Logics of Rational Agency

Lecture 4

Eric Pacuit

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

July 30, 2009
Plan for the Course

✓ Introduction, Motivation and Background

✓ Basic Ingredients for a Logic of Rational Agency

✓ 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
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 $\varphi$ vs. knowing that you can achieve $\varphi$
- Entangling knowledge and preferences
- Planning/intentions (BDI)
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.
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

\[ (s, e_1) \quad P \quad (s, e_2) \quad P \quad (s, e_3) \quad P \quad (t, e_3) \quad \neg P \]
Product Update
Product Update

\[(s, e_1) \models \neg K_B K_A K_B P\]  
\[(s, e_1) \models P\]  
\[(s, e_3) \models P\]  
\[(t, e_3) \models \neg P\]
Product Update

\[(s, e_1) \models \neg K_B K_A K_B P\]

\[(s, e_1) \quad B \quad P \quad (s, e_2)\]

\[(s, e_3) \quad P \quad \neg P \quad (t, e_3)\]
Product Update

\[(s, e_1) \models \neg K_B K_A K_B P\]
Product Update

\[
(s, e_1) \models \neg K_B K_A K_B P
\]

\[
(s, e_1) \quad P \quad B \quad P \quad (s, e_2)
\]

\[
A \quad B
\]

\[
(s, e_3) \quad P \quad B \quad \neg P \quad (t, e_3)
\]
Product Update Details

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

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

Let \( \mathcal{M} = \langle W, R, V \rangle \) be a Kripke model.

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

The **update model** \( \mathcal{M} \otimes \mathcal{A} = \langle W', R', V' \rangle \) where
Let $\mathcal{M} = \langle W, R, V \rangle$ be a Kripke model.

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

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

$W' = \{(w, a) \mid w \models Pre(a)\}$
Product Update Details

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

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

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

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

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

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

The update model $M \otimes A = \langle W', R', V' \rangle$ 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 $\mathcal{M} = \langle W, R, V \rangle$ be a Kripke model.

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

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

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

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


Example: Public Announcement Logic


J. van Benthem. *One is a lonely number*. 2002.
Example: Public Announcement Logic

The **Public Announcement Language** is generated by the following grammar:

\[ p \mid \neg \varphi \mid \varphi \land \varphi \mid K_i \varphi \mid C \varphi \mid [\psi] \varphi \]

where \( p \in \text{At} \) and \( i \in \mathcal{A} \). 

- \([\psi] \varphi\) is intended to mean "After publicly announcing \( \psi \), \( \varphi \) is true".
- \([P] K_i P\): "After publicly announcing \( P \), agent \( i \) knows \( P \)"
- \([\neg K_i P]\) \( C P\): "After announcing that agent \( i \) does not know \( P \), then \( P \) is common knowledge"
- \([\neg K_i P]\) \( K_i P\): "after announcing \( i \) does not know \( P \), then \( i \) knows \( P \). "

Eric Pacuit: LORI, Lecture 4
Example: Public Announcement Logic

The Public Announcement Language is generated by the following grammar:

\[ p \mid \neg \varphi \mid \varphi \land \varphi \mid K_i \varphi \mid C \varphi \mid [\psi] \varphi \]

where \( p \in \text{At} \) and \( i \in A \).

- \([\psi] \varphi\) is intended to mean “After publicly announcing \( \psi \), \( \varphi \) is true”. 

Example: Public Announcement Logic

The Public Announcement Language is generated by the following grammar:

\[ p \mid \neg \varphi \mid \varphi \land \varphi \mid K_i \varphi \mid C \varphi \mid [\psi] \varphi \]

where \( p \in \text{At} \) and \( i \in \mathcal{A} \).

