Logics of Knowledge and Action for Social Software

Eric Pacuit

May 18, 2007

ILLC, University of Amsterdam

staff.science.uva.nl/∼epacuit

epacuit@science.uva.nl

Introduction and Motivation

We are interested in logics for reasoning about rational agents interacting in some (game-theoretic) situation.

Introduction and Motivation

We are interested in logics for reasoning about rational agents interacting in some (game-theoretic) situation.

Motivation

- Social Software

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

- Refine and test our intuitions about interactive situations

- Verify properties of social procedures
  - *Refine existing social procedures or suggest new ones*

Motivating 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.

Motivating 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.*

Motivating 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?

Yes, but why?

Test Bob's information: the procedure forces that

Bob *does not* know that Ann knows that he knows about the talk.

Yes, but why?

Test Bob's information: the procedure forces that

Bob *does not* know that Ann knows that he knows about the talk.

Check Bob's preferences: does Bob *want* to be there?

Yes, but why?

Test Bob's information: the procedure forces that

Bob *does not* know that Ann knows that he knows about the talk.

Check Bob's preferences: does Bob *want* to be there?

Are Ann and Bob part of the same research group? If Ann is a leader, then a direct request from her would be interpreted as a command.

Plan

- Introduction
- Verifying Social Procedures
- Logics for Information Change
  - Dynamic Epistemic Logics, Epistemic Temporal Logics
- Combining Logics is Hard!
- Conclusion

Plan

- Introduction
- Verifying Social Procedures
- Logics for Information Change
  - Dynamic Epistemic Logics, Epistemic Temporal Logics
- Combining Logics is Hard!
- Conclusion

ETL or DEL?

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.

ETL or DEL?

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.

DEL 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.

The ‘Playground’

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

The ‘Playground’

- 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’

- 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’

- 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’

- 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’

- 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 $\mathsf{FinPre}_{-\epsilon}(\mathcal{H}) = \mathsf{FinPre}(\mathcal{H}) - \{\epsilon\}$.

The ‘Playground’

Definition 1 *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* $\mathsf{FinPre}(\mathcal{H}) \subseteq \mathcal{H}$.

The ‘Playground’

Definition 1 *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 2 *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* $\mathcal{H}$*.*

The ‘Playground’

Definition 1 *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 2 *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

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

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:

