report on "Hidden protocols: Modifying our expectations in an evolving world"

Chen Xin

Department of philosophy (ZhuHai)
Sun Yat-Sen University

Shuo-Zhi Seminars, Session 7
Nov.2, 2021

Outline

Outline

Overview

- agents know a protocol $\Longrightarrow$ expectations about future observations.
    - update their knowledge.
- protocols (need not be common knowledge)
- $\Longrightarrow$ semantics-driven logical framework
- answer:
    - What does it mean that we know a protocol?
    - How does protocol affect our knowledge of reality?
    - ......

Overview

- agents know a protocol $\Longrightarrow$ expectations about future observations.
  - update their knowledge.
- protocols (need not be common knowledge)
- $\Longrightarrow$ semantics-driven logical framework
- answer:
  - What does it mean that we know a protocol?
  - How does protocol affect our knowledge of reality?
  - ......

Overview

- agents know a protocol $\Longrightarrow$ expectations about future observations.
  - update their knowledge.
- protocols (need not be common knowledge)
- $\Longrightarrow$ semantics-driven logical framework
- answer:
  - What does it mean that we know a protocol?
  - How does protocol affect our knowledge of reality?
  - ......

Overview

- agents know a protocol $\Longrightarrow$ expectations about future observations.
  - update their knowledge.
- protocols (need not be common knowledge)
- $\Longrightarrow$ semantics-driven logical framework
- answer:
  - What does it mean that we know a protocol?
  - How does protocol affect our knowledge of reality?
  - ......

Overview

- agents know a protocol $\Longrightarrow$ expectations about future observations.
  - update their knowledge.
- protocols (need not be common knowledge)
- $\Longrightarrow$ semantics-driven logical framework
- answer:
  - What does it mean that we know a protocol?
  - How does protocol affect our knowledge of reality?
  - ......

Overview

- two PDL-style epistemic logic $\begin{cases} 1.POL \\ 2.EPL \end{cases}$
- epistemic expectation models $\mathcal{M}_{exp}$
- epistemic protocol model $\mathcal{A}$
- $\mathcal{M}_{exp} = \mathcal{A} \otimes \mathcal{N}_{exp}$
- Sources $\begin{cases} 1.DEL \\ 2.\text{protocol changes} \quad (Wang\ Yanjing) \end{cases}$

Overview

- two PDL-style epistemic logic $\begin{cases} 1.POL \\ 2.EPL \end{cases}$
- epistemic expectation models $\mathcal{M}_{exp}$
- epistemic protocol model $\mathcal{A}$
- $\mathcal{M}_{exp} = \mathcal{A} \otimes \mathcal{N}_{exp}$
- Sources $\begin{cases} 1.DEL \\ 2.\text{protocol changes} \quad (\textit{Wang Yanjing}) \end{cases}$

Overview

- two PDL-style epistemic logic $\begin{cases} 1.POL \\ 2.EPL \end{cases}$
- epistemic expectation models $\mathcal{M}_{exp}$
- epistemic protocol model $\mathcal{A}$
- $\mathcal{M}_{exp} = \mathcal{A} \otimes \mathcal{N}_{exp}$
- Sources $\begin{cases} 1.DEL \\ 2.\text{protocol changes} \quad (\textit{Wang Yanjing}) \end{cases}$

Overview

- two PDL-style epistemic logic $\begin{cases} 1.POL \\ 2.EPL \end{cases}$
- epistemic expectation models $\mathcal{M}_{exp}$
- epistemic protocol model $\mathcal{A}$
- $\mathcal{M}_{exp} = \mathcal{A} \otimes \mathcal{N}_{exp}$
- Sources $\begin{cases} 1.DEL \\ 2.\text{protocol changes} \quad (\textit{Wang Yanjing}) \end{cases}$

Overview

- two PDL-style epistemic logic $\begin{cases} 1.POL \\ 2.EPL \end{cases}$
- epistemic expectation models $\mathcal{M}_{exp}$
- epistemic protocol model $\mathcal{A}$
- $\mathcal{M}_{exp} = \mathcal{A} \otimes \mathcal{N}_{exp}$
- Sources $\begin{cases} 1.DEL \\ 2.\text{protocol changes} \quad (Wang\ Yanjing) \end{cases}$

Outline

Example 1. *Gay*

- Background: caffe, Amsterdam, 1950s. { Kate, Jane, Ann}
- ⇒Kate is gay $\xrightarrow{\text{Kate wants to know}}$ Jane? Ann?
- ⇒Kate: “ *I'm musical, I like Kathleen Ferrier's voice*”.
- Jane(gay): Kate is gay; Ann: Kate's taste in music
- ⟹a hidden protocol: *In 1950s Amsterdam, “musical” was indeed a code term for “gay”, known almost exclusively by gay people.*