- \( [P]K_i P \): “After publicly announcing \( P \), agent \( i \) knows \( P \)”
Example: Public Announcement Logic

The Public Announcement Language is generated by the following grammar:

\[ p \mid \neg \varphi \mid \varphi \land \varphi \mid K_i \varphi \mid C \varphi \mid [\psi] \varphi \]

where \( p \in \text{At} \) 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:

\[ p \mid \neg \varphi \mid \varphi \land \varphi \mid K_i \varphi \mid C \varphi \mid [\psi] \varphi \]

where \( p \in \text{At} \) 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 $\mathcal{M} = \langle W, \{R_i\}_{i \in A}, V \rangle$ is a multi-agent Kripke Model

$\mathcal{M}, w \models [\psi] \varphi$ iff $\mathcal{M}, w \models \psi$ 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 $p \in At$, $V'(p) = V(p) \cap W'$
Example: Public Announcement Logic

\[ [\psi] p \iff (\psi \rightarrow p) \]
Example: Public Announcement Logic

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

\[
[\psi]\neg \phi \leftrightarrow (\psi \rightarrow \neg [\psi]\phi)
\]
Example: Public Announcement Logic

\[
[\psi]p \iff (\psi \rightarrow p)
\]

\[
[\psi]\neg \varphi \iff (\psi \rightarrow \neg [\psi]\varphi)
\]

\[
[\psi](\psi \land \chi) \iff ([\varphi]\psi \land [\varphi]\chi)
\]
Example: Public Announcement Logic

\[
\begin{align*}
[\psi]p & \iff \ (\psi \to p) \\
[\psi]\neg \varphi & \iff \ (\psi \to \neg [\psi] \varphi) \\
[\psi](\psi \land \chi) & \iff \ ([\varphi] \psi \land [\varphi] \chi) \\
[\psi][\varphi] \chi & \iff \ [\psi \land [\psi] \varphi] \chi
\end{align*}
\]
Example: Public Announcement Logic

\[
[\psi]p \leftrightarrow (\psi \rightarrow p)
\]
\[
[\psi]\neg \varphi \leftrightarrow (\psi \rightarrow \neg[\psi]\varphi)
\]
\[
[\psi](\psi \land \chi) \leftrightarrow ([\varphi]\psi \land [\varphi]\chi)
\]
\[
[\psi][\varphi]\chi \leftrightarrow [\psi \land [\psi]\varphi]\chi
\]
\[
[\psi]K_i\varphi \leftrightarrow (\psi \rightarrow K_i[\psi]\varphi)
\]
Example: Public Announcement Logic

\[
\begin{align*}
[\psi]p & \iff (\psi \to p) \\
[\psi]\neg \varphi & \iff (\psi \to \neg[\psi]\varphi) \\
[\psi](\psi \land \chi) & \iff ([\varphi]\psi \land [\varphi]\chi) \\
[\psi][\varphi]\chi & \iff [\psi \land [\psi]\varphi]\chi \\
[\psi]K_i\varphi & \iff (\psi \to K_i[\psi]\varphi)
\end{align*}
\]

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

\[
\begin{align*}
[\psi)p & \iff (\psi \rightarrow p) \\
[\psi]\neg \varphi & \iff (\psi \rightarrow \neg [\psi]\varphi) \\
[\psi](\psi \land \chi) & \iff (\varphi)\psi \land [\psi]\chi) \\
[\psi][\varphi]\chi & \iff [\psi \land [\psi]\varphi]\chi \\
[\psi]K_i\varphi & \iff (\psi \rightarrow K_i[\psi]\varphi)
\end{align*}
\]

The situation is more complicated with common knowledge.

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?

**Observation:** By repeatedly updating an epistemic model with event models, the machinery of DEL creates ETL models.
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 $P$, $\text{forest}(M, P)$, represents all possible evolutions of the system obtained by updating $M$ with sequences from $P$. 
Example: Initial Model and Protocol
Example

\[
\begin{align*}
\text{Example} & : & \text{LORI, Lecture 4} & 17
\end{align*}
\]
Example

(s, P, Q) \rightarrow (t, P, R) \leftarrow (u, P, R) \rightarrow (v, P, R)

\( i \)

\( j \)

\( s \rightarrow t \rightarrow u \rightarrow v \)

\(!P\)

\(!Q\)

\(!R\)
Example

\[(s) \quad (t) \quad (u) \quad (v)\]

\[\neg P \quad \neg P \quad \neg P\]

\[(s, \neg P) \quad (t, \neg P) \quad (u, \neg P)\]

\[\neg Q \quad \neg Q \quad \neg R\]

\[(s, \neg P, \neg Q) \quad (t, \neg P, \neg Q) \quad (u, \neg P, \neg R) \quad (t, \neg P, \neg R) \quad (u, \neg P, \neg R)\]
Example