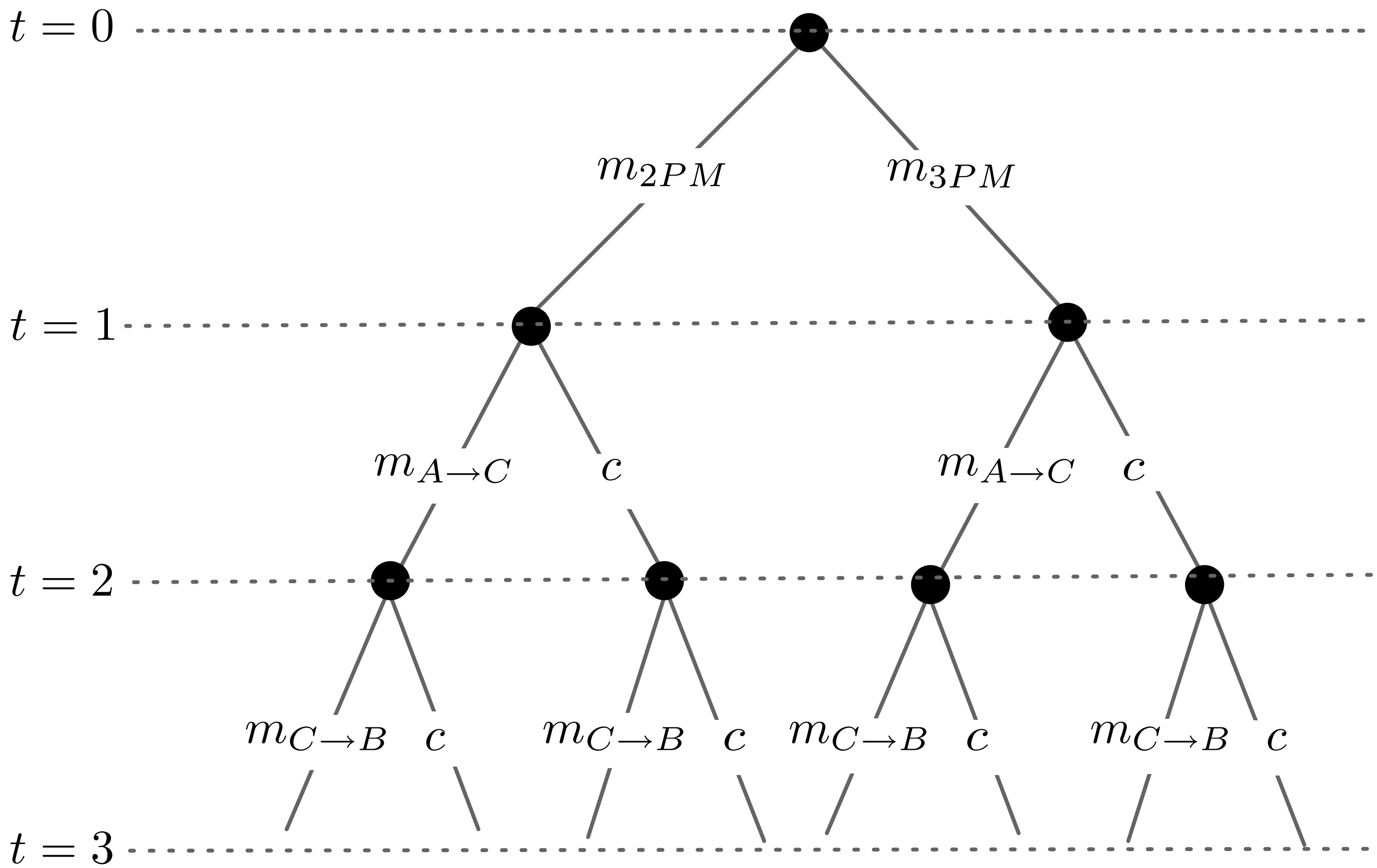
$$H, t \models \phi$$

Truth in a Model

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

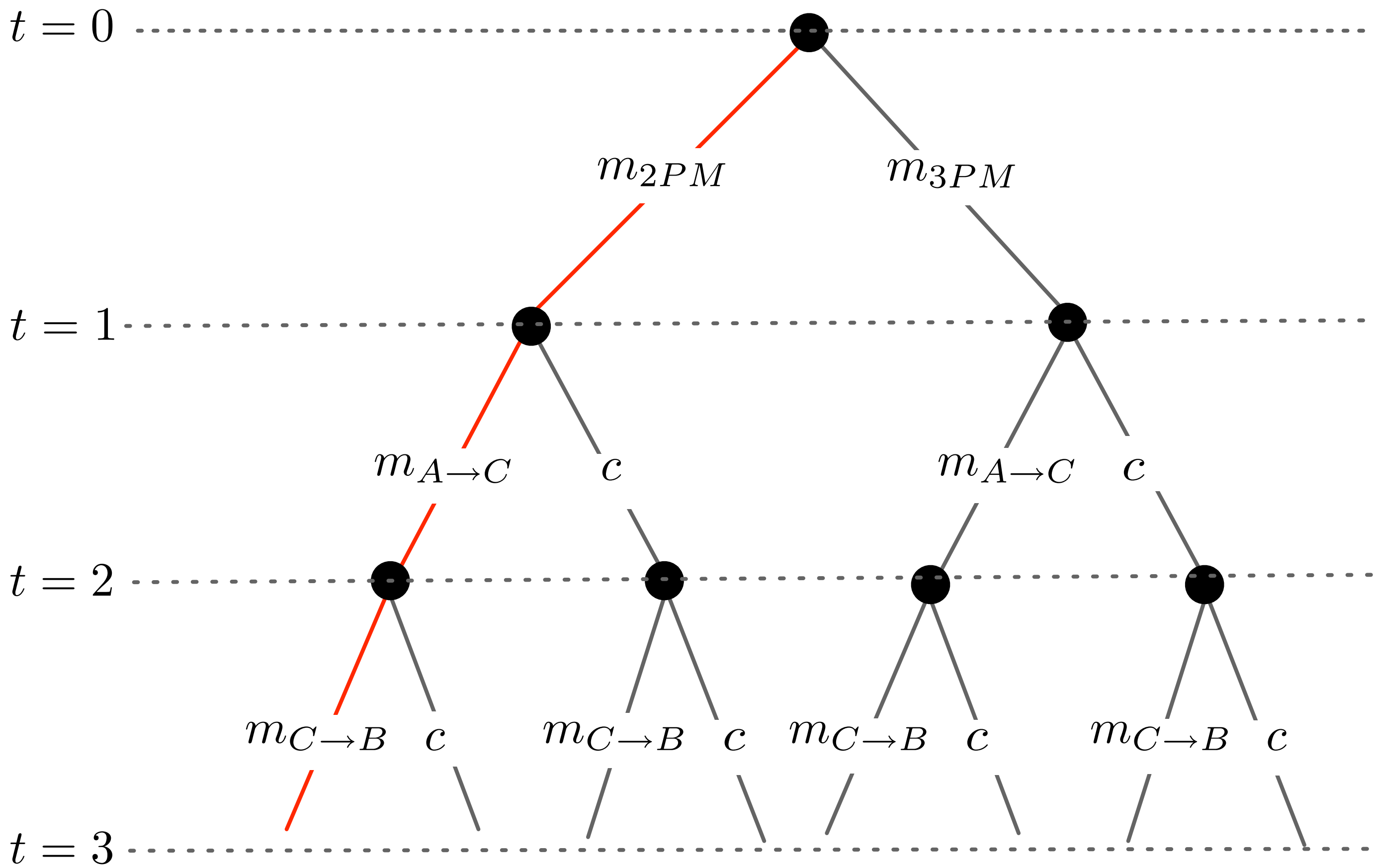
where $\sim_*$ is the reflexive transitive closure of the union of the $\sim_i$.

