Logics of Rational Agency

Lecture 3

Eric Pacuit

Tilburg Institute for Logic and Philosophy of Science
Tilburg Univeristy
ai.stanford.edu/~epacuit

July 29, 2009

✓ Introduction, Motivation and Background

Lecture 2: Basic Ingredients for a Logic of Rational Agency

Lecture 3: Logics of Rational Agency and Social Interaction, Part I

Lecture 4: Logics of Rational Agency and Social Interaction, Part II

Lecture 5: Conclusions and General Issues

Logics of Rational Agency

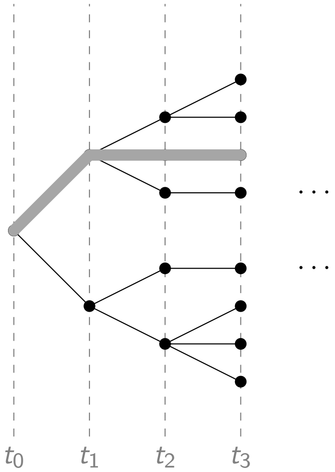
Basic Ingredients

- ✓ informational attitudes (eg., knowledge, belief, certainty)
- ✓ group notions (eg., common knowledge and coalitional ability)
- ✓ time, actions and ability
- ✓ motivational attitudes (eg., preferences)
- ✓ normative attitudes (eg., obligations)

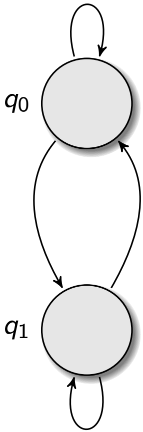
Basic Ingredients

- ✓ informational attitudes (eg., knowledge, belief, certainty)
- ✓ group notions (eg., common knowledge and coalitional ability)
- ✓ time, actions and ability
- ✓ motivational attitudes (eg., preferences)
- ✓ normative attitudes (eg., obligations)

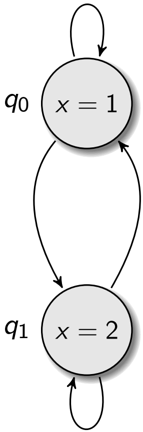
Temporal Logics



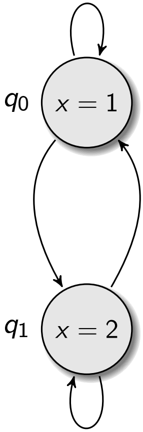
Computational vs. Behavioral Structures

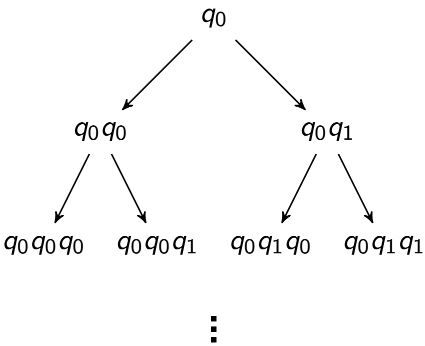


Computational vs. Behavioral Structures



Computational vs. Behavioral Structures





Temporal Logics

Temporal Logics

- *Linear Time Temporal Logic*: Reasoning about computation paths:

$$\Diamond\varphi: \varphi \text{ is true some time in } \textit{the} \text{ future.}$$

A. Pnuelli. *A Temporal Logic of Programs*. in *Proc. 18th IEEE Symposium on Foundations of Computer Science* (1977).

Temporal Logics

- *Linear Time Temporal Logic*: Reasoning about computation paths:

$$\Diamond\varphi\text{: } \varphi \text{ is true some time in } \textit{the} \text{ future.}$$

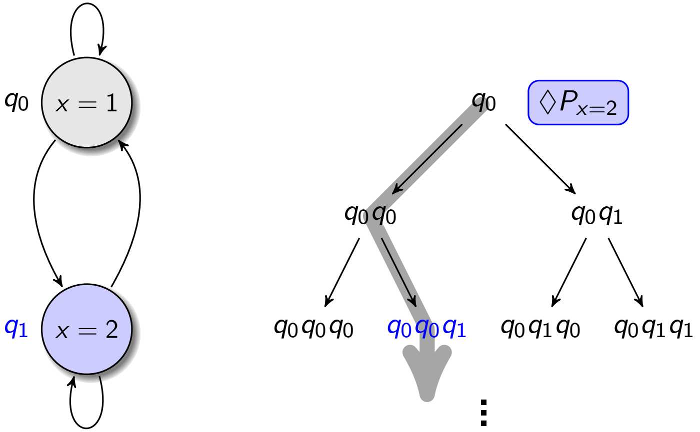
A. Pnuelli. *A Temporal Logic of Programs*. in *Proc. 18th IEEE Symposium on Foundations of Computer Science* (1977).

- *Branching Time Temporal Logic*: Allows quantification over paths:

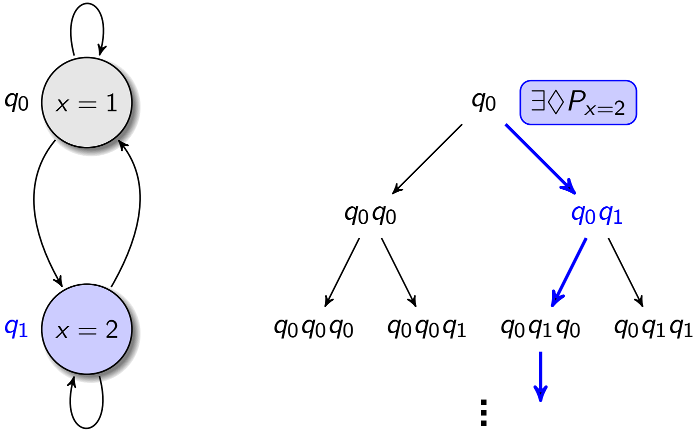
$$\exists\Diamond\varphi\text{: there is a path in which } \varphi \text{ is eventually true.}$$

E. M. Clarke and E. A. Emerson. *Design and Synthesis of Synchronization Skeletons using Branching-time Temproal-logic Specifications*. In *Proceedings Workshop on Logic of Programs*, LNCS (1981).

