Introduction to Formal Epistemology

Lecture 3

Eric Pacuit and Rohit Parikh

August 14, 2007

✓ Introduction, Motivation and Basic Epistemic Logic

✓ Other models of Knowledge, Knowledge in Groups and Group Knowledge

Lecture 3: Adding Dynamics, Reasoning about Knowledge in Games

Lecture 4: Logical Omniscience and Other Problems

Lecture 5: Reasoning about Knowledge in the Context of Social Software

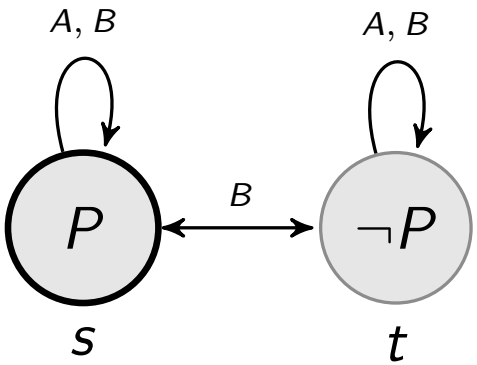
Many of the examples we have discussed are *dynamic* — they involve announcements, actions, etc.

How should we add this *dynamics* to our models?

Many of the examples we have discussed are *dynamic* — they involve announcements, actions, etc.

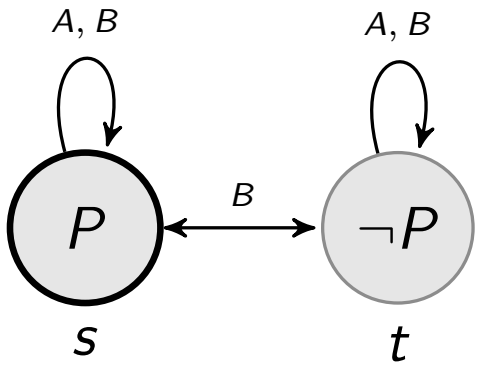
How should we add this *dynamics* to our models?

Public Announcement



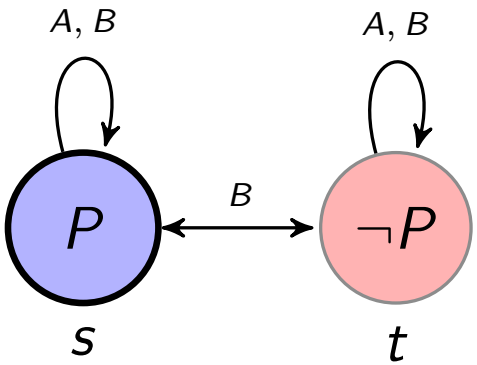
P means "The talk is at 2PM".

Public Announcement



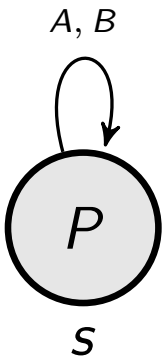
What happens if Ann publicly announces P?

Public Announcement



What happens if Ann publicly announces P?

Public Announcement



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

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.

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

- $[\psi]\varphi$ is intended to mean "After publicly announcing ψ, φ is true".
- $[P]K_iP$: "After publicly announcing P, agent i knows P"
- $[\neg K_iP]CP$: "After announcing that agent i does not know P, then P is common knowledge"
- $[\neg K_iP]K_iP$: "after announcing i does not know P, then i knows P. "

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

- $[\psi]\varphi$ is intended to mean "After publicly announcing ψ, φ is true".
- $[P]K_iP$: "After publicly announcing P, agent i knows P"
- $[\neg K_iP]CP$: "After announcing that agent i does not know P, then P is common knowledge"
- $[\neg K_iP]K_iP$: "after announcing i does not know P, then i knows P. "

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

- $[\psi]\varphi$ is intended to mean "After publicly announcing ψ, φ is true".
- $[P]K_iP$: "After publicly announcing P, agent i knows P"
- $[\neg K_iP]CP$: "After announcing that agent i does not know P, then P is common knowledge"
- $[\neg K_iP]K_iP$: "after announcing i does not know P, then i knows P. "

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

- $[\psi]\varphi$ is intended to mean "After publicly announcing ψ, φ is true".
- $[P]K_iP$: "After publicly announcing P, agent i knows P"
- $[\neg K_iP]CP$: "After announcing that agent i does not know P, then P is common knowledge"
- $[\neg K_iP]K_iP$: "after announcing i does not know P, then i knows P. "

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

- $[\psi]\varphi$ is intended to mean "After publicly announcing ψ, φ is true".
- $[P]K_iP$: "After publicly announcing P, agent i knows P"
- $[\neg K_iP]CP$: "After announcing that agent i does not know P, then P is common knowledge"
- $[\neg K_iP]K_iP$: "after announcing i does not know P, then i knows P. "

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 $p \in \text{At}$, $V'(p) = V(p) \cap W'$

Public Announcement Logic

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

Public Announcement Logic

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

Public Announcement Logic

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

Public Announcement Logic

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

Public Announcement Logic

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

Public Announcement Logic

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

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

Public Announcement Logic

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

The situation is more complicated with common knowledge.

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

Public Announcement Logic

An announcement if successful if after it is publicly announced, it becomes common knowledge. $[\varphi]C\varphi$.

An announcement is unsuccessful if after it is publicly announced, it becomes false. $[\varphi]\neg\varphi$.

P means "it is raining."

$[P \wedge \neg KP]\neg(P \wedge \neg KP)$

Question: Which formulas are successful? unsuccessful?

Public Announcement Logic

An announcement if successful if after it is publicly announced, it becomes common knowledge. $[\varphi]C\varphi$.
An announcement is unsuccessful if after it is publicly announced, it becomes false. $[\varphi]\neg\varphi$.
P means "it is raining."
$[P \wedge \neg KP]\neg(P \wedge \neg KP)$
Question: Which formulas are successful? unsuccessful?

Public Announcement Logic

An announcement if successful if after it is publicly announced, it becomes common knowledge. $[\varphi]C\varphi$.
An announcement is unsuccessful if after it is publicly announced, it becomes false. $[\varphi]\neg\varphi$.
P means "it is raining."
$[P \wedge \neg KP]\neg(P \wedge \neg KP)$
Question: Which formulas are successful? unsuccessful?

Public Announcement Logic

An announcement if successful if after it is publicly announced, it becomes common knowledge. $[\varphi]C\varphi$.
An announcement is unsuccessful if after it is publicly announced, it becomes false. $[\varphi]\neg\varphi$.
P means "it is raining."
$[P \wedge \neg KP]\neg(P \wedge \neg KP)$
Question: Which formulas are successful? unsuccessful?

Dynamic Epistemic Logic

What about more complicated announcements?

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

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

Dynamic Epistemic Logic

What about more complicated announcements?

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

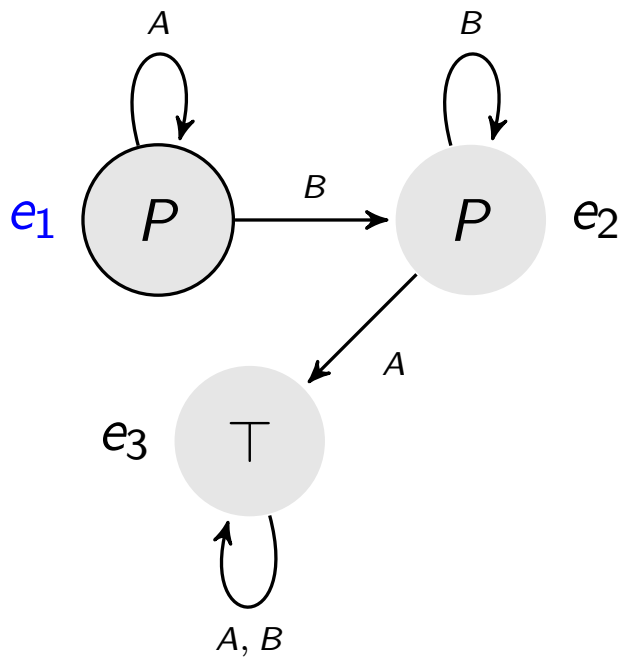
W. van der Hoek, H. van Ditmarsch and B. Kooi. *Dynamic Logic*. 2007.

Dynamic Epistemic Logic

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

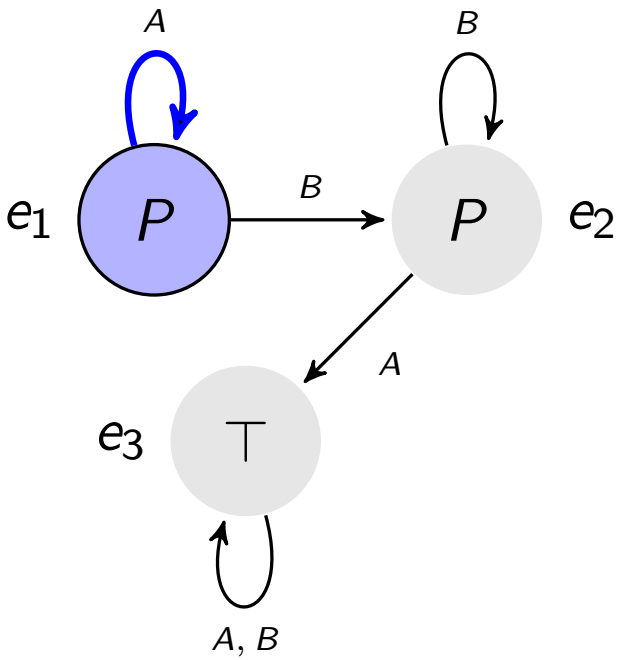
Dynamic Epistemic Logic

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



Dynamic Epistemic Logic

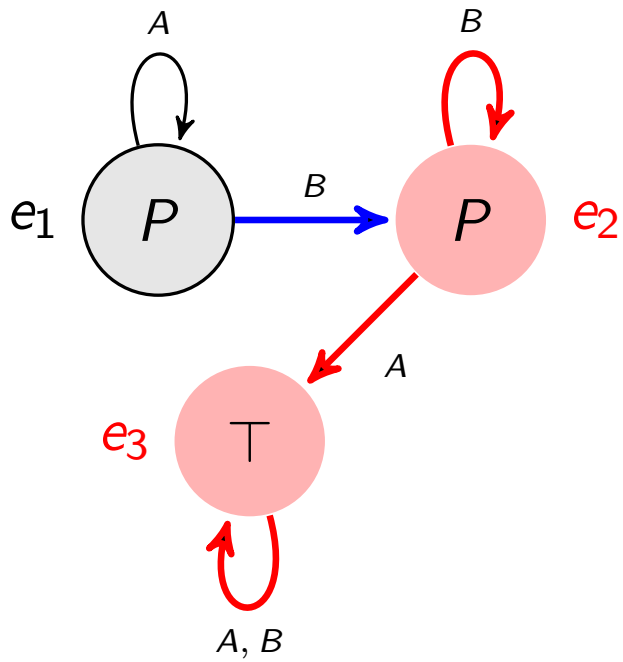
Recall the Ann and Bob example: Charles tells Bob that the talk is at 2PM.



Ann knows which event took place.

Dynamic Epistemic Logic

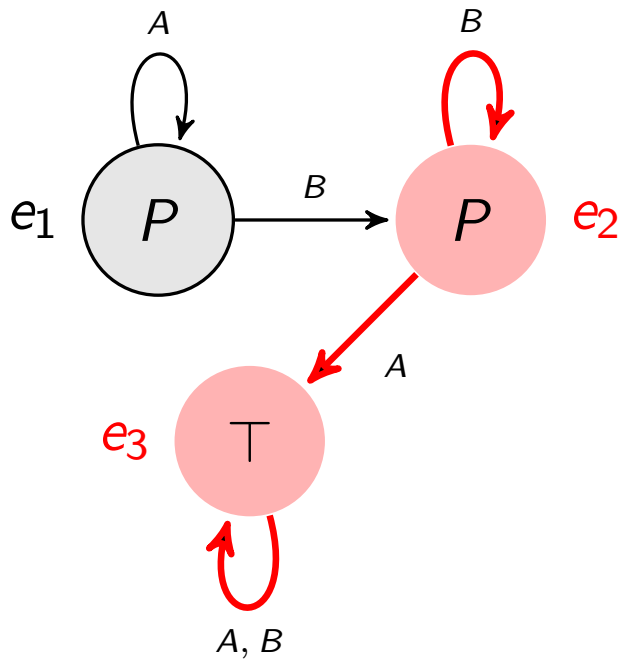
Recall the Ann and Bob example: Charles tells Bob that the talk is at 2PM.



Bob thinks a different event took place.

Dynamic Epistemic Logic

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



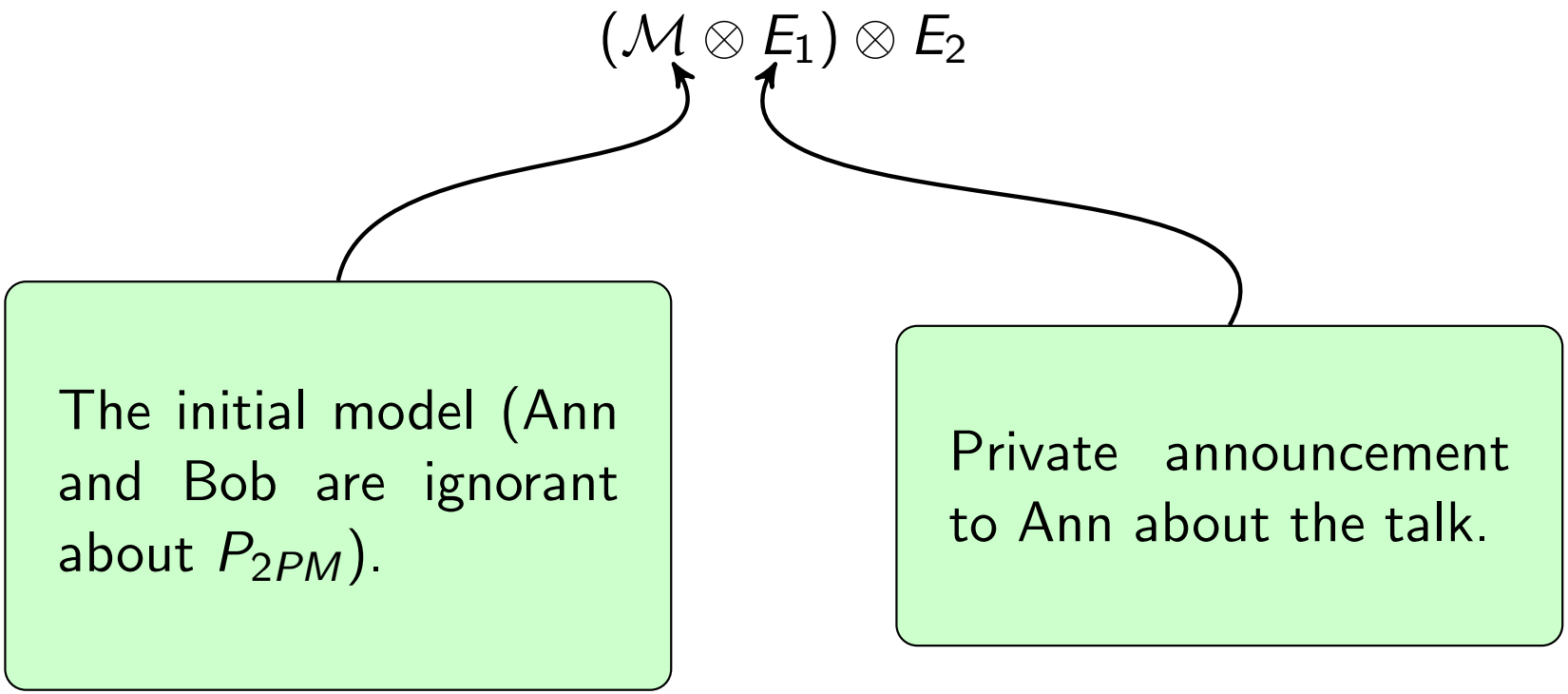
That is, Bob learns the time of the talk, but Ann learns nothing.

