The Tree of Knowledge in Action

UCLA Logic Colloquium

Eric Pacuit

Computer Science Department, Stanford University
ai.stanford.edu/~epacuit

May 30, 2008

Introduction and Motivation

We are interested in reasoning about rational agents interacting in *social* situations.

Introduction and Motivation

We are interested in reasoning about rational agents interacting in *social* situations.

- Philosophy (social philosophy, epistemology)
- Game Theory
- Social Choice Theory
- AI (multiagent systems)

Introduction and Motivation

We are interested in reasoning about rational agents interacting in *social* situations.

What is a rational agent?

- maximize expected utility (instrumentally rational)
- react to observations
- revise beliefs when learning a *surprising* piece of information
- understand higher-order information
- plans for the future
- ????

Introduction and Motivation

We are interested in reasoning about rational agents interacting in *social* situations.

There is a jungle of formal systems!

- logics of informational attitudes (knowledge, beliefs, certainty)
- logics of action & agency
- temporal logics/dynamic logics
- logics of motivational attitudes (preferences, intentions)

(*Not to mention various game-theoretic/social choice models*)

Introduction and Motivation

We are interested in reasoning about rational agents interacting in *social* situations.

There is a jungle of formal systems!

- How do we compare different logical systems studying the same phenomena?
- How *complex* is it to reason about rational agents?
- How should we *combine* the various logical systems?
- What do the logical frameworks contribute to the discussion on rational agency?

(Not to mention various game-theoretic/social choice models)

Introduction and Motivation

We are interested in reasoning about rational agents interacting in *social* situations.

There is a jungle of formal systems!

- How do we compare different logical systems studying the same phenomena?
- How *complex* is it to reason about rational agents?
- How should we *combine* the various logical systems?
- What do the logical frameworks contribute to the discussion on rational agency?

(Not to mention various game-theoretic/social choice models)

Introduction and Motivation

We are interested in reasoning about rational agents interacting in *social* situations.

What do the logical frameworks contribute to the discussion on rational agency?

- refine and test our intuitions
- (epistemic) foundations of game theory
  Logic *and* Game Theory, not Logic in place of Game Theory.
- Social Software: Verify properties of social procedures
  - *Refine existing social procedures or suggest new ones*

R. Parikh. *Social Software. Synthese* **132** (2002).

Logics *of* rational agents in social situations.

vs.

Logics *about* rational agents in social situations.

Question: How do we model (and reason about) *information change* in a social situation?

Question: How do we model (and reason about) *information change* in a social situation?

Plan for today:

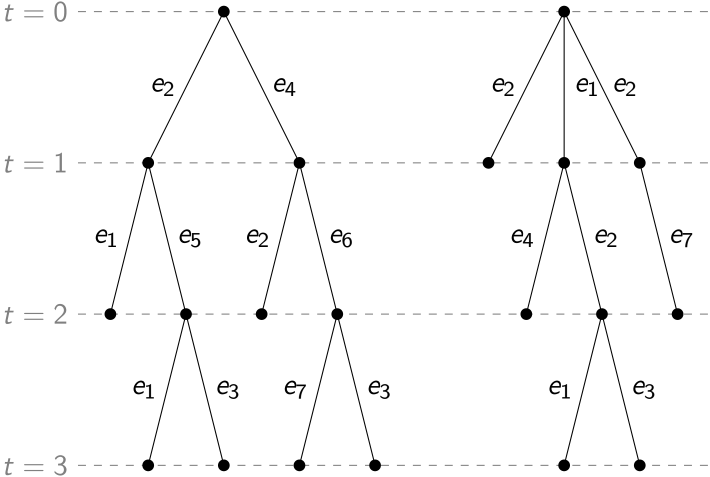
- Method 1: Epistemic Temporal Logic (ETL)
  - Basic framework
  - Survey of decidability/undecidability results
- Method 2: Dynamic Epistemic Logic (DEL)
- Comparing DEL and ETL
- Some further questions

Epistemic Temporal Logic

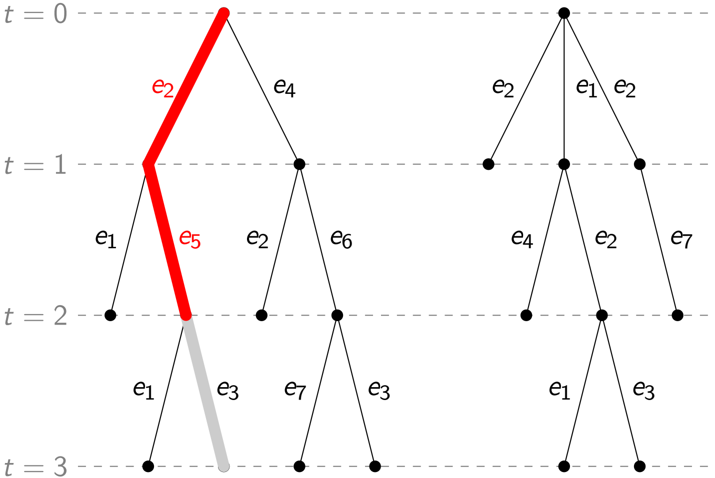
R. Parikh and R. Ramanujam. *A Knowledge Based Semantics of Messages*. *Journal of Logic, Language and Information*, 12: 453 – 467, 1985, 2003.

FHMV. Reasoning about Knowledge. MIT Press, 1995.

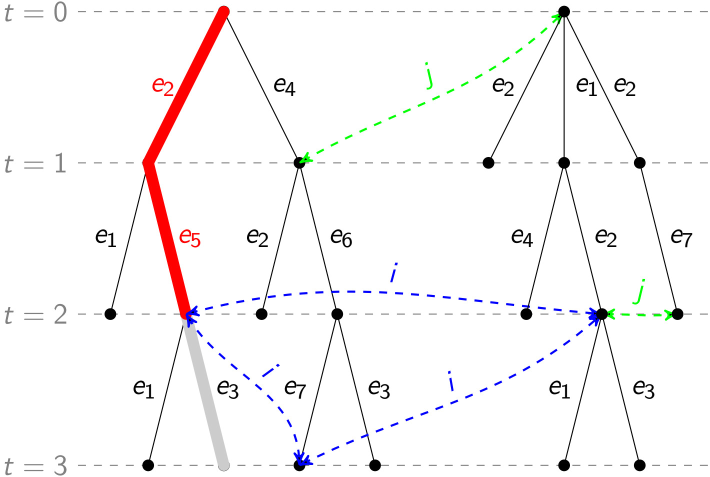
The 'Playground'



The 'Playground'



The ‘Playground’



The ‘Playground’: Notation

The ‘Playground’: Notation

- Let Σ be any set. The elements of Σ are called **events**.

The 'Playground': Notation

- Let Σ be any set. The elements of Σ are called **events**.
- Given any set X, X^* is the set of finite strings over X and X^ω the set of infinite strings over X. Elements of $\Sigma^* \cup \Sigma^\omega$ will be called **histories**.

The 'Playground': Notation

- Let Σ be any set. The elements of Σ are called **events**.
- Given any set X, X^* is the set of finite strings over X and X^ω the set of infinite strings over X. Elements of $\Sigma^* \cup \Sigma^\omega$ will be called **histories**.
- Given $H \in \Sigma^* \cup \Sigma^\omega$, $\mathsf{len}(H)$ is the **length** of H.

The 'Playground': Notation

- Let Σ be any set. The elements of Σ are called **events**.
- Given any set X, X^* is the set of finite strings over X and X^ω the set of infinite strings over X. Elements of $\Sigma^* \cup \Sigma^\omega$ will be called **histories**.
- Given $H \in \Sigma^* \cup \Sigma^\omega$, $\mathsf{len}(H)$ is the **length** of H.
- Given $H, H' \in \Sigma^* \cup \Sigma^\omega$, we write $H \preceq H'$ if H is a *finite* prefix of H'.

The 'Playground': Notation

- Let Σ be any set. The elements of Σ are called **events**.
- Given any set X, X^* is the set of finite strings over X and X^ω the set of infinite strings over X. Elements of $\Sigma^* \cup \Sigma^\omega$ will be called **histories**.
- Given $H \in \Sigma^* \cup \Sigma^\omega$, $\mathsf{len}(H)$ is the **length** of H.
- Given $H, H' \in \Sigma^* \cup \Sigma^\omega$, we write $H \preceq H'$ if H is a *finite* prefix of H'.
- $\mathsf{FinPre}(\mathcal{H}) = \{H \mid \exists H' \in \mathcal{H} \text{ such that } H \preceq H'\}$ is the set of finite prefixes of the elements of $\mathcal{H}$.

The 'Playground': Notation

- Let Σ be any set. The elements of Σ are called **events**.
- Given any set X, X^* is the set of finite strings over X and X^ω the set of infinite strings over X. Elements of $\Sigma^* \cup \Sigma^\omega$ will be called **histories**.
- Given $H \in \Sigma^* \cup \Sigma^\omega$, $\mathsf{len}(H)$ is the **length** of H.
- Given $H, H' \in \Sigma^* \cup \Sigma^\omega$, we write $H \preceq H'$ if H is a *finite* prefix of H'.
- $\mathsf{FinPre}(\mathcal{H}) = \{H \mid \exists H' \in \mathcal{H} \text{ such that } H \preceq H'\}$ is the set of finite prefixes of the elements of $\mathcal{H}$.
- ϵ is the empty string and $\mathsf{FinPre}_{-\epsilon}(\mathcal{H}) = \mathsf{FinPre}(\mathcal{H}) - \{\epsilon\}$.

History-based Frames

Definition

Let Σ be any set of events. A set $\mathcal{H} \subseteq \Sigma^* \cup \Sigma^\omega$ is called a protocol provided $\mathsf{FinPre}_{-\epsilon}(\mathcal{H}) \subseteq \mathcal{H}$. A rooted protocol is any set $\mathcal{H} \subseteq \Sigma^* \cup \Sigma^\omega$ where $\mathsf{FinPre}(\mathcal{H}) \subseteq \mathcal{H}$.

History-based Frames

Definition

Let Σ be any set of events. A set $\mathcal{H} \subseteq \Sigma^* \cup \Sigma^\omega$ is called a protocol provided $\mathsf{FinPre}_{-\epsilon}(\mathcal{H}) \subseteq \mathcal{H}$. A rooted protocol is any set $\mathcal{H} \subseteq \Sigma^* \cup \Sigma^\omega$ where $\mathsf{FinPre}(\mathcal{H}) \subseteq \mathcal{H}$.

