, v) \in R_a$ and $e \rightarrow_a f$.

The action of whispering a card

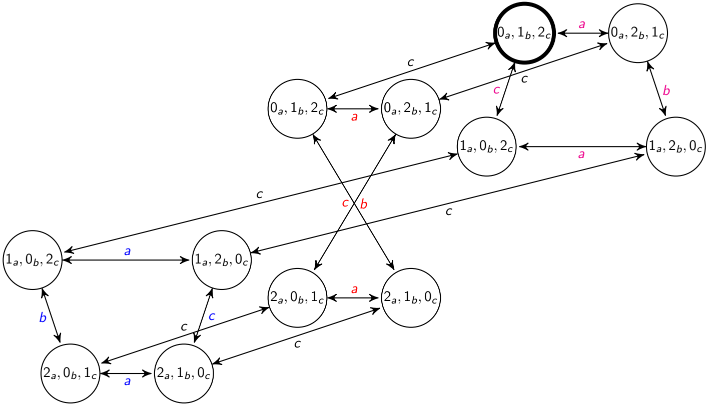
Anne whispers in Bill's ear "I do not have card 2".
Cath notices that Anne reveals she does not have some card, but cannot hear which card.

The action of whispering a card

Anne whispers in Bill's ear "I do not have card 2".
Cath notices that Anne reveals she does not have some card, but cannot hear which card.

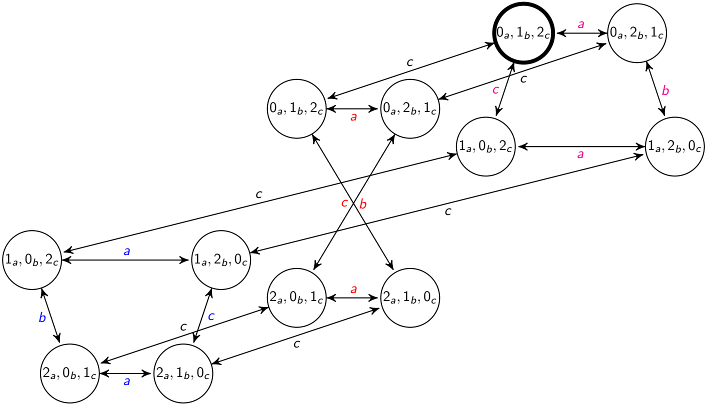


Anne whispers “not 2” to Bill

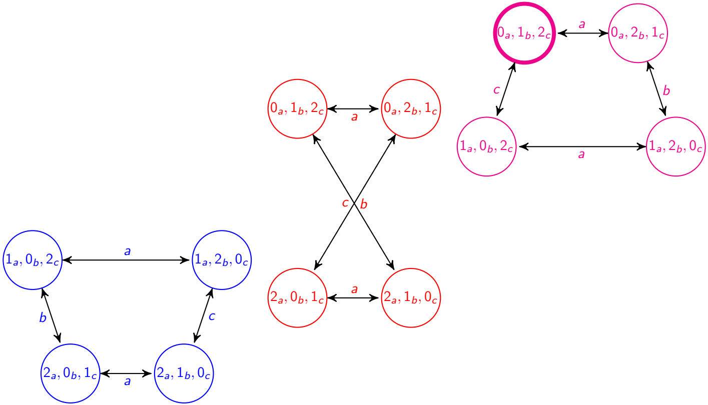


A picture of our epistemic state now

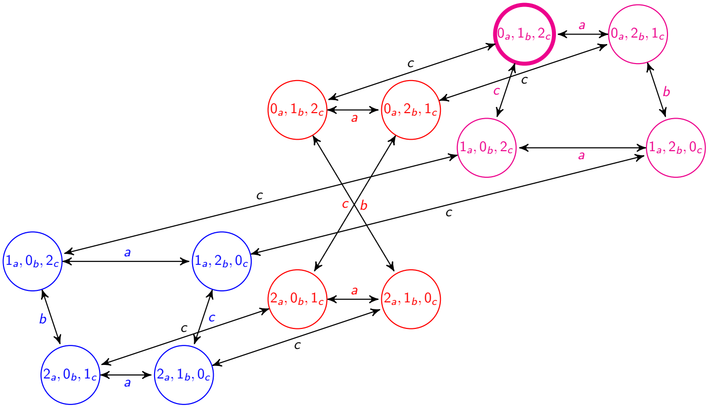
Anne whispers “not 2” to Bill



Anne whispers “not 2” to Bill



Anne whispers “not 2” to Bill



Model and Modal...

We have seen that certain semantics-like objects are included into modalities. It raises the question: Are the syntax and semantics clearly separated??

Model and Modal...

We have seen that certain semantics-like objects are included into modalities. It raises the question: Are the syntax and semantics clearly separated??

Let's explore this formula:

$$[\mathcal{E}, e]\varphi,$$

where $\mathcal{E}, e$ is the action model which represents Anne showing card 0 to Bill.

Model and Modal...

But the formula

$$[\mathcal{E}, e]\varphi$$

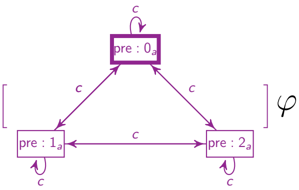
is

Model and Modal...

But the formula

$$[\mathcal{E}, \mathsf{e}]\varphi$$

is

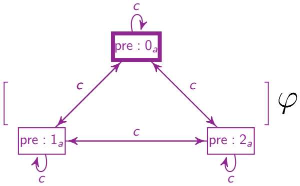


Model and Modal...

But the formula

$$[\mathcal{E}, \mathsf{e}]\varphi$$

is



We have an action model inside the box, and usually models represent semantic information!!

Model and Modal... the answer

Are the syntax and semantics clearly separated??