Example 1. *Gay*

- Background: caffe, Amsterdam, 1950s. { Kate, Jane, Ann}
- $\Rightarrow$Kate is gay $\xrightarrow{\text{Kate wants to know}}$ Jane? Ann?
- $\Rightarrow$Kate: " *I'm musical, I like Kathleen Ferrier's voice*".
- Jane(gay): Kate is gay; Ann: Kate's taste in music
- $\Longrightarrow$a hidden protocol: *In 1950s Amsterdam, "musical" was indeed a code term for "gay", known almost exclusively by gay people.*

Example 1. *Gay*

- Background: caffe, Amsterdam, 1950s. { Kate, Jane, Ann}
- ⇒Kate is gay $\xrightarrow{\text{Kate wants to know}}$ Jane? Ann?
- ⇒Kate: " *I'm musical, I like Kathleen Ferrier's voice*".
- Jane(gay): Kate is gay; Ann: Kate's taste in music
- ⟹a hidden protocol: *In 1950s Amsterdam, "musical" was indeed a code term for "gay", known almost exclusively by gay people.*

Example 1. *Gay*

- Background: caffe, Amsterdam, 1950s. { Kate, Jane, Ann}
- ⇒Kate is gay $\xrightarrow{\text{Kate wants to know}}$ Jane? Ann?
- ⇒Kate: " *I'm musical, I like Kathleen Ferrier's voice*".
- Jane(gay): Kate is gay; Ann: Kate's taste in music
- ⟹a hidden protocol: *In 1950s Amsterdam, "musical" was indeed a code term for "gay", known almost exclusively by gay people.*

Example 2. *Dutch? or not*

- In the Netherlands, people kissing three times on the cheek (left-right-left)
- the rest of Europe, people kiss each other only twice (left-right)
- Is Simon a Dutch?

Outline

Basic settings in this paper

1. **knowing a protocol** $\Rightarrow$ understanding the underlying meaning of the actions inclued by the protocol.
2. protocols need not be common knowledge;
3. fix a protocol specification language;
4. protocol is given by nature;

Outline

epistemic model

- **I**: a finite set of agents; **P**: a finite set of propositions; $Bool(\mathbf{P})$: all Boolean formulas over **P**;

Definition

(***Epistemic Model***): $\mathcal{M}_e = (S, \sim_i, V)$

epistemic model

- **I**: a finite set of agents; **P**: a finite set of propositions; *Bool*(**P**): all Boolean formulas over **P**;

Definition

(***Epistemic Model***): $\mathcal{M}_e = (S, \sim_i, V)$

epistemic model

- **I**: a finite set of agents; **P**: a finite set of propositions; *Bool*(**P**): all Boolean formulas over **P**;

Definition

(***Epistemic Model***): $\mathcal{M}_e = (S, \sim_i, V)$

- *epistemic expectation model* $\Rightarrow$ the expected observations of agents.
- agents observe what is happening around them and reason based on these observations. (e.g.a public announcement)
- $\begin{cases} \text{observation of action} \\ \text{observation of facts} \end{cases}$

preparations for $\mathcal{M}_{exp}$

- a finite set of ***actions*** Σ
- an ***observation*** is a finite string of actions, e.g.: abcd

Definition

Observation expressions π: strings over Σ(regular expressions)

$$\mathcal{L}_{obs} \ni \pi ::= \delta \mid \varepsilon \mid a \mid \pi \cdot \pi \mid \pi + \pi \mid \pi^*$$

$$\delta \rightsquigarrow \emptyset; \quad \epsilon \rightsquigarrow \text{empty string}; \quad a \in \Sigma.$$

Definition

Observation (a set $\mathcal{L}(\pi)$ generated by π):

$\mathcal{L}(\delta) = \emptyset$
$\mathcal{L}(\varepsilon) = \{\epsilon\}$
$\mathcal{L}(a) = \{a\}$
$\mathcal{L}(\pi \cdot \pi') = \{wv \mid w \in \mathcal{L}(\pi), v \in \mathcal{L}(\pi')\}$
$\mathcal{L}(\pi + \pi') = \mathcal{L}(\pi) \cup \mathcal{L}(\pi')$
$\mathcal{L}(\pi^*) = \{\epsilon\} \cup \bigcup_{n>0} \mathcal{L}(\underbrace{\pi \cdot \dots \cdot \pi}_{n})$

epistemic expectation model

Definition

epistemic expectation model

$\mathcal{M}_{exp} := < S, \sim_i, V, Exp > = (\mathcal{M}_e, Exp)$

- $Exp : s \mapsto \pi \in \mathcal{L}_{obs}$ s.t. $\mathcal{L}(\pi) \neq \emptyset$