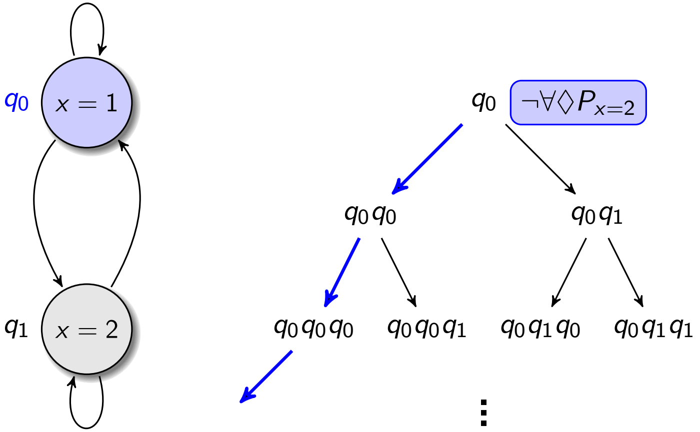
Temporal Logics



Temporal Logics



Temporal Logics



Many Agents

The previous model assumes there is *one* agent that “controls” the transition system.

Many Agents

The previous model assumes there is *one* agent that "controls" the transition system.

What if there is more than one agent?

Many Agents

The previous model assumes there is *one* agent that "controls" the transition system.

What if there is more than one agent?

Example: Suppose that there are two agents: a server (s) and a client (c). The client asks to set the value of x and the server can either grant or deny the request. Assume the agents make simultaneous moves.

Many Agents

The previous model assumes there is *one* agent that "controls" the transition system.

What if there is more than one agent?

Example: Suppose that there are two agents: a server (s) and a client (c). The client asks to set the value of x and the server can either grant or deny the request. Assume the agents make simultaneous moves.

	deny	*grant*
*set*1		
*set*2		

Many Agents

The previous model assumes there is *one* agent that "controls" the transition system.

What if there is more than one agent?

Example: Suppose that there are two agents: a server (s) and a client (c). The client asks to set the value of x and the server can either grant or deny the request. Assume the agents make simultaneous moves.

	deny	*grant*
*set*1		$q_0 \Rightarrow q_0,\ q_1 \Rightarrow q_0$
*set*2		$q_0 \Rightarrow q_1,\ q_1 \Rightarrow q_1$

Many Agents

The previous model assumes there is *one* agent that "controls" the transition system.

What if there is more than one agent?

Example: Suppose that there are two agents: a server (s) and a client (c). The client asks to set the value of x and the server can either grant or deny the request. Assume the agents make simultaneous moves.

	deny	*grant*
*set*1	$q \Rightarrow q$	$q_0 \Rightarrow q_0$, $q_1 \Rightarrow q_0$
*set*2	$q \Rightarrow q$	$q_0 \Rightarrow q_1$, $q_1 \Rightarrow q_1$

From Temporal Logic to Strategy Logic

- *Coalitional Logic*: Reasoning about (local) group power.

$[C]\varphi$: coalition C has a **joint action** to bring about φ.

M. Pauly. *A Modal Logic for Coalition Powers in Games*. *Journal of Logic and Computation* **12** (2002).

From Temporal Logic to Strategy Logic

- *Coalitional Logic*: Reasoning about (local) group power.

  $[C]\varphi$: coalition C has a **joint action** to bring about φ.

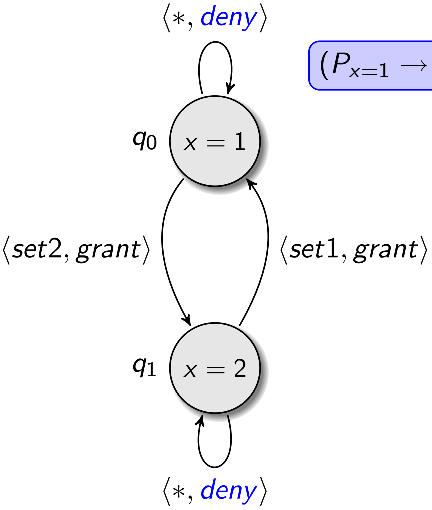
M. Pauly. *A Modal Logic for Coalition Powers in Games*. *Journal of Logic and Computation* **12** (2002).

- *Alternating-time Temporal Logic*: Reasoning about (local and global) group power:

  $\langle\langle A\rangle\rangle\Box\varphi$: The coalition A has a **joint action** to ensure that φ will remain true.

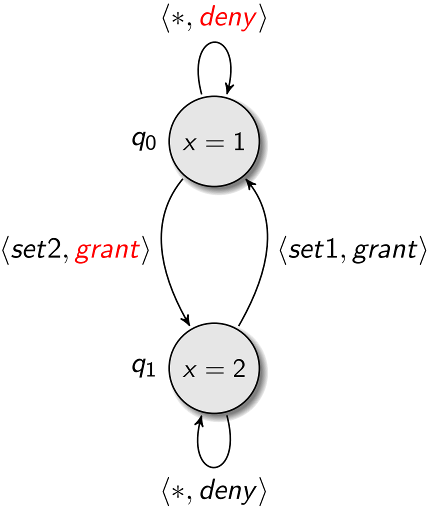
R. Alur, T. Henzinger and O. Kupferman. *Alternating-time Temproal Logic*. *Jouranl of the ACM* (2002).