Model and Modal... the answer

Are the syntax and semantics clearly separated?? **YES.**
Remember the restriction:

> *"$\mathcal{E}, e$ is a pointed action model such that* pre(e) *is a formula built in a previous step of the induction."*

Model and Modal... the answer

Are the syntax and semantics clearly separated?? **YES.**
Remember the restriction:

> *"$\mathcal{E}, e$ is a pointed action model such that* $\mathrm{pre}(e)$ *is a formula built in a previous step of the induction."*

This makes the definition well defined.

Expressive Power

Again we can found a translation from action model logic to the basic modal logic:

1. $[\mathcal{E}, e]p \;\leftrightarrow\; (\mathsf{pre}(e) \rightarrow p)$
2. $[\mathcal{E}, e]\neg\varphi \;\leftrightarrow\; (\mathsf{pre}(e) \rightarrow \neg[\mathcal{E}, e]\varphi)$
3. $[\mathcal{E}, e](\varphi \wedge \psi) \;\leftrightarrow\; [\mathcal{E}, e]\varphi \wedge [\mathcal{E}, e]\psi$
4. $[\mathcal{E}, e]\Diamond_a\varphi \;\leftrightarrow\; (\mathsf{pre}(e) \rightarrow \bigwedge_{e \rightarrow_a f} \Diamond_a[\mathcal{E}, f]\varphi)$
5. $[\mathcal{E}, e][\mathcal{E}', e']\varphi \;\leftrightarrow\; [(\mathcal{E}, e); (\mathcal{E}', e')]\varphi$

Composition of action models

Given two action models $\mathcal{E} = \langle E, \rightarrow, \mathrm{pre}\rangle$ and $\mathcal{E}' = \langle E', \rightarrow', \mathrm{pre}'\rangle$, we define their *composition product* as $\mathcal{E};\mathcal{E}' = \langle E^\times, \rightarrow^\times, \mathrm{pre}^\times\rangle$, where:

$$
\begin{array}{lcl}
E^\times & = & E \times E' \\
\rightarrow_a^\times & = & \{((e,e'),(f,f')) \mid (e,f) \in \rightarrow_a \text{ and } (e',f') \in \rightarrow_a\} \\
\mathrm{pre}^\times(e,e') & = & \mathrm{pre}(e) \wedge [\mathcal{E}, e]\mathrm{pre}'(e')
\end{array}
$$

The Translation to BML

Theorem

Let Tr *be the translation defined as the successive application of the reduction axioms, then we have Action Model Logic and the Basic Modal Logic are equally expressive.*

Proof.

The successive application of the reduction axioms leads us to a formula without action models, i.e., a formula in BML. □

What about the size?

We said that any formula with action models can be translated into a one without action models.

What about the size?

We said that any formula with action models can be translated into a one without action models.

However, this is not for free: the translation introduces a blow up in the size of the formula.

What about the size?

We said that any formula with action models can be translated into a one without action models.

However, this is not for free: the translation introduces a blow up in the size of the formula.

Let us analyze for instance, the following axiom:

$$[\mathcal{E}, e]\Diamond_a\varphi \;\leftrightarrow\; (\mathsf{pre}(e) \rightarrow \bigwedge_{e \rightarrow_a f} \Diamond_a[\mathcal{E}, f]\varphi),$$

i.e., φ appears as many times as successors of e.

Succinctness and Complexity

Theorem

There are formulas in Action Model Logic such that every BML-formula equivalent is exponentially larger than the original one, i.e., cannot be polynomially translated into BML.

Succinctness and Complexity

Theorem

There are formulas in Action Model Logic such that every BML-formula equivalent is exponentially larger than the original one, i.e., cannot be polynomially translated into BML.

Theorem

The satisfiability problem of Action Model Logic is EXPTIME*-complete.*

Succinctness and Complexity

Theorem

There are formulas in Action Model Logic such that every BML-formula equivalent is exponentially larger than the original one, i.e., cannot be polynomially translated into BML.

Theorem

The satisfiability problem of Action Model Logic is EXPTIME*-complete.*

For AML, succinctness affects complexity!

Relevant Bibliography I

Alexandru Baltag, Larry Moss and Slawomir Solecki, pioneers of Dynamic Epistemic Logics.

`https://sites.google.com/site/thealexandrubaltagsite/`

`http://www.indiana.edu/~iulg/moss/`

`http://www.math.uiuc.edu/~ssolecki/`

Baltag A., Moss L., and Solecki S.; (1998) *The Logic of Common Knowledge, Public Announcements, and Private Suspicions*. Proceedings of TARK–VII.

Relevant Bibliography II

Dynamic Epistemic Logic, H. van Ditmarsch, B. Kooi & W. van der Hoek, is a fundamental book in the field.

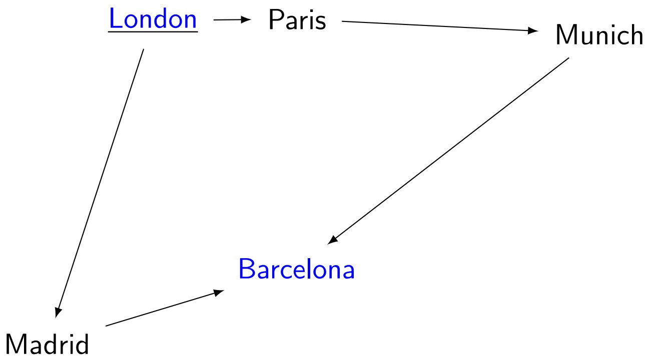
http://sites.google.com/site/hansvanditmarsch/

http://www.philos.rug.nl/~barteld/

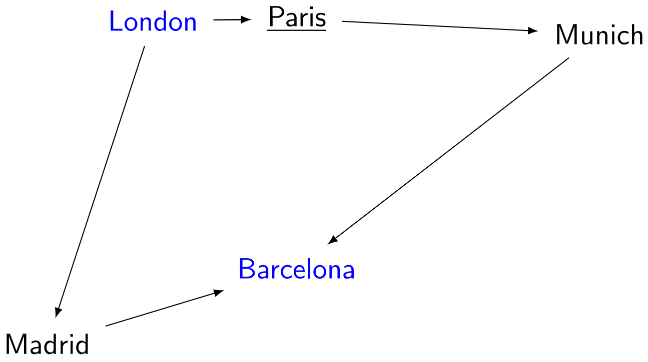
http://cgi.csc.liv.ac.uk/~wiebe/

van Ditmarsch, H., van der Hoek, W., and Kooi, B. (2007) *Dynamic Epistemic Logic*.