- $\mathcal{M}_{exp} \rightsquigarrow \mathcal{M}_e$ (degenerate)
    - $\forall s \in \mathcal{M}_e, Exp(s) = (a_0 + a_1 + \cdots + a_n)^* = \Sigma^*$

epistemic expectation model

Definition

epistemic expectation model

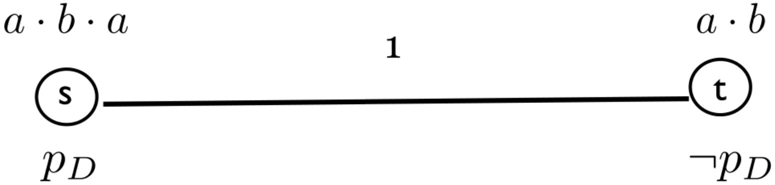
$\mathcal{M}_{exp} := < S, \sim_i, V, Exp >= (\mathcal{M}_e, Exp)$

- $Exp : s \mapsto \pi \in \mathcal{L}_{obs}$ s.t. $\mathcal{L}(\pi) \neq \emptyset$

- $\mathcal{M}_{exp} \rightsquigarrow \mathcal{M}_e$ (degenerate)
  - $\forall s \in \mathcal{M}_e, Exp(s) = (a_0 + a_1 + \cdots + a_n)^* = \Sigma^*$

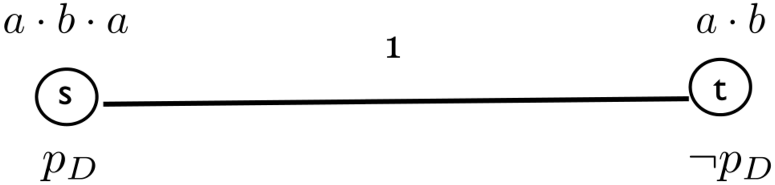
Example 2. *Dutch or not*

- In the Netherlands, people kissing three times on the cheek (left-right-left)
- the rest of Europe, people kiss each other only twice (left-right)
- Is Simon a Dutch?



Example 2. *Dutch or not*

- In the Netherlands, people kissing three times on the cheek (left-right-left)
- the rest of Europe, people kiss each other only twice (left-right)
- Is Simon a Dutch?



Outline

Intuition

- **public announcement logic**: people update their information by deleting impossible scenarios according to what is publicly announced.
- when observing an action, people delete some impossible scenarios where they wouldn't expect that observation to happen.
- we delete the states where the observation w could not have been happened.

Intuition

- **public announcement logic**: people update their information by deleting impossible scenarios according to what is publicly announced.
- when observing an action, people delete some impossible scenarios where they wouldn't expect that observation to happen.
- we delete the states where the observation w could not have been happened.

Intuition

- **public announcement logic**: people update their information by deleting impossible scenarios according to what is publicly announced.
- when observing an action, people delete some impossible scenarios where they wouldn't expect that observation to happen.
- we delete the states where the observation w could not have been happened.

Update by observation

Definition

Update by observation : let $w \in \mathcal{L}_{obs}, \mathcal{M} = (S, \sim, V, Exp)$, the updated model $\mathcal{M}|_w = (S', \sim', V', Exp')$:

- $S' = \{s \mid \mathcal{L}(Exp(s)\backslash w) \neq \emptyset\}$
  - $S' = S - \{s \mid \mathcal{L}(Exp(s)\backslash w) = \emptyset\}$
- $\sim_i' = \sim_i |_{S' \times S'}$
- $V' = V|_{S'}$
- $Exp'(s) = Exp(s)\backslash w$

$\pi \backslash w$

- $\pi \backslash w$:

A regular expression $\pi \backslash w$ is defined with an auxiliary output function o from the set of regular expressions over Σ to $\{\delta, \varepsilon\}$. If $\varepsilon \in \mathcal{L}(\pi)$, the output function o maps a regular expression π to ε; otherwise, it maps π to δ [17,18]:

$$
\begin{array}{ll}
\pi = o(\pi) + \sum_{a \in \Sigma} (a \cdot \pi \backslash a) & \varepsilon \backslash a = \delta \backslash a = b \backslash a = \delta \quad (a \neq b) \\
o(\varepsilon) = \varepsilon & a \backslash a = \varepsilon \\
o(\delta) = o(a) = \delta & (\pi + \pi') \backslash a = \pi \backslash a + \pi' \backslash a \\
o(\pi + \pi') = o(\pi) + o(\pi') & (\pi \cdot \pi') \backslash a = (\pi \backslash a) \cdot \pi' + o(\pi) \cdot (\pi' \backslash a) \\
o(\pi \cdot \pi) = o(\pi) \cdot o(\pi') & \pi^* \backslash a = \pi \backslash a \cdot \pi^* \\
o(\pi^*) = \varepsilon & \pi \backslash a_0 \cdots a_n = \pi \backslash a_0 \backslash a_1 \ldots \backslash a_n
\end{array}
$$

Example 2. Dutch or not

- $\mathcal{M}|_a =?$

POL

Definition

***Public observation logic* POL** (language)

$$\varphi ::= \top \mid p \mid \neg\varphi \mid \varphi \wedge \varphi \mid K_i\varphi \mid [\pi]\varphi$$

$\pi \in \mathcal{L}_{obs}$

- $[\pi]\varphi$: after any observation in π, φ holds.