Eric Pacuit: LORI, Lecture 4
Example

\[ (s) \overset{(s)}{\longrightarrow} (t) \overset{i}{\longrightarrow} (t) \overset{j}{\longrightarrow} (u) \overset{j}{\longrightarrow} (v) \]

\[ !P \quad !P \quad !P \]

\[ (s, !P) \overset{(s)}{\longrightarrow} (t, !P) \overset{i}{\longrightarrow} (u, !P) \]

\[ !Q \quad !Q \quad !R \]

\[ (s, !P, !Q) \overset{(s)}{\longrightarrow} (t, !P, !Q) \overset{i}{\longrightarrow} (u, !P, !R) \]

\[ !Q \quad !R \]
Example
Example

(s) \rightarrow (t) \rightarrow (u) \rightarrow (v)

(!P) \rightarrow (!P) \rightarrow (!P) \rightarrow (!P)

(s, !P) \rightarrow (t, !P) \rightarrow (u, !P)

(!Q) \rightarrow (!Q) \rightarrow (!R) \rightarrow (!R)

(s, !P, !Q) \rightarrow (t, !P, !Q) \rightarrow (t, !P, !R) \rightarrow (u, !P, !R)
Example
Example

(s) \Rightarrow (t) \Rightarrow (u) \Rightarrow (v)

(t) \models R \land \neg \langle!R\rangle \top

(s)!P | (s, !P) | (s, !P, !Q)

(t)!P | (t, !P) | (t, !P, !Q)

(t, !P)!Q \xrightarrow{!Q} \neg | (t, !P, !Q)

(t, !P, !R) \xrightarrow{!R} \neg | (t, !P, !R)

(u, !P)!R | (u, !P, !R)

(u, !P, !Q) \xrightarrow{!Q} \neg | (u, !P, !Q)

(u, !P, !Q, !R) | (u, !P, !Q, !R)
State-Dependent Protocols

The ETL models $\mathcal{F}(M, P)$ in the previous 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 $\text{pre}(e)$ is true at $h$. 

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

Eric Pacuit: LORI, Lecture 4
State-Dependent Protocols

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

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

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

**Theorem (Main Representation Theorem)**

Let $\Sigma$ be a finite set of events and suppose $X_{\text{uni}}^{\text{DEL}}$ is the class of uniform DEL protocols (with a finiteness condition). A model is in $\mathcal{F}(X_{\text{DEL}}^{\text{uni}})$ iff it satisfies propositional stability, synchronicity, perfect recall, local no miracles, and local bisimulation invariance.
Bisimulation Invariance + Finiteness Condition

$t = 0$

$t = 1$

$t = 2$

$t = 3$
Bisimulation Invariance + Finiteness Condition

t = 0

\[ e_2 \]

\[ e_4 \]

\[ e_5 \]

\[ e_1 \]

\[ e_1 \]

\[ e_6 \]

\[ e_7 \]

\[ e_1 \]

\[ e_5 \]

\[ e_2 \]

\[ e_7 \]

\[ e_2 \]

\[ e_1 \]

\[ e_2 \]

\[ e_7 \]
Recall that if $X$ is a set of DEL protocols, we define $F(X) = \{F(M, P) \mid M$ an epistemic model and $P \in X\}$. 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?

1. $A \rightarrow \langle A \rangle^T$ vs. $\langle A \rangle^T \rightarrow A$
Announcement + Protocol Information

1. $A \rightarrow \langle A \rangle^T$ vs. $\langle A \rangle^T \rightarrow A$

2. $\langle A \rangle K_i P \leftrightarrow A \land K_i \langle A \rangle P$
Announcement + Protocol Information

1. \( A \rightarrow \langle A \rangle^T \) vs. \( \langle A \rangle^T \rightarrow A \)

2. \( \langle A \rangle K_i P \leftrightarrow A \land K_i \langle A \rangle P \)

3. \( \langle A \rangle K_i P \leftrightarrow \langle A \rangle^T \land K_i (A \rightarrow \langle A \rangle P) \)
Announcement + Protocol Information