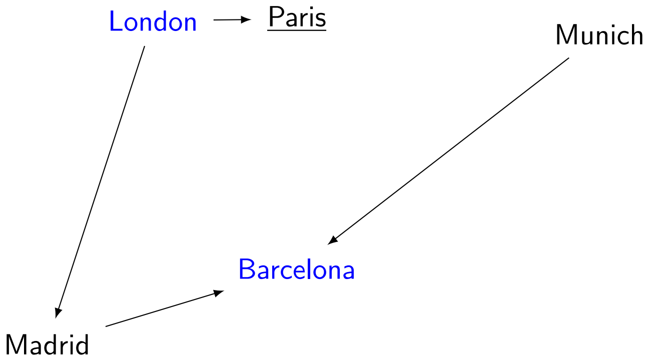
Changing Access



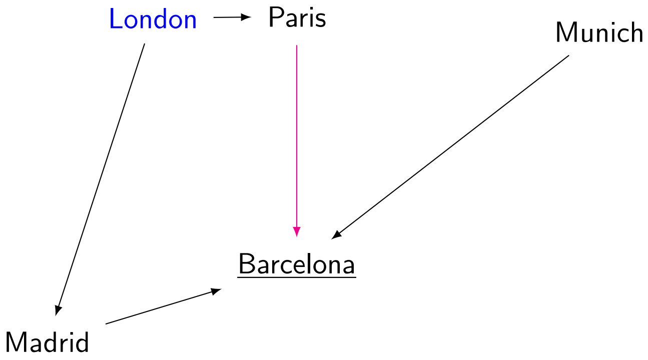
Changing Access



Changing Access



Changing Access



Edge deletion

We can define a dynamic operator which models edge deletion:

$$\mathcal{M}, w \models \langle \mathsf{sb} \rangle \varphi \text{ iff for some } v \in W \text{ s.t. } R(w, v),\ \mathcal{M}^{-}_{wv}, v \models \varphi,$$

Edge deletion

We can define a dynamic operator which models edge deletion:

$$\mathcal{M}, w \models \langle \mathsf{sb} \rangle \varphi \text{ iff for some } v \in W \text{ s.t. } R(w,v),\ \mathcal{M}^-_{wv}, v \models \varphi,$$

where

$$\mathcal{M} = \langle W, R, V \rangle \qquad \mathcal{M}^-_{wv} = \langle W, R^-_{wv}, V \rangle$$

$$R^-_{wv} = R \setminus \{(w,v)\}$$

Edge deletion

We can define a dynamic operator which models edge deletion:

$$\mathcal{M}, w \models \langle \mathsf{sb} \rangle \varphi \;\text{ iff for some } v \in W \text{ s.t. } R(w, v),\; \mathcal{M}^-_{wv}, v \models \varphi,$$

where

$$\mathcal{M} = \langle W, R, V \rangle \qquad \mathcal{M}^-_{wv} = \langle W, R^-_{wv}, V \rangle$$
$$R^-_{wv} = R \setminus \{(w, v)\}$$

We define BML($\langle \mathsf{sb} \rangle$) as the logic BML extended with $\langle \mathsf{sb} \rangle$.

Treeless

Theorem

$BML(\langle \mathsf{sb} \rangle)$ does not have the tree model property.

Treeless

Theorem
$BML(\langle \mathrm{sb} \rangle)$ does not have the tree model property.

Proof.
The formula $\Diamond\Diamond\top \wedge [\mathrm{sb}]\Box\bot$ does not hold at the root of a tree. $\Box$

Treeless

Theorem

$BML(\langle \mathrm{sb} \rangle)$ does not have the tree model property.

Proof.

The formula $\Diamond\Diamond\top \wedge [\mathrm{sb}]\Box\bot$ does not hold at the root of a tree. $\square$

Corollary

$BML(\langle \mathrm{sb} \rangle)$ is strictly more expressive than BML.

Parentheses: using QBF for complexity proofs

Parentheses: using QBF for complexity proofs

Quantified Boolean Formulas (QBF):

- Syntax:

$$\alpha ::= x \mid \neg\alpha \mid \alpha \wedge \alpha \mid \exists x.\alpha$$

Parentheses: using QBF for complexity proofs

Quantified Boolean Formulas (QBF):

- Syntax:

$$\alpha ::= x \mid \neg\alpha \mid \alpha \wedge \alpha \mid \exists x.\alpha$$

- Semantics:

$$\begin{array}{lcl} v \models x & \Leftrightarrow & v(x) = 1 \\ v \models \neg\alpha & \Leftrightarrow & v \not\models \alpha \\ v \models \alpha \wedge \alpha' & \Leftrightarrow & v \models \alpha \text{ and } v \models \alpha' \end{array}$$

Parentheses: using QBF for complexity proofs

Quantified Boolean Formulas (QBF):

- Syntax:

$$\alpha ::= x \mid \neg\alpha \mid \alpha \wedge \alpha \mid \exists x.\alpha$$

- Semantics:

$$\begin{array}{lcl} v \models x & \Leftrightarrow & v(x) = 1 \\ v \models \neg\alpha & \Leftrightarrow & v \not\models \alpha \\ v \models \alpha \wedge \alpha' & \Leftrightarrow & v \models \alpha \text{ and } v \models \alpha' \\ v \models \exists x.\alpha & \Leftrightarrow & v[x \to 1] \models \alpha \text{ or } v[x \to 0] \models \alpha \end{array}$$