POL

Definition

***Public observation logic* POL** (language)

$$\varphi ::= \top \mid p \mid \neg\varphi \mid \varphi \land \varphi \mid K_i\varphi \mid [\pi]\varphi$$

$\pi \in \mathcal{L}_{obs}$

- $[\pi]\varphi$: after any observation in π, φ holds.

Truth for POL

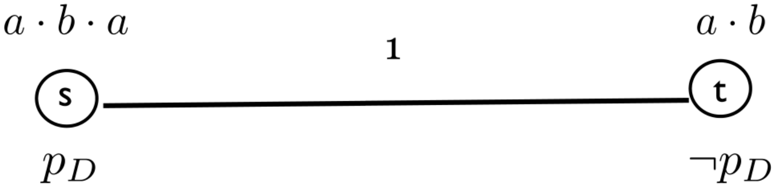
Definition

Truth

- $\mathcal{M} = (S, \sim, V, Exp)$, $s \in \mathcal{M}$
- $\mathcal{M}, s \models [\pi]\varphi$ iff
  $\forall w \in \mathcal{L}(\pi), w \in init(Exp(s)) \Longrightarrow \mathcal{M}|w, s \models \varphi$
  - $w \in init(\pi)$ iff $\exists v \in \Sigma^*, wv \in \mathcal{L}(\pi)$ ($\mathcal{L}(\pi \backslash w) \neq \emptyset$)

Example 2 (Dutch or not)

1. $\mathcal{M}, s \models [a \cdot b](\neg K_1 P_D \wedge [a]K_1 P_D$



Outline

Observation bisimulation

Definition

Observation bisimulation

R is a ***bisimulation*** between $\mathcal{M} = (S, \sim, V, Exp)$ and $\mathcal{N} = (S', \sim', V', Exp')$ if for any $s \in S$, $s' \in S'$, we have that if $(s, s') \in R$, then

- $V(s) = V'(s')$;
- $\mathcal{L}(Exp(s)) = \mathcal{L}(Exp'(s'))$;
- **Zig** : if $s \sim_i t$ then $\exists t' \in \mathcal{N}$ such that$s' \sim_i t'$ and tRt';
- **Zag**: if $s' \sim_i t'$ then $\exists t \in \mathcal{M}$ such that $s \sim_i t$ and tRt'.

$\mathcal{M}, s \,\underline{\leftrightarrow}_o\, \mathcal{N}, s'$

Invariance

Theorem

*(**Bisimulation invariance**)*
For any two finite epistemic expectation states $\mathcal{M}, s$ and $\mathcal{N}, s'$:
$\mathcal{M}, s \,\underline{\leftrightarrow}_o\, \mathcal{N}, s'$ *iff* $\quad \forall \varphi \in \mathbf{POL}: M, s \models \varphi \iff N, s' \models \varphi$.

Outline

- *epistemic expectation models* can be seen as **compact representations** of certain *epistemic temporal models*.

Outline

protocol

- a **protocol** is a *rule* telling us what we should do under what conditions.

Definition

(***Protocol expression***)
$\mathcal{L}_{prot} \ni \eta ::= \delta \mid \varepsilon \mid a \mid ?\varphi \mid \eta \cdot \eta \mid \eta + \eta \mid \eta^*$
$\delta \rightsquigarrow \emptyset$; $\varepsilon \rightsquigarrow$ empty string; $a \in \Sigma$; $\varphi \in Bool(\mathbf{P})$

- adding Boolean tests
- $(?love \cdot stay)^* \cdot (?\neg love \cdot separate)$: we should stay together as long as we are in love.
- "$?K\varphi$"
- A protocol without **tests** corresponds to observations without any conditions.

protocol

- a **protocol** is a *rule* telling us what we should do under what conditions.

Definition

(***Protocol expression***)
$\mathcal{L}_{prot} \ni \eta ::= \delta \mid \varepsilon \mid a \mid ?\varphi \mid \eta \cdot \eta \mid \eta + \eta \mid \eta^*$
$\delta \rightsquigarrow \emptyset$; $\varepsilon \rightsquigarrow$ empty string; $a \in \Sigma$; $\varphi \in Bool(\mathbf{P})$

- adding Boolean tests
- $(?love \cdot stay)^* \cdot (?\neg love \cdot separate)$: we should stay together as long as we are in love.
- "$?K\varphi$"
- A protocol without **tests** corresponds to observations without any conditions.

protocol

- a **protocol** is a *rule* telling us what we should do under what conditions.