Revisiting the Example



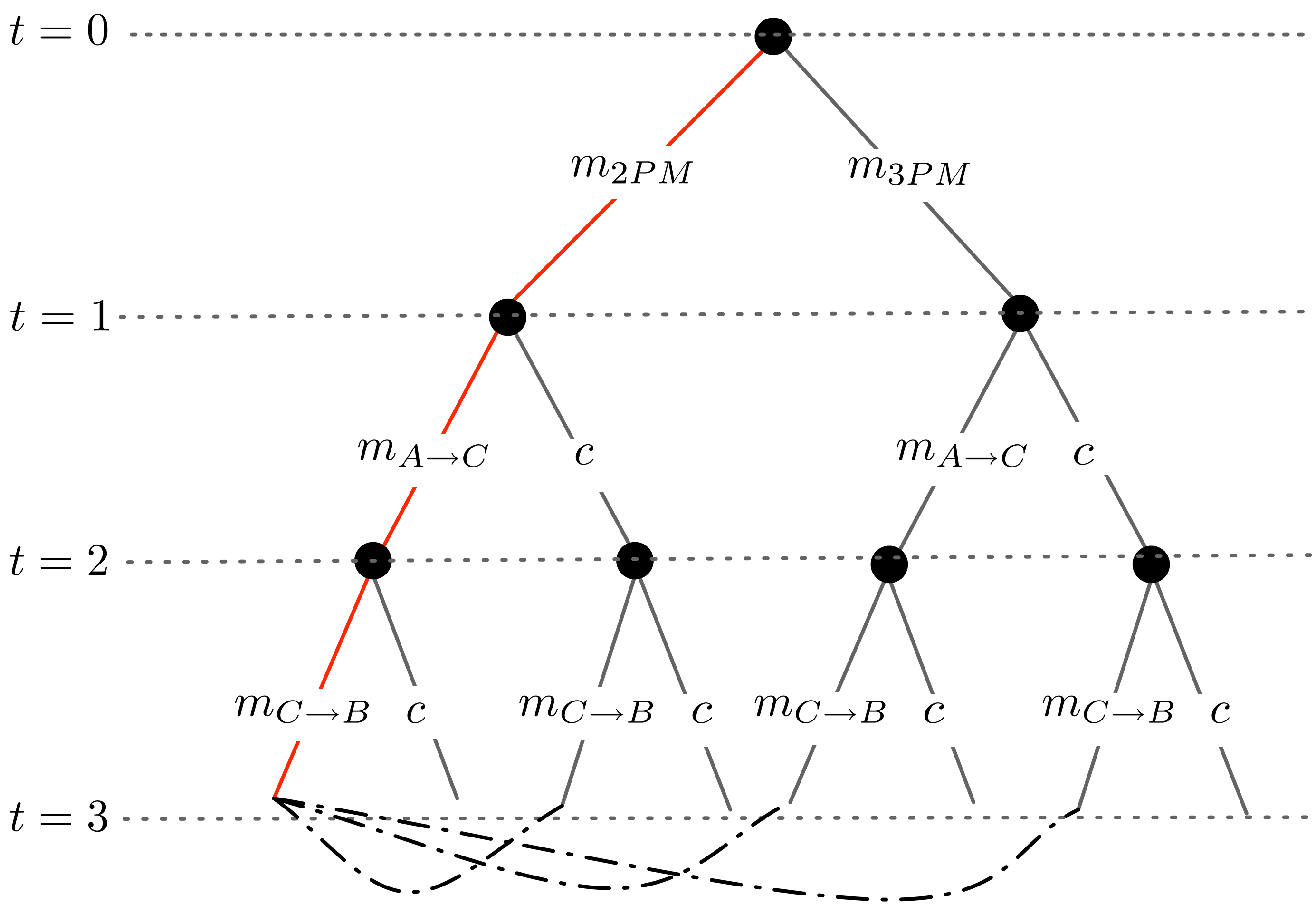
Full Event Tree

Revisiting the Example



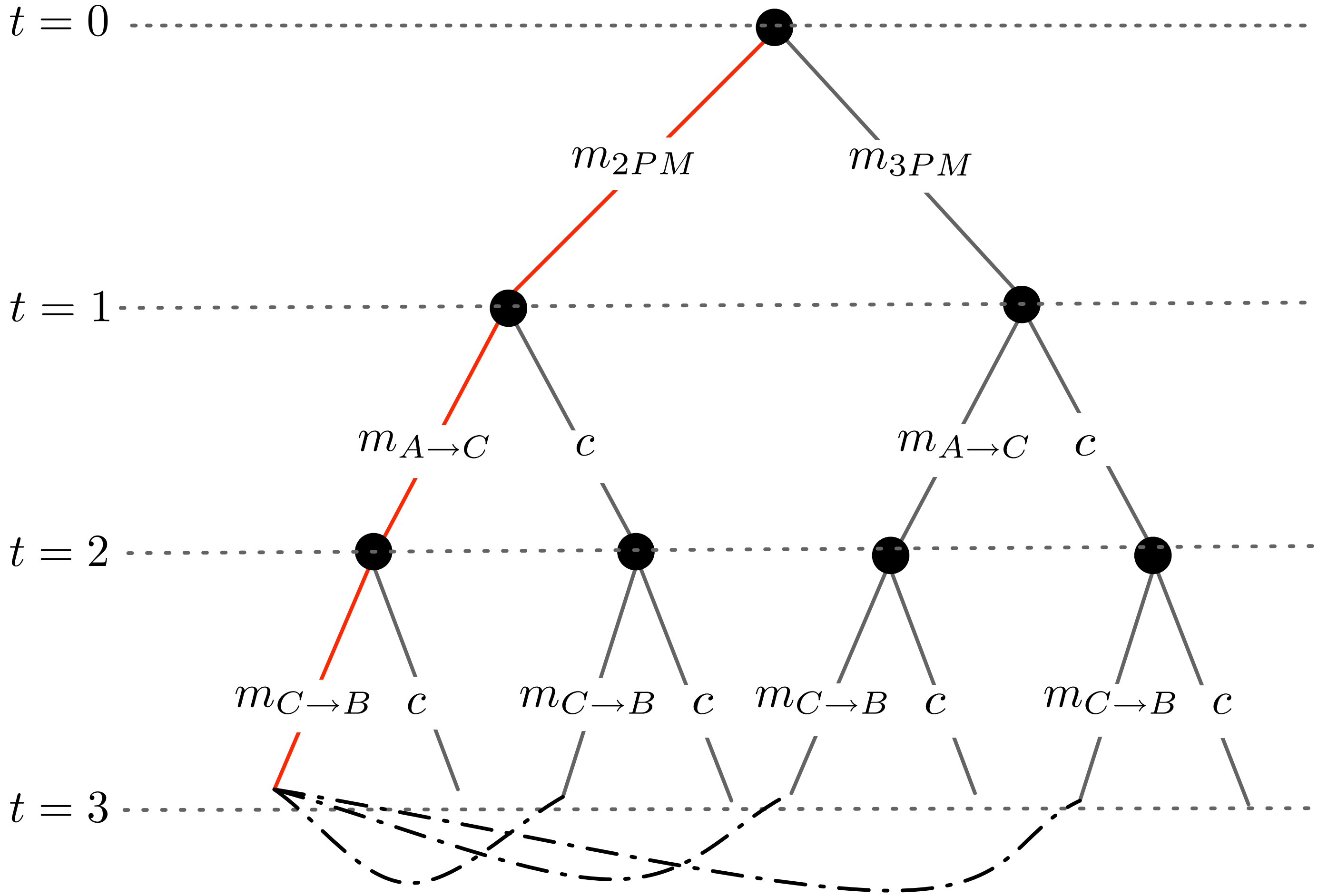
Full Event Tree

Revisiting the Example



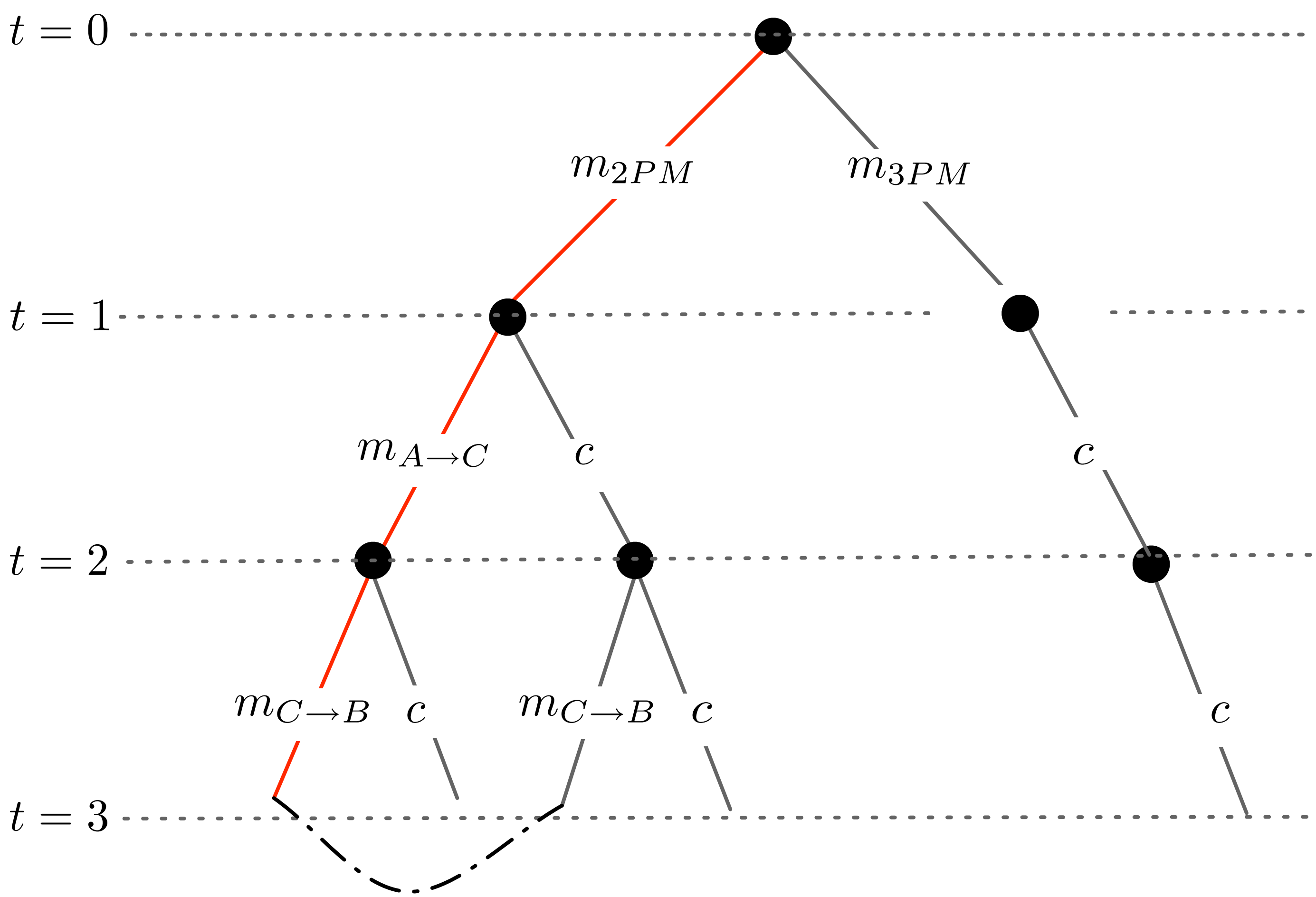
Bob's uncertainty: $H, 3 \models \neg K_B P_{2PM}$