Dynamic Epistemic Logic

Dynamic Epistemic Logic

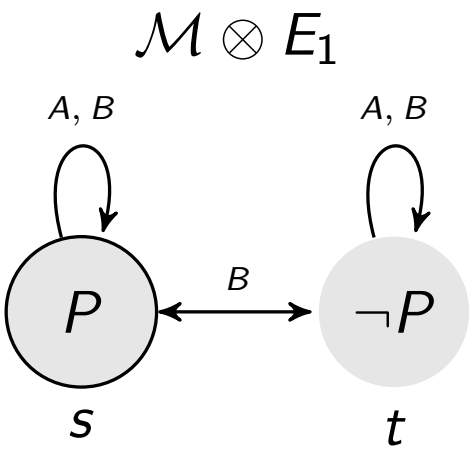
$$(\mathcal{M} \otimes E_1) \otimes E_2$$

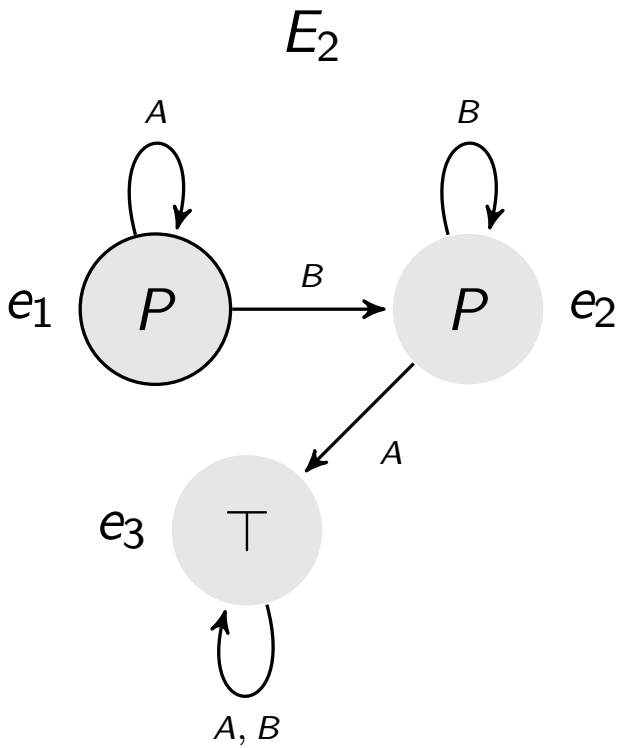
Dynamic Epistemic Logic



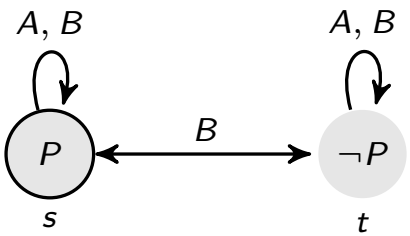
Product Update

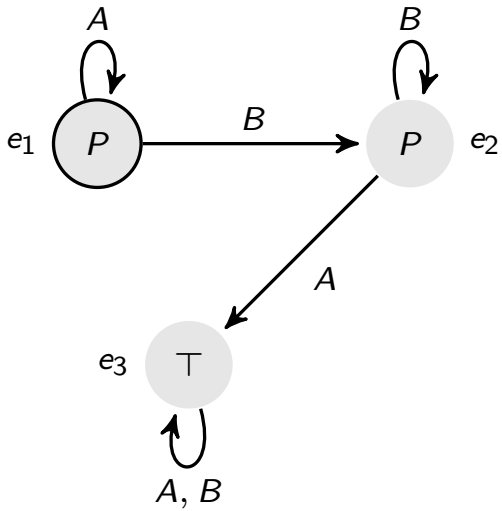
Product Update





Product Update

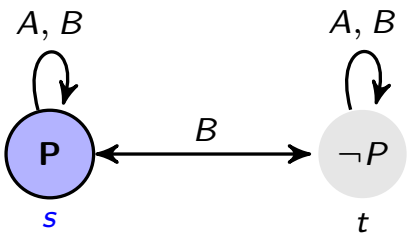


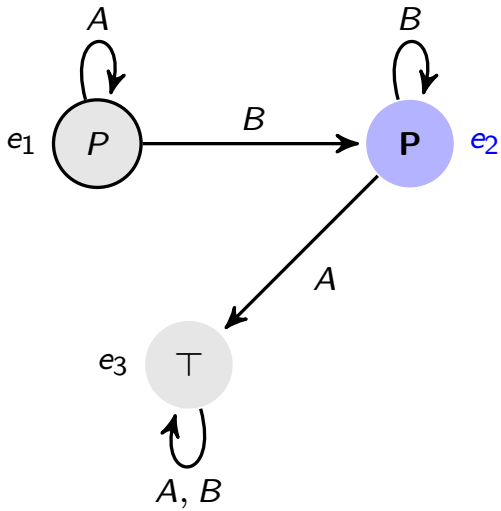


(s, e_1) P P (s, e_2)

(s, e_3) P $\neg P$ (t, e_3)

Product Update





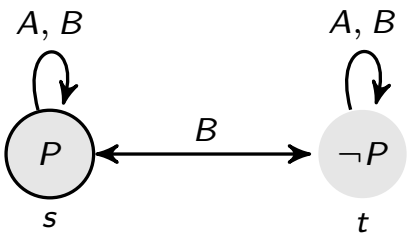
(s, e_1) P

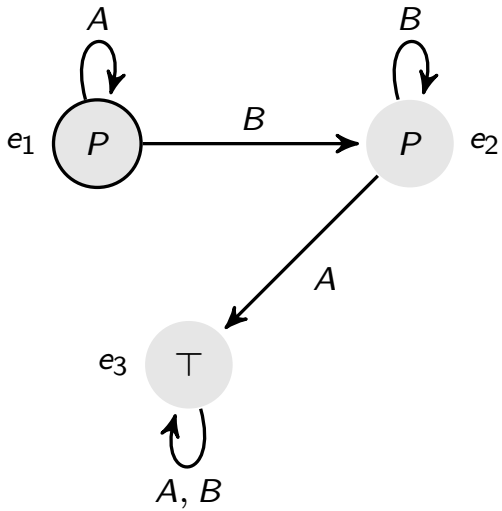
P (s, e_2)

(s, e_3) P

$\neg P$ (t, e_3)

Product Update

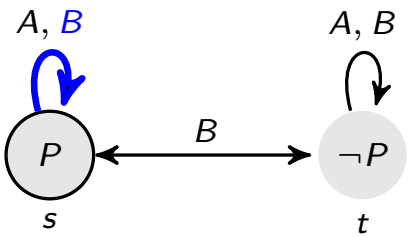


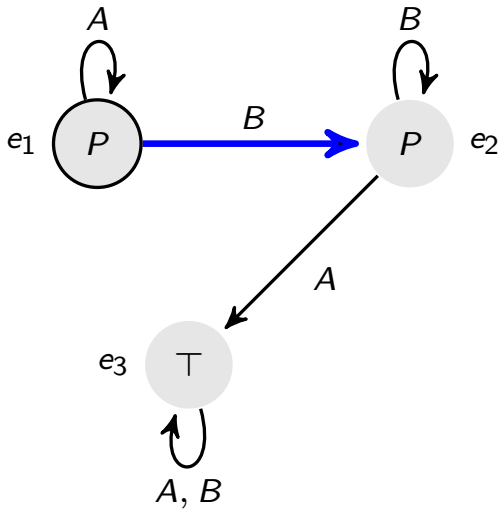


$(s, e_1) \models \neg K_B K_A K_B P$ (s, e_1) P P (s, e_2)

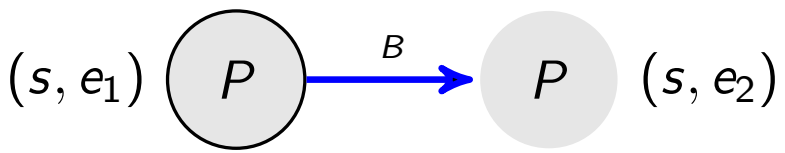
(s, e_3) P $\neg P$ (t, e_3)

Product Update

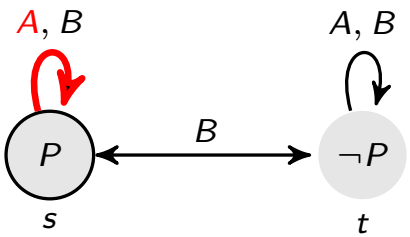


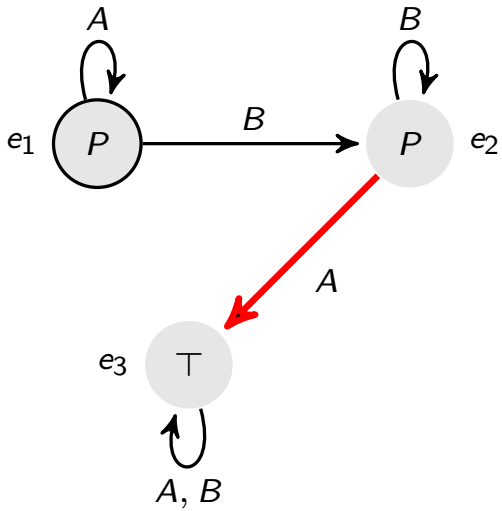


$(s, e_1) \models \neg K_B K_A K_B P$

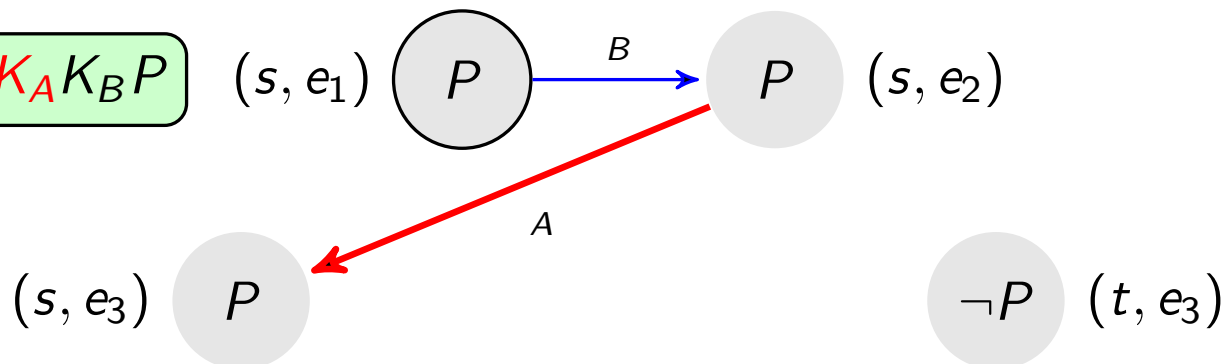


Product Update

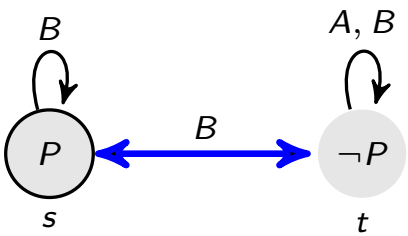


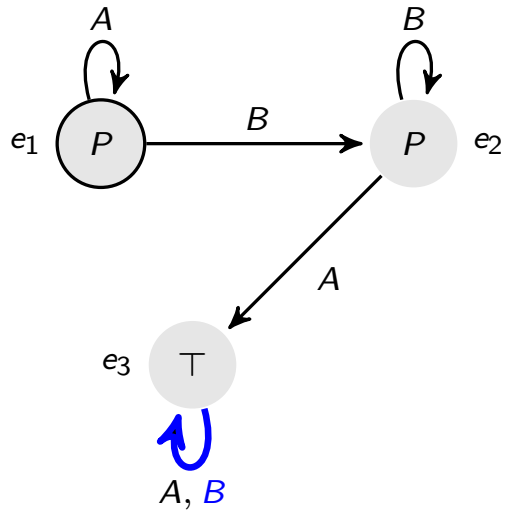


$(s, e_1) \models \neg K_B K_A K_B P$

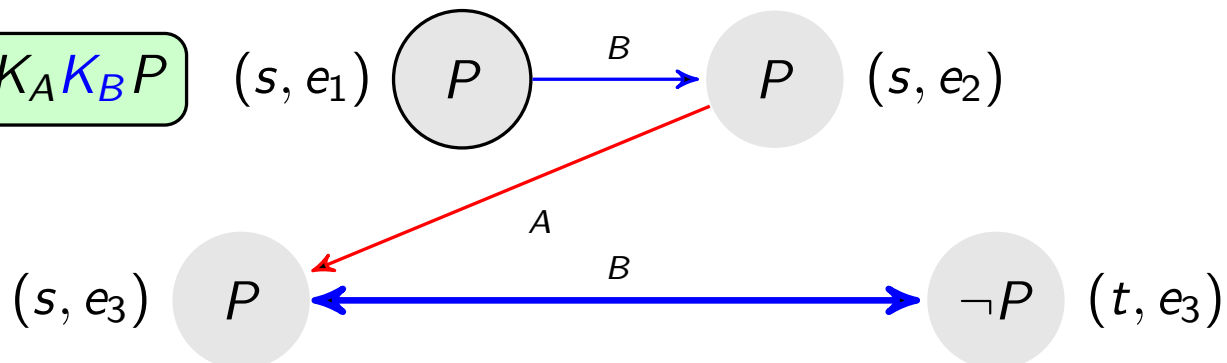


Product Update





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



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 $Pre : \mathcal{L} \rightarrow \wp(A)$.

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 $w \in V(p)$

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

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'
- $(w, a) \in V(p)$ iff $w \in V(p)$

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

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'
- $(w, a) \in V(p)$ iff $w \in V(p)$

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

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'
- $(w, a) \in V(p)$ iff $w \in V(p)$

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

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'
- $(w, a) \in V(p)$ iff $w \in V(p)$

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

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'
- $(w, a) \in V(p)$ iff $w \in V(p)$

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

DEL methodology: when describing a social situation, describe the initial situation, describe the event and provide a method for how events change a model that can be described in the formal language, then construct the event tree as needed.

Alternatively: 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: when describing a social situation, describe the initial situation, describe the event and provide a method for how events change a model that can be described in the formal language, then construct the event tree as needed.