Definition

(***Protocol expression***)
$\mathcal{L}_{prot} \ni \eta ::= \delta \mid \varepsilon \mid a \mid ?\varphi \mid \eta \cdot \eta \mid \eta + \eta \mid \eta^*$
$\delta \rightsquigarrow \emptyset$; $\varepsilon \rightsquigarrow$ empty string; $a \in \Sigma$; $\varphi \in Bool(\mathbf{P})$

- adding Boolean tests
- $(?love \cdot stay)^* \cdot (?\neg love \cdot separate)$: we should stay together as long as we are in love.
- "$?K\varphi$"
- A protocol without **tests** corresponds to observations without any conditions.

protocol

- a **protocol** is a *rule* telling us what we should do under what conditions.

Definition

(***Protocol expression***)
$\mathcal{L}_{prot} \ni \eta ::= \delta \mid \varepsilon \mid a \mid ?\varphi \mid \eta \cdot \eta \mid \eta + \eta \mid \eta^*$
$\delta \rightsquigarrow \emptyset$; $\varepsilon \rightsquigarrow$ empty string; $a \in \Sigma$; $\varphi \in Bool(\mathbf{P})$

- adding Boolean tests
- $(?love \cdot stay)^* \cdot (?\neg love \cdot separate)$: we should stay together as long as we are in love.
- "$?K\varphi$"
- A protocol without **tests** corresponds to observations without any conditions.

protocol

- a **protocol** is a *rule* telling us what we should do under what conditions.

Definition

(***Protocol expression***)
$\mathcal{L}_{prot} \ni \eta ::= \delta \mid \varepsilon \mid a \mid ?\varphi \mid \eta \cdot \eta \mid \eta + \eta \mid \eta^*$
$\delta \rightsquigarrow \emptyset$; $\varepsilon \rightsquigarrow$ empty string; $a \in \Sigma$; $\varphi \in Bool(\mathbf{P})$

- adding Boolean tests
- $(?love \cdot stay)^* \cdot (?\neg love \cdot separate)$: we should stay together as long as we are in love.
- "$?K\varphi$"
- A protocol without **tests** corresponds to observations without any conditions.

Example 2 Dutch or not

- if you are Dutch-related, then you kiss three times; and if you are non-Dutch-related, then you kiss two times.
- π_k: $?p_D \cdot a \cdot b \cdot a + ?\neg p_D \cdot a \cdot b$

Example 2 Dutch or not

- if you are Dutch-related, then you kiss three times; and if you are non-Dutch-related, then you kiss two times.
- π_k: $?p_D \cdot a \cdot b \cdot a + ?\neg p_D \cdot a \cdot b$

guarded observation

Definition

(guarded observations) (a set $\mathcal{L}_g(\eta)$ genetated by η)

$$\mathcal{L}_g(\delta) = \emptyset$$
$$\mathcal{L}_g(\varepsilon) = \mathcal{P}(\mathbf{P})$$
$$\mathcal{L}_g(a) = \{\rho a \rho \mid \rho \subseteq \mathbf{P}\}$$
$$\mathcal{L}_g(?\varphi) = \{\rho \mid \rho \models \varphi, \rho \subseteq \mathbf{P}\}$$
$$\mathcal{L}_g(\eta_1 \cdot \eta_2) = \{w \diamond v \mid w \in \mathcal{L}_g(\eta_1), v \in \mathcal{L}(\eta_2)\}$$
$$\mathcal{L}_g(\eta_1 + \eta_2) = \mathcal{L}_g(\eta_1) \cup \mathcal{L}_g(\eta_2)$$
$$\mathcal{L}_g(\eta^*) = \mathcal{P}(\mathbf{P}) \cup \bigcup_{n>0} \mathcal{L}_g(\underbrace{\eta \cdot \cdots \cdot \eta}_{n})$$

- if $w = w'\rho$, $v = \rho v'$ then $w \diamond v = w'\rho v'$,

guarded observation

Definition

(guarded observations) (a set $\mathcal{L}_g(\eta)$ genetated by η)

$$\mathcal{L}_g(\delta) = \emptyset$$
$$\mathcal{L}_g(\varepsilon) = \mathcal{P}(\mathbf{P})$$
$$\mathcal{L}_g(a) = \{\rho a \rho \mid \rho \subseteq \mathbf{P}\}$$
$$\mathcal{L}_g(?\varphi) = \{\rho \mid \rho \models \varphi, \rho \subseteq \mathbf{P}\}$$
$$\mathcal{L}_g(\eta_1 \cdot \eta_2) = \{w \diamond v \mid w \in \mathcal{L}_g(\eta_1), v \in \mathcal{L}(\eta_2)\}$$
$$\mathcal{L}_g(\eta_1 + \eta_2) = \mathcal{L}_g(\eta_1) \cup \mathcal{L}_g(\eta_2)$$
$$\mathcal{L}_g(\eta^*) = \mathcal{P}(\mathbf{P}) \cup \bigcup_{n>0} \mathcal{L}_g(\underbrace{\eta \cdot \dots \cdot \eta}_{n})$$