Definition

An ETL frame is a tuple $\langle \Sigma, \mathcal{H}, \{\sim_i\}_{i \in \mathcal{A}} \rangle$ where Σ is a (finite or infinite) set of events, $\mathcal{H}$ is a protocol, and for each $i \in \mathcal{A}$, $\sim_i$ is an equivalence relation on the set of finite strings in $\mathcal{H}$.

History-based Frames

Definition

Let Σ be any set of events. A set $\mathcal{H} \subseteq \Sigma^* \cup \Sigma^\omega$ is called a protocol provided $\text{FinPre}_{-\epsilon}(\mathcal{H}) \subseteq \mathcal{H}$. A rooted protocol is any set $\mathcal{H} \subseteq \Sigma^* \cup \Sigma^\omega$ where $\text{FinPre}(\mathcal{H}) \subseteq \mathcal{H}$.

Definition

An ETL frame is a tuple $\langle \Sigma, \mathcal{H}, \{\sim_i\}_{i \in \mathcal{A}} \rangle$ where Σ is a (finite or infinite) set of events, $\mathcal{H}$ is a protocol, and for each $i \in \mathcal{A}$, $\sim_i$ is an equivalence relation on the set of finite strings in $\mathcal{H}$.

Some assumptions:

1. If Σ is assumed to be finite, then we say that $\mathcal{F}$ is **finitely branching**.
2. If $\mathcal{H}$ is a rooted protocol, $\mathcal{F}$ is a **tree frame**.

Formal Languages

- $P\varphi$ (φ is true *sometime* in the past),
- $F\varphi$ (φ is true *sometime* in the future),
- $Y\varphi$ (φ is true at *the* previous moment),
- $N\varphi$ (φ is true at *the* next moment),
- $N_e\varphi$ (φ is true after event e)
- $K_i\varphi$ (agent i knows φ) and
- $C_B\varphi$ (the group $B \subseteq \mathcal{A}$ commonly knows φ).

History-based Models

An ETL **model** is a structure $\langle \mathcal{H}, \{\sim_i\}_{i \in \mathcal{A}}, V \rangle$ where $\langle \mathcal{H}, \{\sim_i\}_{i \in \mathcal{A}} \rangle$ is an ETL frame and

$V : \mathsf{At} \rightarrow 2^{\mathsf{finite}(\mathcal{H})}$ is a valuation function.

Formulas are interpreted at pairs H, t:

$$H, t \models \varphi$$

Truth in a Model

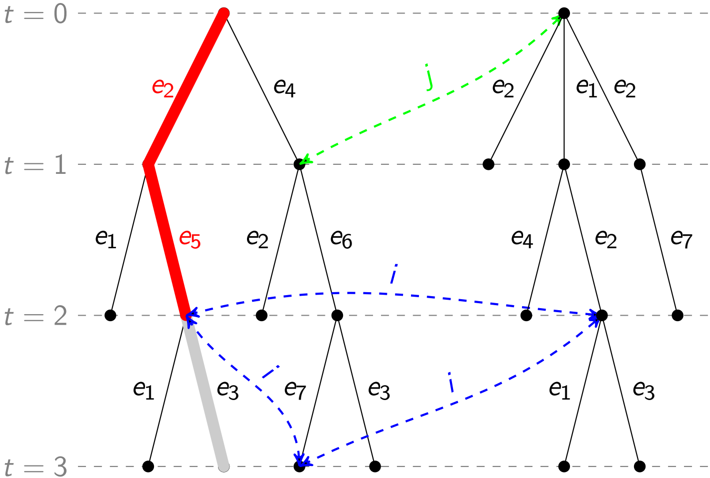
- $H, t \models P\varphi$ iff there exists $t' \leq t$ such that $H, t' \models \varphi$
- $H, t \models F\varphi$ iff there exists $t' \geq t$ such that $H, t' \models \varphi$
- $H, t \models N\varphi$ iff $H, t+1 \models \varphi$
- $H, t \models Y\varphi$ iff $t > 1$ and $H, t-1 \models \varphi$
- $H, t \models K_i\varphi$ iff for each $H' \in \mathcal{H}$ and $m \geq 0$ if $H_t \sim_i H'_m$ then $H', m \models \varphi$
- $H, t \models C\varphi$ iff for each $H' \in \mathcal{H}$ and $m \geq 0$ if $H_t \sim_* H'_m$ then $H', m \models \varphi$.

where $\sim_*$ is the reflexive transitive closure of the union of the $\sim_i$.

Truth in a Model

- $H, t \models P\varphi$ iff there exists $t' \leq t$ such that $H, t' \models \varphi$
- $H, t \models F\varphi$ iff there exists $t' \geq t$ such that $H, t' \models \varphi$
- $H, t \models N\varphi$ iff $H, t+1 \models \varphi$
- $H, t \models Y\varphi$ iff $t > 1$ and $H, t-1 \models \varphi$
- $H, t \models K_i\varphi$ iff for each $H' \in \mathcal{H}$ and $m \geq 0$ if $H_t \sim_i H'_m$ then $H', m \models \varphi$
- $H, t \models C\varphi$ iff for each $H' \in \mathcal{H}$ and $m \geq 0$ if $H_t \sim_* H'_m$ then $H', m \models \varphi$.

where $\sim_*$ is the reflexive transitive closure of the union of the $\sim_i$.


t = 0
t = 1
t = 2
t = 3
e2
e4
j
e2
e1
e2
e1
e5
e2
e6
e4
e2
e7
i
e1
e3
e7
e3
i
e1
e3

An Example

Ann would like Bob to attend her talk; however, she only wants Bob to attend if he is interested in the subject of her talk, not because he is just being polite.

An Example

Ann would like Bob to attend her talk; however, she only wants Bob to attend if he is interested in the subject of her talk, not because he is just being polite.

There is a very simple procedure to solve Ann's problem: *have a (trusted) friend tell Bob the time and subject of her talk.*

An Example

Ann would like Bob to attend her talk; however, she only wants Bob to attend if he is interested in the subject of her talk, not because he is just being polite.

There is a very simple procedure to solve Ann's problem: *have a (trusted) friend tell Bob the time and subject of her talk.*

Is this procedure correct?

An Example

Yes, if

1. Ann knows about the talk.

An Example

Yes, if

1. Ann knows about the talk.
2. Bob knows about the talk.

An Example

Yes, if

1. Ann knows about the talk.
2. Bob knows about the talk.
3. Ann knows that Bob knows about the talk.

An Example

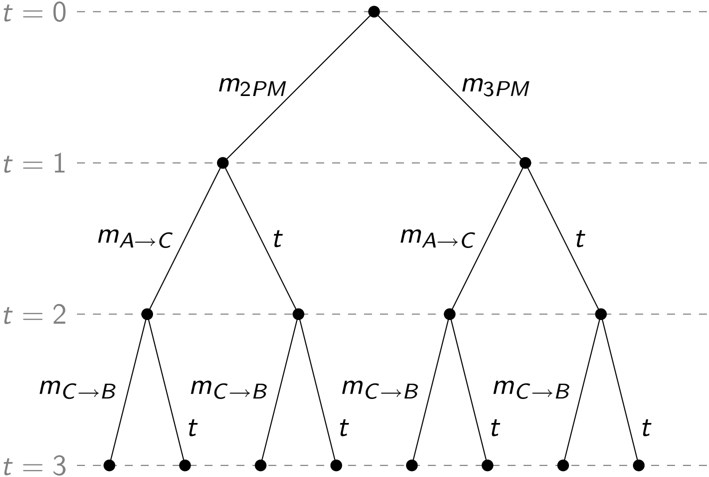
Yes, if

1. Ann knows about the talk.
2. Bob knows about the talk.
3. Ann knows that Bob knows about the talk.
4. Bob *does not* know that Ann knows that he knows about the talk.

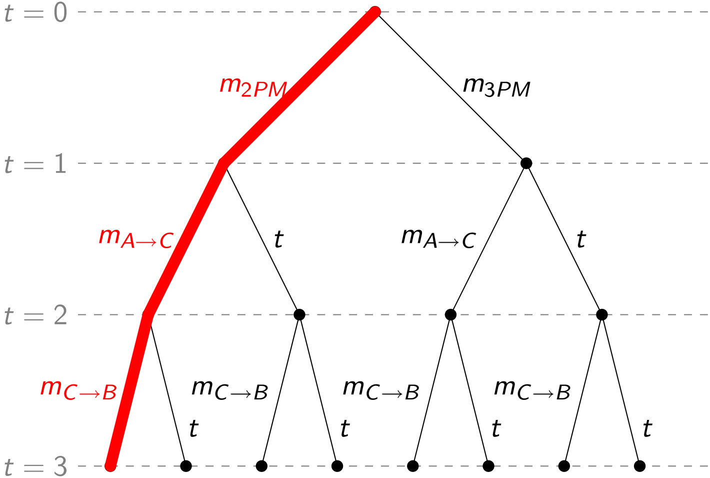
An Example

Yes, if

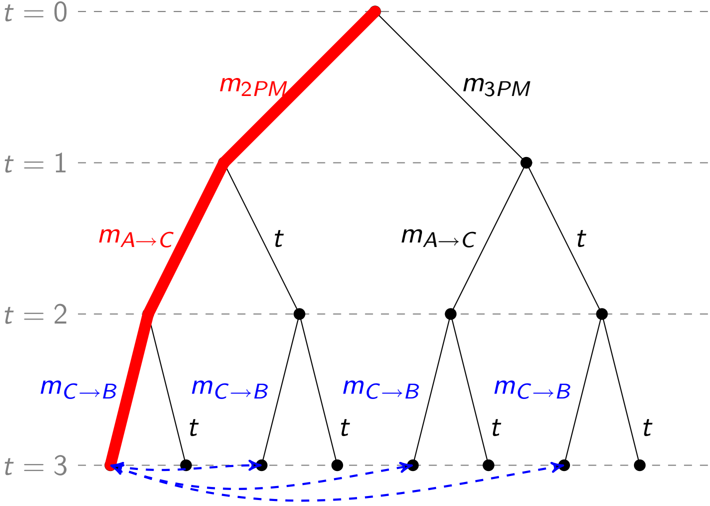
1. Ann knows about the talk.
2. Bob knows about the talk.
3. Ann knows that Bob knows about the talk.
4. Bob *does not* know that Ann knows that he knows about the talk.
5. *And nothing else.*