1. $A \rightarrow \langle A \rangle^\top$ vs. $\langle A \rangle^\top \rightarrow A$

2. $\langle A \rangle K_i P \leftrightarrow A \land K_i \langle A \rangle P$

3. $\langle A \rangle K_i P \leftrightarrow \langle A \rangle^\top \land K_i (A \rightarrow \langle A \rangle P)$

4. $\langle A \rangle K_i P \leftrightarrow \langle A \rangle^\top \land K_i (\langle A \rangle^\top \rightarrow \langle A \rangle P)$
Announcement + Protocol Information

1. \( A \rightarrow \langle A \rangle^T \) vs. \( \langle A \rangle^T \rightarrow A \)

2. \( \langle A \rangle K_i P \leftrightarrow A \land K_i \langle A \rangle P \)

3. \( \langle A \rangle K_i P \leftrightarrow \langle A \rangle^T \land K_i (A \rightarrow \langle A \rangle P) \)

4. \( \langle A \rangle K_i P \leftrightarrow \langle A \rangle^T \land K_i (\langle A \rangle^T \rightarrow \langle A \rangle P) \)

**Theorems** Sound and complete axiomatizations of various generated ETL models.
Reasoning with Protocols
Reasoning with Protocols: An Example

1. Uma is a physician whose neighbour is ill. Uma does not know and has not been informed. Uma has no obligation (as yet) to treat the neighbour.

2. Uma is a physician whose neighbour Sam is ill. The neighbour’s daughter Ann comes to Uma’s house and tells her. Now Uma does have an obligation to treat Sam, or perhaps call in an ambulance or a specialist.

3. Mary is a patient in St. Gibson’s hospital. Mary is having a heart attack. The caveat which applied in case 1. does not apply here. The hospital has an obligation to be aware of Mary’s condition at all times and to provide emergency treatment as appropriate.

Reasoning with Protocols: An Example

1. Uma is a physician whose neighbour is ill. Uma does not know and has not been informed. Uma has no obligation (as yet) to treat the neighbour.

2. Uma is a physician whose neighbour Sam is ill. The neighbour’s daughter Ann comes to Uma’s house and tells her. Now Uma does have an obligation to treat Sam, or perhaps call in an ambulance or a specialist.

3. Mary is a patient in St. Gibson’s hospital. Mary is having a heart attack. The caveat which applied in case 1. does not apply here. The hospital has an obligation to be aware of Mary’s condition at all times and to provide emergency treatment as appropriate.

Reasoning with Protocols: An Example

1. Uma is a physician whose neighbour is ill. Uma does not know and has not been informed. Uma has no obligation (as yet) to treat the neighbour.

2. Uma is a physician whose neighbour Sam is ill. The neighbour’s daughter Ann comes to Uma’s house and tells her. Now Uma does have an obligation to treat Sam, or perhaps call in an ambulance or a specialist.

3. Mary is a patient in St. Gibson’s hospital. Mary is having a heart attack. The caveat which applied in case 1. does not apply here. The hospital has an obligation to be aware of Mary’s condition at all times and to provide emergency treatment as appropriate.

Reasoning with Protocols: An Example

1. Uma is a physician whose neighbour is ill. Uma does not know and has not been informed. Uma has no obligation (as yet) to treat the neighbour.

2. Uma is a physician whose neighbour Sam is ill. The neighbour’s daughter Ann comes to Uma’s house and tells her. Now Uma does have an obligation to treat Sam, or perhaps call in an ambulance or a specialist.

3. Mary is a patient in St. Gibson’s hospital. Mary is having a heart attack. The caveat which applied in case 1. does not apply here. The hospital has an obligation to be aware of Mary’s condition at all times and to provide emergency treatment as appropriate.