- if $w = w'\rho$, $v = \rho v'$ then $w \diamond v = w'\rho v'$,

conversion function

- $f_\rho : \mathcal{L}_{prot} \to \mathcal{L}_{obs}$:

$$f_\rho(\delta) = \delta$$
$$f_\rho(\varepsilon) = \varepsilon$$
$$f_\rho(a) = a$$
$$f_\rho(\eta \cdot \eta') = f_\rho(\eta) \cdot f_\rho(\eta')$$
$$f_\rho(\eta + \eta') = f_\rho(\eta) + f_\rho(\eta')$$
$$f_\rho(\eta^*) = (f_\rho(\eta))^*$$
$$f_\rho(?\varphi) = \begin{cases} \varepsilon & \rho \models \varphi \\ \delta & \rho \not\models \varphi \end{cases}$$

- e.g. $f_{\{p\}}(?p \cdot \mathrm{a} + ?\neg p \cdot \mathrm{b}) = \mathrm{a}$

conversion function

- $f_\rho : \mathcal{L}_{prot} \to \mathcal{L}_{obs}$:

$$f_\rho(\delta) = \delta$$
$$f_\rho(\varepsilon) = \varepsilon$$
$$f_\rho(a) = a$$
$$f_\rho(\eta \cdot \eta') = f_\rho(\eta) \cdot f_\rho(\eta')$$
$$f_\rho(\eta + \eta') = f_\rho(\eta) + f_\rho(\eta')$$
$$f_\rho(\eta^*) = (f_\rho(\eta))^*$$
$$f_\rho(?\varphi) = \begin{cases} \varepsilon & \rho \models \varphi \\ \delta & \rho \not\models \varphi \end{cases}$$

- e.g. $f_{\{p\}}(?p \cdot \mathrm{a} + ?\neg p \cdot \mathrm{b}) = \mathrm{a}$

Definitions

Def. 17 (***Characteristic formula***). Let $\rho \subseteq \mathbf{P}$. the *characteristic formula* for ρ:

$$\varphi_\rho := \bigwedge_{p \in \rho} p \wedge \bigwedge_{p \notin \rho} \neg p$$

e.g. $\varphi_{\{p\}} = p \wedge \neg p$

Theorem

Prop. 18
$\mathcal{L}(f_\rho(\eta)) = \{w \mid w = a_0 \cdot \cdots \cdot a_n, a_i \in \Sigma \cup \{\varepsilon\}, \rho a_0 \rho a_1 \ldots \rho a_k \rho \in \mathcal{L}_g(\eta)\}$;
Every η has a normal form η° as follows:

$$\eta^\circ = \Sigma_{\rho \subseteq \mathbf{P}}(?\varphi_\rho \cdot f_\rho(\eta))$$

s.t. $\mathcal{L}_g(\eta) = \mathcal{L}_g(\eta^\circ)$

- From **Prop. 18**, according to the protocol η, the expected observations on a state s in an epistemic model $\mathcal{M}$ can be computed by $f_{V_{\mathcal{M}(s)}}(\eta)$.

For example:

- $f_{\{p\}}(?p \cdot \mathrm{a} + ?\neg p \cdot \mathrm{b}) = \mathrm{a}$
- $f_{V_{M,s}}(\pi_k) = a \cdot b \cdot a$
- $f_{V_{M,t}}(\pi_k) = a \cdot b$

- From **Prop. 18**, according to the protocol η, the expected observations on a state s in an epistemic model $\mathcal{M}$ can be computed by $f_{V_{\mathcal{M}(s)}}(\eta)$.

For example:

- $f_{\{p\}}(?p \cdot \mathrm{a}+?\neg p \cdot \mathrm{b}) = \mathrm{a}$
- $f_{V_{M,s}}(\pi_k) = a \cdot b \cdot a$
- $f_{V_{M,t}}(\pi_k) = a \cdot b$

Outline

protocol models

Definition

Protocol models $\mathcal{A} = (T, \sim, Prot)$

- T is a domain of abstract objects,
- $Prot : T \to \mathcal{L}_{prot}$ assigns to each domain object a protocol.

an epistemic protocol : $(\mathcal{A}, t)$
a public protocol: $(\mathcal{A}, t)$ and the T of $\mathcal{A}$ is a singleton set.