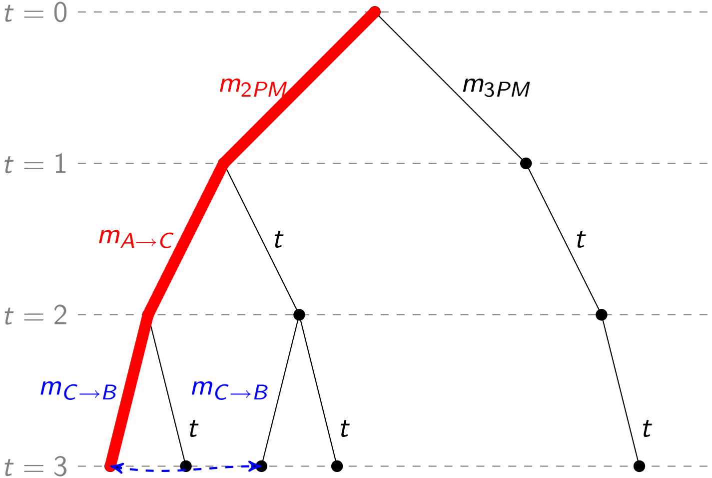
$t = 0$
$t = 1$
$t = 2$
$t = 3$
m_{2PM}
m_{3PM}
$m_{A \to C}$
t
$m_{A \to C}$
t
$m_{C \to B}$
t
$m_{C \to B}$
t
$m_{C \to B}$
t
$m_{C \to B}$
t



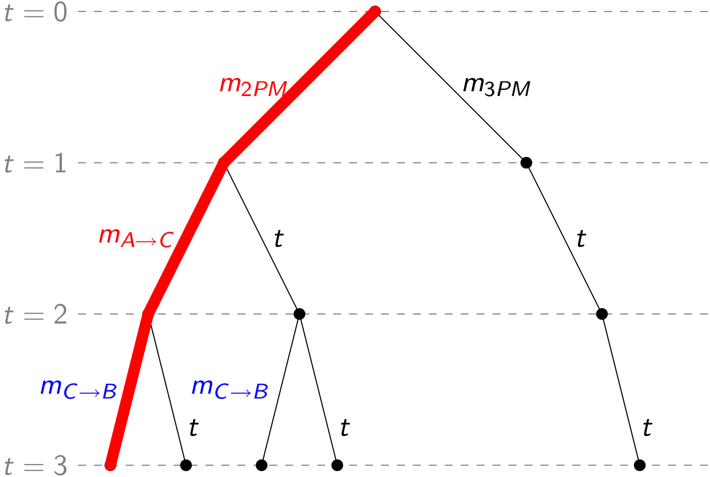
$$H, 3 \models \varphi$$



Bob's uncertainty: $H, 3 \models \neg K_B P_{2PM}$

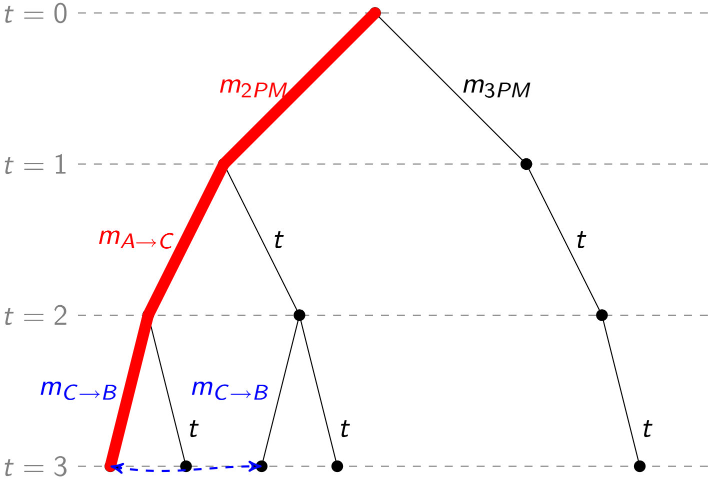


Bob's uncertainty + 'Protocol information': $H, 3 \models K_B P_{2PM}$



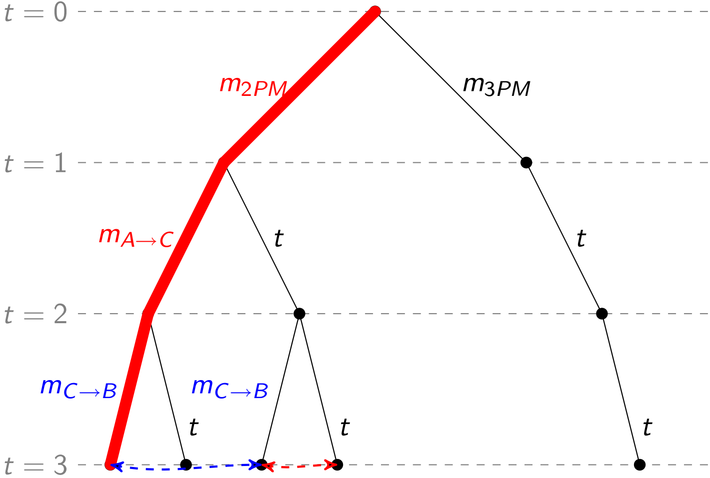
Bob's uncertainty + 'Protocol information':

$$H, 3 \models \neg K_B K_A K_B P_{2PM}$$



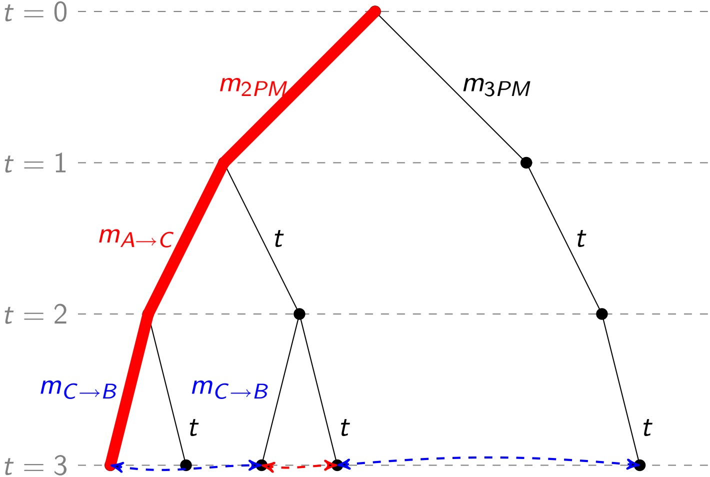
Bob's uncertainty + 'Protocol information':

$$H, 3 \models \neg K_B K_A K_B P_{2PM}$$



Bob's uncertainty + 'Protocol information':

$H, 3 \models \neg K_B K_A K_B P_{2PM}$



Bob's uncertainty + 'Protocol information':

$H, 3 \models \neg K_B K_A K_B P_{2PM}$

Living at the Edge of Decidability

Living at the Edge of Decidability

1. Expressivity of the formal language. Does the language include a common knowledge operator? A future operator? Both?

Living at the Edge of Decidability

1. Expressivity of the formal language. Does the language include a common knowledge operator? A future operator? Both?

2. Structural conditions on the underlying event structure. Do we restrict to protocol frames (finitely branching trees)? Finitely branching forests? Or, arbitrary ETL frames?

Living at the Edge of Decidability

1. Expressivity of the formal language. Does the language include a common knowledge operator? A future operator? Both?

2. Structural conditions on the underlying event structure. Do we restrict to protocol frames (finitely branching trees)? Finitely branching forests? Or, arbitrary ETL frames?

3. Conditions on the reasoning abilities of the agents. Do the agents satisfy perfect recall? No miracles? Do they agents' know what time it is?

Agent Oriented Properties:

- **No Miracles**: For all finite histories $H, H' \in \mathcal{H}$ and events $e \in \Sigma$ such that $He \in \mathcal{H}$ and $H'e \in \mathcal{H}$, if $H \sim_i H'$ then $He \sim_i H'e$.
- **Perfect Recall**: For all finite histories $H, H' \in \mathcal{H}$ and events $e \in \Sigma$ such that $He \in \mathcal{H}$ and $H'e \in \mathcal{H}$, if $He \sim_i H'e$ then $H \sim_i H'$.
- **Synchronous**: For all finite histories $H, H' \in \mathcal{H}$, if $H \sim_i H'$ then $\text{len}(H) = \text{len}(H')$.

Decidability in the Purely Temporal Setting

Theorem (Rabin)

The satisfiable problem for monadic second-order logic of the k-ary tree is decidable.

M. O. Rabin. *Decidability of Second-Order Theories and Automata on Infinite Trees*. *Transactions of the American Mathematical Society*, 141, 1969.

Theorem

The satisfiability problem for $\mathcal{L}_{TL}$ with respect to TL tree models (without epistemic structure) is decidable.

Arbitrary Agents

Theorem

The satisfiability problem (with respect to a language $\mathcal{L}_{ETL}$ with C,F, etc.) is decidable — EXPTIME-complete).

Arbitrary Agents

Theorem

The satisfiability problem (with respect to a language $\mathcal{L}_{ETL}$ with C,F, etc.) is decidable — EXPTIME-complete).

- The theorem holds if we restrict to tree models.

Ideal Agents

Assume there are two agents

Theorem

The satisfiability problem for $\mathcal{L}_{ETL}$ is highly undecidable under certain idealizations.

Ideal Agents

Assume there are two agents

Theorem

The satisfiability problem for $\mathcal{L}_{ETL}$ is highly undecidable under certain idealizations.

For example,

Theorem (Halpern & Vardi)

On interpreted systems that satisfy perfect recall or no learning, the satisfiability problem for $\mathcal{L}_{ETL}$ is Σ^1_1-complete.

(no learning: For $H, H' \in \mathcal{H}$, if $H_t \sim_i H'_{t'}$ then for all $k \geq t$ there exists $k' \geq t'$ such that $H_k \sim_i H'_{k'}$.)

J. Halpern and M. Vardi.. *The Complexity of Reasoning abut Knowledge and Time. J. Computer and Systems Sciences*, 38, 1989.