Parentheses: using QBF for complexity proofs

Quantified Boolean Formulas (QBF):

- Syntax:

$$\alpha ::= x \mid \neg\alpha \mid \alpha \wedge \alpha \mid \exists x.\alpha$$

- Semantics:

$$\begin{array}{lcl} v \models x & \Leftrightarrow & v(x) = 1 \\ v \models \neg\alpha & \Leftrightarrow & v \not\models \alpha \\ v \models \alpha \wedge \alpha' & \Leftrightarrow & v \models \alpha \text{ and } v \models \alpha' \\ v \models \exists x.\alpha & \Leftrightarrow & v[x \to 1] \models \alpha \text{ or } v[x \to 0] \models \alpha \end{array}$$

Theorem

Deciding validity of a QBF is PSPACE*-complete.*

The model checking problem

Definition

Given a formula φ and a pointed model $\mathcal{M}, w$, the model checking problem consists in deciding if $\mathcal{M}, w \models \varphi$.

The model checking problem

Definition

Given a formula φ and a pointed model $\mathcal{M}, w$, the model checking problem consists in deciding if $\mathcal{M}, w \models \varphi$.

We would like to know what is the computational cost of solving this task.

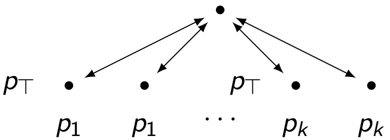
The model checking problem

Definition
Given a formula φ and a pointed model $\mathcal{M}, w$, the model checking problem consists in deciding if $\mathcal{M}, w \models \varphi$.

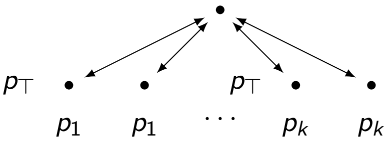
We would like to know what is the computational cost of solving this task.

We'll use QBF to give a complexity bound.

Model checking



Model checking



$\mathcal{M}_k = \langle W, R, V \rangle$ is:

$$\begin{aligned}
W &= \{w\} \cup \{w_i^1, w_i^0 \mid 1 \leq i \leq k\}\\
V(p_i) &= \{w_i^1, w_i^0\}\\
V(p_\top) &= \{w_i^1 \mid 1 \leq i \leq k\}\\
R &= \{(w, w_i^1), (w, w_i^0),\\
&\qquad (w_i^1, w), (w_i^0, w) \mid 1 \leq i \leq k\}
\end{aligned}$$

The translation

Now we will give the translations from QBF to sabotage logic. The lack of an edge pointing from the evaluation point to an $p_\top$ point means that the corresponding variable has to be assigned to 1, otherwise to 0.
Let $(\ )'$ be the following linear translation from QBF to $\mathrm{BML}(\langle \mathsf{sb} \rangle)$:

$$
\begin{aligned}
(\exists x_i.\alpha)' &= \langle \mathsf{sb} \rangle (p_i \wedge \Diamond(\alpha)') \\
(x_i)' &= \neg\Diamond(p_i \wedge p_\top) \\
(\neg\alpha)' &= \neg(\alpha)' \\
(\alpha \wedge \beta)' &= (\alpha)' \wedge (\beta)'.
\end{aligned}
$$

The translation

Now we will give the translations from QBF to sabotage logic. The lack of an edge pointing from the evaluation point to an $p_\top$ point means that the corresponding variable has to be assigned to 1, otherwise to 0.
Let $(\ )'$ be the following linear translation from QBF to $\mathrm{BML}(\langle \mathrm{sb} \rangle)$:

$$\begin{aligned}
(\exists x_i.\alpha)' &= \langle \mathrm{sb} \rangle (p_i \wedge \Diamond(\alpha)') \\
(x_i)' &= \neg\Diamond(p_i \wedge p_\top) \\
(\neg\alpha)' &= \neg(\alpha)' \\
(\alpha \wedge \beta)' &= (\alpha)' \wedge (\beta)'.
\end{aligned}$$

The we have:

$$\alpha \text{ is true iff } \mathcal{M}_k, w \models (\alpha)'.$$

We are almost there...

Theorem

Model checking BML($\langle \mathsf{sb} \rangle$) is $\mathrm{PS{\scriptsize PACE}}$*-hard.*

We are almost there...

Theorem

Model checking BML($\langle \mathsf{sb} \rangle$) is $\mathrm{PS{\scriptsize PACE}}$*-hard.*

The evaluation of the truth of a formula in a model can be done by a polynomial space algorithm that follows the semantics. Then we have:

We are almost there...

Theorem
Model checking BML($\langle \mathsf{sb} \rangle$) is PSPACE*-hard.*

The evaluation of the truth of a formula in a model can be done by a polynomial space algorithm that follows the semantics. Then we have:

Theorem
Model checking BML($\langle \mathsf{sb} \rangle$) is PSPACE*-complete.*

A more general approach...

- So far, we studied some particular dynamic logics.
- But it is possible introduce operators in a more general way.

A more general approach...

- So far, we studied some particular dynamic logics.
- But it is possible introduce operators in a more general way.
- Let W be a domain, $w \in W$ and $R \subseteq W^2$, consider the following function:

$$f_W^{\mathsf{sb}}(w, R) = \{(v, R_{wv}^{-}) \mid (w, v) \in R\}.$$

A more general approach...