- an epistemic observation state uniquely determines an epistemic protocol,
- an epistemic protocol and an epistemic state together uniquely determine an epistemic observation state.
- *modal product operation* of $\mathcal{M}_{exp}$ and $\mathcal{A}$.

- an epistemic observation state uniquely determines an epistemic protocol,
- an epistemic protocol and an epistemic state together uniquely determine an epistemic observation state.
- *modal product operation* of $\mathcal{M}_{exp}$ and $\mathcal{A}$.

- an epistemic observation state uniquely determines an epistemic protocol,
- an epistemic protocol and an epistemic state together uniquely determine an epistemic observation state.
- *modal product operation* of $\mathcal{M}_{exp}$ and $\mathcal{A}$.

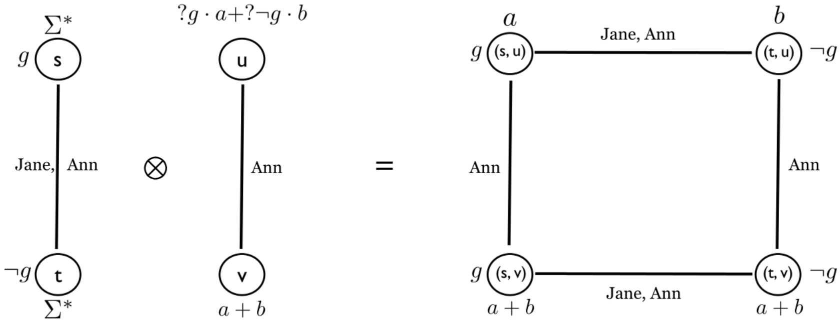
protocol update

Definition

protocol update: $\mathcal{M}_{exp} = (S, \sim, V, Exp)$, $\mathcal{A} = (T, \sim, Prot)$; $\mathcal{M}_{exp} \otimes \mathcal{A} = (S', \sim', V', Exp')$:

- $S' = \{(s,t) \in S \times T \mid \mathcal{L}(f_{V_{\mathcal{M}(s)}}(Prot(t))) \neq \emptyset\}$
- $(s,t) \sim_i (s',t')$ iff $s \sim_i s'$ in $\mathcal{M}_{exp}$ and $t \sim_i t'$ in $\mathcal{A}$
- $V'(s,t) = V(s)$
- $Exp'((s,t)) = f_{V_{\mathcal{M}(s)}}(Prot(t))$

Example 1 Gay



*some properties about $\otimes$

Theorem

Given $\mathcal{M}_{exp} = (\mathcal{M}_e, Exp)$, there is an epistemic model $\mathcal{M}'_e$ and an epistemic protocol model $\mathcal{A}$ such that $\mathcal{M} \underline{\leftrightarrow}_o \mathcal{M}'_e \otimes \mathcal{A}$

- ask: " $\exists \mathcal{A}, \mathcal{M}_{exp} \underline{\leftrightarrow}_o \mathcal{M}_e \otimes \mathcal{A}$ " ïij§
- $\Longrightarrow$ ***Theorem 29***:
  - if $\mathcal{M}_{exp} = (\mathcal{M}_e, Exp)$ observationally saturated, then there is a $\mathcal{A}$ s.t. $\mathcal{M}_e \otimes \mathcal{A} \underline{\leftrightarrow}_o \mathcal{M}_{exp}$.

*effective equivalence

- **Def. 30 *effective equivalence***: Two protocol models $\mathcal{A}$ and $\mathcal{B}$ are said to be *effectively equivalent* ($\mathcal{A} \equiv_{ef} \mathcal{B}$) if for any epistemic expectation model $\mathcal{M}_{exp}$: $\mathcal{M}_{exp} \otimes \mathcal{A} \underline{\leftrightarrow} \mathcal{M}_{exp} \otimes \mathcal{B}$.
- **Def. 31 *Protocol emulation*** (When restricted to public protocols, $\eta \approx \eta' \Leftrightarrow \mathcal{L}_g(\eta) = \mathcal{L}_g(\eta')$.)
- **Theorem 32:** For all *finite* protocol models $\mathcal{A}$ and $\mathcal{B}$: $\mathcal{A} \equiv_{ef} \mathcal{B} \Leftrightarrow \mathcal{A} \approx B$

*effective equivalence

- **Def. 30** ***effective equivalence***: Two protocol models $\mathcal{A}$ and $\mathcal{B}$ are said to be *effectively equivalent* ($\mathcal{A} \equiv_{ef} \mathcal{B}$) if for any epistemic expectation model $\mathcal{M}_{exp}$: $\mathcal{M}_{exp} \otimes \mathcal{A} \underline{\leftrightarrow} \mathcal{M}_{exp} \otimes \mathcal{B}$.
- **Def. 31** ***Protocol emulation*** (When restricted to public protocols, $\eta \approx \eta' \Leftrightarrow \mathcal{L}_g(\eta) = \mathcal{L}_g(\eta')$.)
- **Theorem 32:** For all *finite* protocol models $\mathcal{A}$ and $\mathcal{B}$: $\mathcal{A} \equiv_{ef} \mathcal{B} \Leftrightarrow \mathcal{A} \approx B$