High Undecidability over ETL Structures

Define the language $\mathcal{L}_\Sigma(\mathcal{A})$ inductively as follows:

$$\varphi := p \mid \neg\varphi \mid \varphi \wedge \psi \mid \langle\alpha\rangle\varphi$$

$$\alpha := a \mid ?p \mid \alpha;\beta \mid \alpha \cup \beta \mid \alpha^*$$

High Undecidability over ETL Structures

Define the language $\mathcal{L}_\Sigma(\mathcal{A})$ inductively as follows:

$$\varphi := p \mid \neg\varphi \mid \varphi \wedge \psi \mid \langle\alpha\rangle\varphi$$

$$\alpha := a \mid ?p \mid \alpha;\beta \mid \alpha \cup \beta \mid \alpha^*$$

- $HR_\sigma H'$ iff $H' = H\sigma$ if $\sigma \in \Sigma$
- $HR_i H'$ iff $H \sim_i H'$
- $HR_{\sigma^-} H'$ iff $\text{len}(H) \geq 1$ and $H = H'\sigma$
- $HR_{?p} H'$ iff $H = H'$ and $p \in V(H)$.
- $HR_{\alpha;\beta} H'$ iff there exist H'' such that $HR_\alpha H''$ and $H'' R_\beta H'$
- $HR_{\alpha^*} H'$ iff for some n, $HR^n_\alpha H'$ (i.e. $R_{\alpha^*} = R^*_\alpha$)

High Undecidability over ETL Structures

Define the language $\mathcal{L}_\Sigma(\mathcal{A})$ inductively as follows:

$$\varphi := p \mid \neg\varphi \mid \varphi \wedge \psi \mid \langle\alpha\rangle\varphi$$

$$\alpha := a \mid ?p \mid \alpha;\beta \mid \alpha \cup \beta \mid \alpha^*$$

- $HR_\sigma H'$ iff $H' = H\sigma$ if $\sigma \in \Sigma$
- $HR_i H'$ iff $H \sim_i H'$
- $HR_{\sigma^-} H'$ iff $\text{len}(H) \geq 1$ and $H = H'\sigma$
- $HR_{?p} H'$ iff $H = H'$ and $p \in V(H)$.
- $HR_{\alpha;\beta} H'$ iff there exist H'' such that $HR_\alpha H''$ and $H'' R_\beta H'$
- $HR_{\alpha^*} H'$ iff for some n, $HR^n_\alpha H'$ (i.e. $R_{\alpha^*} = R^*_\alpha$)

$H, t \models \langle\alpha\rangle\varphi$ iff there exists $H' \in \mathcal{H}$ and $m \in \mathbb{N}$ such that $H_t R_\alpha H'_m$ and $H', m \models \varphi$

High Undecidability over Trees

Theorem

The satisfiable problem of $\mathcal{L}_\Sigma(\mathcal{A})$ with respect to ETL tree frames that satisfy no miracles is Σ_1^1-complete.

Theorem

The satisfiability problem with respect to synchronous ETL tree frames that satisfy perfect recall is Σ_1^1-complete.

High Undecidability over Trees

Theorem

The satisfiable problem of $\mathcal{L}_\Sigma(\mathcal{A})$ with respect to ETL tree frames that satisfy no miracles is Σ^1_1-complete.

Theorem

The satisfiability problem with respect to synchronous ETL tree frames that satisfy perfect recall is Σ^1_1-complete.

Theorem

The satisfiability problem with respect to the language $\mathcal{L}_\Sigma(\mathcal{A})^-$ with respect to ETL tree frames that satisfy perfect recall is Σ^1_1 complete.

Theorem

The validity problem for bounded agents is decidable.

Recurrent Tiling

Let $\mathcal{T}$ be a finite set of tile types such that each edge is associated with a color. A **tiling** of the plane is a function $t : \mathbb{N} \times \mathbb{N} \rightarrow \mathcal{T}$ such that the colors of successive tiles match.

A **recurrent tiling** is a tiling such that from some $t_0 \in \mathcal{T}$, t_0 appears infinitely often along the x-axis.

Theorem

The recurrent tiling problem (does there exist a recurrent tiling of the first-quadrant?) *is* $\Sigma^1_1 -$ *complete.*

Theorem

The satisfiable problem of $\mathcal{L}_\Sigma(\mathcal{A})$ with respect to ETL tree frames that satisfy no miracles is Σ^1_1-complete.

Theorem

The satisfiable problem of $\mathcal{L}_\Sigma(\mathcal{A})$ with respect to ETL tree frames that satisfy no miracles is Σ^1_1-complete.

Brief! Sketch of the Proof: Find a formula φ such that φ is satisfiable iff there is a recurrent tiling of the plane.

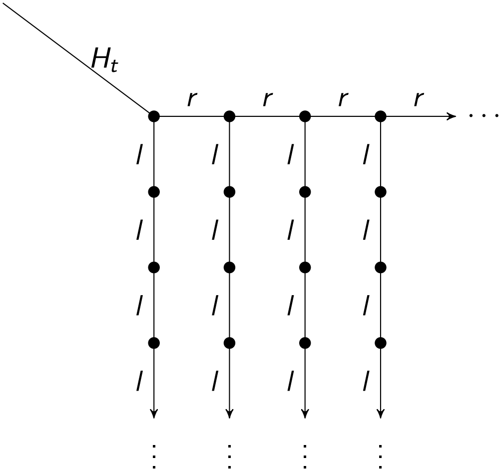
Theorem

The satisfiable problem of $\mathcal{L}_\Sigma(\mathcal{A})$ with respect to ETL tree frames that satisfy no miracles is Σ^1_1-complete.

Brief! Sketch of the Proof: Find a formula φ such that φ is satisfiable iff there is a recurrent tiling of the plane.

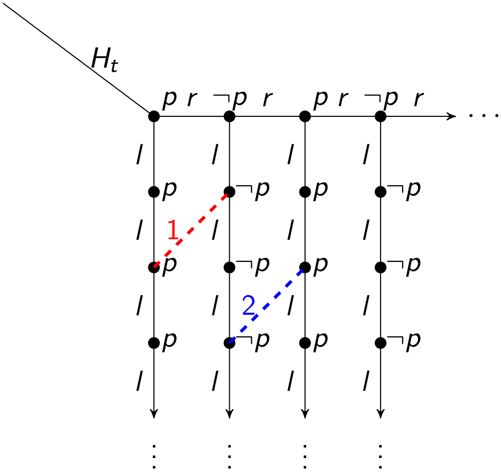
Brief! Sketch of the proof

$H, t \models [r^*; l; l^*]\neg\langle r\rangle\top$
$H, t \models [r^*; l^*]\langle l\rangle\top$
and so on



Brief! Sketch of the proof

$p \rightarrow [1]\neg p$
$\neg p \rightarrow [2]p$



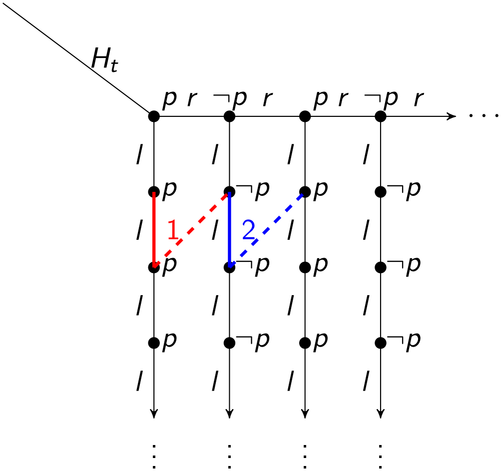
Brief! Sketch of the proof

"up-step":

$\alpha_u :=$

$?p; I; 1; ?\neg p; I; 2; ?p.$

$[\alpha_u^*]\langle\alpha_u\rangle\top$



Questions

- The argument mixes temporal and epistemic steps under the scope of the iteration operator. The Halpern & Vardi results use a weaker language, but stronger structural and agent properties.

Questions

- The argument mixes temporal and epistemic steps under the scope of the iteration operator. The Halpern & Vardi results use a weaker language, but stronger structural and agent properties.
- What about languages with common knowledge and arbitrary past?

Questions

- The argument mixes temporal and epistemic steps under the scope of the iteration operator. The Halpern & Vardi results use a weaker language, but stronger structural and agent properties.
- What about languages with common knowledge and arbitrary past?
- What corresponds to the *finite model property*?

J. van Benthem and EP. *The Tree of Knowledge in Action*. Proceedings of AiML, 2006.

Two Methodologies

ETL methodology: when describing a social situation, first write down all possible sequences of events, then at each moment write down the agents' uncertainty, from that infer how the agents' knowledge changes from one moment to the next.

Two Methodologies

ETL methodology: when describing a social situation, first write down all possible sequences of events, then at each moment write down the agents' uncertainty, from that infer how the agents' knowledge changes from one moment to the next.

Alternative methodology: describe an initial situations, provide a method for how events change a model that can be described in the formal language, then construct the event tree as needed.

Two Methodologies

ETL methodology: when describing a social situation, first write down all possible sequences of events, then at each moment write down the agents' uncertainty, from that infer how the agents' knowledge changes from one moment to the next.

Alternative methodology: describe an initial situations, provide a method for how events change a model that can be described in the formal language, then construct the event tree as needed.

Dynamic Epistemic Logic

Returning to the Example: DEL

Returning to the Example: DEL

$$(\mathcal{M} \otimes E_1) \otimes E_2$$

Returning to the Example: DEL

$$(\mathcal{M} \otimes E_1) \otimes E_2$$

The initial model (Ann and Bob are ignorant about P_{2PM}).

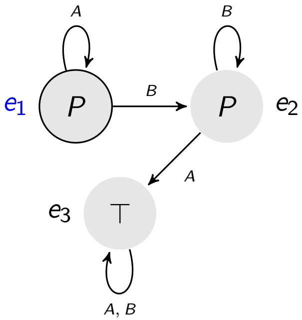
Private announcement to Ann about the talk.

Abstract Description of the Event

Recall the Ann and Bob example: Charles tells Bob that the talk is at 2PM.

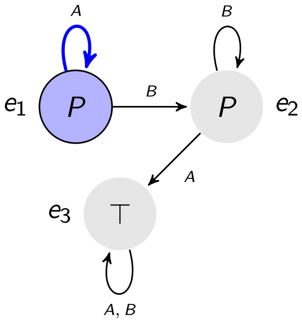
Abstract Description of the Event

Recall the Ann and Bob example: Charles tells Bob that the talk is at 2PM.



Abstract Description of the Event

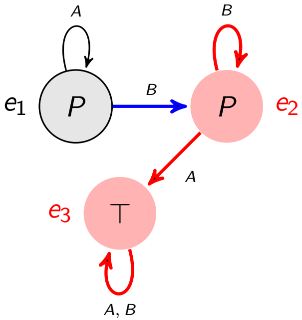
Recall the Ann and Bob example: Charles tells Bob that the talk is at 2PM.



Ann knows which event took place.

Abstract Description of the Event

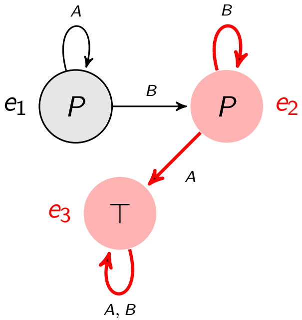
Recall the Ann and Bob example: Charles tells Bob that the talk is at 2PM.



Bob thinks a different event took place.

Abstract Description of the Event

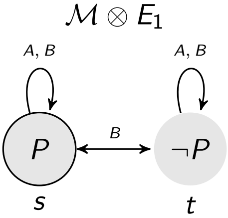
Recall the Ann and Bob example: Charles tells Bob that the talk is at 2PM.