- So far, we studied some particular dynamic logics.
- But it is possible introduce operators in a more general way.
- Let W be a domain, $w \in W$ and $R \subseteq W^2$, consider the following function:

$$f^{\mathsf{sb}}_W(w, R) = \{(v, R^-_{wv}) \mid (w, v) \in R\}.$$

- Let $\mathcal{C}$ be a class of models, consider the following family of functions:

$$f^{\mathsf{sb}} = \{f^{\mathsf{sb}}_W \mid \langle W, R, V\rangle \in \mathcal{C}\}.$$

A more general approach...

- So far, we studied some particular dynamic logics.
- But it is possible introduce operators in a more general way.
- Let W be a domain, $w \in W$ and $R \subseteq W^2$, consider the following function:

$$f_W^{\mathsf{sb}}(w, R) = \{(v, R^-_{wv}) \mid (w, v) \in R\}.$$

- Let $\mathcal{C}$ be a class of models, consider the following family of functions:

$$f^{\mathsf{sb}} = \{f_W^{\mathsf{sb}} \mid \langle W, R, V\rangle \in \mathcal{C}\}.$$

- Now we can change the semantics definition:

$\langle W, R, V\rangle, w \models \langle \mathsf{sb}\rangle\varphi$ iff for some $(v, S) \in f_W^{\mathsf{sb}}(w, R)$, $\langle W, S, V\rangle, v \models \varphi$.

Model Update Functions

- f^{sb} is the family of functions associated to $\langle \mathsf{sb} \rangle$.
- It is easy to change the definition and obtain other primitives, not just sabotage (e.g., add or swap edges).
- This families of functions are families of model update functions, and we can use them to define dynamic modal operators.
- This way of defining operations results useful to obtain general results for a family of logics.

Defining model update operators

Consider the following operations:

$$\begin{aligned} R^{+}_{wv} &= R \cup wv \\ R^{*}_{wv} &= (R \backslash vw) \cup wv. \end{aligned}$$

Defining model update operators

Consider the following operations:

$$\begin{aligned} R^+_{wv} &= R \cup wv \\ R^*_{wv} &= (R \backslash vw) \cup wv. \end{aligned}$$

Let's define new model update functions:

$$f^{\text{br}}_W(w, R) = \{(v, R^+_{wv}) \mid (w, v) \notin R\}$$

Defining model update operators

Consider the following operations:

$$\begin{array}{rcl} R^{+}_{wv} & = & R \cup wv \\ R^{*}_{wv} & = & (R \backslash vw) \cup wv. \end{array}$$

Let's define new model update functions:

$$\begin{array}{rcl} f^{\mathsf{br}}_{W}(w,R) & = & \{(v, R^{+}_{wv}) \mid (w,v) \notin R\} \\ f^{\mathsf{sw}}_{W}(w,R) & = & \{(v, R^{*}_{vw}) \mid (w,v) \in R\} \end{array}$$

Defining model update operators

Consider the following operations:

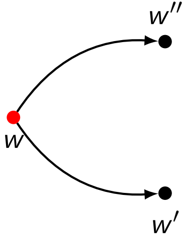
$$
\begin{array}{rcl}
R^{+}_{wv} & = & R \cup wv \\
R^{*}_{wv} & = & (R \backslash vw) \cup wv.
\end{array}
$$

Let's define new model update functions:

$$
\begin{array}{rcl}
f^{\text{br}}_{W}(w, R) & = & \{(v, R^{+}_{wv}) \mid (w, v) \notin R\} \\
f^{\text{sw}}_{W}(w, R) & = & \{(v, R^{*}_{vw}) \mid (w, v) \in R\}
\end{array}
$$

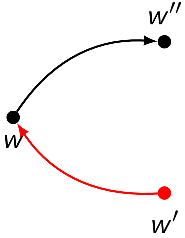
We define the operators $\langle \text{br} \rangle$ and $\langle \text{sw} \rangle$ using these model update functions.

Relation-Changing Modal Logics - Swap and Bridge



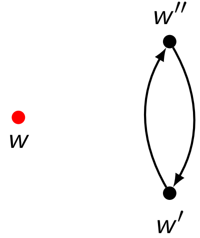
$$\mathcal{M}, w \models \langle \mathsf{sw} \rangle \varphi$$

Relation-Changing Modal Logics - Swap and Bridge



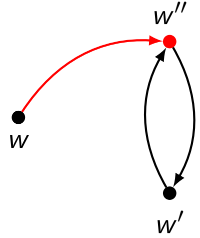
$$\mathcal{M}^*_{w'w}, w' \models \varphi$$

Relation-Changing Modal Logics - Swap and Bridge



$$\mathcal{M}, w \models \langle \mathsf{br} \rangle \varphi$$

Relation-Changing Modal Logics - Swap and Bridge



$$\mathcal{M}^{+}_{ww''}, w'' \models \varphi$$

Characterizing formula

Let f be a family of model update functions, and let $\delta_f(v_1, V_1, v_2, V_2)$ be a formula over the appropriate correspondence language with only the first-order variables $v1, v2$ and the second-order binary variables $V1, V2$ free. We say that δ_f defines f if in every model $\mathcal{M} = \langle W, R, V\rangle$, for every $w \in W$, and for every second-order assignment g,

$$\text{for all } v, S,\ (v, S) \in f_W(w, R) \text{ iff } \mathcal{M}, (((g_w^{v1})_R^{V_1})_v^{v_2})_S^{V_2} \models \delta_f.$$

How the changes are characterized

$$(v, S) \in f_W(w, R) \text{ iff } \mathcal{M}, (((g_w^{v1})_R^{V_1})_v^{v_2})_S^{V_2} \models \delta_f.$$

- v_1 represents the current point of evaluation;
- V_1 the current accessibility relation;
- v_2 is the point after the evaluation of the formula;
- V_2 is the accessibility relation after the update.