Example 1 & 2

$t = 0$

$t = 1$

$t = 2$

$t = 3$
Example 1 & 2

\[ t = 0 \]

\[ t = 1 \]

\[ t = 2 \]

\[ t = 3 \]
Example 1 & 2

\[ \lambda_u(v) = \lambda_u(c) \]

\( t = 0 \)

\( t = 1 \)

\( t = 2 \)

\( t = 3 \)
Example 1 & 2
Example 1 & 2

$H, 1 \models \neg K_u S$

$t = 0$

$t = 1$

$t = 2$

$t = 3$
Example 1 & 2

\[ \lambda_u(vm) = \lambda_u(cm) \]
Example 2

\[ \lambda_u(vm) = \lambda_u(cm) \]
Example 2

\[ t = 0 \]

\[ t = 1 \]

\[ t = 2 \]

\[ t = 3 \]

2 1 2 1 2
Example 2

$t = 0$

$t = 1$

$t = 2$

$t = 3$

$H, 2 \models K_u G(t)$
Ann has the (knowledge based) obligation to tell Uma about her father’s illness ($K_a G(m)$).
Ann has the (knowledge based) obligation to tell Uma about her father’s illness ($K_a G(m)$).

Clearly, Ann will not be under any obligation to tell Uma that her father is ill, if Ann justifiably believes that Uma would not treat her father even if she knew of his illness.
Ann has the (knowledge based) obligation to tell Uma about her father’s illness \((K_a G(m))\).

Clearly, Ann will not be under any obligation to tell Uma that her father is ill, if Ann justifiably believes that Uma would not treat her father even if she knew of his illness.

Thus, to carry out a deduction we will need to assume

\[ K_a(K_u \text{sick} \leftrightarrow \circ \text{treat}) \]
A similar assumption is needed to derive that Jill has an obligation to treat Sam.
A similar assumption is needed to derive that Jill has an obligation to treat Sam.

Obviously, if Uma has a good reason to believe that Ann always lies about her father being ill, then she is under no obligation to treat Sam.
A similar assumption is needed to derive that Jill has an obligation to treat Sam.

Obviously, if Uma has a good reason to believe that Ann always lies about her father being ill, then she is under no obligation to treat Sam.

In other words, we need to assume

$$K_u(msg \leftrightarrow \text{sick})$$
Common Knowledge of Ethicality

These formulas can all be derived for one common assumption which we call *Common Knowledge of Ethicality*. 
Common Knowledge of Ethicality

These formulas can all be derived for one common assumption which we call Common Knowledge of Ethicality.

1. The agents must (commonly) know the protocol.
2. The agents are all of the same “type” (social utility maximizers)
Issue: Group Knowledge

Communication/observation + protocol information leads to group knowledge.
Achieving Group Knowledge

- $\mathcal{M}, w \models C\varphi$ iff for each $w'$, if $w \sim_* w'$ then $\mathcal{M}, w' \models \varphi$ ($\sim_*$ is the reflexive transitive closure of the union of each agent’s accessibility relation)

- $\mathcal{M}, w \models D\varphi$ iff for each $w' \in D(\mathcal{M})$, if $w \sim_i w'$ for each $i \in A$, then $\mathcal{M}, w' \models \varphi$. 

Theorem If every agent ‘says all she knows’ (i.e., ‘I am in this partition cell’) then distributed knowledge is turned into common knowledge.

J. van Benthem.

One is a lonely number. 2002.
Achieving Group Knowledge

\[ \mathcal{M}, w \models \Sigma \varphi \text{ iff for each } w', \text{ if } w \sim_* w' \text{ then } \mathcal{M}, w' \models \varphi \text{ (} \sim_* \text{ is the reflexive transitive closure of the union of each agent's accessibility relation)} \]

\[ \mathcal{M}, w \models D\varphi \text{ iff for each } w' \in D(\mathcal{M}), \text{ if } w \sim_i w' \text{ for each } i \in \mathcal{A}, \text{ then } \mathcal{M}, w' \models \varphi. \]

**Theorem** If every agent ‘says all she knows’ (i.e., ‘I am in this partition cell’) then distributed knowledge is turned into common knowledge.

J. van Benthem. *One is a lonely number*. 2002.
Achieving Group Knowledge

“honest” public announcement: the speaker of the announcement believes what he announces (preconditions of $\varphi$ is $\varphi \land K_i\varphi$)
Achieving Group Knowledge

“honest” public announcement: the speaker of the announcement believes what he announces (preconditions of \( \varphi \) is \( \varphi \land K_i \varphi \))

We denote the protocol of honest communication, that uses all and only public announcements with preconditions of this form by ProtocolHonest.
Achieving Group Knowledge

“honest” public announcement: the speaker of the announcement believes what he announces (preconditions of $\varphi$ is $\varphi \land K_i\varphi$)

We denote the protocol of honest communication, that uses all and only public announcements with preconditions of this form by ProtocolHonest.