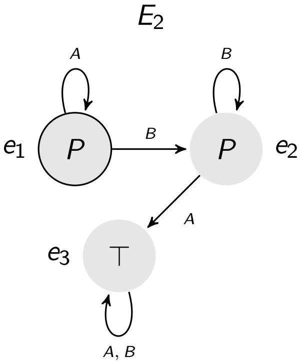


That is, Bob learns the time of the talk, but Ann learns nothing.

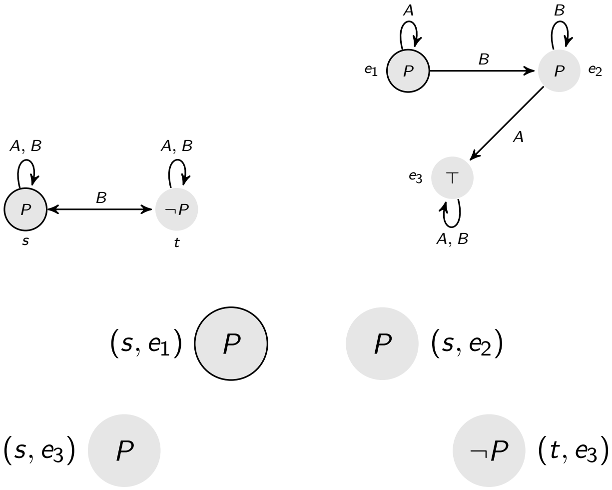
Product Update

Product Update

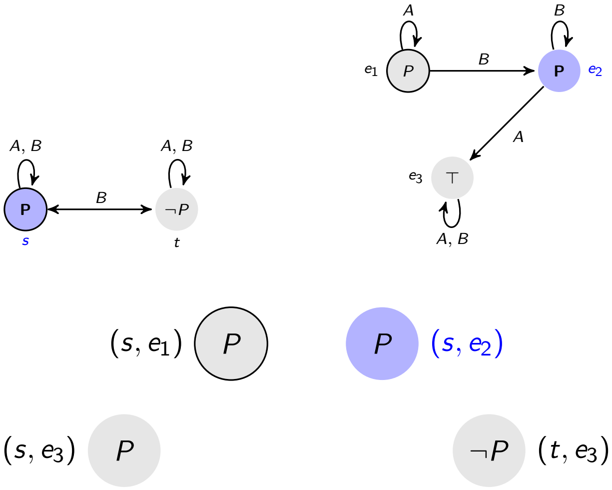




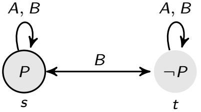
Product Update

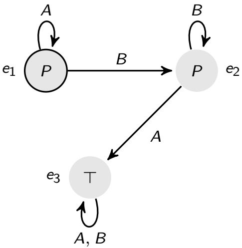


Product Update



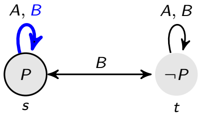
Product Update

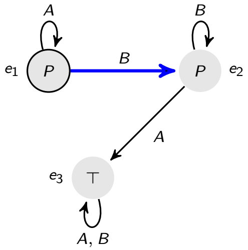




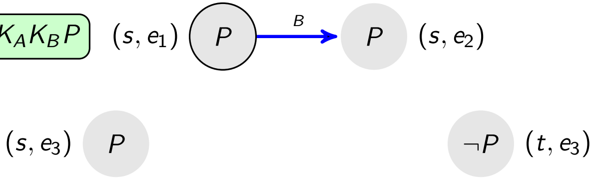
$(s, e_1) \models \neg K_B K_A K_B P$

Product Update

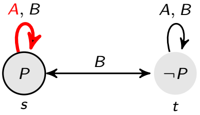


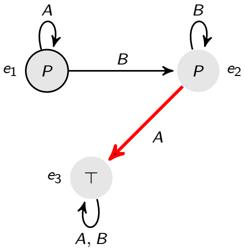


$(s, e_1) \models \neg K_B K_A K_B P$

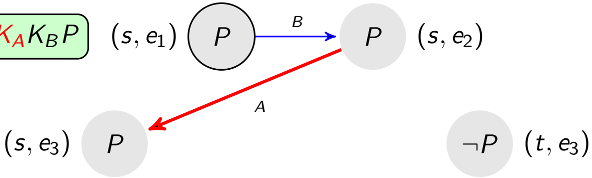


Product Update

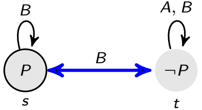


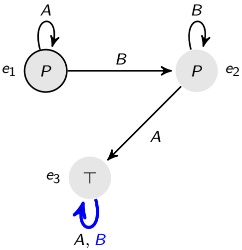


$(s, e_1) \models \neg K_B K_A K_B P$

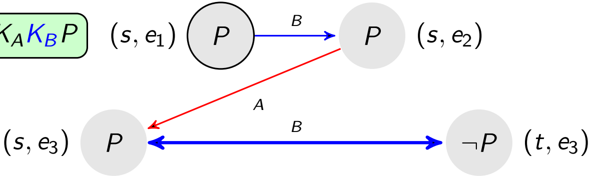


Product Update





$$(s, e_1) \models \neg K_B K_A K_B P$$



Product Update Details

Let $\mathbb{M} = \langle W, R, V\rangle$ be a Kripke model.

An event model is a tuple $\mathbb{A} = \langle A, S, Pre\rangle$, where $S \subseteq A \times A$ and $Pre : \mathcal{L} \rightarrow \wp(A)$.

Product Update Details

Let $\mathbb{M} = \langle W, R, V \rangle$ be a Kripke model.

An event model is a tuple $\mathbb{A} = \langle A, S, Pre \rangle$, where $S \subseteq A \times A$ and $Pre : \mathcal{L} \to \wp(A)$.

The update model $\mathbb{M} \otimes \mathbb{A} = \langle W', R', V' \rangle$ where

Product Update Details

Let $\mathbb{M} = \langle W, R, V \rangle$ be a Kripke model.

An event model is a tuple $\mathbb{A} = \langle A, S, Pre \rangle$, where $S \subseteq A \times A$ and $Pre : \mathcal{L} \rightarrow \wp(A)$.

The update model $\mathbb{M} \otimes \mathbb{A} = \langle W', R', V' \rangle$ where

- $W' = \{(w, a) \mid w \models Pre(a)\}$

Product Update Details

Let $\mathbb{M} = \langle W, R, V \rangle$ be a Kripke model.

An event model is a tuple $\mathbb{A} = \langle A, S, Pre \rangle$, where $S \subseteq A \times A$ and $Pre : \mathcal{L} \rightarrow \wp(A)$.

The update model $\mathbb{M} \otimes \mathbb{A} = \langle W', R', V' \rangle$ where

- $W' = \{(w, a) \mid w \models Pre(a)\}$
- $(w, a)R'(w', a')$ iff wRw' and aSa'

Product Update Details

Let $\mathbb{M} = \langle W, R, V \rangle$ be a Kripke model.

An event model is a tuple $\mathbb{A} = \langle A, S, Pre \rangle$, where $S \subseteq A \times A$ and $Pre : \mathcal{L} \to \wp(A)$.

The update model $\mathbb{M} \otimes \mathbb{A} = \langle W', R', V' \rangle$ where

- $W' = \{(w, a) \mid w \models Pre(a)\}$
- $(w, a) R' (w', a')$ iff wRw' and aSa'
- $(w, a) \in V(p)$ iff $w \in V(p)$

Product Update Details

Let $\mathbb{M} = \langle W, R, V \rangle$ be a Kripke model.

An event model is a tuple $\mathbb{A} = \langle A, S, Pre \rangle$, where $S \subseteq A \times A$ and $Pre : \mathcal{L} \to \wp(A)$.

The update model $\mathbb{M} \otimes \mathbb{A} = \langle W', R', V' \rangle$ where

- $W' = \{(w, a) \mid w \models Pre(a)\}$
- $(w, a)R'(w', a')$ iff wRw' and aSa'
- $(w, a) \in V(p)$ iff $w \in V(p)$

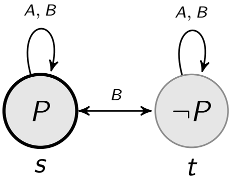
$\mathcal{M}, w \models [\mathrm{A}, a]\varphi$ iff $\mathcal{M}, w \models Pre(a)$ implies $\mathcal{M} \otimes A, (w, a) \models \varphi$.

Literarture

A. Baltag and L. Moss. *Logics for Epistemic Programs*. 2004.

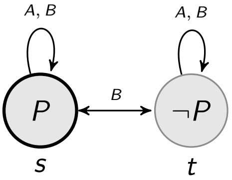
W. van der Hoek, H. van Ditmarsch and B. Kooi. *Dynamic Episetmic Logic*. 2007.

Example: Public Announcement



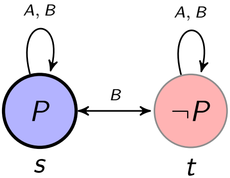
P means "The talk is at 2PM".

Example: Public Announcement



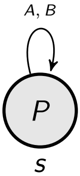
What happens if Ann publicly announces P?

Example: Public Announcement



What happens if Ann publicly announces P?

Example: Public Announcement



What happens if Ann publicly announces P? $s \models CP$

Example: Public Announcement Logic

J. Plaza. *Logics of Public Communications*. 1989.

J. Gerbrandy. *Bisimulations on Planet Kripke*. 1999.

J. van Benthem. *One is a lonely number*. 2002.

Example: Public Announcement Logic

The Public Announcement Language is generated by the following grammar:

$$p \mid \neg\varphi \mid \varphi \wedge \varphi \mid K_i\varphi \mid C\varphi \mid [\psi]\varphi$$

where $p \in \mathsf{At}$ and $i \in \mathcal{A}$.

Example: Public Announcement Logic

The Public Announcement Language is generated by the following grammar:

$$p \mid \neg\varphi \mid \varphi \wedge \varphi \mid K_i\varphi \mid C\varphi \mid [\psi]\varphi$$

where $p \in \mathsf{At}$ and $i \in \mathcal{A}$.

- $[\psi]\varphi$ is intended to mean "After publicly announcing ψ, φ is true".

Example: Public Announcement Logic

The Public Announcement Language is generated by the following grammar:

$$p \mid \neg\varphi \mid \varphi \wedge \varphi \mid K_i\varphi \mid C\varphi \mid [\psi]\varphi$$

where $p \in \mathsf{At}$ and $i \in \mathcal{A}$.

- $[P]K_iP$: "After publicly announcing P, agent i knows P"

Example: Public Announcement Logic

The Public Announcement Language is generated by the following grammar:

$$p \mid \neg\varphi \mid \varphi \wedge \varphi \mid K_i\varphi \mid C\varphi \mid [\psi]\varphi$$

where $p \in \mathsf{At}$ and $i \in \mathcal{A}$.