Revisiting the Example



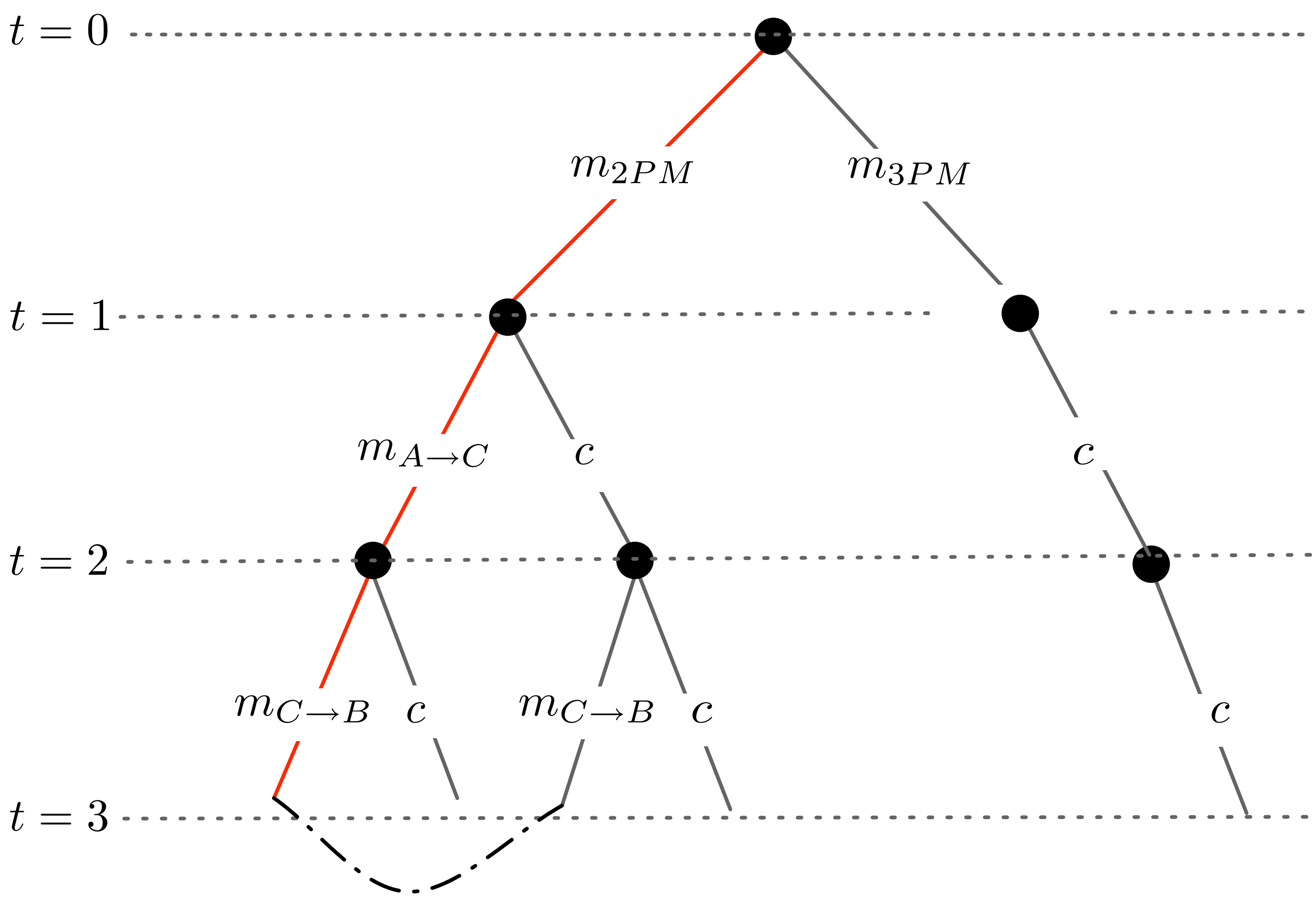
Two assumptions: Ann and Bob do not lie.

Revisiting the Example



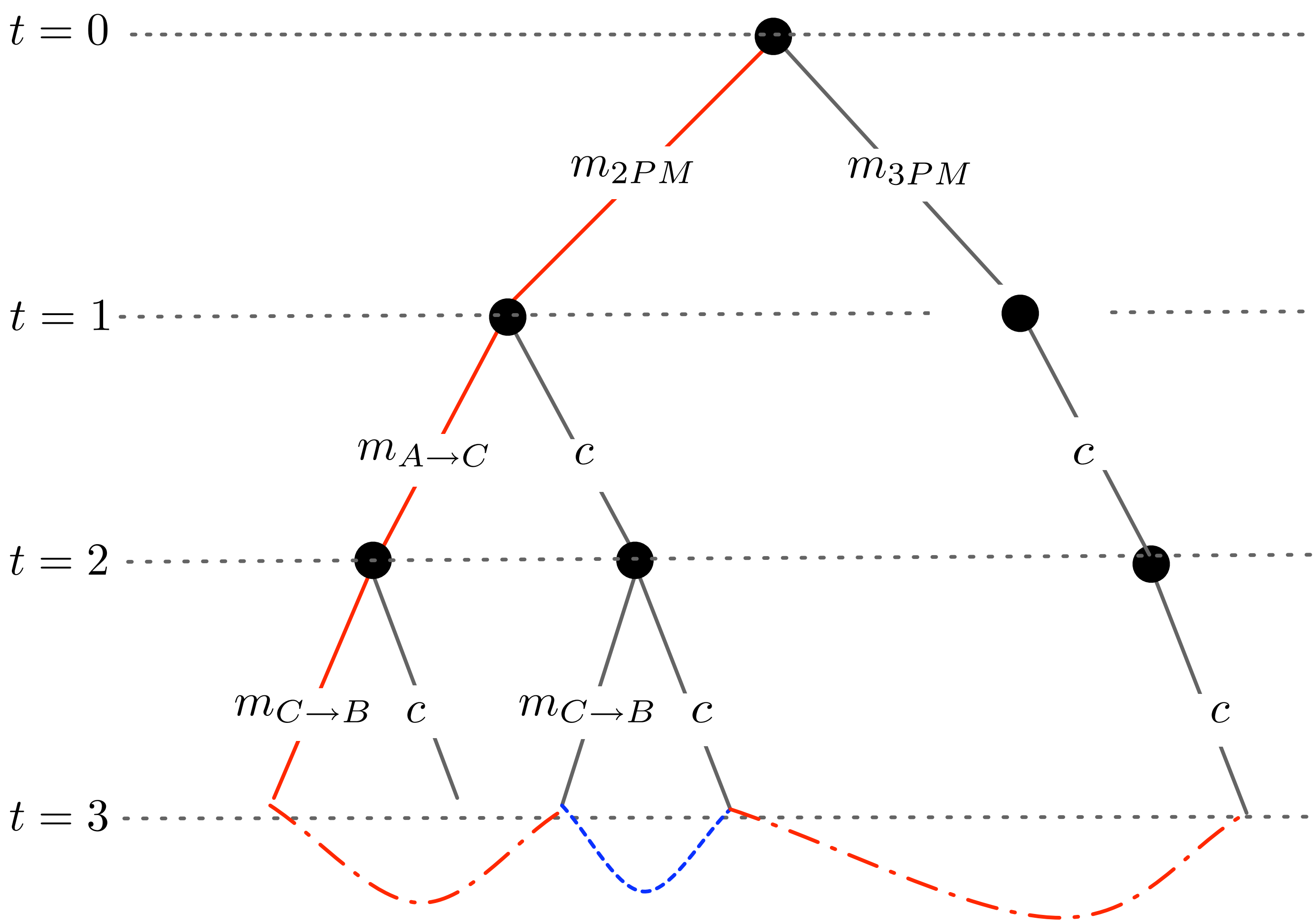
Two assumptions: Ann and Bob do not lie.