Second-order translation

Given a family of model update functions f, and δ_f a formula that defines f, define $\mathsf{ST}_{x,r}$ as follows

$$\begin{array}{rcl}
\mathsf{ST}_{x,r}(p) & = & p(x) \\
\mathsf{ST}_{x,r}(\neg\varphi) & = & \neg\mathsf{ST}_{x,r}(\varphi) \\
\mathsf{ST}_{x,r}(\varphi \wedge \psi) & = & \mathsf{ST}_{x,r}(\varphi) \wedge \mathsf{ST}_{x,r}(\psi) \\
\mathsf{ST}_{x,r}(\Diamond\varphi) & = & \exists y.(r(x,y) \wedge \mathsf{ST}_{y,r}(\varphi))
\end{array}$$

Second-order translation

Given a family of model update functions f, and δ_f a formula that defines f, define $\mathsf{ST}_{x,r}$ as follows

$$
\begin{array}{rcl}
\mathsf{ST}_{x,r}(p) & = & p(x) \\
\mathsf{ST}_{x,r}(\neg\varphi) & = & \neg\mathsf{ST}_{x,r}(\varphi) \\
\mathsf{ST}_{x,r}(\varphi \wedge \psi) & = & \mathsf{ST}_{x,r}(\varphi) \wedge \mathsf{ST}_{x,r}(\psi) \\
\mathsf{ST}_{x,r}(\Diamond\varphi) & = & \exists y.(r(x,y) \wedge \mathsf{ST}_{y,r}(\varphi)) \\
\mathsf{ST}_{x,r}(\langle f\rangle\varphi) & = & \exists y.\exists s.(\delta_f[{}^{v_1}/_{x}, {}^{V_1}/_{r}, {}^{v_2}/_{y}, {}^{V_2}/_{s}] \wedge \mathsf{ST}_{y,s}(\varphi)),
\end{array}
$$

where $\theta[{}^{x}/_{y}]$ is the formula obtained by replacing all free occurrences of x by y in θ, and y, s are variables which have not been used yet in the translation.

And as we expected...

Theorem

Let $\varphi \in BML(\langle f \rangle)$ *and let* δ_f *be a formula defining* f*. Then*

$$\mathcal{M}, w \models \varphi \text{ iff } \mathcal{M}, g^x_w \models \mathsf{ST}_{x,r}(\varphi),$$

where g *is an arbitrary second-order assignment and* g^x_w *is identical to* g *except perhaps in that* $g^x_w(x) = w$*.*

Characterizing $\Diamond$

For instance, it is easy to define the formula $\delta_\Diamond$ which characterizes the operator $\Diamond$:

Characterizing $\Diamond$

For instance, it is easy to define the formula $\delta_{\Diamond}$ which characterizes the operator $\Diamond$:

$$\delta_{\Diamond} \doteq V_1(v_1, v_2) \land \forall z.\forall z'.(V_1(z, z') \leftrightarrow V_2(z, z')).$$

Characterizing $\Diamond$

For instance, it is easy to define the formula $\delta_\Diamond$ which characterizes the operator $\Diamond$:

$$\delta_\Diamond \doteq V_1(v_1, v_2) \wedge \forall z.\forall z'.(V_1(z, z') \leftrightarrow V_2(z, z')).$$

The formula above clearly establishes that the current state has a successor, and that the accessibility relation does not change.

Characterizing $\langle \mathsf{sb} \rangle$

A formula characterizing $\langle \mathsf{sb} \rangle$ is:

$$\begin{aligned}\delta_{\langle \mathsf{sb} \rangle} \doteq \quad & V_1(v_1, v_2) \wedge \neg V_2(v_1, v_2) \wedge \\ & \forall z. \forall z'. ((v_1, v_2) \neq (z, z') \rightarrow (V_1(z, z') \leftrightarrow V_2(z, z'))).\end{aligned}$$

Characterizing $\langle \mathsf{sb} \rangle$

A formula characterizing $\langle \mathsf{sb} \rangle$ is:

$$\begin{aligned}\delta_{\langle \mathsf{sb} \rangle} \doteq \quad & V_1(v_1, v_2) \wedge \neg V_2(v_1, v_2) \wedge \\ & \forall z.\forall z'.((v_1, v_2) \neq (z, z') \rightarrow (V_1(z, z') \leftrightarrow V_2(z, z'))).\end{aligned}$$

- First part says that there are two elements which were related, and after the update are not longer related.
- The second part establishes that the rest of the relation continues exactly as before.

Characterizing $\langle \mathsf{br} \rangle$

For $\langle \mathsf{br} \rangle$ we have:

$$\begin{aligned}\delta_{\langle \mathsf{br} \rangle} \doteq \quad & \neg V_1(v_1, v_2) \wedge V_2(v_1, v_2) \wedge \\ & \forall z. \forall z'. ((v_1, v_2) \neq (z, z') \rightarrow (V_1(z, z') \leftrightarrow V_2(z, z'))).\end{aligned}$$

Characterizing $\langle \mathsf{br} \rangle$

For $\langle \mathsf{br} \rangle$ we have:

$$\begin{aligned}\delta_{\langle \mathsf{br} \rangle} \doteq \quad & \neg V_1(v_1, v_2) \wedge V_2(v_1, v_2) \wedge \\ & \forall z. \forall z'. ((v_1, v_2) \neq (z, z') \rightarrow (V_1(z, z') \leftrightarrow V_2(z, z'))).\end{aligned}$$

- First part says that there are two elements which weren't related, that after the update are related.
- The second part establishes that the rest of the relation continues exactly as before.

First-order vs. Second-order

- The standard translation for the basic modal logic BML returns a first-order formula.
- However, for operators based on model update functions we define a second-order translation.
- If you are thinking about first-order translations, that's the next step.
- We'll see it doesn't result obvious having first-order translations.