- $[\neg K_i P]CP$: "After announcing that agent i does not know P, then P is common knowledge"

Example: Public Announcement Logic

The Public Announcement Language is generated by the following grammar:

$$p \mid \neg\varphi \mid \varphi \wedge \varphi \mid K_i\varphi \mid C\varphi \mid [\psi]\varphi$$

where $p \in \mathsf{At}$ and $i \in \mathcal{A}$.

- $[\neg K_i P]K_i P$: "after announcing i does not know P, then i knows P. "

Example: Public Announcement Logic

Suppose $\mathcal{M} = \langle W, \{R_i\}_{i\in\mathcal{A}}, V\rangle$ is a multi-agent Kripke Model

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

where $\mathcal{M}|_\psi = \langle W', R', V'\rangle$ with

- $W' = W \cap \{w \mid \mathcal{M}, w \models \psi\}$
- $R' = R \cap W' \times W'$
- for all $p \in \mathsf{At}$, $V'(p) = V(p) \cap W'$

Example: Public Announcement Logic

$$[\psi]p \quad \leftrightarrow \quad (\psi \rightarrow p)$$

Example: Public Announcement Logic

$$\begin{aligned} [\psi]p &\leftrightarrow (\psi \rightarrow p) \\ [\psi]\neg\varphi &\leftrightarrow (\psi \rightarrow \neg[\psi]\varphi) \end{aligned}$$

Example: Public Announcement Logic

$$
\begin{array}{rcl}
[\psi]p & \leftrightarrow & (\psi \rightarrow p) \\
[\psi]\neg\varphi & \leftrightarrow & (\psi \rightarrow \neg[\psi]\varphi) \\
[\psi](\psi \wedge \chi) & \leftrightarrow & ([\varphi]\psi \wedge [\varphi]\chi)
\end{array}
$$

Example: Public Announcement Logic

$$
\begin{array}{rcl}
[\psi]p & \leftrightarrow & (\psi \rightarrow p) \\
[\psi]\neg\varphi & \leftrightarrow & (\psi \rightarrow \neg[\psi]\varphi) \\
[\psi](\psi \wedge \chi) & \leftrightarrow & ([\varphi]\psi \wedge [\varphi]\chi) \\
[\psi][\varphi]\chi & \leftrightarrow & [\psi \wedge [\psi]\varphi]\chi
\end{array}
$$

Example: Public Announcement Logic

$$
\begin{array}{rcl}
[\psi]p & \leftrightarrow & (\psi \rightarrow p) \\
[\psi]\neg\varphi & \leftrightarrow & (\psi \rightarrow \neg[\psi]\varphi) \\
[\psi](\psi \wedge \chi) & \leftrightarrow & ([\varphi]\psi \wedge [\varphi]\chi) \\
[\psi][\varphi]\chi & \leftrightarrow & [\psi \wedge [\psi]\varphi]\chi \\
[\psi]K_i\varphi & \leftrightarrow & (\psi \rightarrow K_i[\psi]\varphi)
\end{array}
$$

Example: Public Announcement Logic

$$\begin{array}{rcl} [\psi]p & \leftrightarrow & (\psi \rightarrow p) \\ [\psi]\neg\varphi & \leftrightarrow & (\psi \rightarrow \neg[\psi]\varphi) \\ [\psi](\psi \wedge \chi) & \leftrightarrow & ([\varphi]\psi \wedge [\varphi]\chi) \\ [\psi][\varphi]\chi & \leftrightarrow & [\psi \wedge [\psi]\varphi]\chi \\ [\psi]K_i\varphi & \leftrightarrow & (\psi \rightarrow K_i[\psi]\varphi) \end{array}$$

Theorem Every formula of Public Announcement Logic is equivalent to a formula of Epistemic Logic.

Example: Public Announcement Logic

$$
\begin{array}{rcl}
[\psi]p & \leftrightarrow & (\psi \rightarrow p) \\
[\psi]\neg\varphi & \leftrightarrow & (\psi \rightarrow \neg[\psi]\varphi) \\
[\psi](\psi \wedge \chi) & \leftrightarrow & ([\varphi]\psi \wedge [\varphi]\chi) \\
[\psi][\varphi]\chi & \leftrightarrow & [\psi \wedge [\psi]\varphi]\chi \\
[\psi]K_i\varphi & \leftrightarrow & (\psi \rightarrow K_i[\psi]\varphi)
\end{array}
$$

The situation is more complicated with common knowledge.

J. van Benthem, J. van Eijk, B. Kooi. *Logics of Communication and Change*. 2006.

Some Questions

- How do we relate the ETL-style analysis with the DEL-style analysis?
- In the DEL setting, what are the underlying assumptions about the reasoning abilities of the agents?
- Can we axiomatize interesting subclasses of ETL frames?

J. van Benthem, J. Gerbrandy, EP. *Merging Frameworks for Interaction: DEL and ETL*. TARK 2007.

J. van Benthem, J. Gerbrandy, T. Hoshi, EP. *Merging Frameworks for Interaction*. manuscript.

DEL *and* ETL

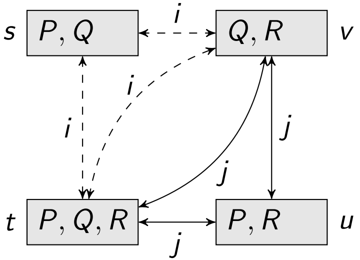
Observation: By repeatedly updating an epistemic model with event models, the machinery of DEL creates ETL models.

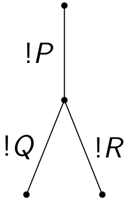
DEL *and* ETL

Observation: By repeatedly updating an epistemic model with event models, the machinery of DEL creates ETL models.

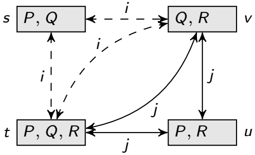
Let M be an epistemic model, and P a DEL protocol (tree of event models). The ETL model generated by M and P, forest(M, P), represents all possible evolutions of the system obtained by updating M with sequences from P.

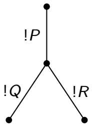
Example: Initial Model and Protocol



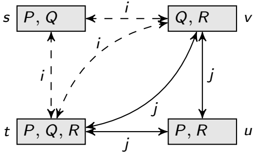


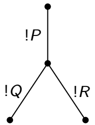
Example

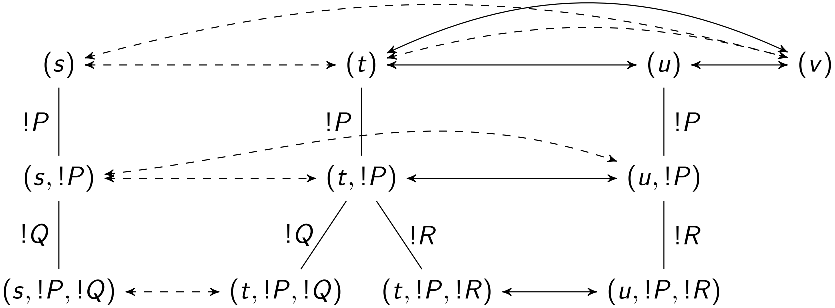




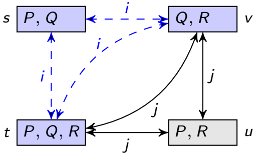
Example

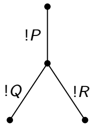

s
P, Q
Q, R
v
t
P, Q, R
P, R
u
i
i
i
j
j
j

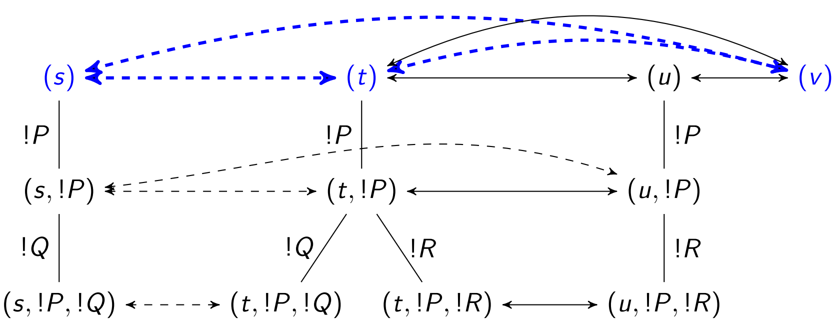

!P
!Q
!R


(s)
(t)
(u)
(v)
!P
!P
!P
(s, !P)
(t, !P)
(u, !P)
!Q
!Q
!R
!R
(s, !P, !Q)
(t, !P, !Q)
(t, !P, !R)
(u, !P, !R)

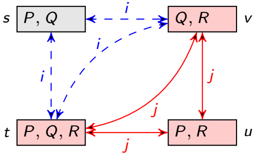
Example

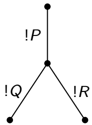

s
P, Q
Q, R
v
i
i
i
j
j
t
P, Q, R
P, R
u
j

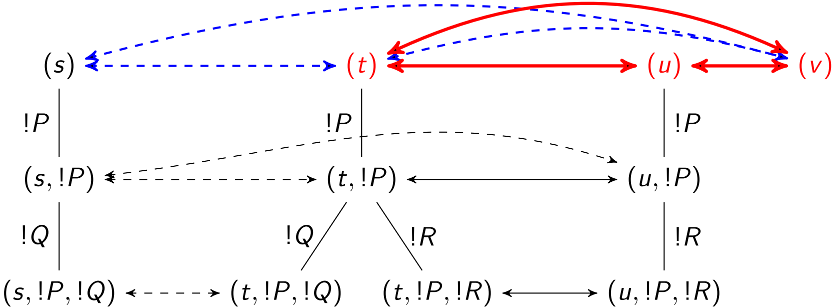

!P
!Q
!R


(s)
(t)
(u)
(v)
!P
!P
!P
(s, !P)
(t, !P)
(u, !P)
!Q
!Q
!R
!R
(s, !P, !Q)
(t, !P, !Q)
(t, !P, !R)
(u, !P, !R)