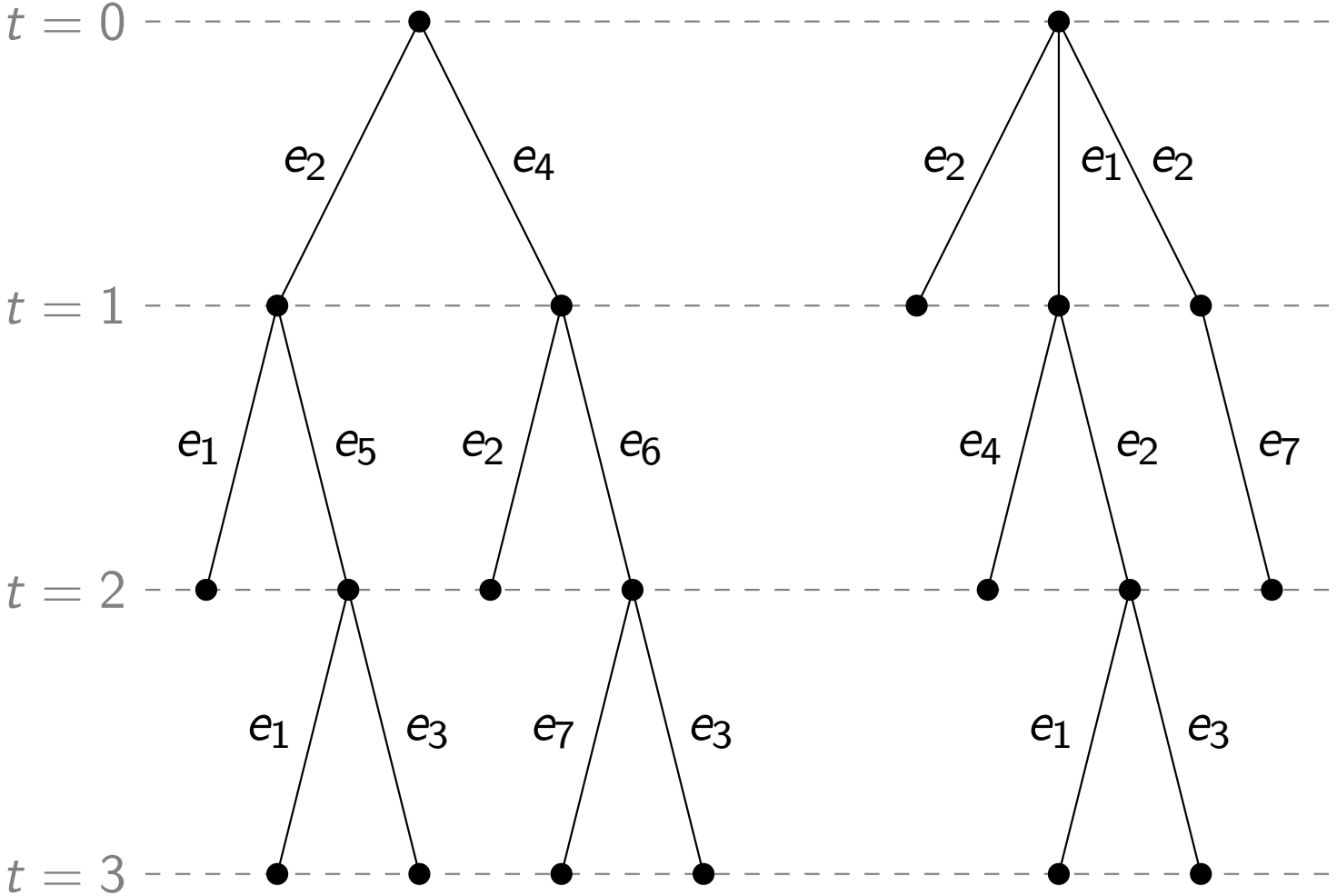
Alternatively: when describing a social situation, first write down all possible sequences of events, then at each moment write down the agents' uncertainty, from that infer how the agents' knowledge changes from one moment to the next.

Epistemic Temporal Logic

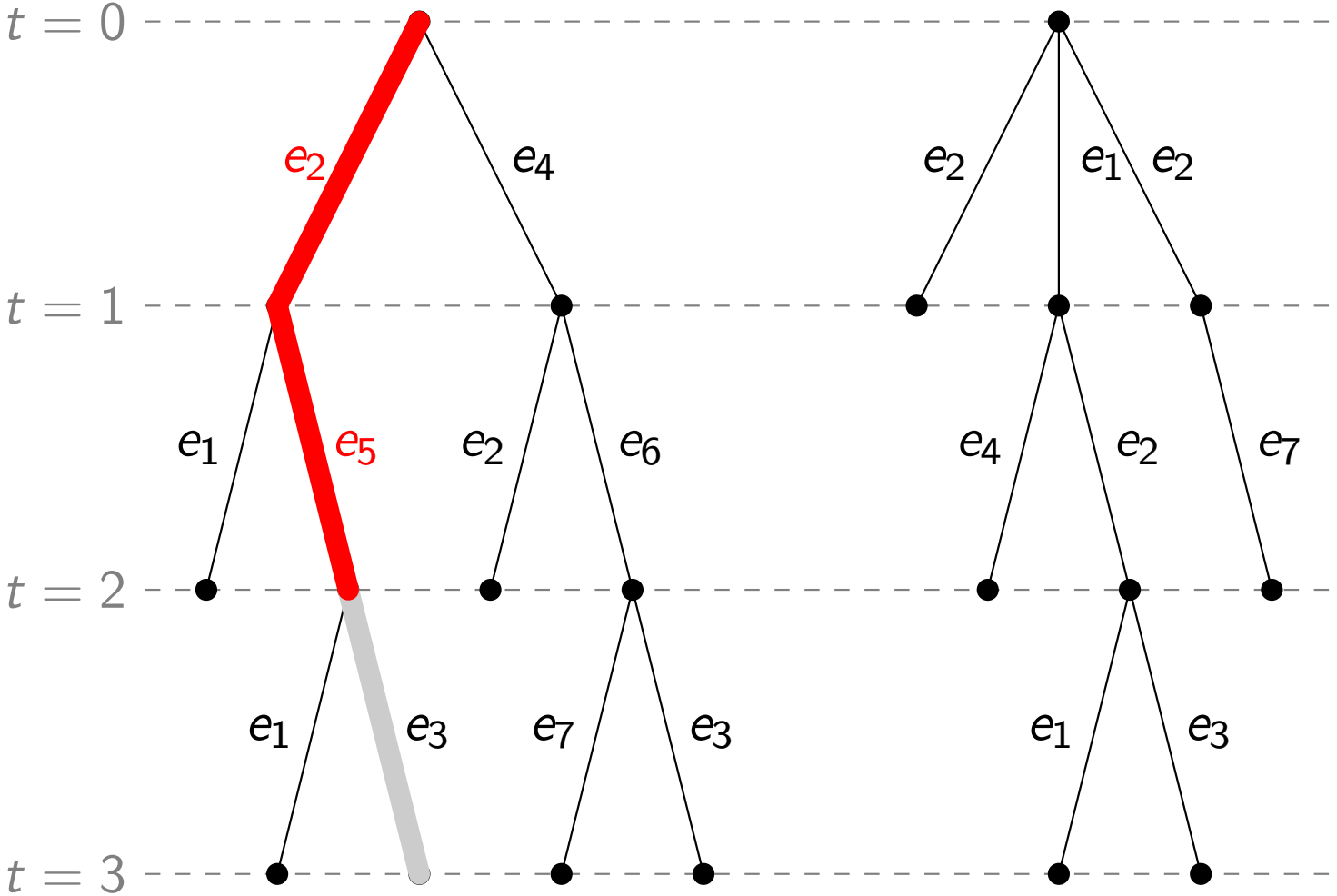
R. Parikh and R. Ramanujam. *A Knowledge Based Semantics of Messages*. *Journal of Logic, Language and Information*, 12: 453 – 467, 1985, 2003.

FHMV. Reasoning about Knowledge. MIT Press, 1995.

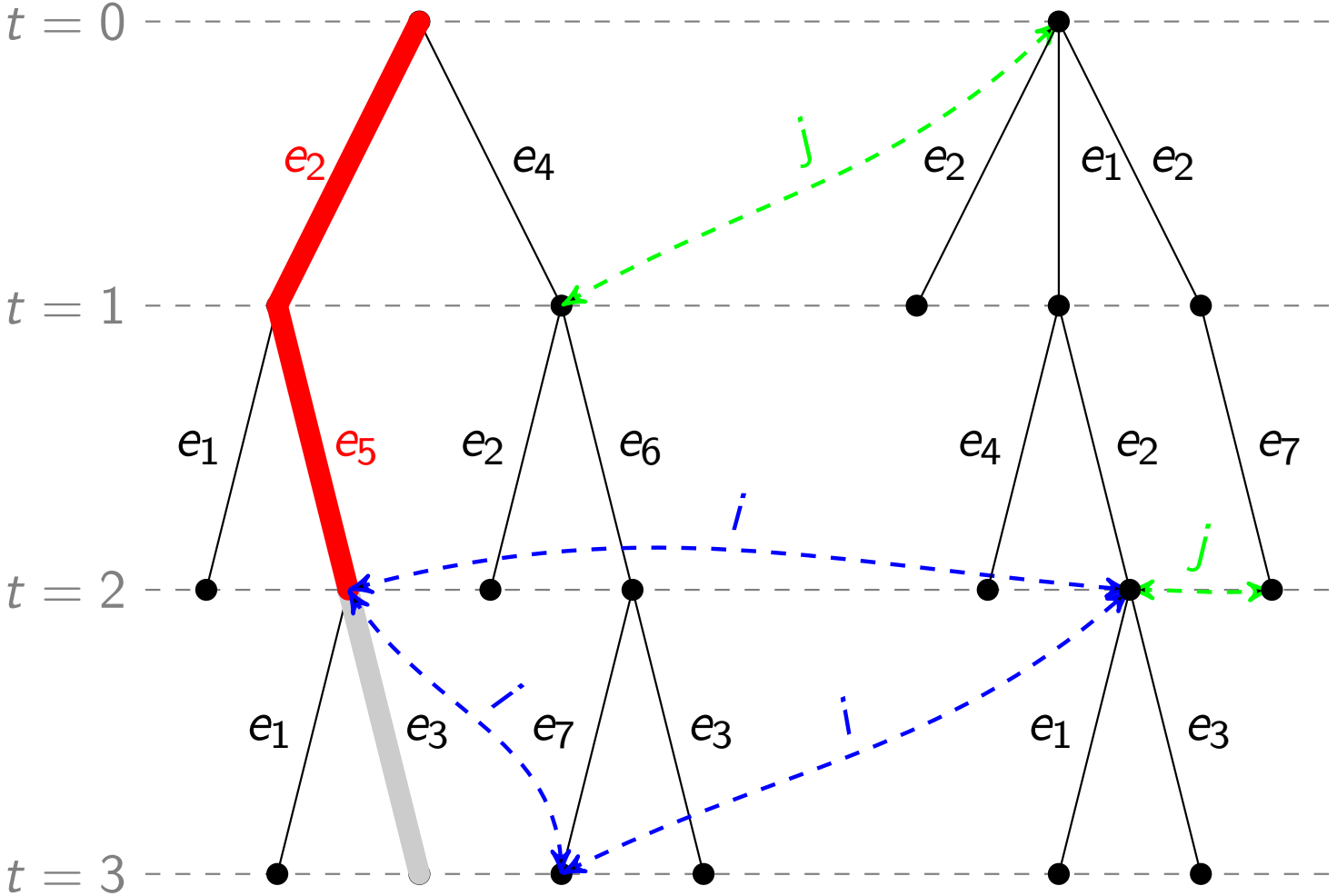
The 'Playground'



The 'Playground'



The 'Playground'



The 'Playground': Notation

- 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 $H \in \Sigma^* \cup \Sigma^\omega$, $\text{len}(H)$ 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'.
- $\text{FinPre}(\mathcal{H}) = \{H \mid \exists H' \in \mathcal{H} \text{ such that } H \preceq H'\}$ is the set of finite prefixes of the elements of $\mathcal{H}$.
- ϵ is the empty string and $\text{FinPre}_{-\epsilon}(\mathcal{H}) = \text{FinPre}(\mathcal{H}) - \{\epsilon\}$.

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 $H \in \Sigma^* \cup \Sigma^\omega$, $\text{len}(H)$ 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'.
- $\text{FinPre}(\mathcal{H}) = \{H \mid \exists H' \in \mathcal{H} \text{ such that } H \preceq H'\}$ is the set of finite prefixes of the elements of $\mathcal{H}$.
- ϵ is the empty string and $\text{FinPre}_{-\epsilon}(\mathcal{H}) = \text{FinPre}(\mathcal{H}) - \{\epsilon\}$.

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 $H \in \Sigma^* \cup \Sigma^\omega$, $\text{len}(H)$ 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'.
- $\text{FinPre}(\mathcal{H}) = \{H \mid \exists H' \in \mathcal{H} \text{ such that } H \preceq H'\}$ is the set of finite prefixes of the elements of $\mathcal{H}$.
- ϵ is the empty string and $\text{FinPre}_{-\epsilon}(\mathcal{H}) = \text{FinPre}(\mathcal{H}) - \{\epsilon\}$.

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 $H \in \Sigma^* \cup \Sigma^\omega$, $\text{len}(H)$ 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'.
- $\text{FinPre}(\mathcal{H}) = \{H \mid \exists H' \in \mathcal{H} \text{ such that } H \preceq H'\}$ is the set of finite prefixes of the elements of $\mathcal{H}$.
- ϵ is the empty string and $\text{FinPre}_{-\epsilon}(\mathcal{H}) = \text{FinPre}(\mathcal{H}) - \{\epsilon\}$.

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 $H \in \Sigma^* \cup \Sigma^\omega$, $\text{len}(H)$ 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'.
- $\text{FinPre}(\mathcal{H}) = \{H \mid \exists H' \in \mathcal{H} \text{ such that } H \preceq H'\}$ is the set of finite prefixes of the elements of $\mathcal{H}$.
- ϵ is the empty string and $\text{FinPre}_{-\epsilon}(\mathcal{H}) = \text{FinPre}(\mathcal{H}) - \{\epsilon\}$.

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 $H \in \Sigma^* \cup \Sigma^\omega$, $\text{len}(H)$ 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'.
- $\text{FinPre}(\mathcal{H}) = \{H \mid \exists H' \in \mathcal{H} \text{ such that } H \preceq H'\}$ is the set of finite prefixes of the elements of $\mathcal{H}$.
- ϵ is the empty string and $\text{FinPre}_{-\epsilon}(\mathcal{H}) = \text{FinPre}(\mathcal{H}) - \{\epsilon\}$.

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 $H \in \Sigma^* \cup \Sigma^\omega$, $\text{len}(H)$ 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'.
- $\text{FinPre}(\mathcal{H}) = \{H \mid \exists H' \in \mathcal{H} \text{ such that } H \preceq H'\}$ is the set of finite prefixes of the elements of $\mathcal{H}$.
- ϵ is the empty string and $\text{FinPre}_{-\epsilon}(\mathcal{H}) = \text{FinPre}(\mathcal{H}) - \{\epsilon\}$.

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 $\text{FinPre}_{-\epsilon}(\mathcal{H}) \subseteq \mathcal{H}$. 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**,.

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**,.