Revisiting the Example



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

Revisiting the Example



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

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 $\mathsf{len}(H) = \mathsf{len}(H')$.

Very Quick Intro to DEL

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

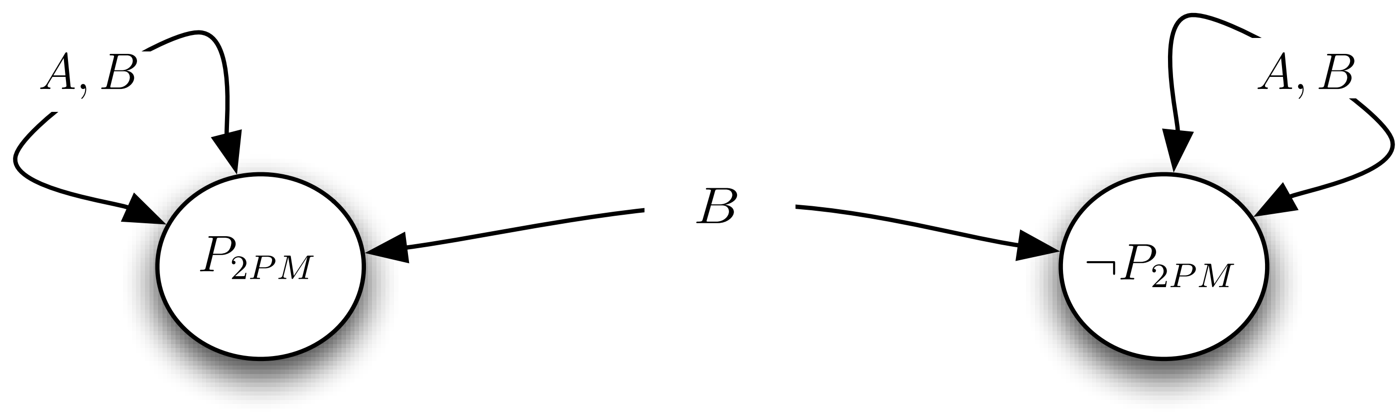
An **event model** is a tuple $\mathbb{A} = \langle A, S, Pre \rangle$.

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 $w \in V(p)$

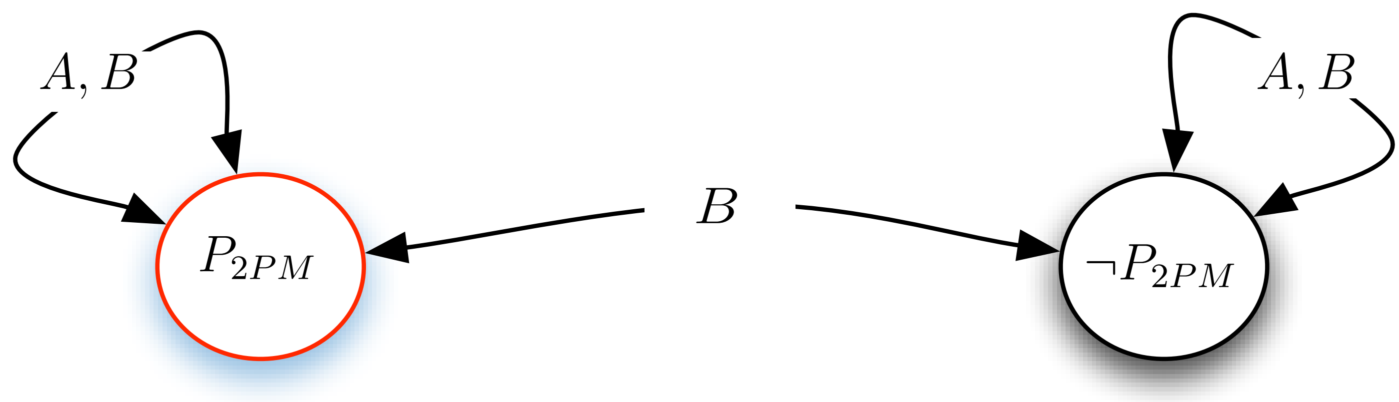
$\mathcal{M}, w \models [\mathsf{A}, a]\phi$ iff $\mathcal{M}, w \models Pre(a)$ implies $\mathcal{M} \otimes A, (w, a) \models \phi$.