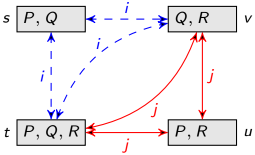
Example

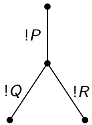

s
P, Q
i
Q, R
v
i
i
j
j
t
P, Q, R
P, R
u
j

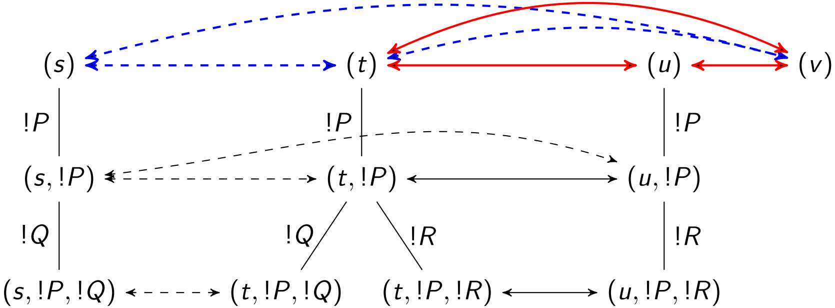

!P
!Q
!R


(s)
(t)
(u)
(v)
!P
!P
!P
(s, !P)
(t, !P)
(u, !P)
!Q
!Q
!R
!R
(s, !P, !Q)
(t, !P, !Q)
(t, !P, !R)
(u, !P, !R)

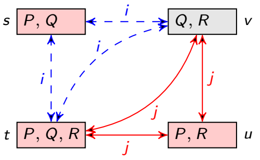
Example

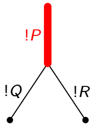

s
P, Q
Q, R
v
t
P, Q, R
P, R
u
i
i
i
j
j
j

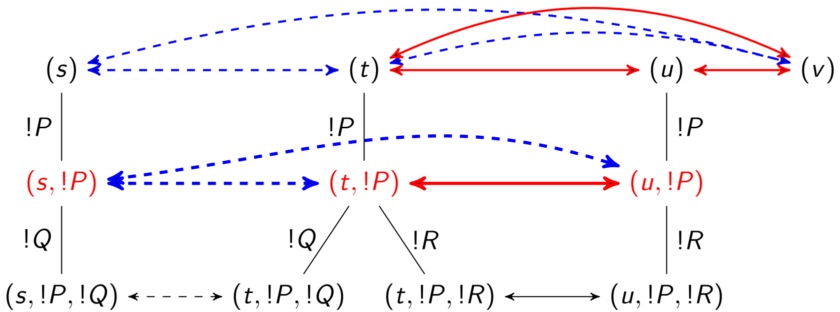

!P
!Q
!R


(s)
(t)
(u)
(v)
!P
!P
!P
(s, !P)
(t, !P)
(u, !P)
!Q
!Q
!R
!R
(s, !P, !Q)
(t, !P, !Q)
(t, !P, !R)
(u, !P, !R)

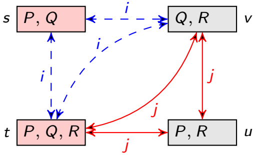
Example

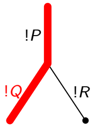

s
P, Q
Q, R
v
i
i
i
j
j
t
P, Q, R
P, R
u
j

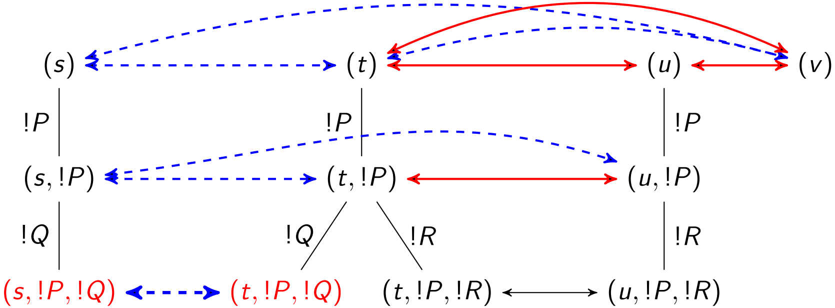

!P
!Q
!R


(s)
(t)
(u)
(v)
!P
!P
!P
(s, !P)
(t, !P)
(u, !P)
!Q
!Q
!R
!R
(s, !P, !Q)
(t, !P, !Q)
(t, !P, !R)
(u, !P, !R)

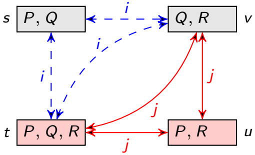
Example

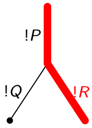

s
P, Q
Q, R
v
i
i
i
j
j
t
P, Q, R
P, R
u
j

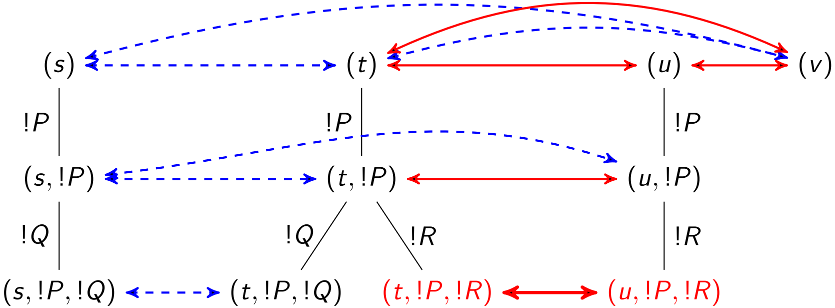

!P
!Q
!R


(s)
(t)
(u)
(v)
!P
!P
!P
(s, !P)
(t, !P)
(u, !P)
!Q
!Q
!R
!R
(s, !P, !Q)
(t, !P, !Q)
(t, !P, !R)
(u, !P, !R)

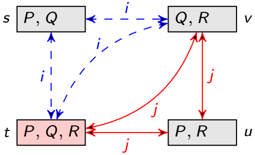
Example


s
P, Q
i
Q, R
v
i
i
j
j
t
P, Q, R
P, R
u
j

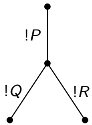

!P
!Q
!R


(s)
(t)
(u)
(v)
!P
!P
!P
(s, !P)
(t, !P)
(u, !P)
!Q
!Q
!R
!R
(s, !P, !Q)
(t, !P, !Q)
(t, !P, !R)
(u, !P, !R)

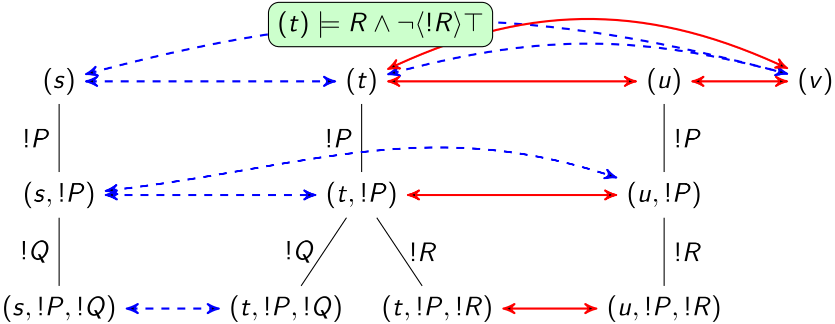
Example



s
P, Q
Q, R
v
t
P, Q, R
P, R
u
i
i
i
j
j
j



!P
!Q
!R



$(t) \models R \wedge \neg\langle !R\rangle\top$
(s)
(t)
(u)
(v)
$!P$
$!P$
$!P$
$(s, !P)$
$(t, !P)$
$(u, !P)$
$!Q$
$!Q$
$!R$
$!R$
$(s, !P, !Q)$
$(t, !P, !Q)$
$(t, !P, !R)$
$(u, !P, !R)$

State-Dependent Protocols

The ETL models $\mathbb{F}\mathcal{M}$P in the previus example satisfies a rather strong *uniformity condition*: if $(\mathcal{E}, e)$ is allowable according to the protocol P then for all histories h, the epistemic action $(\mathcal{E}, e)$ can be executed at h iff pre(e) is true at h.

Definition

State-Dependent DEL Protocol Let $\mathcal{M}$ be an epistemic model. A **state-dependent DEL protocol on** $\mathcal{M}$ is a function $p : D(\mathcal{M}) \rightarrow Ptcl(\mathbb{E})$.

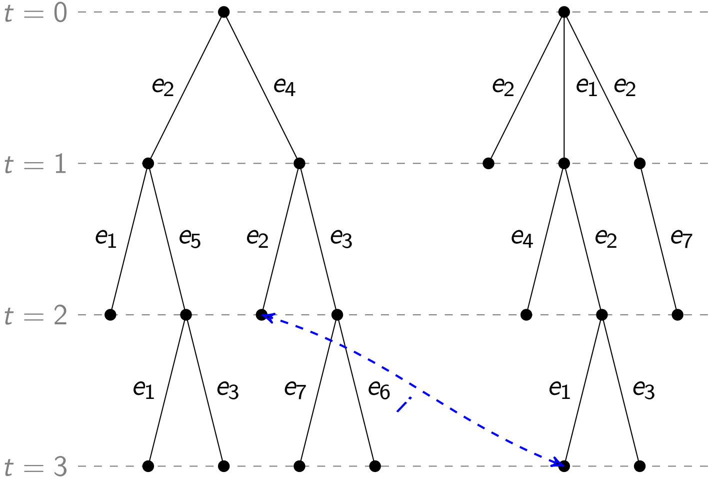
Representation Result

Given a set of DEL protocols $\mathbf{X}$, let $\mathbb{F}(\mathbf{X})$ be the class of ETL frames generated by protocols from $\mathbf{X}$.

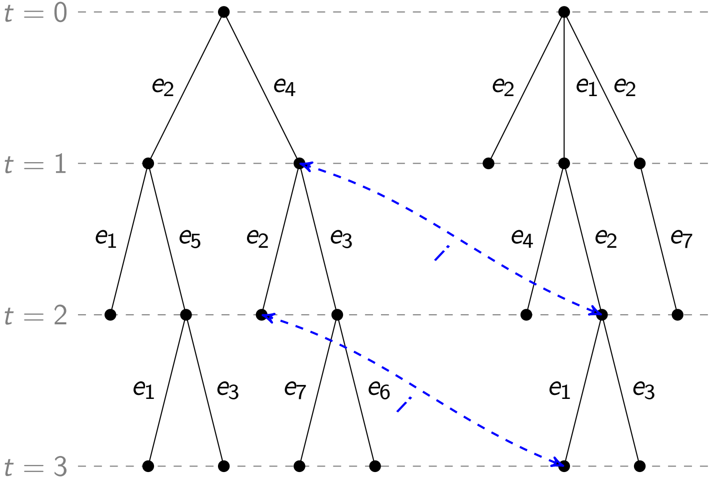
Theorem (Main Representation Theorem)

Let Σ be a finite set of events and suppose $\mathbf{X}_{DEL}^{uni}$ is the class of uniform DEL protocols. A model is in $\mathbb{F}(\mathbf{X}_{DEL}^{uni})$ iff it satisfies propositional stability, synchronicity, perfect recall, local no miracles, and local bisimulation invariance.