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

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

Formulas are interpreted at pairs H, t:

$$H, t \models \varphi$$

Truth in a Model

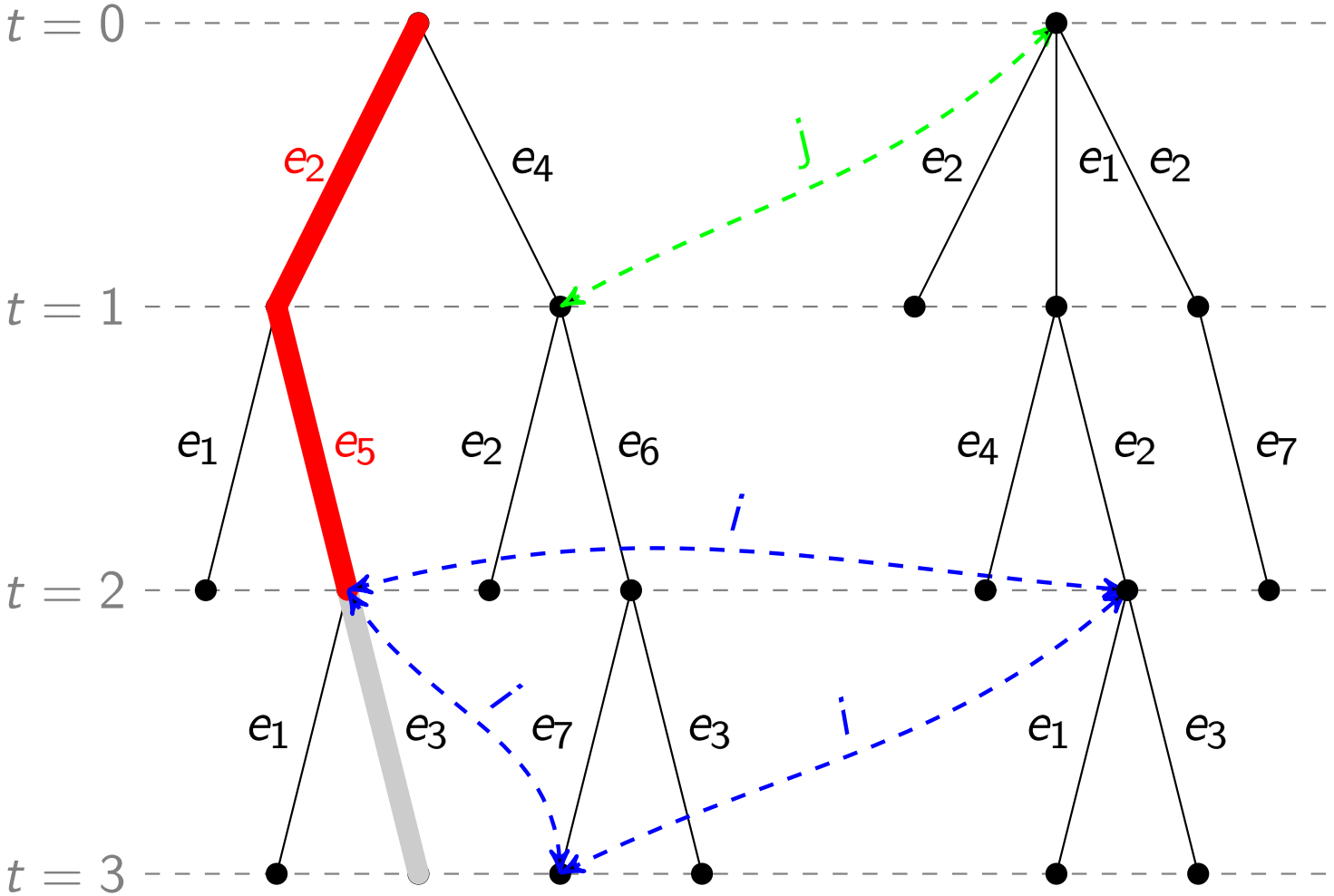
- $H, t \models P\varphi$ iff there exists $t' \leq t$ such that $H, t' \models \varphi$
- $H, t \models F\varphi$ iff there exists $t' \geq t$ such that $H, t' \models \varphi$
- $H, t \models N\varphi$ iff $H, t+1 \models \varphi$
- $H, t \models Y\varphi$ iff $t > 1$ and $H, t-1 \models \varphi$
- $H, t \models K_i\varphi$ 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 $H', m \models \varphi$.

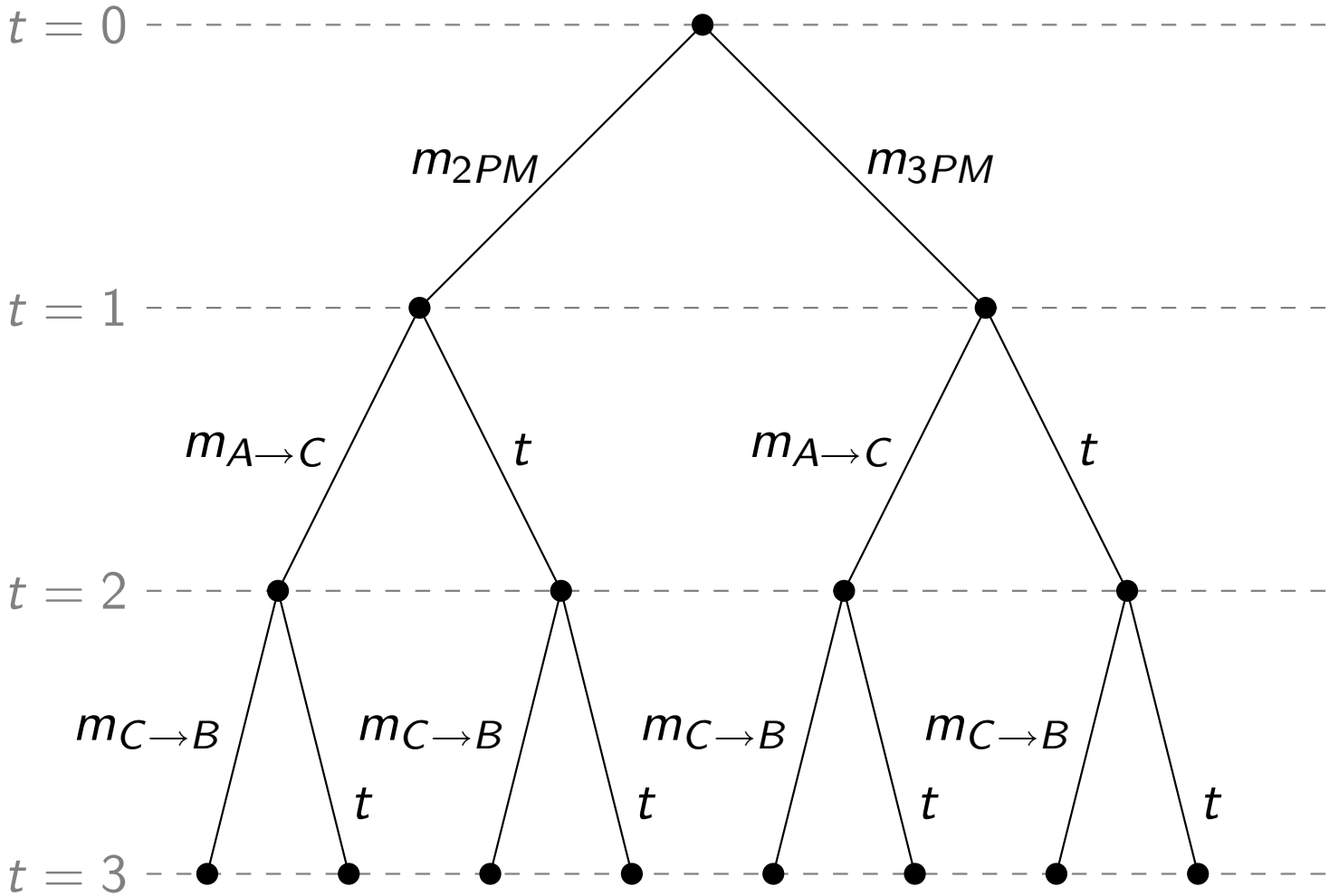
where $\sim_*$ is the reflexive transitive closure of the union of the $\sim_i$.

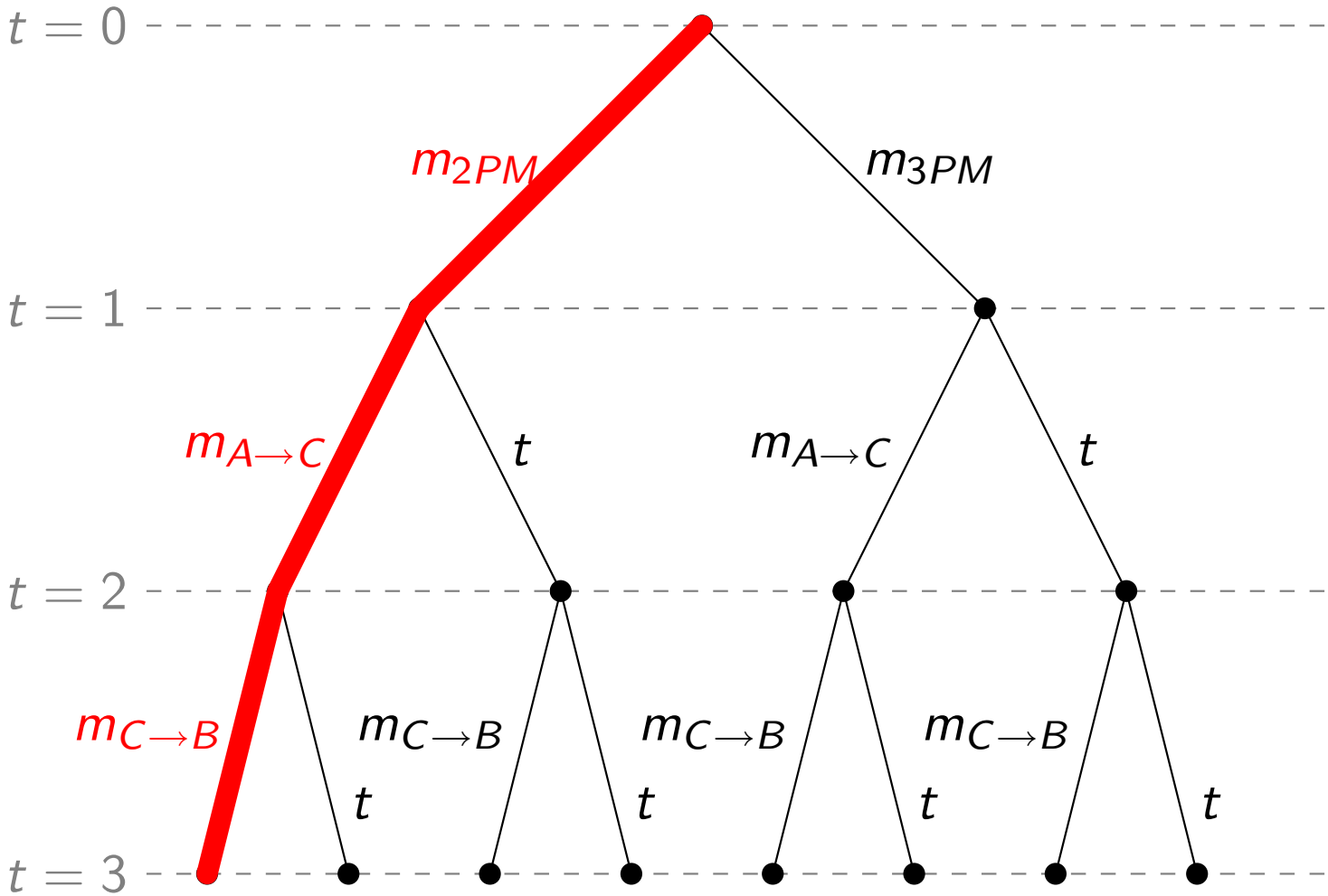

t = 0
t = 1
t = 2
t = 3
e2
e4
j
e2
e1
e2
e1
e5
e2
e6
e4
e2
e7
i
e1
e3
e7
e3
i
e1
e3

Returning to the Example

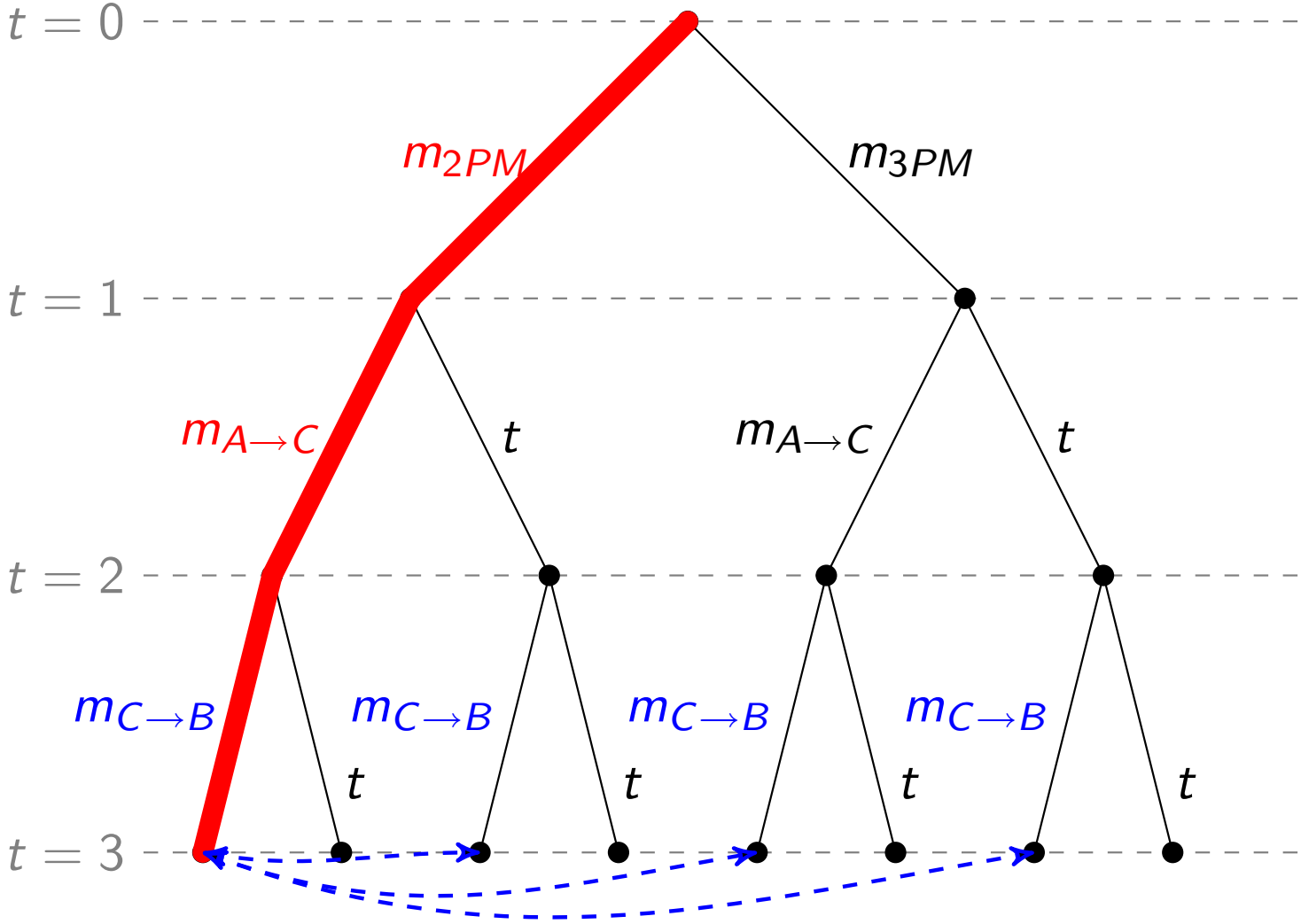
The events are

1. Ann receives a message that the talk is at 2PM (m_{2PM})
2. Ann receives a message that the talk is at 3PM (m_{3PM})
3. Ann tells Charles the talk is at 2PM ($m_{A \to C}$)
4. Charles tells Bob the talk is at 2PM ($m_{C \to B}$)
5. Nothing happens (t)