Revisiting the Example, again



Initial Kripke Structure

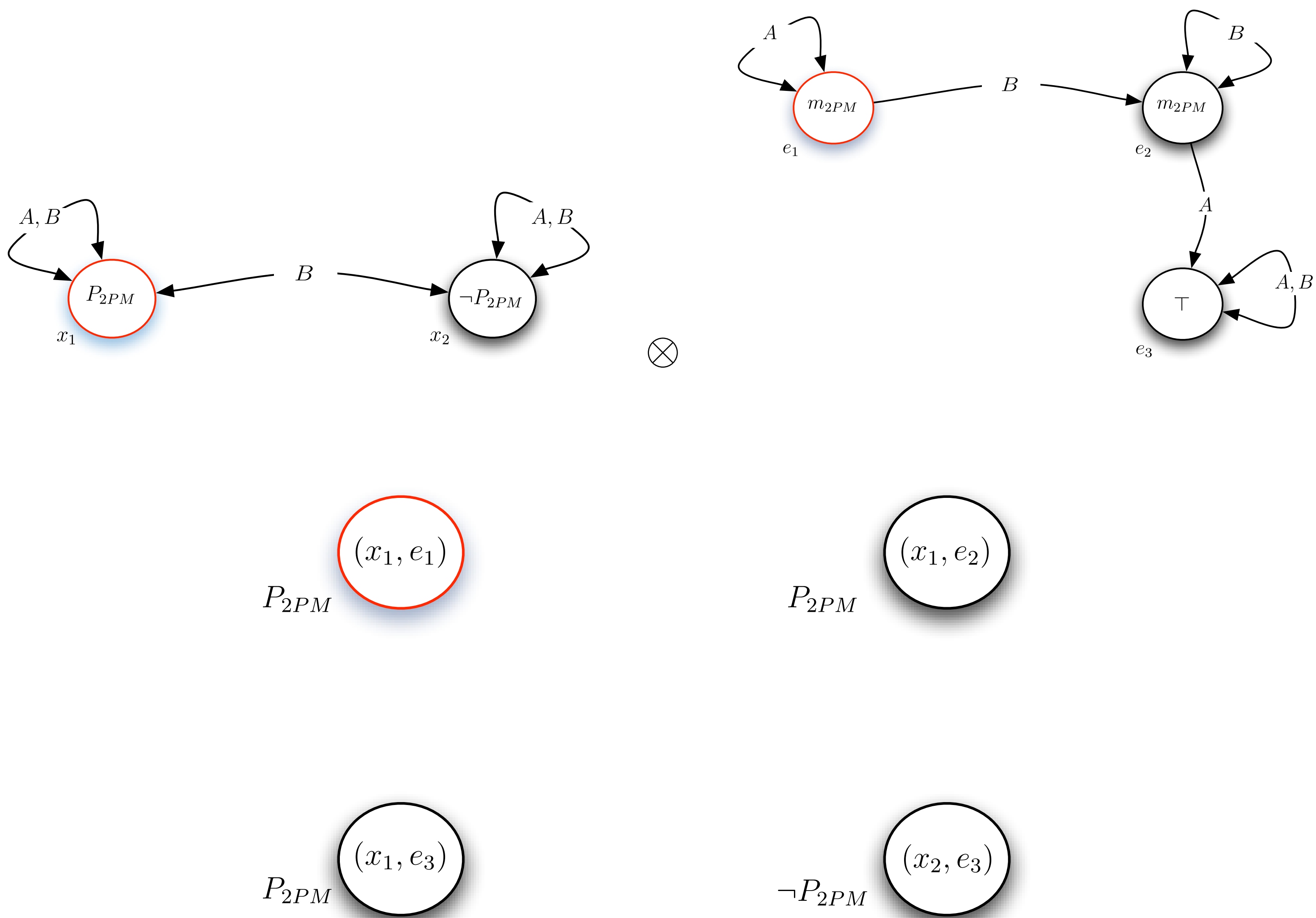
Revisiting the Example, again



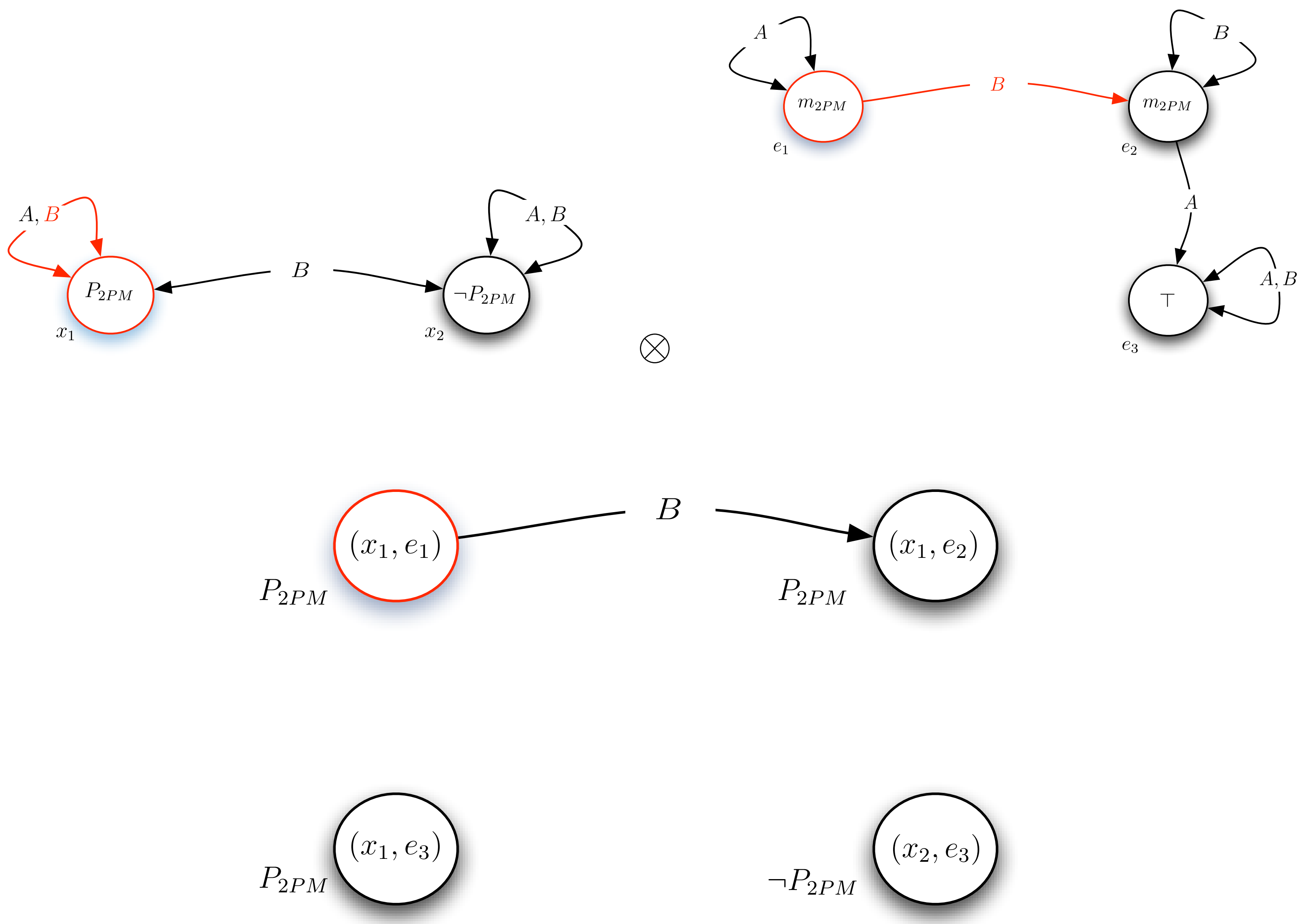
Initial Kripke Structure

- $K_A P_{2PM}$
- $\neg K_B P_{2PM}$
- $K_A \neg K_B P_{2PM}$
- $\ldots$

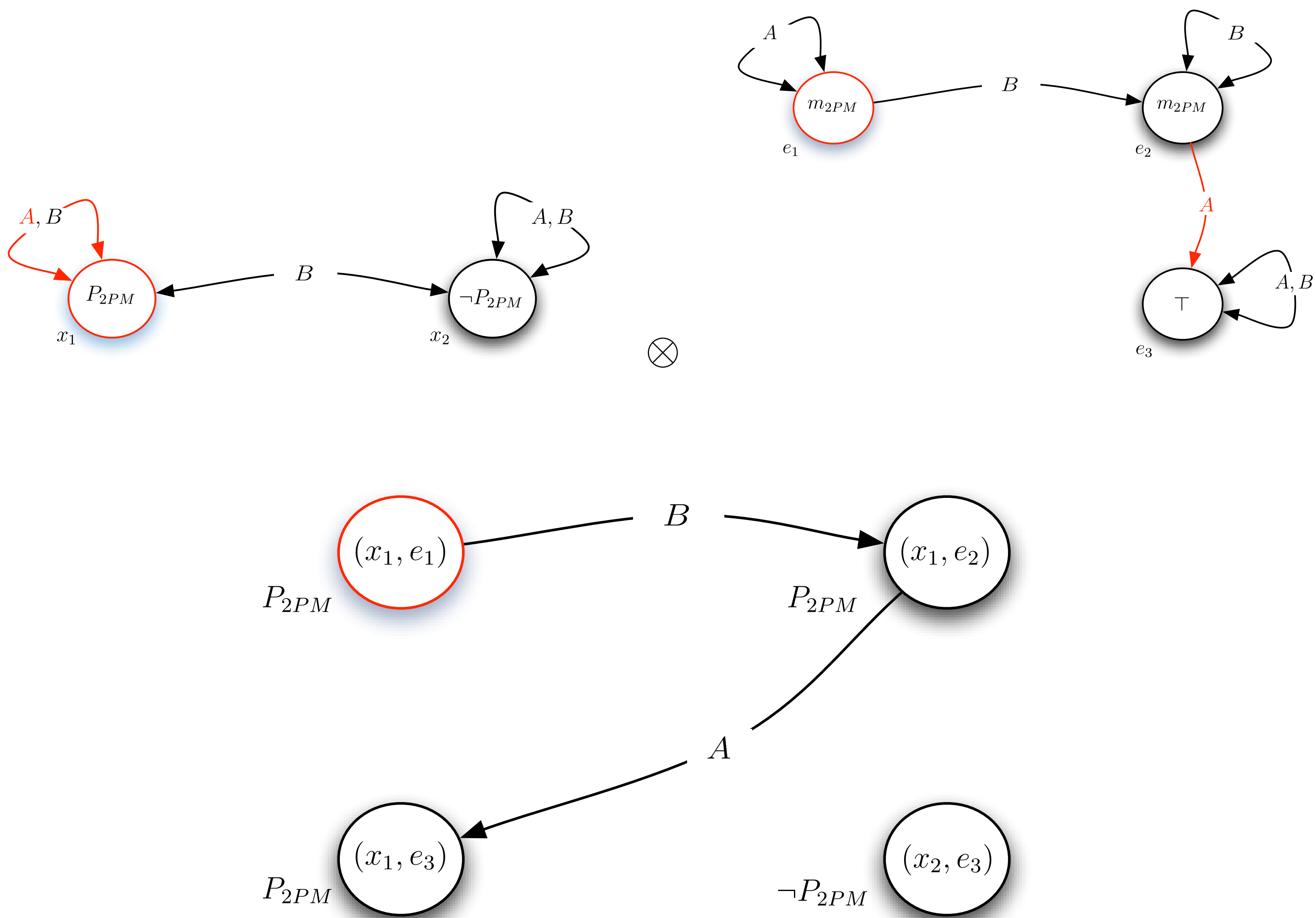
Revisiting the Example, again



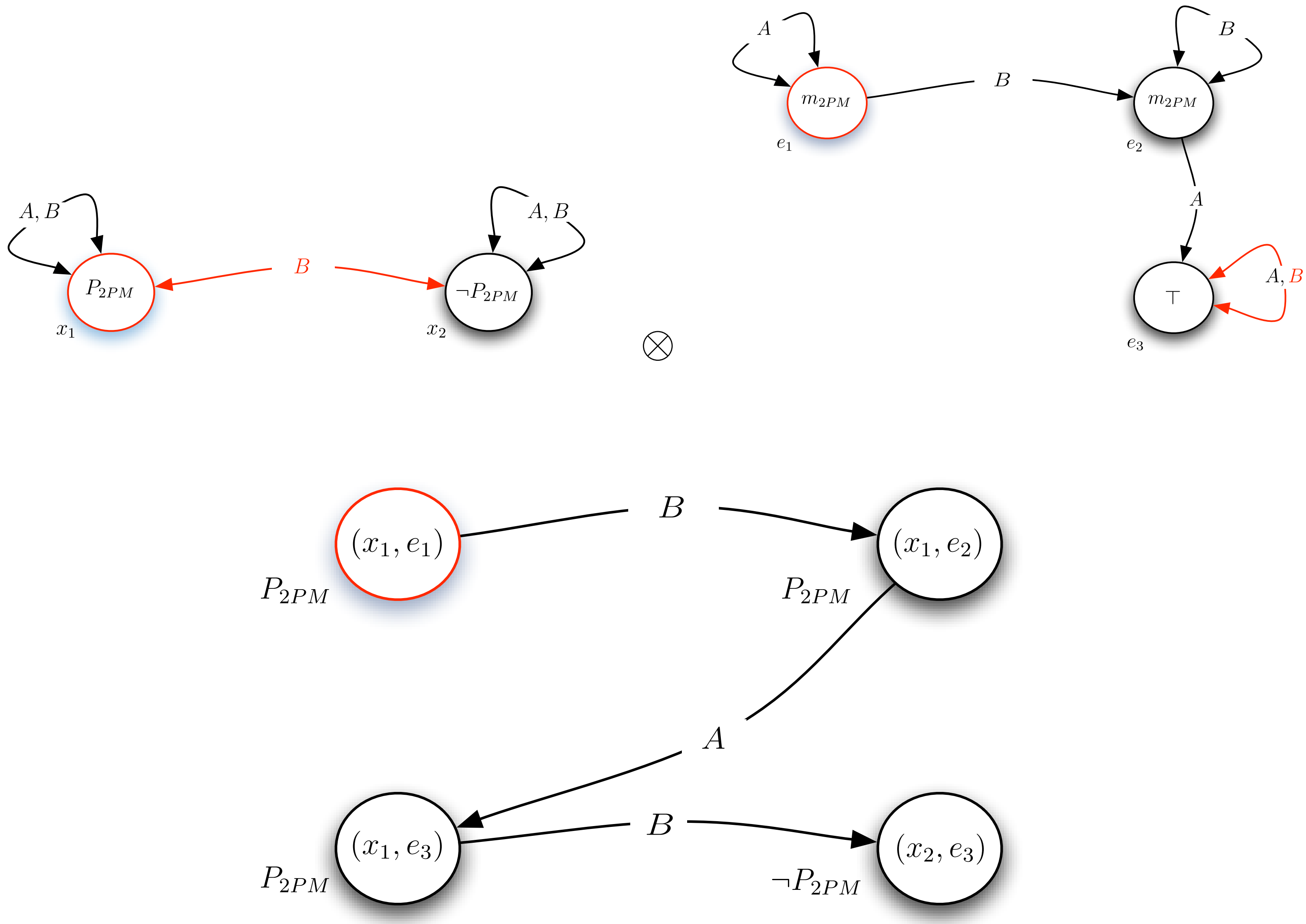
Revisiting the Example, again



Revisiting the Example, again



Revisiting the Example, again

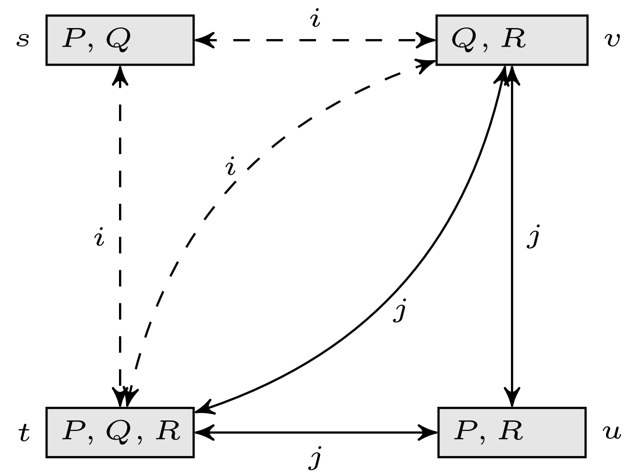


DEL *and* ETL

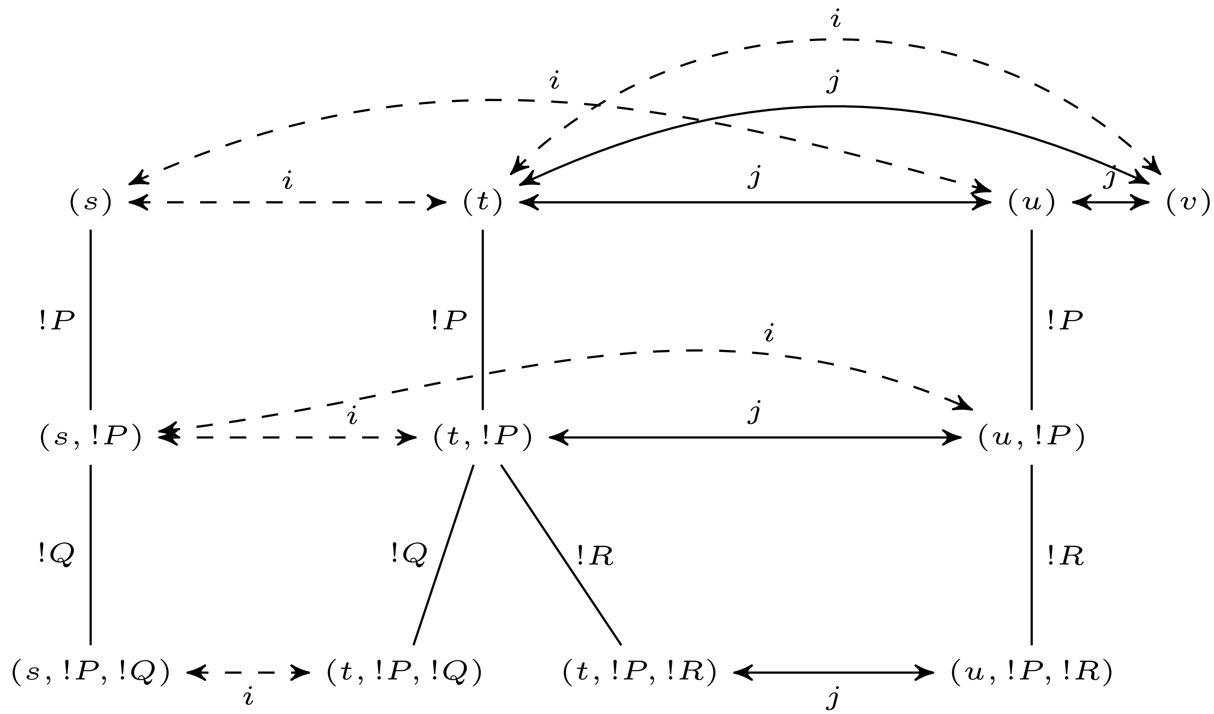
Observation: by repeatedly updating an epistemic model with event models, the machinery of DEL in effect creates ETL models.

Let M be an epistemic model, and $\mathcal{E}$ a DEL protocol. The ETL model generated by M and $\mathcal{E}$, $\mathsf{Forest}M\mathcal{E}$, represents all possible evolutions of the system obtained by updating M with sequences from $\mathcal{E}$.

Proposition For any formula $\phi \in \mathcal{L}_{DEL}$, $M, w \models \phi$ iff $\mathsf{Forest}M\mathcal{DEL}, (w) \models \phi^{\#}$.



$$\mathcal{E} = \{!P, !P!Q, !P!R\}$$



New Questions

Theorem Let $\mathcal{DEL}$ be the class of *all* DEL protocols. A model is in $\mathbb{F}(\mathcal{DEL})$ iff it satisfies synchronicity, perfect recall, local uniform no miracles, and local bisimulation invariance.