Multi-agent Transition Systems



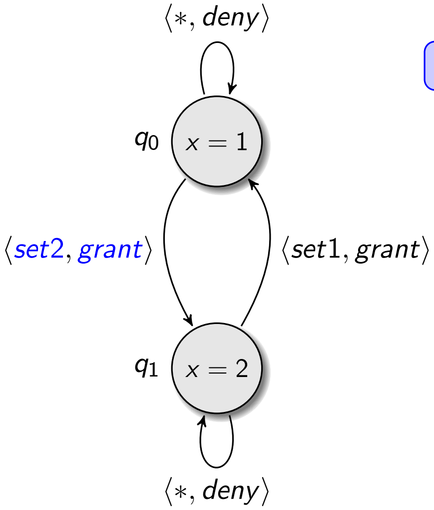
$(P_{x=1} \rightarrow [s]P_{x=1}) \wedge (P_{x=2} \rightarrow [s]P_{x=2})$

Multi-agent Transition Systems



$$P_{x=1} \rightarrow \neg[s]P_{x=2}$$

Multi-agent Transition Systems



$P_{x=1} \longrightarrow \neg[s, c]P_{x=2}$

Preference (Modal) Logics

x, y objects

$x \succeq y$: x is at least as good as y

Preference (Modal) Logics

x, y objects

$x \succeq y$: x is at least as good as y

1. $x \succeq y$ and $y \not\succeq x$ ($x \succ y$)
2. $x \not\succeq y$ and $y \succeq x$ ($y \succ x$)
3. $x \succeq y$ and $y \succeq x$ ($x \sim y$)
4. $x \not\succeq y$ and $y \not\succeq x$ ($x \perp y$)

Preference (Modal) Logics

x, y objects

$x \succeq y$: x is at least as good as y

1. $x \succeq y$ and $y \not\succeq x$ ($x \succ y$)
2. $x \not\succeq y$ and $y \succeq x$ ($y \succ x$)
3. $x \succeq y$ and $y \succeq x$ ($x \sim y$)
4. $x \not\succeq y$ and $y \not\succeq x$ ($x \perp y$)

Properties: transitivity, connectedness, etc.

Preference (Modal) Logics

Modal betterness model $\mathcal{M} = \langle W, \succeq, V \rangle$

Preference (Modal) Logics

Modal betterness model $\mathcal{M} = \langle W, \succeq, V \rangle$

Preference Modalities $\langle \succeq \rangle \varphi$: "there is a world at least as good (as the current world) satisfying φ"

$\mathcal{M}, w \models \langle \succeq \rangle \varphi$ iff there is a $v \succeq w$ such that $\mathcal{M}, v \models \varphi$

Preference (Modal) Logics

Modal betterness model $\mathcal{M} = \langle W, \succeq, V \rangle$

Preference Modalities $\langle \succeq \rangle \varphi$: "there is a world at least as good (as the current world) satisfying φ"

$\mathcal{M}, w \models \langle \succeq \rangle \varphi$ iff there is a $v \succeq w$ such that $\mathcal{M}, v \models \varphi$

$\mathcal{M}, w \models \langle \succ \rangle \varphi$ iff there is $v \succeq w$ and $w \not\succeq v$ such that $\mathcal{M}, v \models \varphi$

Preference (Modal) Logics

1. $\langle \succ \rangle \varphi \rightarrow \langle \succeq \rangle \varphi$
2. $\langle \succeq \rangle \langle \succ \rangle \varphi \rightarrow \langle \succ \rangle \varphi$
3. $\varphi \wedge \langle \succeq \rangle \psi \rightarrow (\langle \succ \rangle \psi \vee \langle \succeq \rangle (\psi \wedge \langle \succeq \rangle \varphi))$
4. $\langle \succ \rangle \langle \succeq \rangle \varphi \rightarrow \langle \succ \rangle \varphi$

Theorem The above logic (with Necessitation and Modus Ponens) is sound and complete with respect to the class of preference models.

J. van Benthem, O. Roy and P. Girard. *Everything else being equal: A modal logic approach to* ceteris paribus *preferences*. JPL, 2008.

Preference Modalities

$\varphi \geq \psi$: the state of affairs φ is at least as good as ψ (ceteris paribus)

G. von Wright. *The logic of preference*. Edinburgh University Press (1963).

From worlds to sets and back

Lifting

- $X \geq_{\forall\exists} Y$ if $\forall y \in Y\ \exists x \in X$: $x \succeq y$

From worlds to sets and back

Lifting

- $X \geq_{\forall\exists} Y$ if $\forall y \in Y\ \exists x \in X$: $x \succeq y$
  $A(\varphi \rightarrow \langle\succeq\rangle\psi)$

From worlds to sets and back

Lifting

- $X \geq_{\forall\exists} Y$ if $\forall y \in Y\ \exists x \in X$: $x \succeq y$
  $A(\varphi \rightarrow \langle \succeq \rangle \psi)$
- $X \geq_{\forall\forall} Y$ if $\forall y \in Y\ \forall x \in X$: $x \succeq y$
  $A(\varphi \rightarrow [\succ] \neg \psi)$

From worlds to sets and back

Lifting

- $X \geq_{\forall\exists} Y$ if $\forall y \in Y\ \exists x \in X$: $x \succeq y$
  $A(\varphi \rightarrow \langle\succeq\rangle\psi)$
- $X \geq_{\forall\forall} Y$ if $\forall y \in Y\ \forall x \in X$: $x \succeq y$
  $A(\varphi \rightarrow [\succ]\neg\psi)$

Deriving
$P_1 \gg P_2 \gg P_3 \gg \cdots \gg P_n$
$x > y$ iff x and y differ in at least one P_i and the first P_i where this happens is one with $P_i x$ and $\neg P_i y$

F. Liu and D. De Jongh. *Optimality, belief and preference*. 2006.

✓ Introduction, Motivation and Background

✓ Basic Ingredients for a Logic of Rational Agency

Lecture 3: Logics of Rational Agency and Social Interaction, Part I

Lecture 4: Logics of Rational Agency and Social Interaction, Part II

Lecture 5: Conclusions and General Issues

General Issues

Once a semantics and language are fixed, then standard questions can be asked: eg. develop a proof theory, completeness, decidability, model checking.