**Theorem** For all $\mathcal{M}$ in which all $\sim_i$ are equivalence relations, and each $\varphi$ that is purely epistemic (that is, it does not contain temporal operators) it holds that:

$$\text{Forest}(\mathcal{M}, \text{ProtocolHonest}) \models I\varphi \leftrightarrow GI\varphi$$
Achieving Group Knowledge (unreliable messages)

Classic example: email, generals problem.
Achieving Group Knowledge (unreliable messages)

Classic example: email, generals problem.

Theorem
In all $S_5$ models $M$, it holds for all $\phi$ in which epistemic operators occur only positively:

$$\text{Forest}(M, \text{ProtocolInsecure}) \models \leftrightarrow \text{GC}$$
Achieving Group Knowledge (unreliable messages)

Classic example: email, generals problem.

![Diagram](image)

**Theorem** In all S5 models $\mathcal{M}$, it holds for all $\varphi$ in which epistemic operators occur only positively:

$\text{Forest}(\mathcal{M}, \text{ProtocolInsecure}) \models C\varphi \leftrightarrow GC\varphi$
Many Issues!

▶ Can group knowledge be achieved in a finite number of steps?
Can group knowledge be achieved in a finite number of steps? (Parikh; Heifetz and Samet: No! : )
Many Issues!

- Can group knowledge be achieved in a finite number of steps? (Parikh; Heifetz and Samet: No! :)

- Protocol involves not only the type of announcement, but who can say what to whom...
Many Issues!

- Can group knowledge be achieved in a finite number of steps? (Parikh; Heifetz and Samet: No!)

- Protocol involves not only the type of announcement, but who can say what to whom...

- What is the logic of specific protocols (in languages with group knowledge operators)?
Many Issues!

- Can group knowledge be achieved in a finite number of steps? (Parikh; Heifetz and Samet: No!)

- Protocol involves not only the type of announcement, but who can say what to whom...

- What is the logic of specific protocols (in languages with group knowledge operators)?

- New notions of group knowledge?
Reasoning about protocols

What type of events *change* the protocol?

Do the agents *know* the protocol?
What is a Protocol?

Given the full tree $T$ of events, a protocol is any subtree of $T$. 

- **Physical properties:** every message is eventually answered, no message is received before it is sent.
- **Agent types:** agent $i$ is the type of agent who always lies, agent $j$ is the type who always tells the truth.

A protocol is the set of histories of an extensive game consistent with a (partial) strategy profile.
What is a Protocol?

Given the full tree $T$ of events, a protocol is any subtree of $T$.

- A protocol is the set of histories compatible with some process, i.e., it is the “unwinding” of a (multi-agent) state transition system.
What is a Protocol?

Given the full tree $T$ of events, a protocol is any subtree of $T$.

- A protocol is the set of histories compatible with some process, i.e., it is the “unwinding” of a (multi-agent) state transition system.

- A protocol is the set of histories satisfying some property:
  - Physical properties: every message is eventually answered, no message is received before it is sent
  - Agent types: agent $i$ is the type of agent who always lies, agent $j$ is the type who always tells the truth
What is a Protocol?

Given the full tree $T$ of events, a protocol is any subtree of $T$.

- A protocol is the set of histories compatible with some process, i.e., it is the “unwinding” of a (multi-agent) state transition system.