New Questions

Theorem Let $\mathcal{DEL}$ be the class of *all* DEL protocols. A model is in $\mathbb{F}(\mathcal{DEL})$ iff it satisfies synchronicity, perfect recall, local uniform no miracles, and local bisimulation invariance.

Correspondence theory w.r.t. to generated ETL trees and Event Models.

New Questions

Theorem Let $\mathcal{DEL}$ be the class of *all* DEL protocols. A model is in $\mathbb{F}(\mathcal{DEL})$ iff it satisfies synchronicity, perfect recall, local uniform no miracles, and local bisimulation invariance.

Correspondence theory w.r.t. to generated ETL trees and Event Models.

Theorem Sound and complete axiomatization of $\mathbb{F}\mathcal{PAL}$.

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

Plan

- Introduction
- Verifying Social Procedures
- Logics for Information Change
  - Dynamic Epistemic Logics, Epistemic Temporal Logics
- Combining Logics is Hard!
- Conclusion

Living at the Edge of Decidability

- Assumptions about the structure of the "playground" (eg. tree, finite tree).
- Assumptions about the players (eg. perfect recall).
- What can the language express (eg. common knowledge, arbitrary future).

Living at the Edge of Decidability

- Assumptions about the structure of the "playground" (eg. tree, finite tree).
- Assumptions about the players (eg. perfect recall).