General Issues

How should we *compare* the different logical systems?

- Embedding one logic in another:

General Issues

How should we *compare* the different logical systems?

- Embedding one logic in another: *coalition logic* is a fragment of ATL ($tr([C]\varphi) = \langle\langle C\rangle\rangle \bigcirc \varphi$)

General Issues

How should we *compare* the different logical systems?

- Embedding one logic in another: *coalition logic* is a fragment of ATL ($tr([C]\varphi) = \langle\langle C \rangle\rangle \bigcirc \varphi$)

- Compare different models for a fixed language:

General Issues

How should we *compare* the different logical systems?

- Embedding one logic in another: *coalition logic* is a fragment of ATL ($tr([C]\varphi) = \langle\langle C \rangle\rangle \bigcirc \varphi$)

- Compare different models for a fixed language:
  - Alternating-Time Temporal Logics: Three different semantics for the ATL language.

V. Goranko and W. Jamroga. *Comparing Semantics of Logics for Multiagent Systems*. KRA, 2004.

General Issues

How should we *compare* the different logical systems?

- Embedding one logic in another: *coalition logic* is a fragment of ATL ($tr([C]\varphi) = \langle\langle C\rangle\rangle \bigcirc \varphi$)

- Compare different models for a fixed language:
  - Alternating-Time Temporal Logics: Three different semantics for the ATL language.

V. Goranko and W. Jamroga. *Comparing Semantics of Logics for Multiagent Systems*. KRA, 2004.

- Comparing different frameworks:

General Issues

How should we *compare* the different logical systems?

- Embedding one logic in another: *coalition logic* is a fragment of ATL ($tr([C]\varphi) = \langle\langle C \rangle\rangle \bigcirc \varphi$)

- Compare different models for a fixed language:
  - Alternating-Time Temporal Logics: Three different semantics for the ATL language.

V. Goranko and W. Jamroga. *Comparing Semantics of Logics for Multiagent Systems*. KRA, 2004.

- Comparing different frameworks: eg. PDL vs. Temporal Logic, PDL vs. STIT, STIT vs. ATL, etc.

General Issues

How should we *merge* the different logical systems?

General Issues

How should we *merge* the different logical systems?

- Combining logics is hard!

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

General Issues

How should we *merge* the different logical systems?

- Combining logics is hard!

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

Theorem $\Box\varphi \leftrightarrow \varphi$ is provable in combinations of Epistemic Logics and PDL with certain "cross axioms" ($\Box[a]\varphi \leftrightarrow [a]\Box\varphi$) (and full substitution).

R. Schmidt and D. Tishkovsky. *On combinations of propositional dynamic logic and doxastic modal logics*. JOLLI, 2008.

Merging logics of rational agency

- Reasoning about information change (knowledge and time/actions)
- Knowledge, beliefs and certainty
- "Epistemizing" logics of action and ability: *knowing how to achieve* φ vs. *knowing that you can achieve* φ
- Entangling knowledge and preferences
- Planning/intentions (BDI)

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?

Example

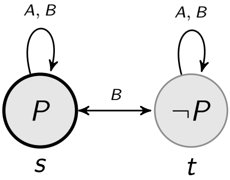
Ann would like Bob to attend her talk; however, she only wants Bob to attend if he is interested in the subject of her talk, not because he is just being polite.

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, if

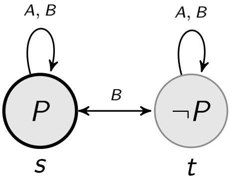
1. Ann knows about the talk.
2. Bob knows about the talk.
3. Ann knows that Bob knows about the talk.
4. Bob *does not* know that Ann knows that he knows about the talk.
5. *And nothing else.*

Example



P means "The talk is at 2PM".

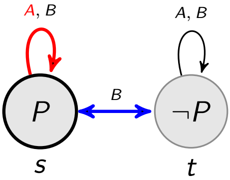
Example



P means "The talk is at 2PM".

$\mathcal{M}, s \models K_A P \wedge \neg K_B P$

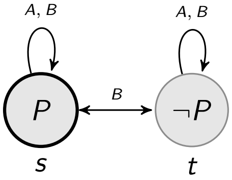
Example

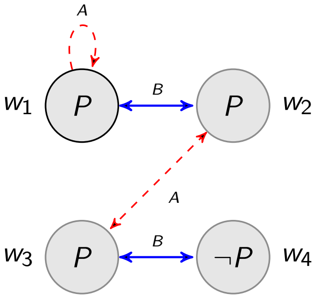


P means “The talk is at 2PM”.