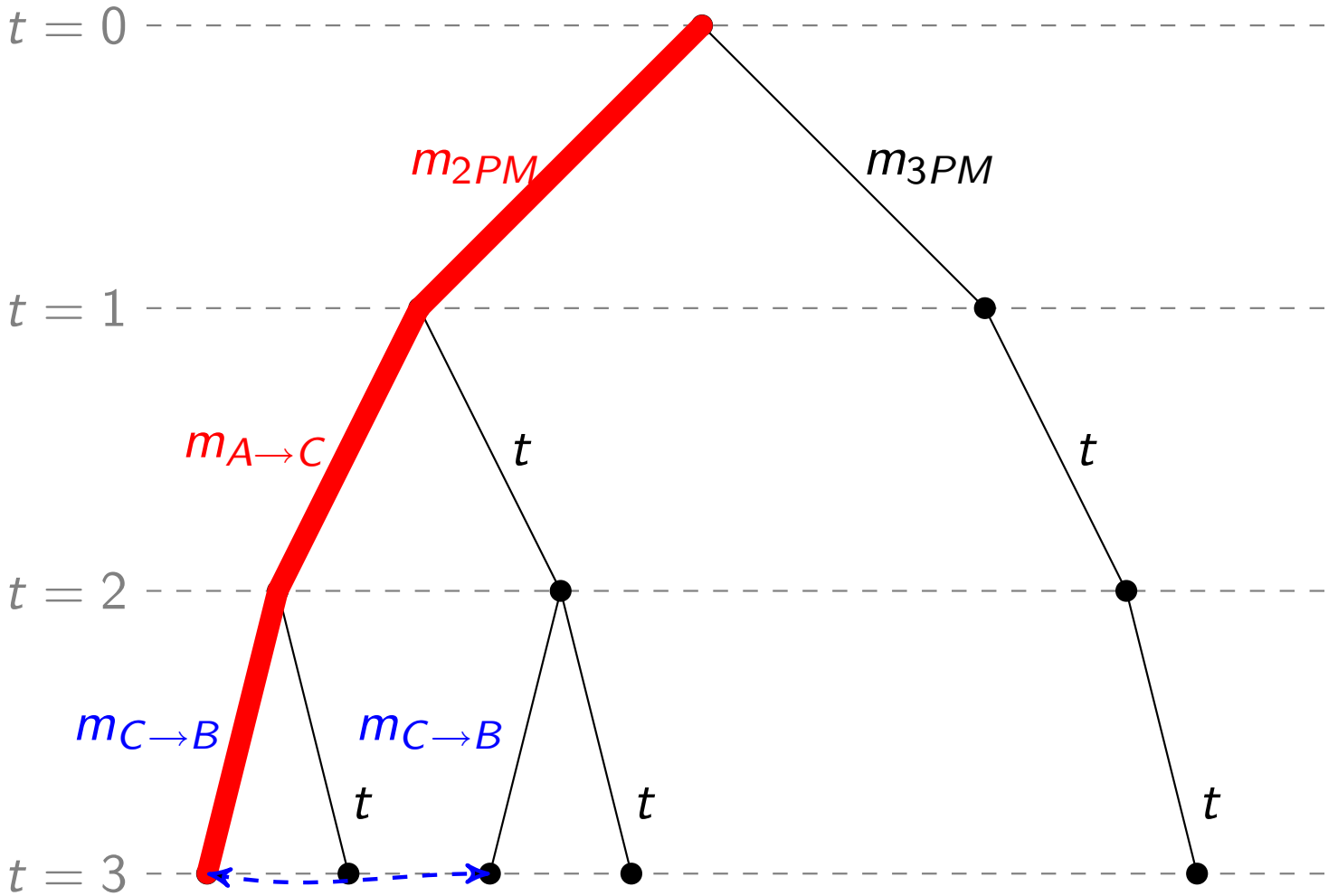

$t = 0$
$t = 1$
$t = 2$
$t = 3$
m_{2PM}
m_{3PM}
$m_{A\to C}$
t
$m_{A\to C}$
t
$m_{C\to B}$
$m_{C\to B}$
$m_{C\to B}$
$m_{C\to B}$
t
t
t
t



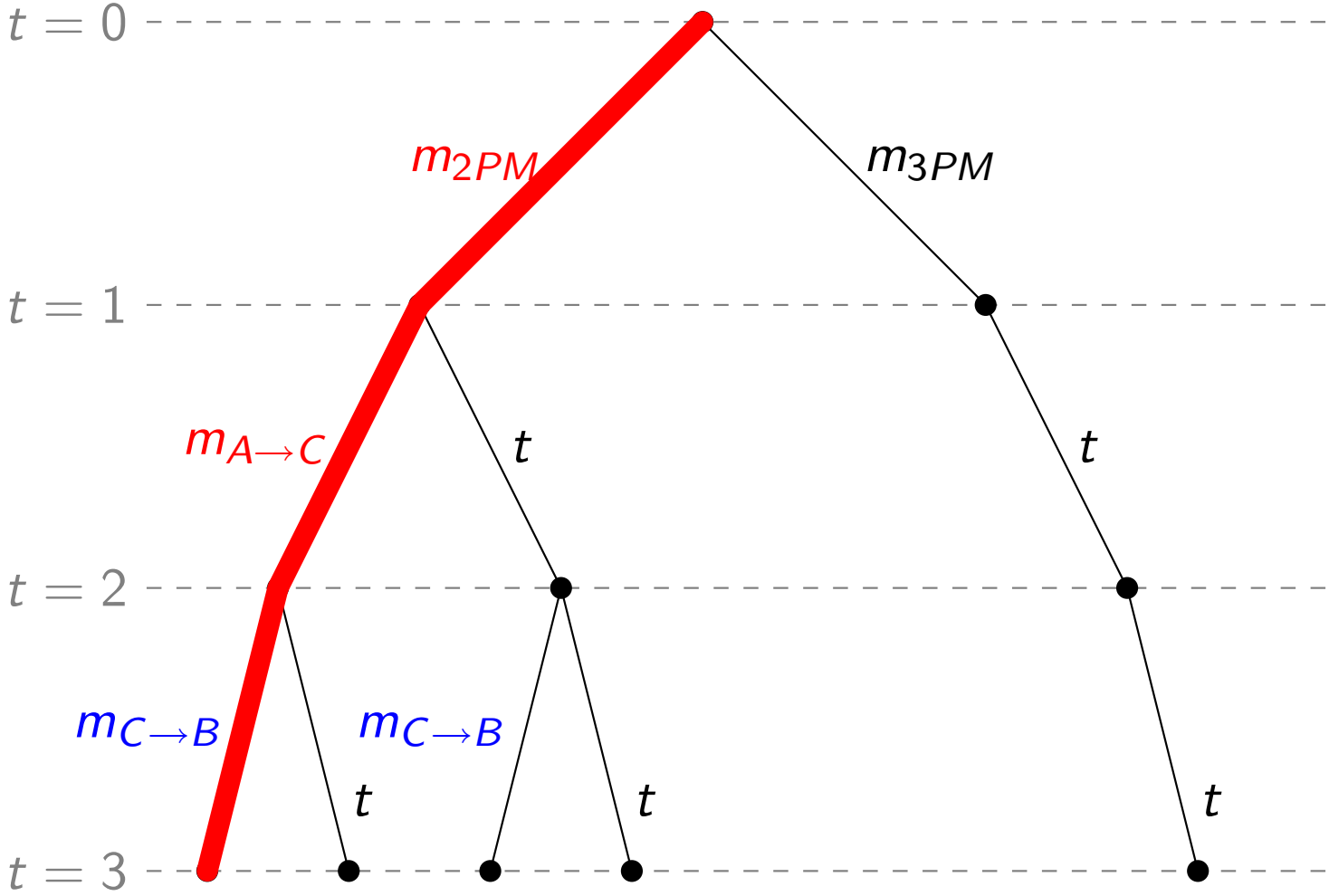
$$H, 3 \models \varphi$$



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

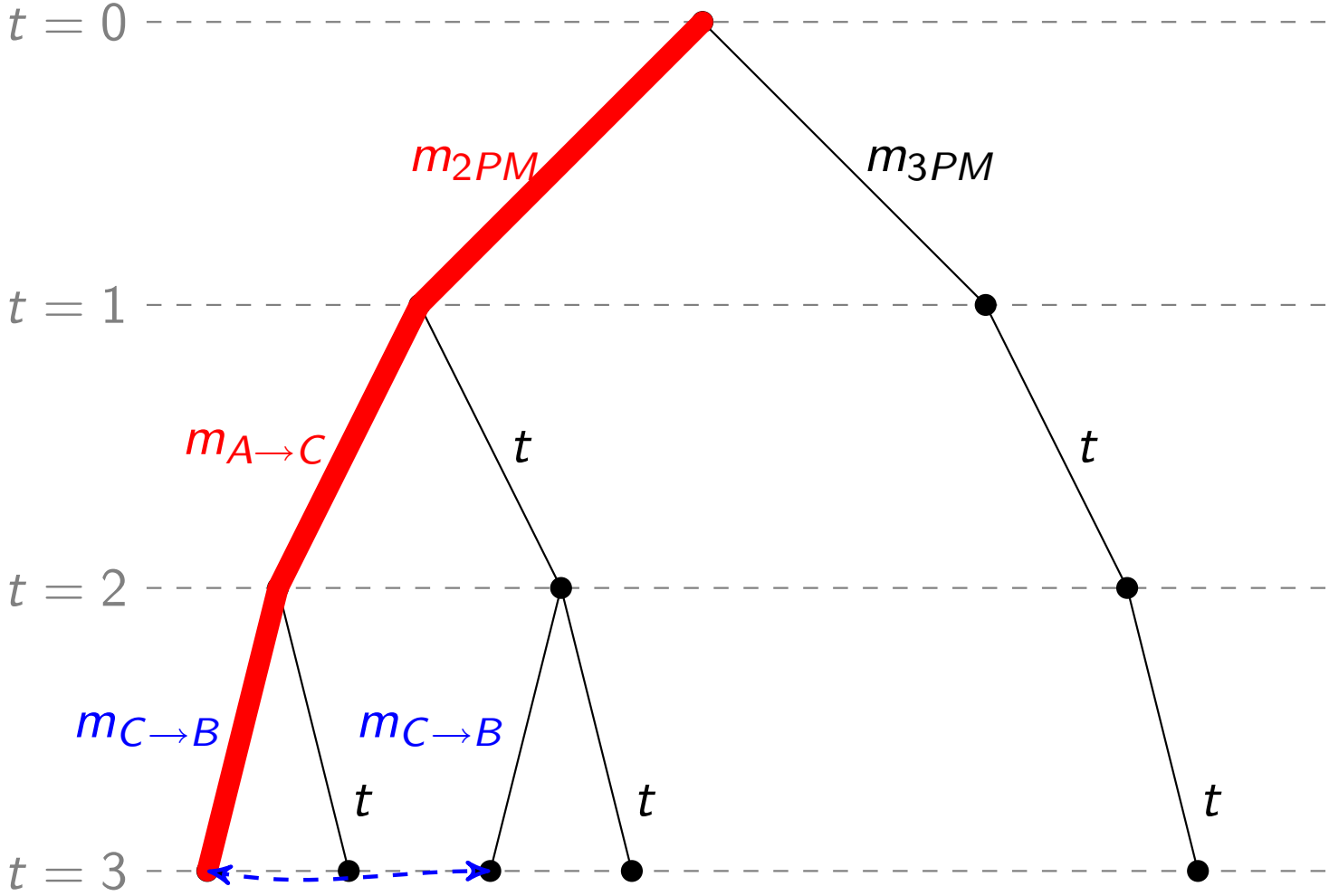


Bob's uncertainty + 'Protocol information': $H, 3 \models K_B P_{2PM}$



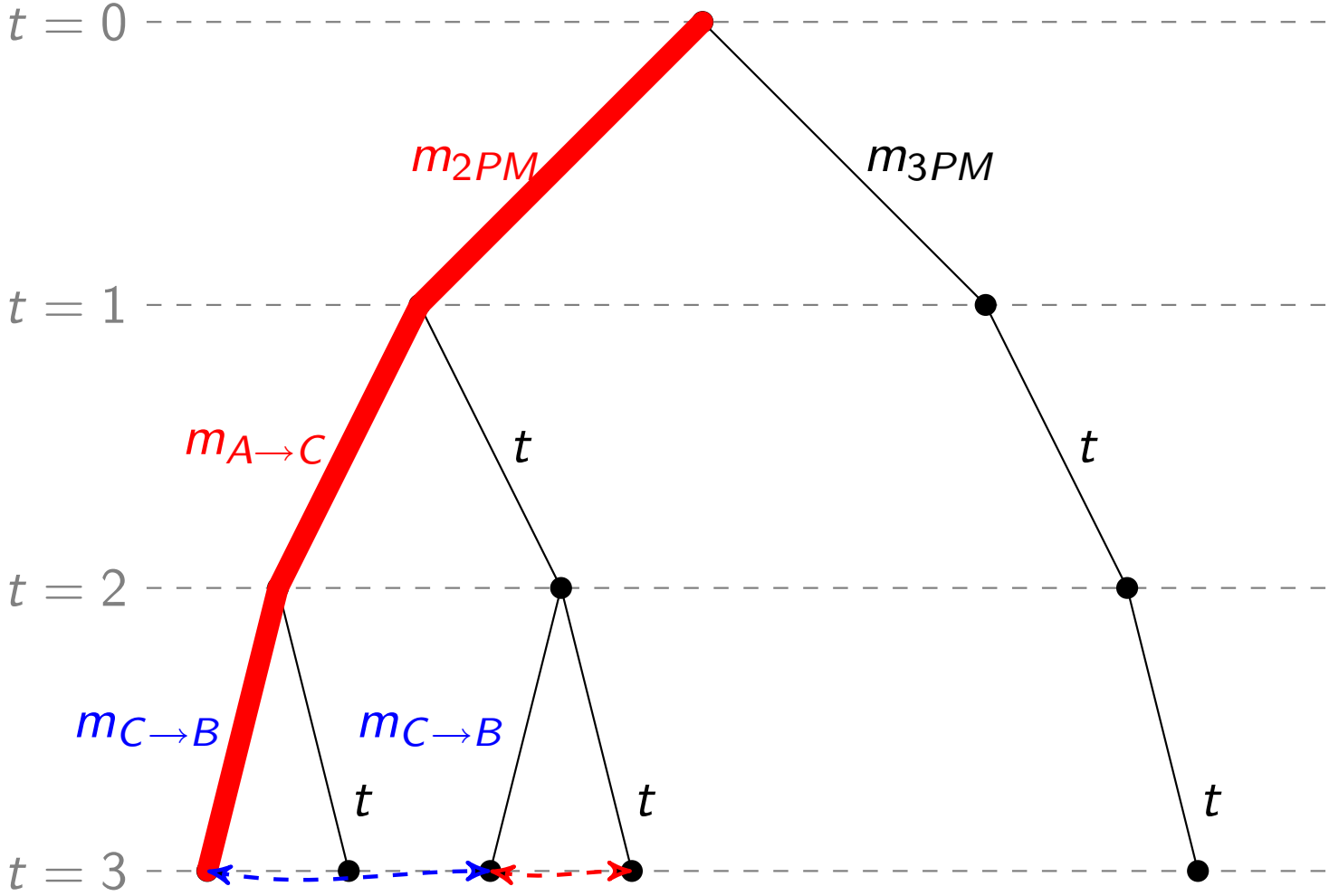
Bob's uncertainty + 'Protocol information':