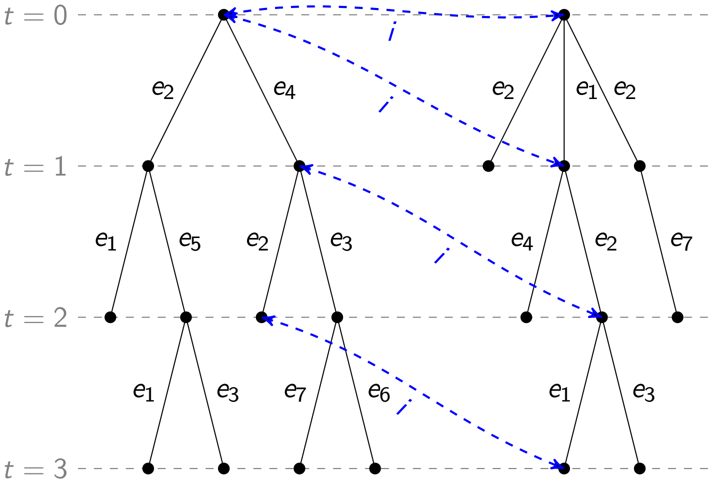
Perfect Recall



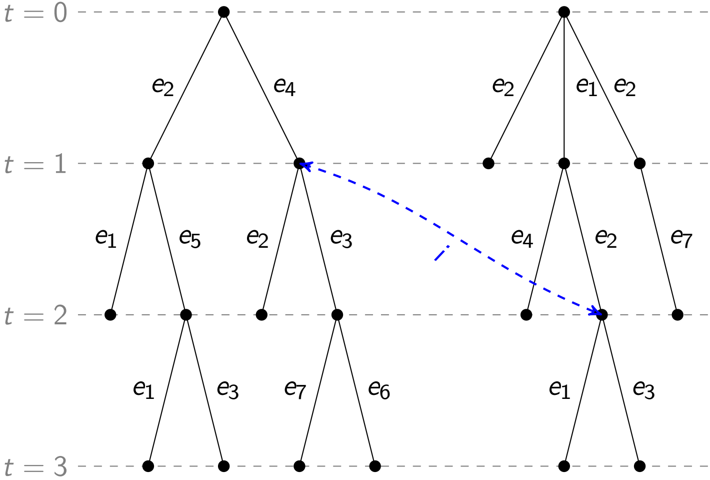
Perfect Recall



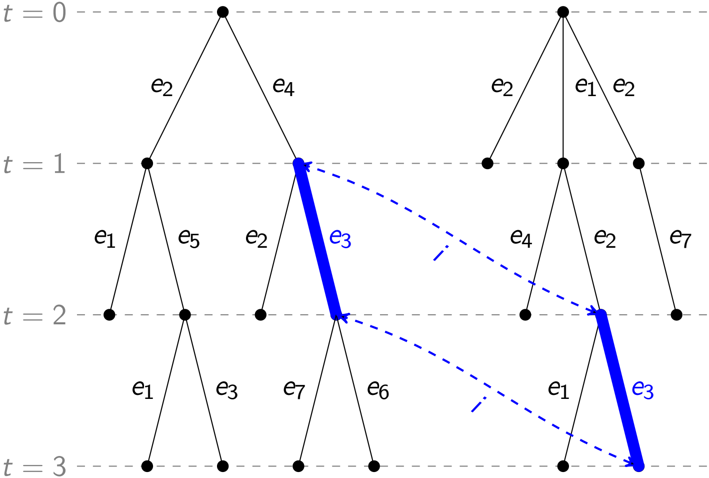
Perfect Recall



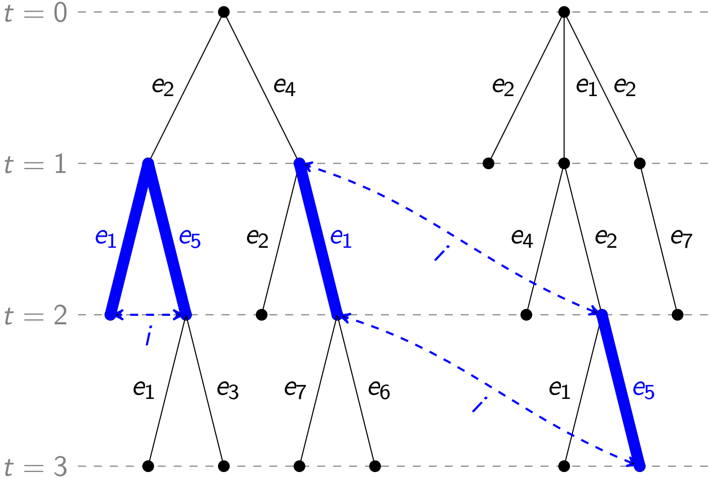
No Miracles



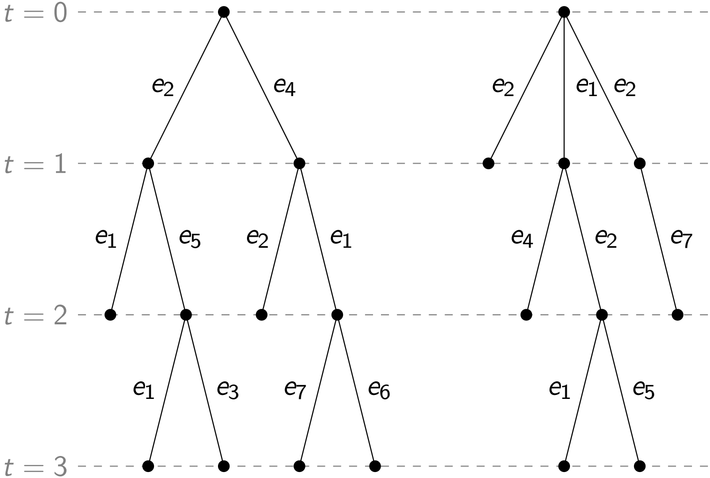
No Miracles



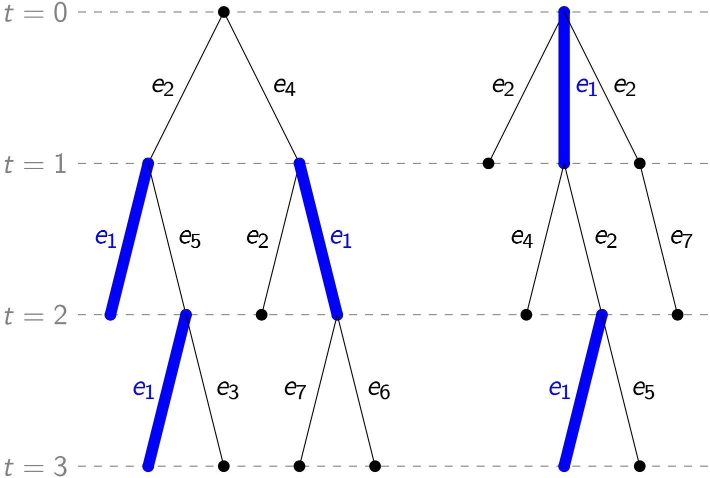
No Miracles



Bisimulation Invariance



Bisimulation Invariance



Why is DEL Decidable?

Theorem

The logic of the epistemic language with only operators $\langle e \rangle$ on models with Perfect Recall and No Miracles is decidable.

Derived from decidability of $\mathbf{PDL} \times \mathbf{K}_m$

D. Gabbay, A. Kurucz, F. Wolter and M. Zakharyaschev. *Many Dimensional Modal Logics: Theory and Applications*. 2003.

Why is DEL Decidable?

Theorem

The logic of the epistemic language with only operators $\langle e \rangle$ on models with Perfect Recall and No Miracles is decidable.

Derived from decidability of $\mathbf{PDL} \times \mathbf{K}_m$

D. Gabbay, A. Kurucz, F. Wolter and M. Zakharyaschev. *Many Dimensional Modal Logics: Theory and Applications*. 2003.

Theorem (Miller & Moss)

The dynamic-epistemic logic of public announcement with program iterations is undecidable.

J. Miller and L. Moss. *The Undecidability of Iterated Modal Relativization*. *Studia Logica*, 79:**3**, 2005.

Recall that if **X** is a set of DEL protocols, we define $\mathbb{F}(\mathbf{X}) = \{\mathbb{F}(\mathcal{M}, \mathsf{P}) \mid \mathcal{M}$ an epistemic model and $\mathsf{P} \in \mathbf{X}\}$. This construction suggests the following natural questions:

- Which DEL protocols generate interesting ETL models?
- Which modal languages are most suitable to describe these models?
- Can we axiomatize interesting classes DEL-generated ETL models?

J. van Benthem, J. Gerbrandy, T. Hoshi, EP. *Merging Frameworks for Interaction*. manuscript.

Conclusions

- Epistemic temporal models are a natural meeting place for logical studies from many different directions (Rabin's Theorem, products of logics, DEL, etc.).

Conclusions

- Epistemic temporal models are a natural meeting place for logical studies from many different directions (Rabin's Theorem, products of logics, DEL, etc.).
- Decidability/Undecidability results: on ETL frames, the grid is there, so be careful what you can say about it.

Conclusions

- Epistemic temporal models are a natural meeting place for logical studies from many different directions (Rabin's Theorem, products of logics, DEL, etc.).
- Decidability/Undecidability results: on ETL frames, the grid is there, so be careful what you can say about it.
- Merging DEL and ETL: New axiomatization results ($\langle !P\rangle\top \leftrightarrow P$)

Conclusions

- Epistemic temporal models are a natural meeting place for logical studies from many different directions (Rabin's Theorem, products of logics, DEL, etc.).
- Decidability/Undecidability results: on ETL frames, the grid is there, so be careful what you can say about it.
- Merging DEL and ETL: New axiomatization results ($\langle !P\rangle\top \leftrightarrow P$)
- Part of larger discussion on logics of model change: belief revision, preference change

Conclusions

- Epistemic temporal models are a natural meeting place for logical studies from many different directions (Rabin's Theorem, products of logics, DEL, etc.).
- Decidability/Undecidability results: on ETL frames, the grid is there, so be careful what you can say about it.
- Merging DEL and ETL: New axiomatization results ($\langle !P\rangle\top \leftrightarrow P$)
- Part of larger discussion on logics of model change: belief revision, preference change
- Definability of protocols: where do the protocols come from? What do the agents know about the protocol?

Conclusions: Logics of Rational Agency

- What's going on in the area:
  www.illc.uva.nl/wordpress

- Upcoming Workshop: Logic and Intelligent Interaciton
  ai.stanford.edu/∼epacuit/LaII

- Upcoming special issue of the Journal of Logic, Language and Information edited by J. van Benthem and EP.

Thank You!