$\mathcal{M}, s \models K_A P \wedge \neg K_B P$

Example



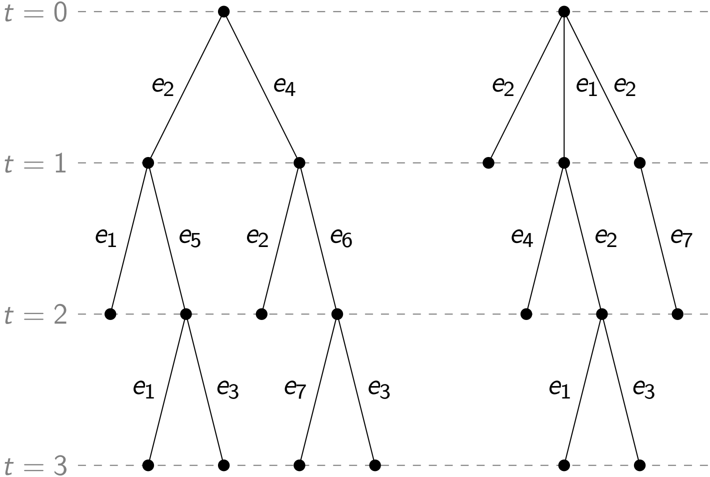


Epistemic Temporal Logic

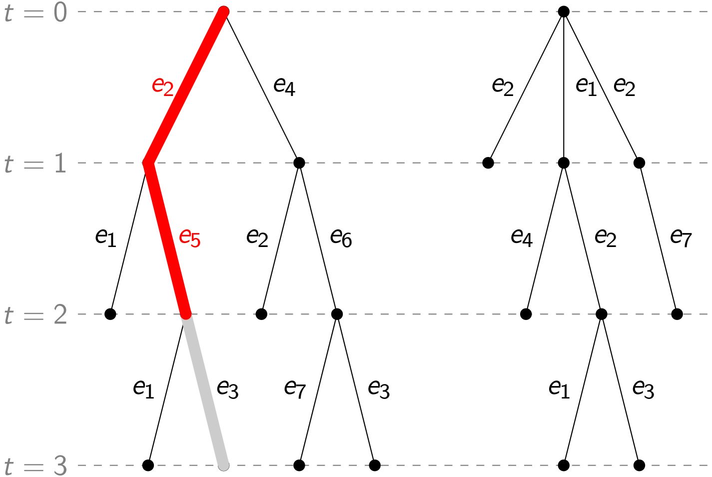
R. Parikh and R. Ramanujam. *A Knowledge Based Semantics of Messages*. *Journal of Logic, Language and Information*, 12: 453 – 467, 1985, 2003.

FHMV. Reasoning about Knowledge. MIT Press, 1995.

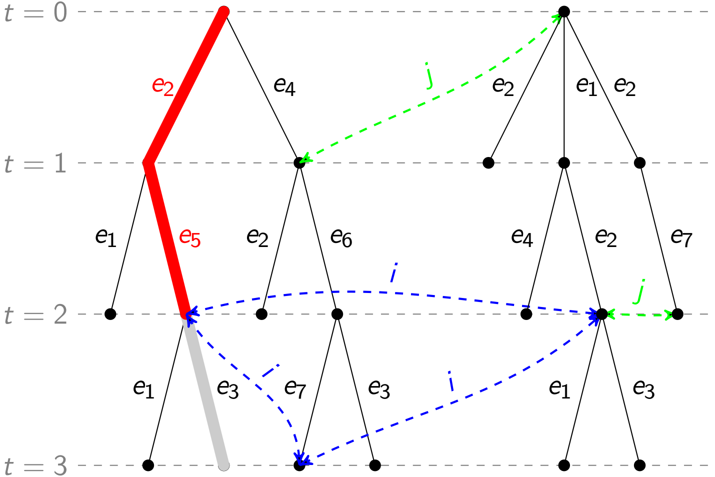
The ‘Playground’



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'.
- $\mathsf{FinPre}(\mathcal{H}) = \{H \mid \exists H' \in \mathcal{H}$ 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 $\text{FinPre}_{-\epsilon}(\mathcal{H}) \subseteq \mathcal{H}$. 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 $\mathcal{H}$.

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 $\mathcal{H}$.

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

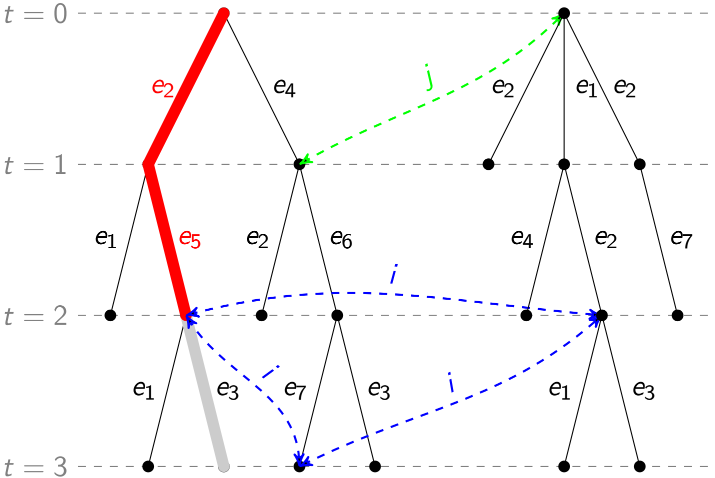
- $H, t \models P\varphi$ iff there exists $t' \leq t$ such that $H, t' \models \varphi$
- $H, t \models F\varphi$ iff there exists $t' \geq t$ such that $H, t' \models \varphi$
- $H, t \models N\varphi$ iff $H, t+1 \models \varphi$
- $H, t \models Y\varphi$ iff $t > 1$ and $H, t-1 \models \varphi$
-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



$t = 0$
$t = 1$
$t = 2$
$t = 3$
e_2
e_4
j
e_2
e_1
e_2
e_1
e_5
e_2
e_6
e_4
e_2
e_7
i
e_1
e_3
i
e_7
e_3
i
e_1
e_3

Returning to the 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.

Returning to the 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.