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



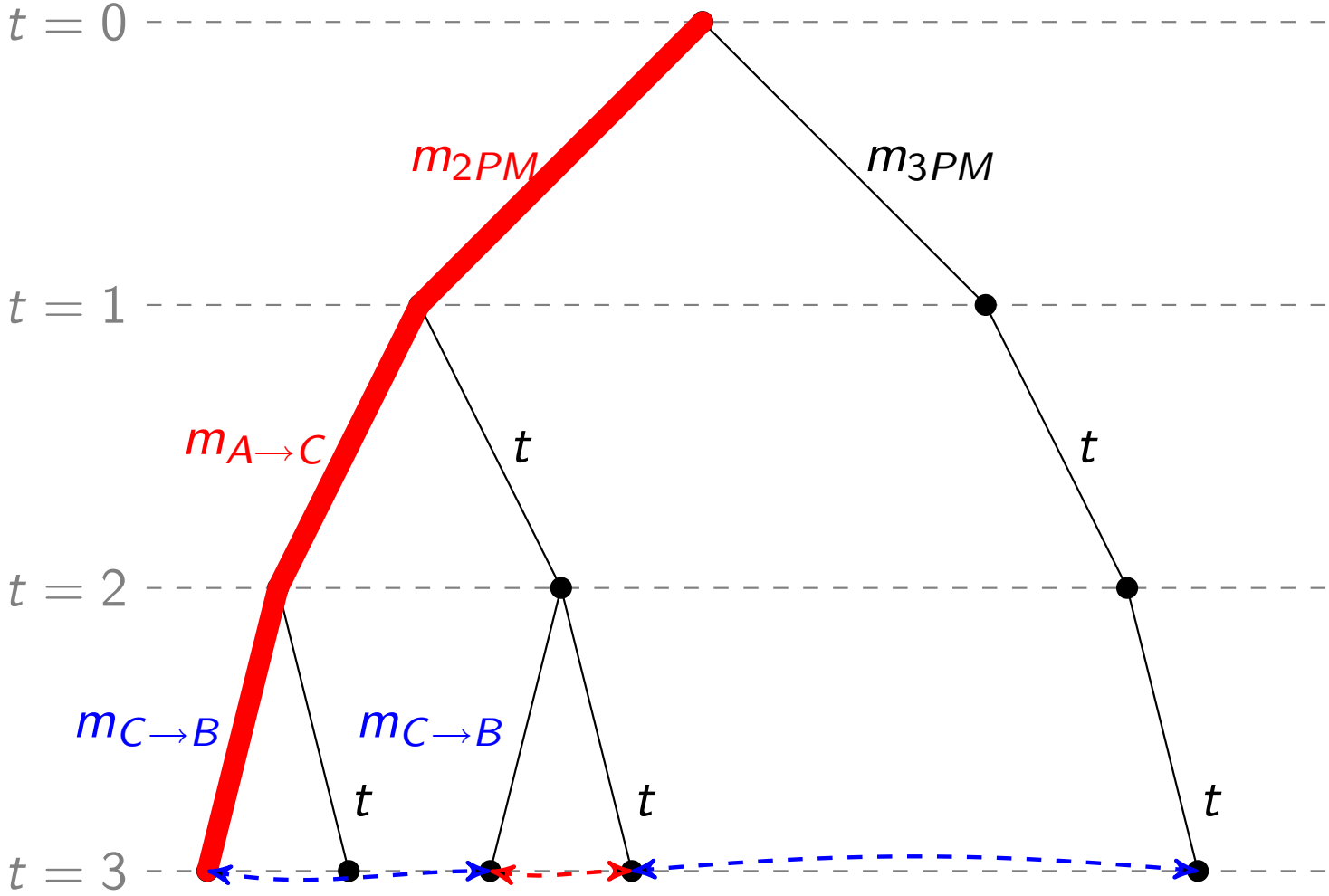
Bob's uncertainty + 'Protocol information':

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



Bob's uncertainty + 'Protocol information':

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

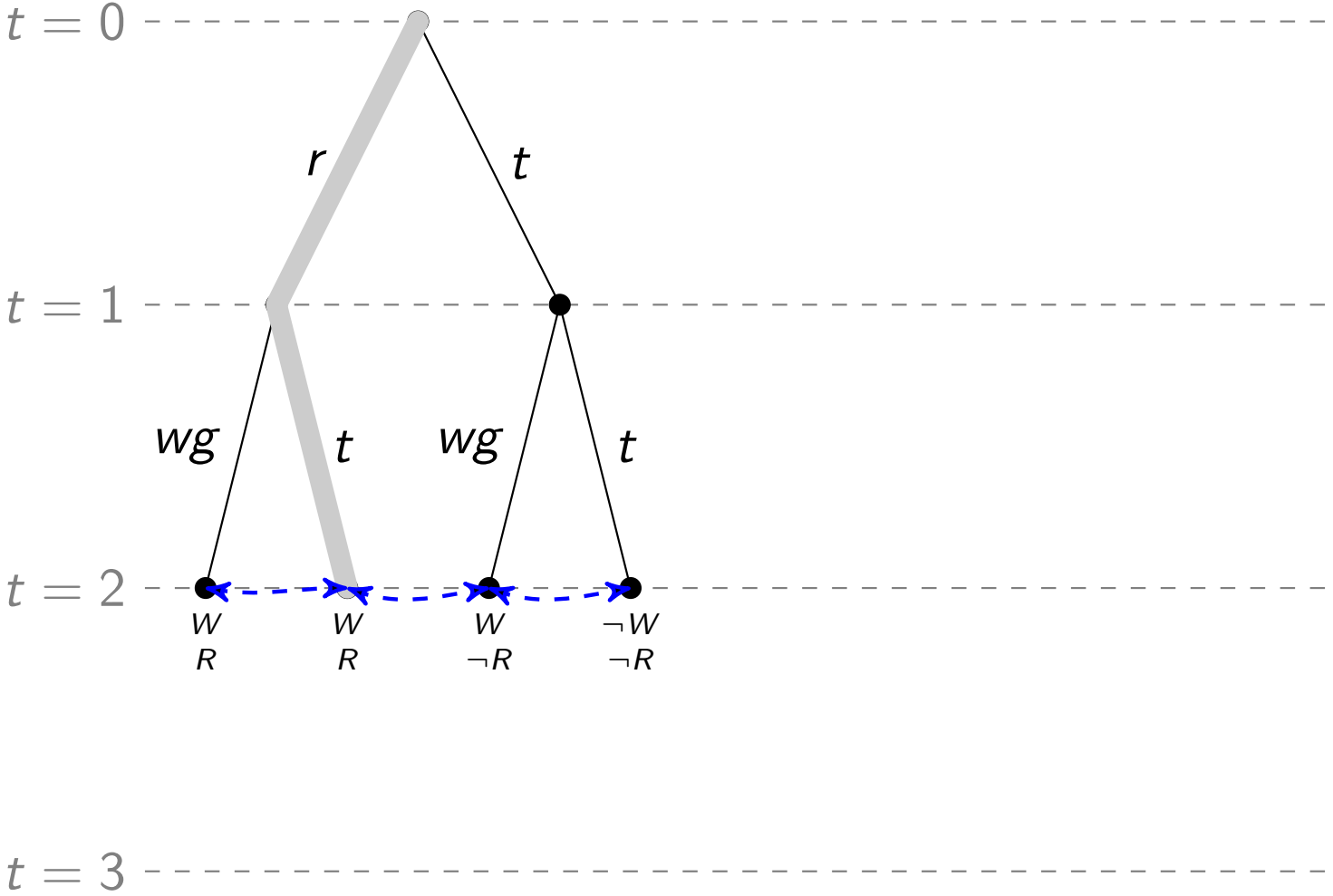


Bob's uncertainty + 'Protocol information':

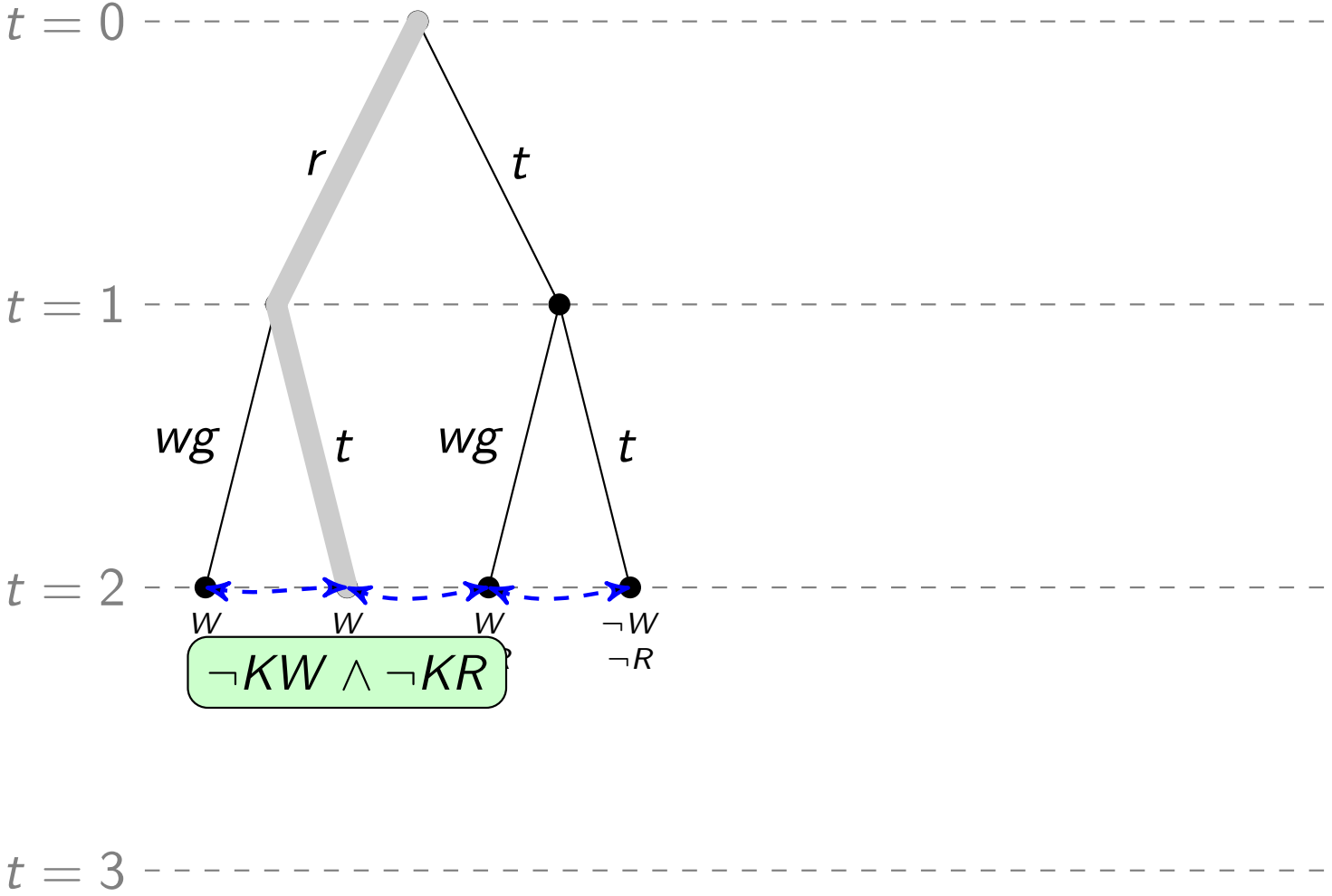
$$H, 3 \models \neg K_B K_A K_B P_{2PM}$$