- A protocol is the set of histories satisfying some property:
  - Physical properties: every message is eventually answered, no message is received before it is sent
  - Agent types: agent $i$ is the type of agent who always lies, agent $j$ is the type who always tells the truth

- A protocol is the set of histories of an extensive game consistent with a (partial) strategy profile.
Defining a Protocol

1. What formal language should we use to define the protocol?
2. What models do we have in mind?

Given a formula $\phi$, two ways to think about defining a protocol:

- Set of histories: the set $P$ in the full event tree $T$ such that $h \in P$ iff $h| = \phi$

- Set of models: the set $\text{Mod}(\phi)$ (the set of models of $\phi$)

A Liar: $((K_i \phi) ?; !\neg \phi) \cup (K_i \neg \phi) ?; !\phi) \cup \text{skip}^*$
Defining a Protocol

1. What formal language should we use to define the protocol?
2. What models do we have in mind?
Defining a Protocol

1. What formal language should we use to define the protocol?
2. What models do we have in mind?

Given a formula $\varphi$, two ways to think about defining a protocol:

**Set of histories:** the set of histories $P$ in the full event tree $T$ such that $h \in P$ iff $h \models \varphi$

**Set of models:** the set $\mathrm{Mod}(\varphi)$ (the set of models of $\varphi$)
Defining a Protocol

1. What formal language should we use to define the protocol?

2. What models do we have in mind?

Given a formula $\varphi$, two ways to think about defining a protocol:

**Set of histories:** the set of histories $P$ in the full event tree $T$ such that $h \in P$ iff $h \models \varphi$

**Set of models:** the set $\text{Mod}(\varphi)$ (the set of models of $\varphi$)

A Liar: $((K_1 \varphi; \not\varphi) \cup (K_1 \neg \varphi; \not\varphi) \cup \text{skip})^*$

Two types of uncertainty?

Given two finite histories $h$ and $h'$,

$h \sim_i h'$ means given the events $i$ has observed, $h$ and $h'$ are indistinguishable.
Two types of uncertainty?

Given two maximal histories $H$ and $H'$,

agent $i$ may be uncertain which of the two will be the final outcome.
Protocol/Procedural information

▶ What type of events *change* the protocol?

▶ How should we model the protocol information?
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 \( \varphi \) vs. knowing that you can achieve \( \varphi \)
- Entangling knowledge and preferences
- Planning/intentions (BDI)
Logics of Beliefs and Preference

\[ K(\varphi \succeq \psi) : \text{“Ann knows that } \varphi \text{ is at least as good as } \psi \text{”} \]

\[ K\varphi \succeq K\psi : \text{“knowing } \varphi \text{ is at least as good as knowing } \psi \text{”} \]
Logics of Beliefs and Preference

$K(\varphi \succeq \psi)$: “Ann knows that $\varphi$ is at least as good as $\psi$”

$K\varphi \succeq K\psi$: “knowing $\varphi$ is at least as good as knowing $\psi$”

$\mathcal{M} = \langle W, \sim, \succeq, V \rangle$
Logics of Beliefs and Preference

\[ K(\varphi \succeq \psi) : \text{“Ann knows that } \varphi \text{ is at least as good as } \psi \text{”} \]

\[ K\varphi \succeq K\psi : \text{“knowing } \varphi \text{ is at least as good as knowing } \psi \]

\[ \mathcal{M} = \langle W, \sim, \succeq, V \rangle \]


\[ A(\psi \rightarrow \langle \preceq \rangle \varphi) \quad \text{vs.} \quad K(\psi \rightarrow \langle \preceq \rangle \varphi) \]
\[ A(\psi \rightarrow \langle \preceq \rangle \varphi) \text{ vs. } K(\psi \rightarrow \langle \preceq \rangle \varphi) \]

Should preferences be restricted to information sets?
A(ψ → ⟨⪰⟩φ) vs. K(ψ → ⟨⪰⟩φ)

Should preferences be restricted to information sets?

M, w \models ⟨⪰ \cap ∼⟩φ iff there is a v with w ∼ v and w ⪯ v such that M, v \models φ

K(ψ → ⟨⪰ \cap ∼⟩φ)
Defining Beliefs from Preferences

Starting with the work of Savage (based on Ramsey and de Finetti), there is a tradition in game theory and decision theory to define beliefs and utilities in terms of the agent’s preferences.
Defining Beliefs from Preferences

- Starting with the work of Savage (based on Ramsey and de Finetti), there is a tradition in game theory and decision theory to define beliefs and utilities in terms of the agent’s preferences.

- Typically the results come in the form of a representation theorem:

  *If the agents preferences satisfy such-and-such properties, then there is a set of conditional probability functions and a (state independent) utility function such that the agent can be assumed to act as an expected utility maximizer.*
End of lecture 4.