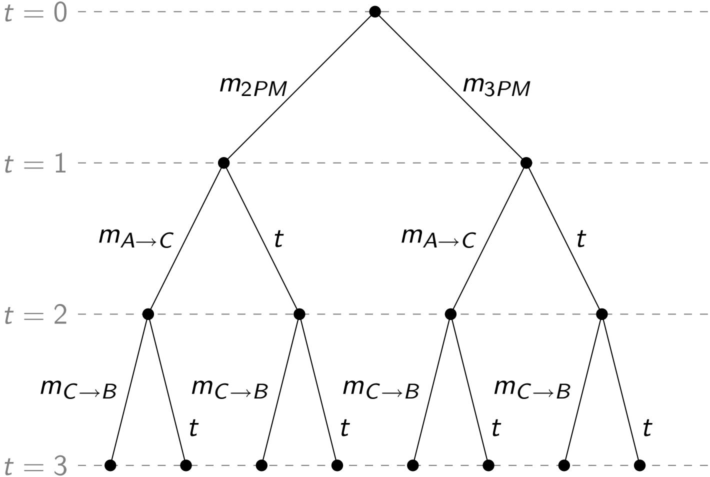
There is a very simple procedure to solve Ann's problem: *have a (trusted) friend tell Bob the time and subject of her talk.*

Returning to the Example

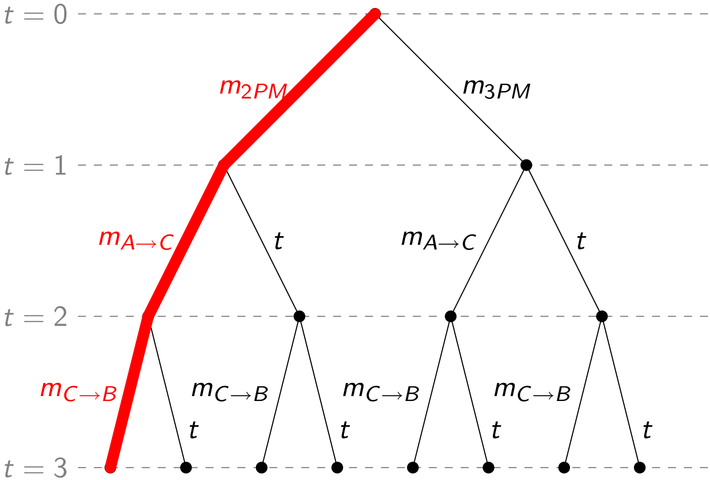
Ann would like Bob to attend her talk; however, she only wants Bob to attend if he is interested in the subject of her talk, not because he is just being polite.

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

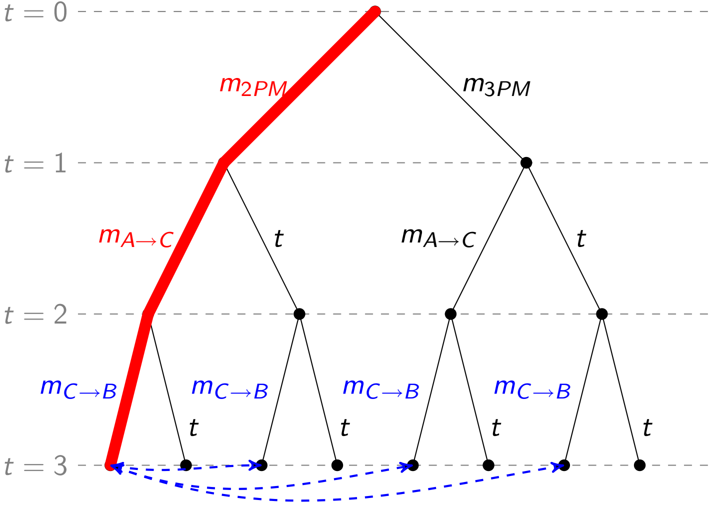
Is this procedure correct?



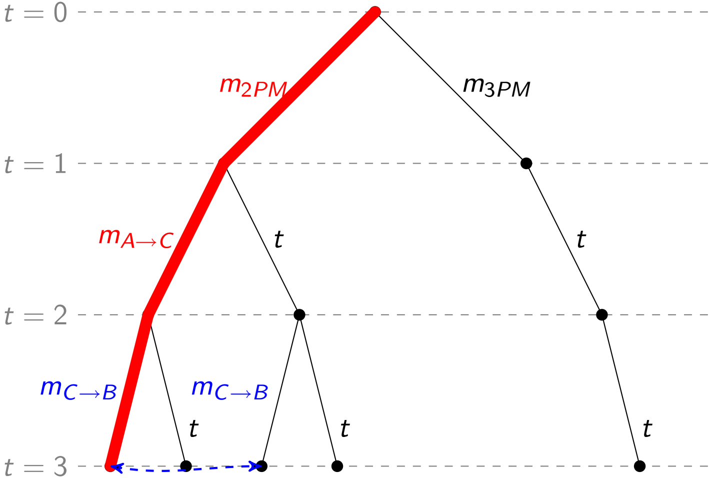
$t = 0$
$t = 1$
$t = 2$
$t = 3$
m_{2PM}
m_{3PM}
$m_{A \to C}$
t
$m_{A \to C}$
t
$m_{C \to B}$
t
$m_{C \to B}$
t
$m_{C \to B}$
t
$m_{C \to B}$
t