- What can the language express (eg. common knowledge, arbitrary future).

Theorem(s) [Halpern & Vardi] Certain idealizations can lead to high complexity (Π^1_1-completeness).

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

See also,

J. van Benthem and E. Pacuit. *The Tree of Knowledge in Action: Towards a Common Perspective*. forthcoming AiML 2006.

Plan

- Introduction
- Verifying Social Procedures
- Logics for Information Change
  - Dynamic Epistemic Logics, Epistemic Temporal Logics
- Combining Logics is Hard!
- Conclusion

Conclusion

Logic *and* Game Theory (not Logic *instead of* Game Theory)

- *Who* are we trying to model? I.e., what does it mean to be a *rational* agent?
- Which aspects of social situations should we focus on? Knowledge, Beliefs, Group Knowledge, Preferences, Desires, Ability, Actions, etc.
- One grand system, or many smaller systems that loosely "fit" together?

See a forthcoming special issue of JoLLI on logics for social, interactive situations edited by Johan van Benthem and EP.

The sciences do not try to explain, they hardly even try to interpret, they mainly make models. By a model is meant a mathematical construct which, with the addition of certain verbal interpretations, describes observed phenomena. The justification of such a mathematical construct is solely and precisely that it is expected to work. — Johann Von Neumann

Dankje Wel!