Learning from the Protocol

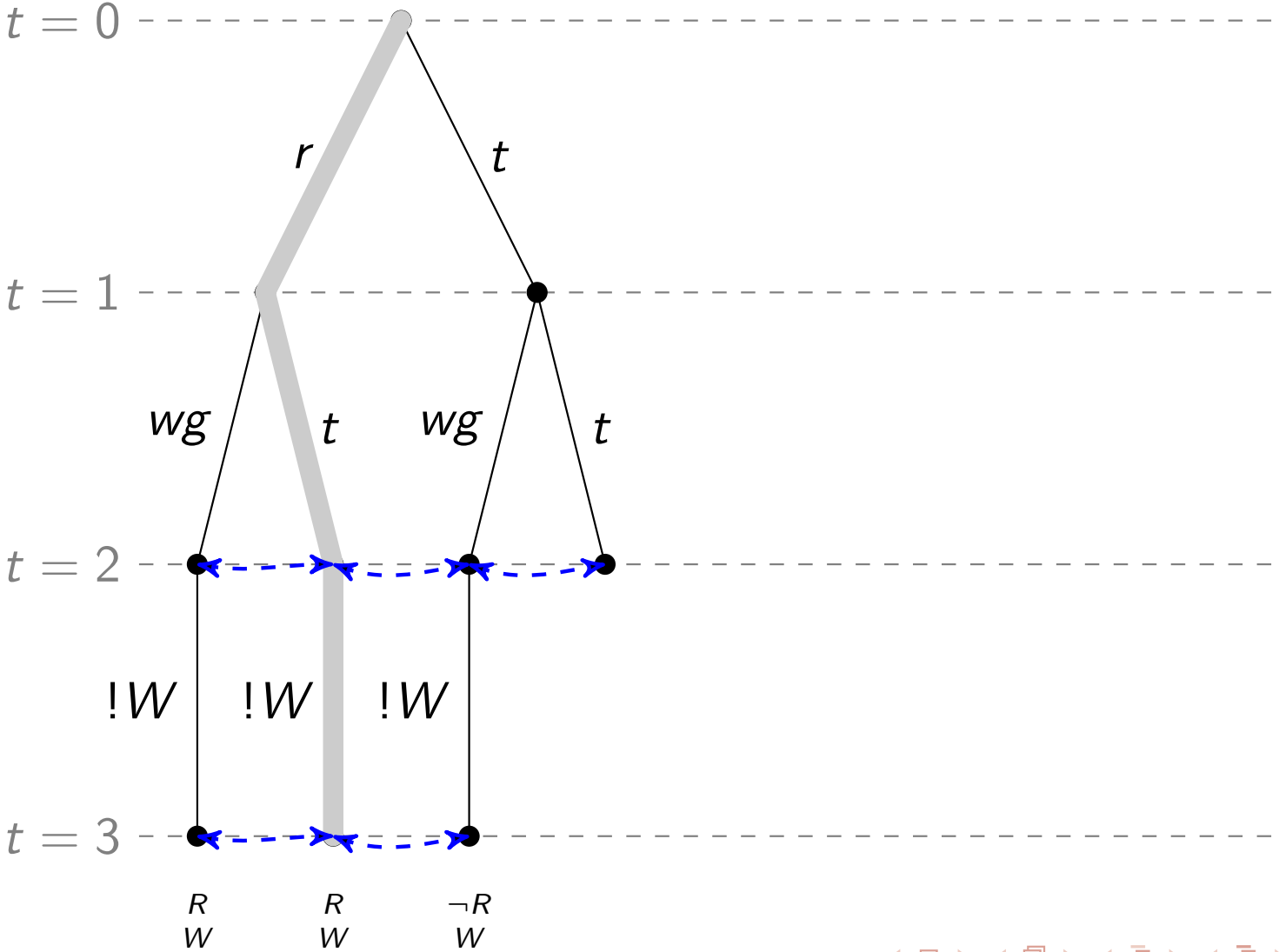
Learning from the Protocol



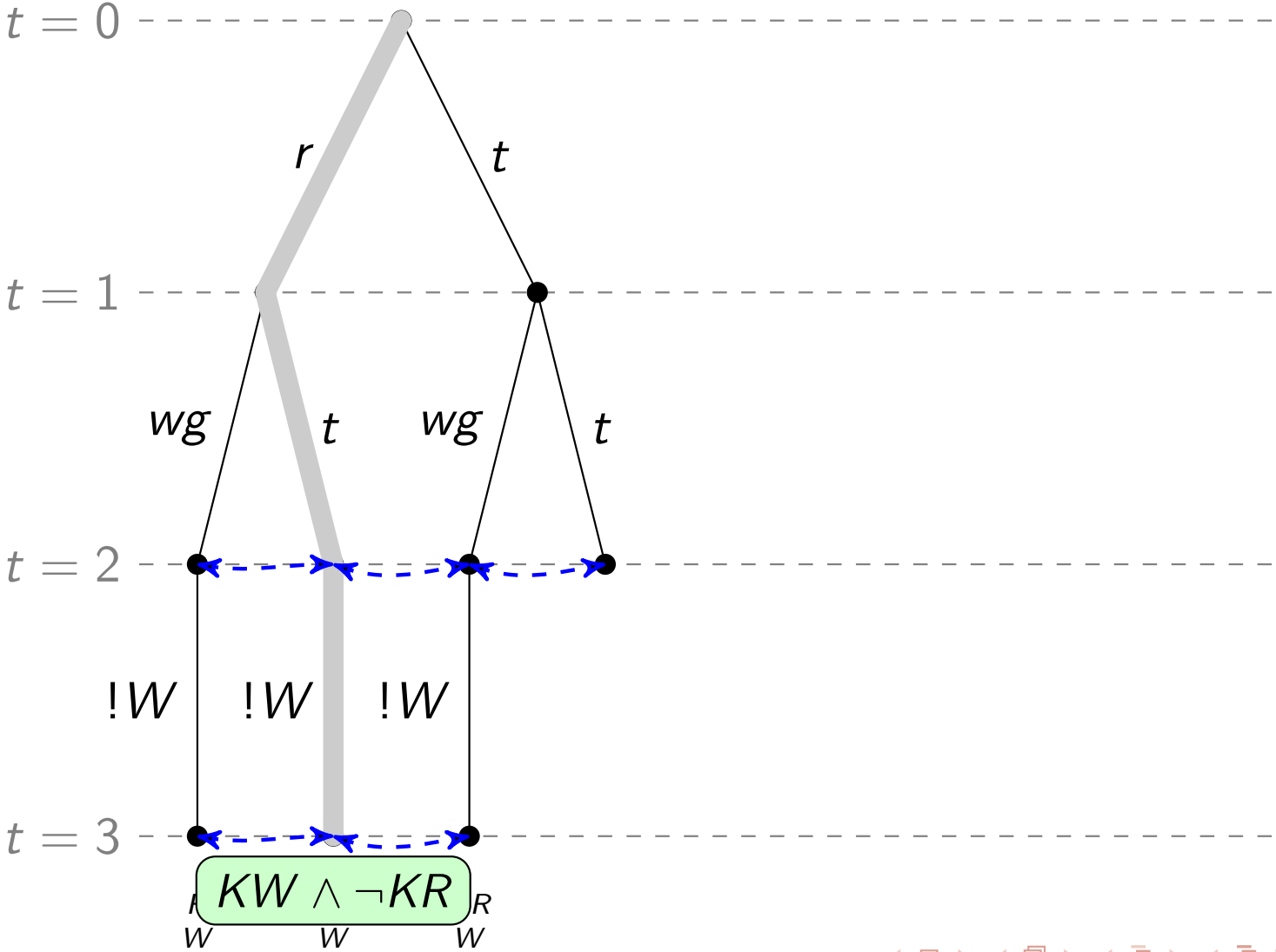
Learning from the Protocol



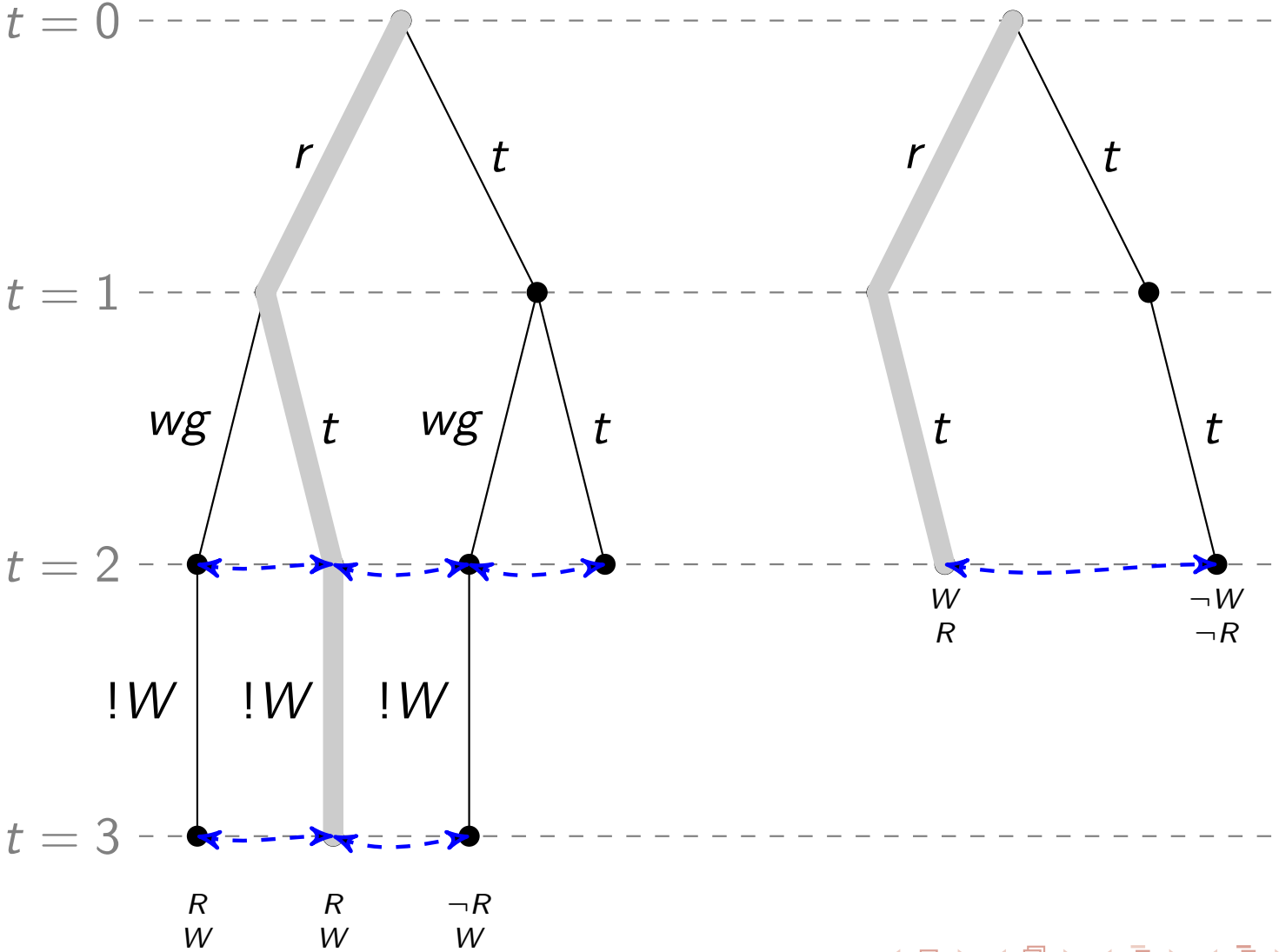
Learning from the Protocol



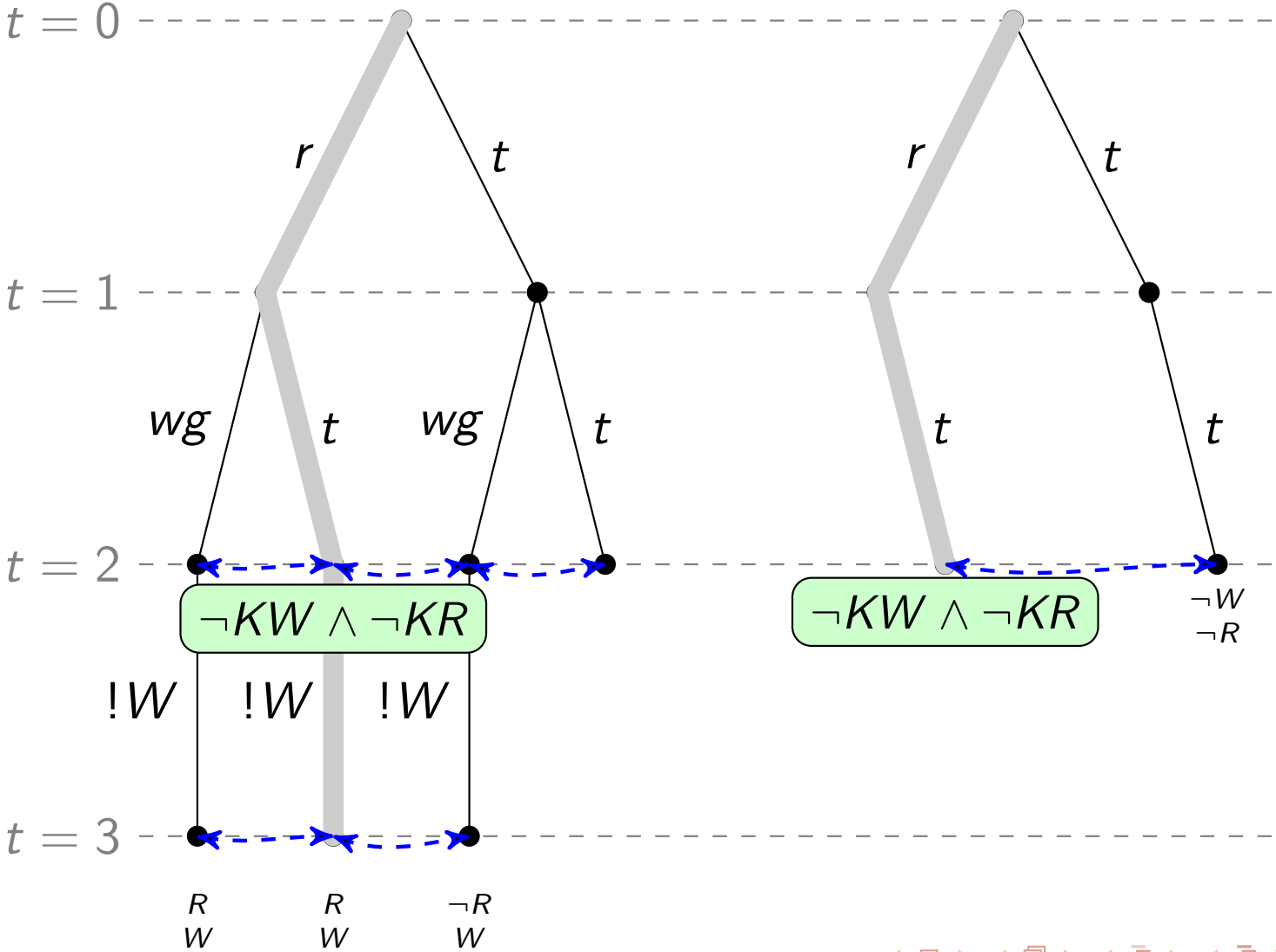
Learning from the Protocol



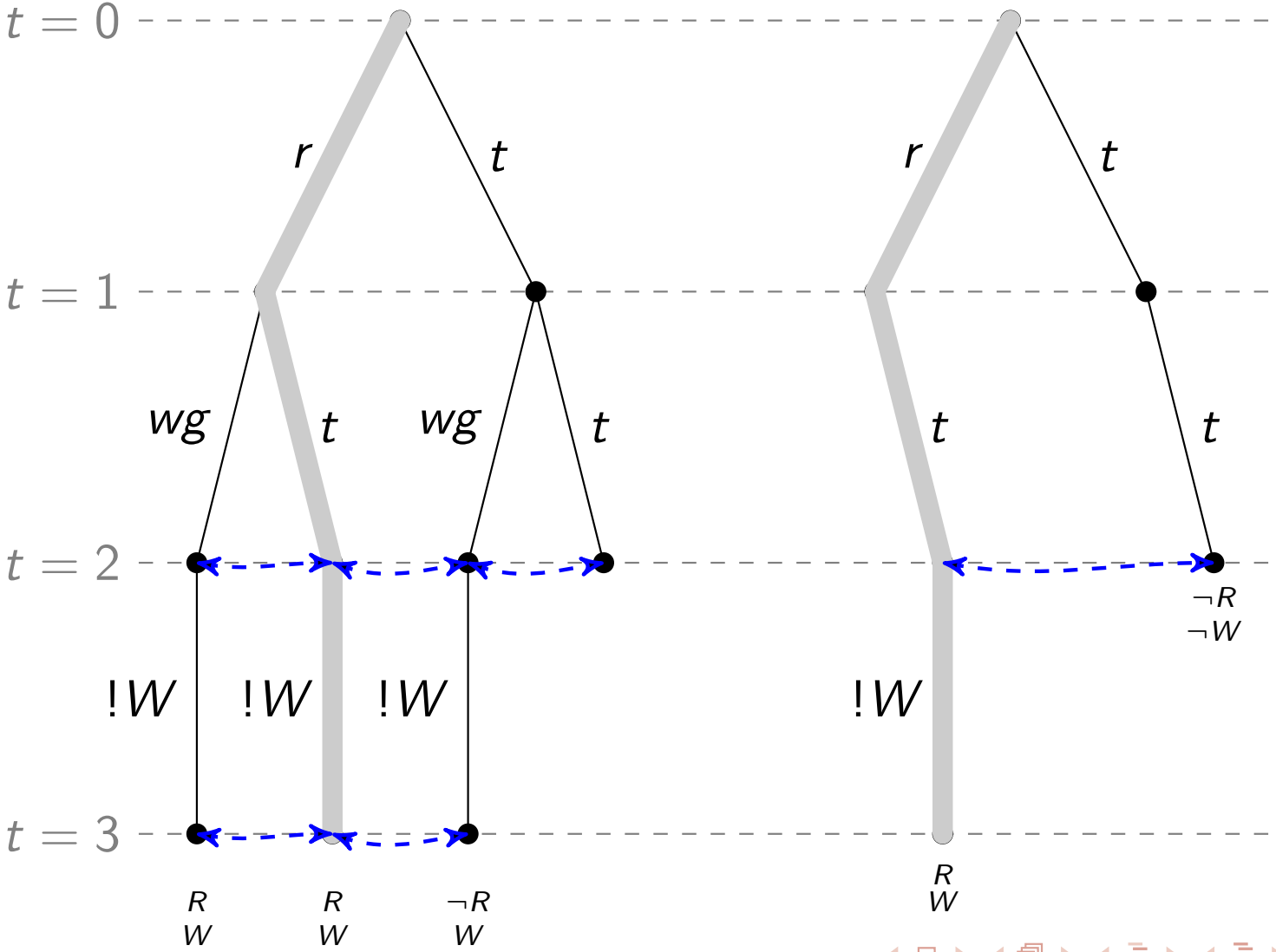
Learning from the Protocol



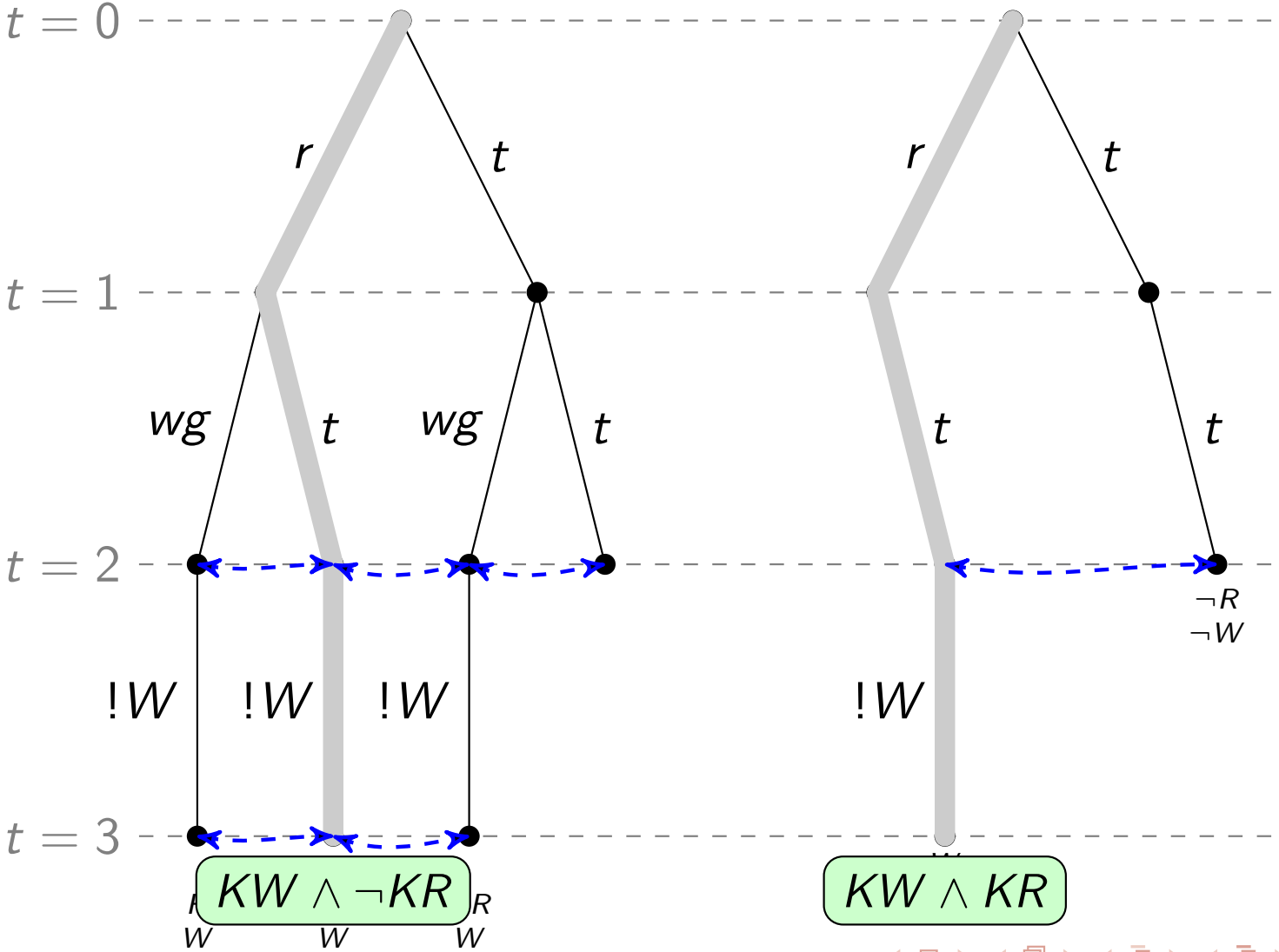
Learning from the Protocol



Learning from the Protocol



Learning from the Protocol



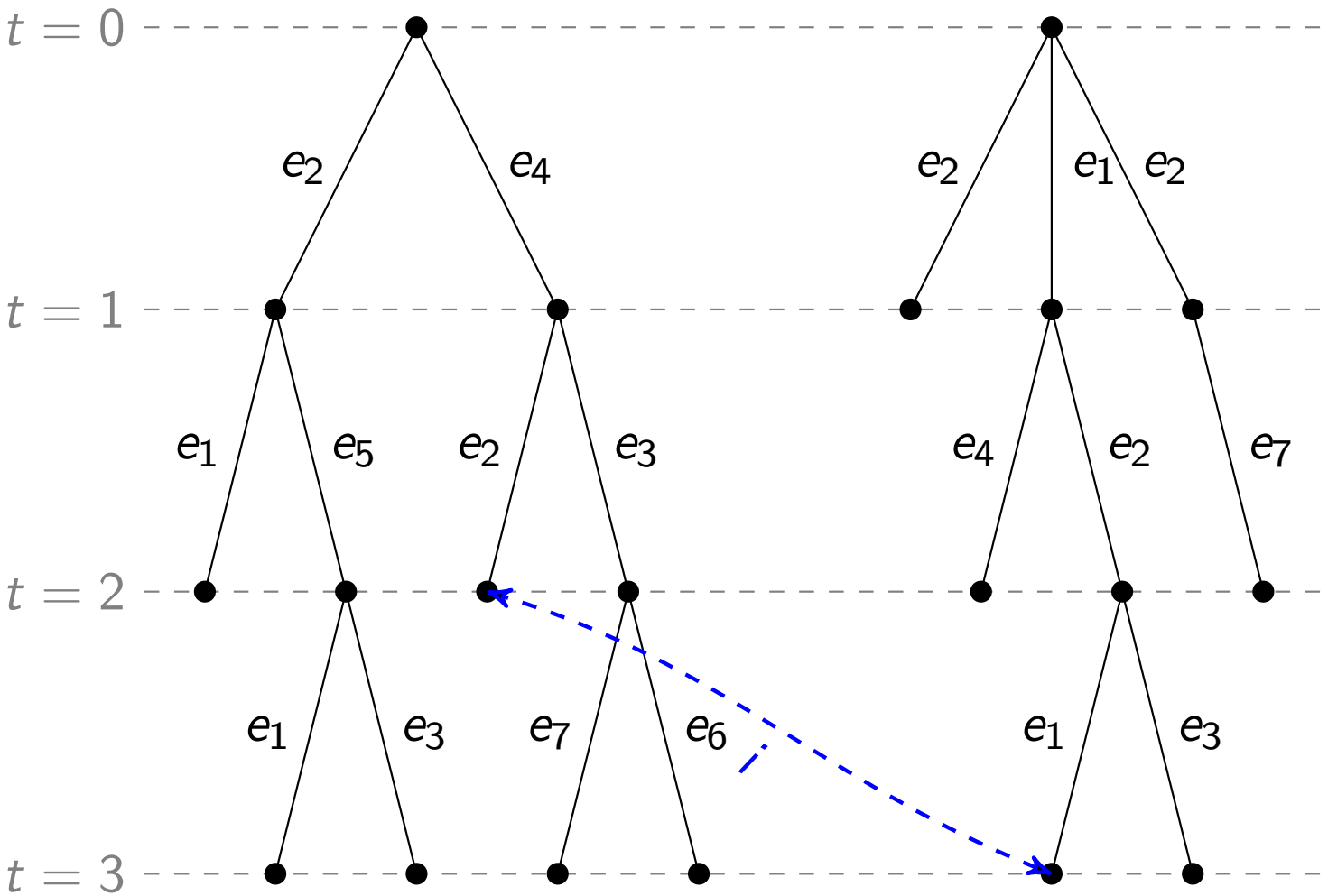
Parameters of The Model

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

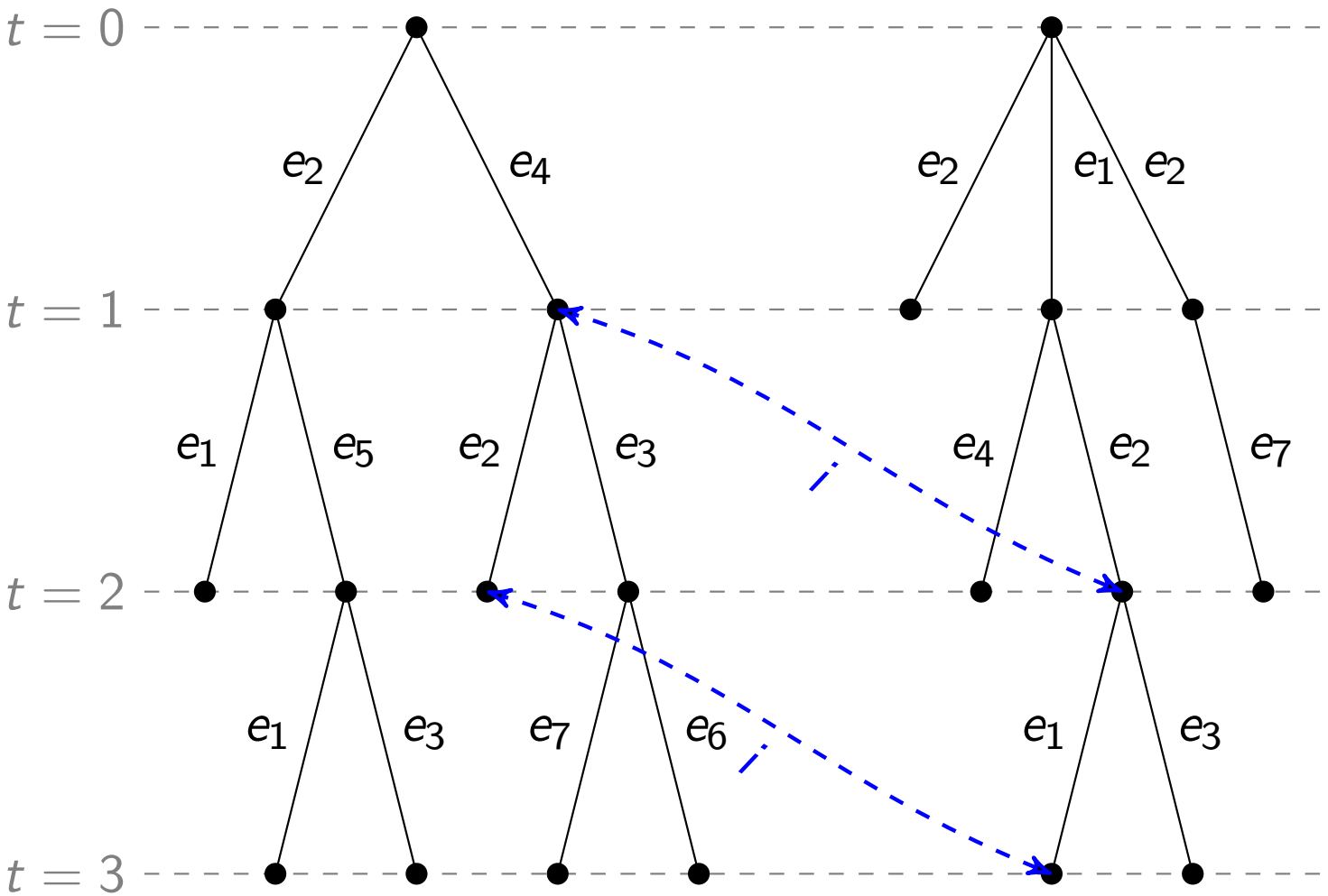