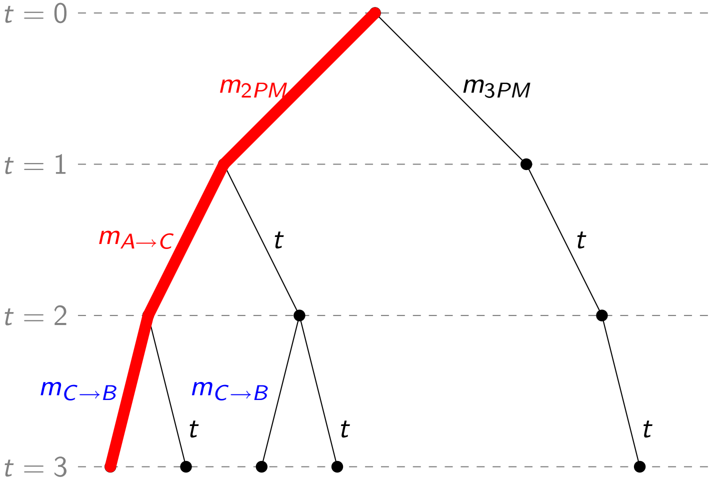
$$H, 3 \models \varphi$$



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

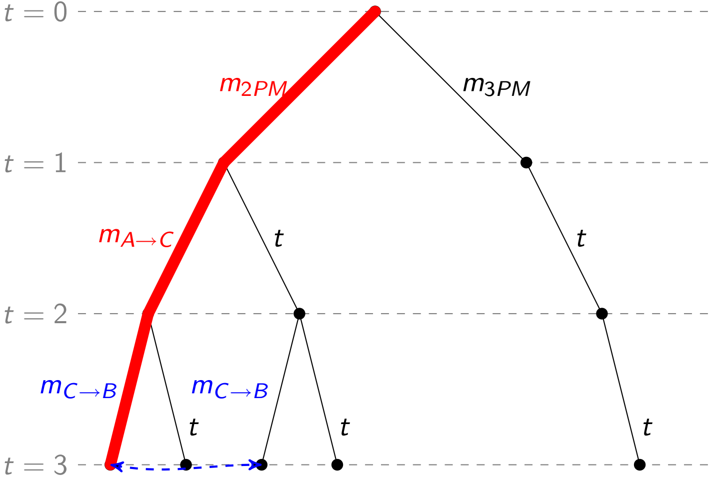


Bob's uncertainty + 'Protocol information': $H, 3 \models K_B P_{2PM}$



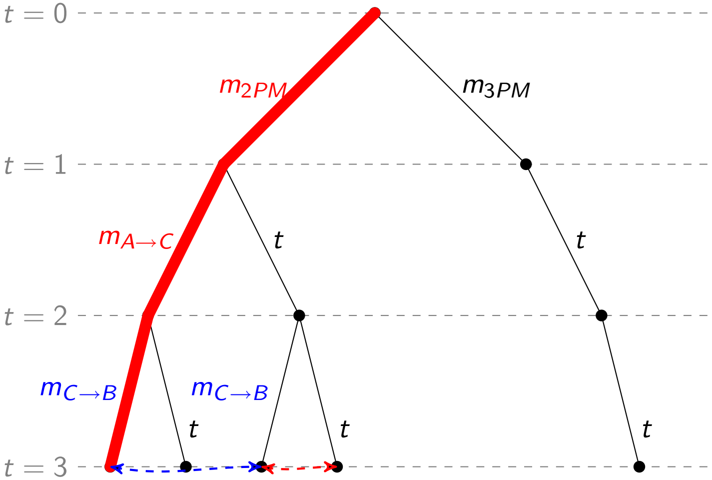
Bob's uncertainty + 'Protocol information':

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



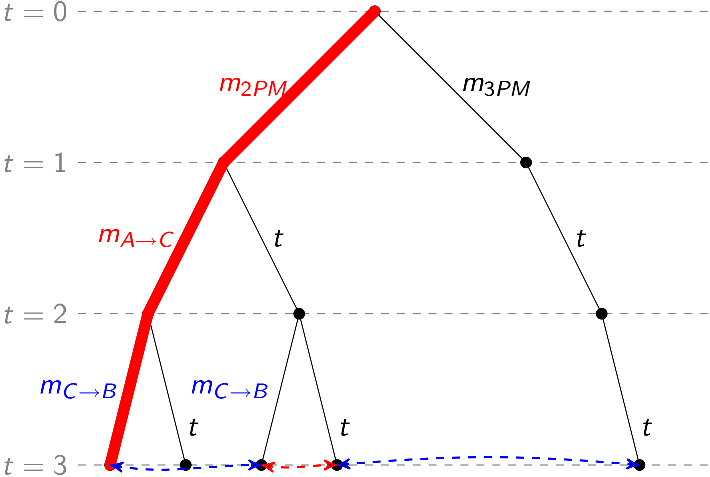
Bob's uncertainty + 'Protocol information':

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



Bob's uncertainty + 'Protocol information':

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



Bob's uncertainty + 'Protocol information':

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

Parameters of the Logical Framework

Parameters of the Logical Framework

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

Parameters of the Logical Framework

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?

Parameters of the Logical Framework

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

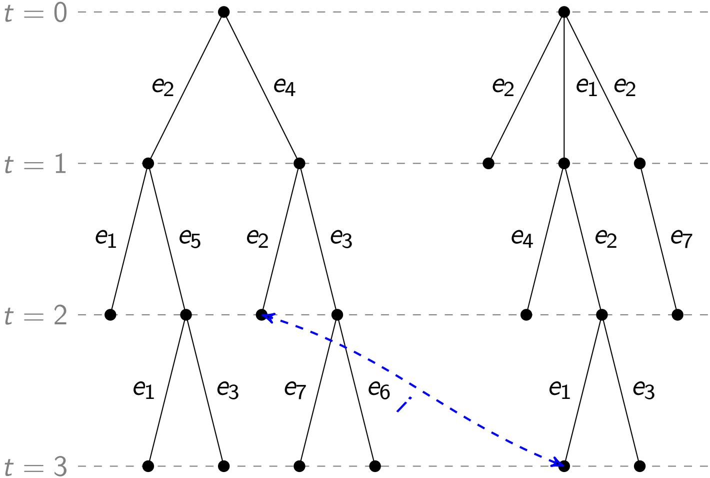
2. Structural conditions on the underlying event structure. Do we restrict to protocol frames (finitely branching trees)? Finitely branching forests? Or, arbitrary ETL frames?

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?