A very expressive operator

Consider the modal logic BML extended with the following operator, whose intuitive semantics is that φ is evaluated after replacing the current accessibility relation by its transitive closure:

$$\langle W, R, V\rangle, w \models \circledcirc^{+}\varphi \text{ iff } \langle W, R^{+}, V\rangle, w \models \varphi.$$

A very expressive operator

Consider the modal logic BML extended with the following operator, whose intuitive semantics is that φ is evaluated after replacing the current accessibility relation by its transitive closure:

$$\langle W, R, V\rangle, w \models \circledcirc^{+}\varphi \text{ iff } \langle W, R^{+}, V\rangle, w \models \varphi.$$

It is possible to translate $\circledcirc^{+}$ to first-order logic?

$\mathrm{BML}(\circledcirc^+)$ is not compact

Proof.

Consider the infinite set $\Gamma = \{\circledcirc^+\Diamond p\} \cup \{\Box^n \neg p \mid n \geq 0\}$. Every finite subset of Γ is satisfiable, but Γ is not. $\Box$

BML($\circledcirc^+$) is not compact

Proof.

Consider the infinite set $\Gamma = \{\circledcirc^+\Diamond p\} \cup \{\Box^n\neg p \mid n \geq 0\}$. Every finite subset of Γ is satisfiable, but Γ is not. $\Box$

- As a conclusion, we have that certain operators that can be defined in the framework we introduced cannot be translated into first-order logic.
- But we can still do something: explicit translations for some operators.

Some notation

We write xy for (x, y), and use the following notation:

$$
\begin{array}{lll}
nm = xy & \text{is defined as} & n = x \wedge m = y \\
nm \neq xy & \text{is defined as} & n \neq x \ \vee \ m \neq y \\
nm \in S & \text{is defined as} & \displaystyle\bigvee_{xy \in S} nm = xy, \text{ and} \\
nm \notin S & \text{is defined as} & \displaystyle\bigwedge_{xy \in S} nm \neq xy,
\end{array}
$$

where S is a finite set of pairs of variables. In particular $nm \in \emptyset$ is a notation for $\bot$ and $nm \notin \emptyset$ is a notation for $\top$. For S a set of pairs of variables, define $S^{-1} = \{mn \mid nm \in S\}$.

Explicit translation for BML($\langle \mathsf{sb} \rangle$)

The non-trivial cases of the translation for BML($\langle \mathsf{sb} \rangle$):

$$
\begin{array}{rcl}
\mathsf{ST}_{x,S}(\Diamond\varphi) & = & \exists y.(r(x,y) \wedge xy \notin S \wedge \mathsf{ST}_{y,S}(\varphi)) \\
\mathsf{ST}_{x,S}(\langle \mathsf{sb} \rangle\varphi) & = & \exists y.(r(x,y) \wedge xy \notin S \wedge \mathsf{ST}_{y,S\cup xy}(\varphi))
\end{array}
$$

where y and z are variables which have not been used yet in the translation.

Explicit translation for BML($\langle \mathsf{sb} \rangle$)

The non-trivial cases of the translation for BML($\langle \mathsf{sb} \rangle$):

$$
\begin{array}{rcl}
\mathrm{ST}_{x,S}(\Diamond\varphi) & = & \exists y.(r(x,y) \wedge xy \notin S \wedge \mathrm{ST}_{y,S}(\varphi)) \\
\mathrm{ST}_{x,S}(\langle \mathsf{sb} \rangle\varphi) & = & \exists y.(r(x,y) \wedge xy \notin S \wedge \mathrm{ST}_{y,S\cup xy}(\varphi))
\end{array}
$$

where y and z are variables which have not been used yet in the translation.

- S keeps the set of pairs that has been sabotaged.

Explicit translation for BML($\langle \mathsf{sb} \rangle$)

The non-trivial cases of the translation for BML($\langle \mathsf{sb} \rangle$):

$$\begin{array}{lcl} \mathsf{ST}_{x,S}(\Diamond\varphi) & = & \exists y.(r(x,y) \wedge xy \notin S \wedge \mathsf{ST}_{y,S}(\varphi)) \\ \mathsf{ST}_{x,S}(\langle \mathsf{sb} \rangle\varphi) & = & \exists y.(r(x,y) \wedge xy \notin S \wedge \mathsf{ST}_{y,S\cup xy}(\varphi)) \end{array}$$

where y and z are variables which have not been used yet in the translation.

- S keeps the set of pairs that has been sabotaged.
- For $\Diamond$ we need to add the condition $xy \notin S$.

Explicit translation for BML($\langle sb \rangle$)

The non-trivial cases of the translation for BML($\langle sb \rangle$):

$$
\begin{array}{lcl}
\mathrm{ST}_{x,S}(\Diamond\varphi) & = & \exists y.(r(x,y) \wedge xy \notin S \wedge \mathrm{ST}_{y,S}(\varphi)) \\
\mathrm{ST}_{x,S}(\langle \mathrm{sb} \rangle\varphi) & = & \exists y.(r(x,y) \wedge xy \notin S \wedge \mathrm{ST}_{y,S\cup xy}(\varphi))
\end{array}
$$

where y and z are variables which have not been used yet in the translation.

- S keeps the set of pairs that has been sabotaged.
- For $\Diamond$ we need to add the condition $xy \notin S$.
- Remember $nm \notin S = \bigwedge_{xy \in S} nm \neq xy$, not so easy...

Explicit translation for BML($\langle sb \rangle$)

The non-trivial cases of the translation for BML($\langle sb \rangle$):

$$\begin{array}{lcl} \mathsf{ST}_{x,S}(\Diamond\varphi) & = & \exists y.(r(x,y) \wedge xy \notin S \wedge \mathsf{ST}_{y,S}(\varphi)) \\ \mathsf{ST}_{x,S}(\langle \mathsf{sb} \rangle\varphi) & = & \exists y.(r(x,y) \wedge xy \notin S \wedge \mathsf{ST}_{y,S\cup xy}(\varphi)) \end{array}$$

where y and z are variables which have not been used yet in the translation.