2. Structural conditions on the underlying event structure. Do we restrict to protocol frames (finitely branching trees)? Finitely branching forests? Or, arbitrary ETL frames?

3. Conditions on the reasoning abilities of the agents. Do the agents satisfy perfect recall? No miracles? Synchronization?

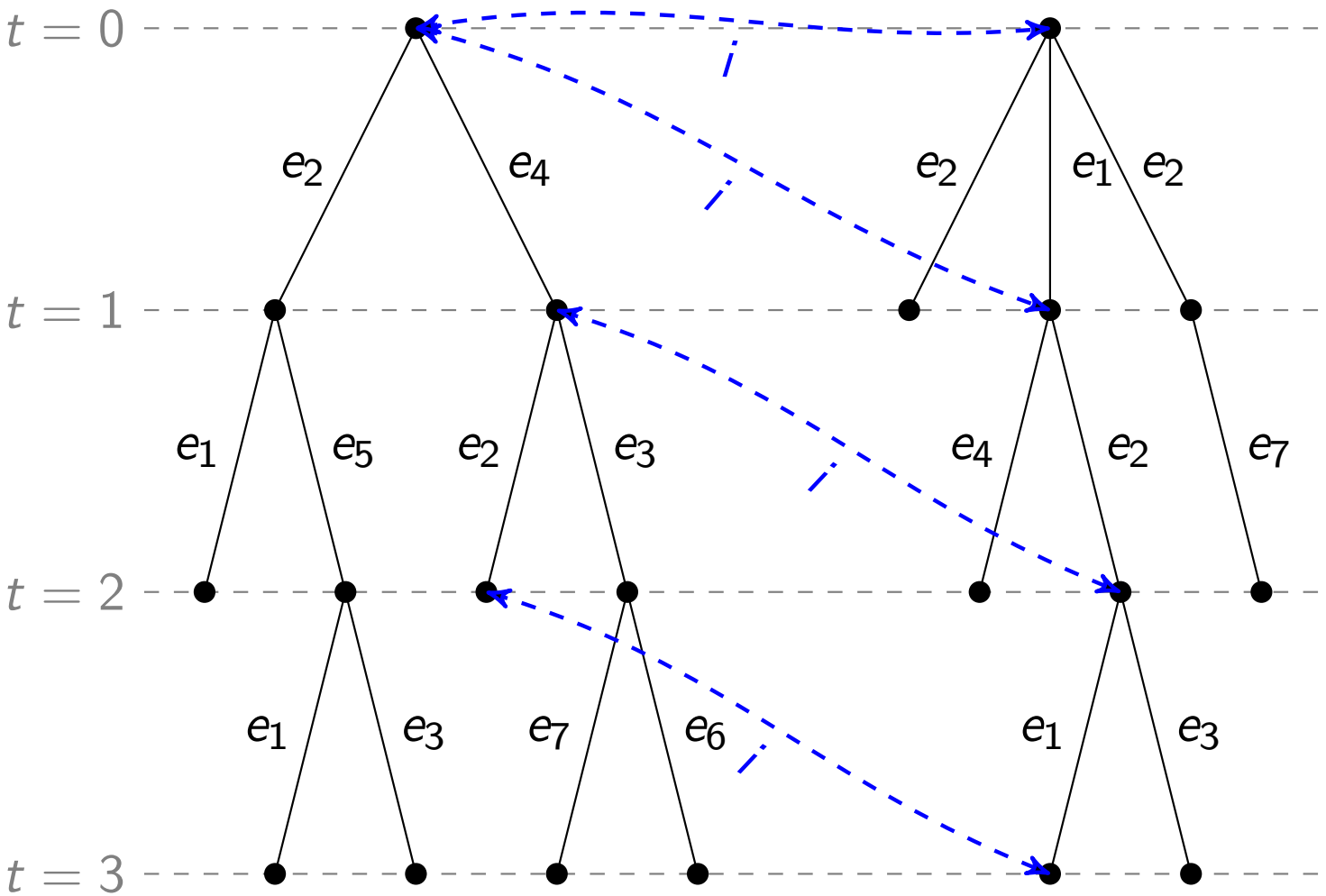
Perfect Recall



Perfect Recall



Perfect Recall



Some Literature

J. Halpern and M. Vardi. *The Complexity of Reasoning about Knowledge and Time I*. Journal of Computer and System Science (1989).

J. van Benthem and EP. *The Tree of Knowledge in Action: Towards a Common Perspective*. Proceedings of Advances in Modal Logic (2006).

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