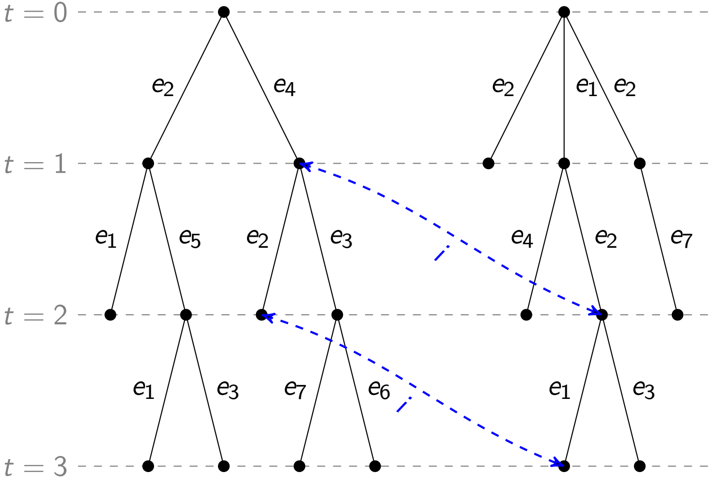
Agent Oriented Properties:

- **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 $\text{len}(H) = \text{len}(H')$.

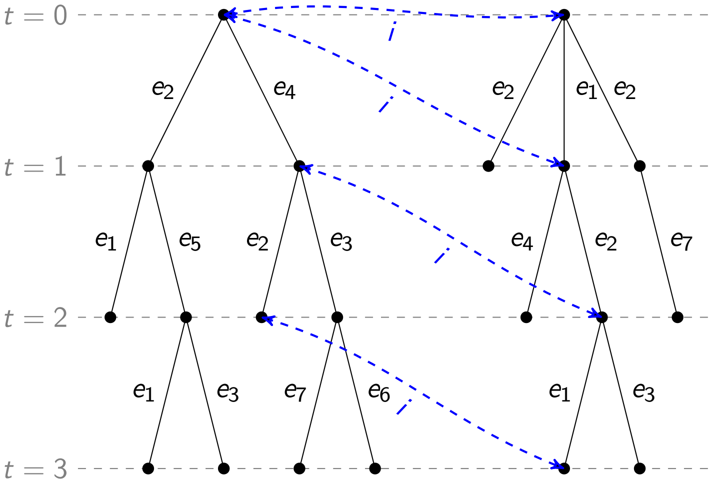
Perfect Recall



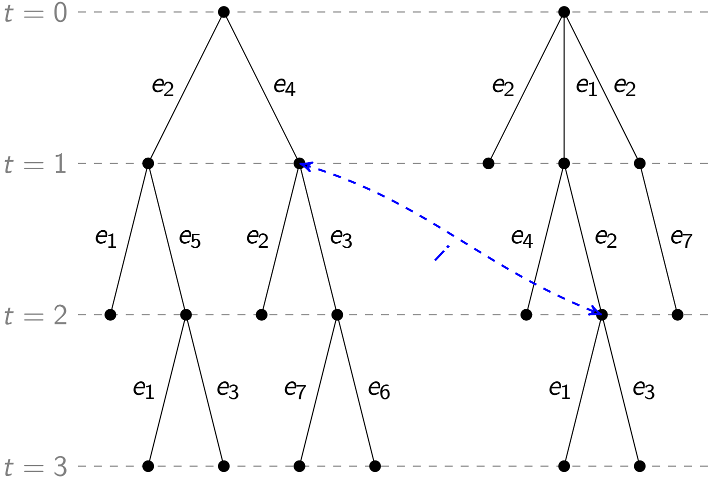
Perfect Recall



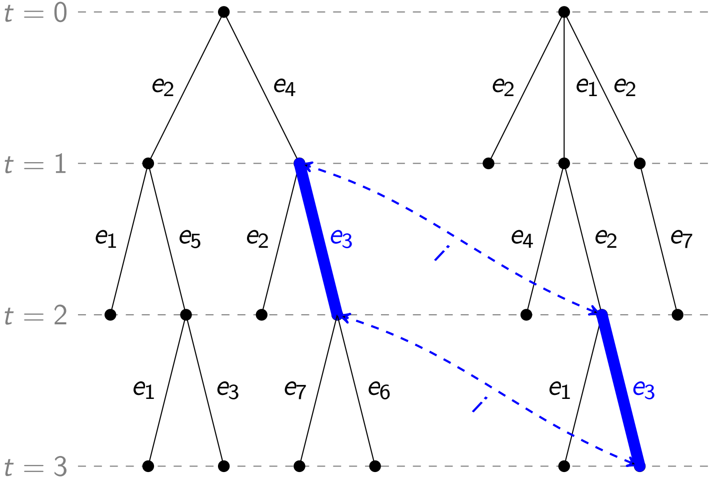
Perfect Recall



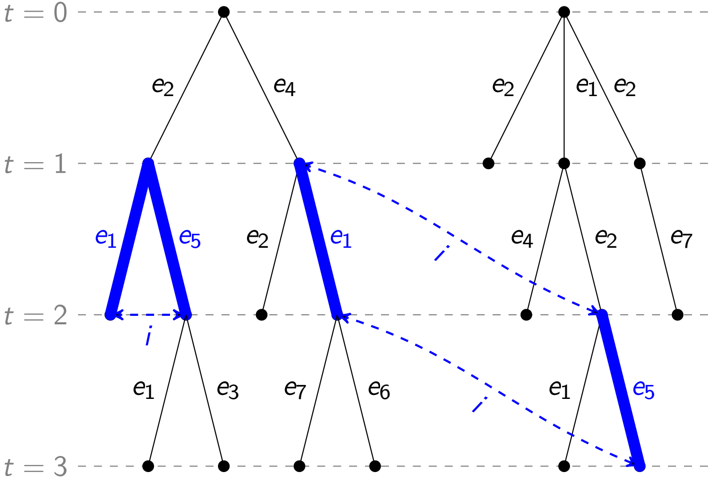
No Miracles



No Miracles



No Miracles



Ideal Agents

Assume there are two agents

Theorem

The logic of ideal *agents with respect to a language with common knowledge and future is highly undecidable (for example, by assuming perfect recall).*

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

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

End of lecture 3.