- S keeps the set of pairs that has been sabotaged.
- For $\Diamond$ we need to add the condition $xy \notin S$.
- Remember $nm \notin S = \bigwedge_{xy \in S} nm \neq xy$, not so easy...
- Same for sabotage, also adding the new sabotaged pair to S.

Explicit translation for BML($\langle \mathsf{br} \rangle$)

The non-trivial cases of the translation for BML($\langle \mathsf{br} \rangle$):

$$
\begin{array}{lcl}
\mathsf{ST}_{x,S}(\Diamond\varphi) & = & \exists y.((r(x,y) \vee xy \in S) \wedge \mathsf{ST}_{y,S}(\varphi)) \\
\mathsf{ST}_{x,S}(\langle \mathsf{br} \rangle\varphi) & = & \exists y.(\neg(r(x,y) \vee xy \in S) \wedge \mathsf{ST}_{y,S\cup xy}(\varphi))
\end{array}
$$

Explicit translation for BML($\langle \text{br} \rangle$)

The non-trivial cases of the translation for BML($\langle \text{br} \rangle$):

$$
\begin{array}{lcl}
\mathsf{ST}_{x,S}(\Diamond\varphi) & = & \exists y.((r(x,y) \vee xy \in S) \wedge \mathsf{ST}_{y,S}(\varphi)) \\
\mathsf{ST}_{x,S}(\langle \text{br} \rangle\varphi) & = & \exists y.(\neg(r(x,y) \vee xy \in S) \wedge \mathsf{ST}_{y,S\cup xy}(\varphi))
\end{array}
$$

- S keeps the set of pairs that has been added.

Explicit translation for BML($\langle\mathsf{br}\rangle$)

The non-trivial cases of the translation for BML($\langle\mathsf{br}\rangle$):

$$
\begin{array}{lcl}
\mathsf{ST}_{x,S}(\Diamond\varphi) & = & \exists y.((r(x,y) \vee xy \in S) \wedge \mathsf{ST}_{y,S}(\varphi)) \\
\mathsf{ST}_{x,S}(\langle\mathsf{br}\rangle\varphi) & = & \exists y.(\neg(r(x,y) \vee xy \in S) \wedge \mathsf{ST}_{y,S\cup xy}(\varphi))
\end{array}
$$

- S keeps the set of pairs that has been added.
- Look for similarities...

Explicit translation for BML($\langle sw \rangle$)

The non-trivial cases of the translation for BML($\langle sw \rangle$):

$$
\begin{array}{rcl}
\mathrm{ST}_{x,S}(\Diamond\varphi) & = & \exists y.(((r(x,y) \wedge xy \notin S) \vee xy \in S^{-1}) \wedge \mathrm{ST}_{y,S}(\varphi)) \\
\mathrm{ST}_{x,S}(\langle sw \rangle\varphi) & = & \quad (r(x,x) \wedge \mathrm{ST}_{x,S}(\varphi)) \\
& & \vee \;\; \exists y.(x \neq y \wedge r(x,y) \wedge xy \notin (S \cup S^{-1}) \wedge \\
& & \qquad \mathrm{ST}_{y,S \cup xy}(\varphi)) \\
& & \vee \;\; \bigvee_{yz \in S}(x = z \wedge \mathrm{ST}_{y,S \setminus yz \cup zy}(\varphi))
\end{array}
$$

Explicit translation for BML($\langle \mathsf{sw} \rangle$)

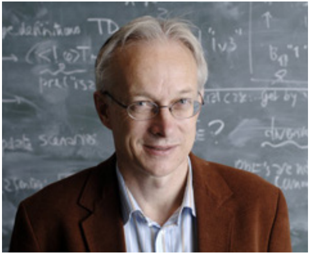
The non-trivial cases of the translation for BML($\langle \mathsf{sw} \rangle$):

$$
\begin{array}{rcl}
\mathrm{ST}_{x,S}(\Diamond\varphi) & = & \exists y.(((r(x,y) \wedge xy \notin S) \vee xy \in S^{-1}) \wedge \mathrm{ST}_{y,S}(\varphi)) \\
\mathrm{ST}_{x,S}(\langle \mathsf{sw} \rangle\varphi) & = & \quad (r(x,x) \wedge \mathrm{ST}_{x,S}(\varphi)) \\
 & & \vee \;\; \exists y.(x \neq y \wedge r(x,y) \wedge xy \notin (S \cup S^{-1}) \wedge \\
 & & \qquad \mathrm{ST}_{y,S \cup xy}(\varphi)) \\
 & & \vee \;\; \bigvee_{yz \in S}(x = z \wedge \mathrm{ST}_{y,S \setminus yz \cup zy}(\varphi))
\end{array}
$$

- S keeps the set of pairs that has been swapped.
- A more involved treatment is needed: we have to take care of reflexive edges, and edges swapped twice.

Relevant Bibliography I

Johan van Benthem. Besides being responsible of many contributions in modal logics, he introduced the first ideas about the sabotage operator.



van Benthem, J.; (2005) *An essay on sabotage and obstruction*. In: D. Hutter, W. Stephan (Eds.), Mechanizing Mathematical Reasoning, Vol. 2605 of Lecture Notes in Computer Science, Springer, pp. 268–276.

Relevant Bibliography II

- Areces, C., Fervari, R., and Hoffmann, G.; (2012) *Moving Arrows and Four Model Checking Results.* In Proceedings of the 19th International Workshop on Logic, Language, Information and Computation (WoLLIC 2012).
- Fervari R.; (2014) *Relation-Changing Modal Logics.* PhD Thesis, Universidad Nacional de Córdoba, Facultad de Matemática, Astronomía y Física. Córdoba, Argentina, March 2014.
- Areces, C., Fervari, R., and Hoffmann, G.; (2015) *Relation-Changing Modal Operators.* Logic Journal of the IGPL.