*effective equivalence

- **Def. 30 *effective equivalence***: Two protocol models $\mathcal{A}$ and $\mathcal{B}$ are said to be *effectively equivalent* ($\mathcal{A} \equiv_{ef} \mathcal{B}$) if for any epistemic expectation model $\mathcal{M}_{exp}$: $\mathcal{M}_{exp} \otimes \mathcal{A} \underline{\leftrightarrow} \mathcal{M}_{exp} \otimes \mathcal{B}$.
- **Def. 31 *Protocol emulation*** (When restricted to public protocols, $\eta \approx \eta' \Leftrightarrow \mathcal{L}_g(\eta) = \mathcal{L}_g(\eta')$.)
- **Theorem 32:** For all *finite* protocol models $\mathcal{A}$ and $\mathcal{B}$: $\mathcal{A} \equiv_{ef} \mathcal{B} \Leftrightarrow \mathcal{A} \approx B$

Outline

EPL

Definition

***Language of* EPL**

$$\varphi ::= \top \mid p \mid \neg\varphi \mid \varphi \wedge \varphi \mid K_i\varphi \mid [\pi]\varphi \mid [!\mathcal{A}_e]\varphi$$

truth

Definition

Truth

- $\mathcal{M}, s \models [!\mathcal{A}_e]\varphi$ iff $\mathcal{L}(f_{V(s)}(Prot(e))) \neq \emptyset \Longrightarrow \mathcal{M} \otimes \mathcal{A}_e, (s, e) \models \varphi$
  - $\mathcal{M}, s \models [\pi]\varphi$ iff $\forall w \in \mathcal{L}(\pi), w \in init(Exp(s)) \Longrightarrow \mathcal{M}|w, s \models \varphi$
  - $w \in init(\pi) iff \exists v \in \Sigma^*, wv \in \mathcal{L}(\pi)$ (i.e.$\mathcal{L}(\pi \backslash w) \neq \emptyset$)
- after installing the new epistemic protocol $\mathcal{A}_e$, the formula φ is true.

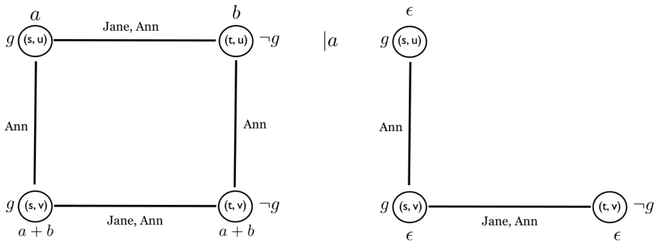
truth

Definition

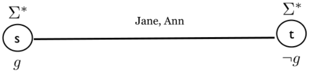
Truth

- $\mathcal{M}, s \models [!\mathcal{A}_e]\varphi$ iff $\mathcal{L}(f_{V(s)}(Prot(e))) \neq \emptyset \Longrightarrow \mathcal{M} \otimes \mathcal{A}_e, (s, e) \models \varphi$
    - $\mathcal{M}, s \models [\pi]\varphi$ iff $\forall w \in \mathcal{L}(\pi), w \in init(Exp(s)) \Longrightarrow \mathcal{M}|w, s \models \varphi$
    - $w \in init(\pi) iff \exists v \in \Sigma^*, wv \in \mathcal{L}(\pi)$ (i.e.$\mathcal{L}(\pi \backslash w) \neq \emptyset$)
- after installing the new epistemic protocol $\mathcal{A}_e$, the formula φ is true.

Example 1 Gay



Recall the original model $\mathcal{M}$:



Now we can verify for the actual state s:

$$\mathcal{M}, s \vDash [!\mathcal{A}_e][a](K_{Jane}g \wedge \neg K_{Ann}g), \quad \text{and}$$

$$\mathcal{M}, s \vDash [!\mathcal{A}_e][a]\neg K_{Ann}(K_{Jane}g \vee K_{Jane}\neg g).$$

motivation

- many actions used in protocols also change the facts.
- e.g. "turn on the light if you see that the light is off".

motivation

- many actions used in protocols also change the facts.
- e.g. "turn on the light if you see that the light is off".

fact-changing actions

Definition

(***Fact-changing actions***). A set of fact-changing actions (*fc-actions*) is a tuple (Σ, ι) such that $\iota : \Sigma \times \mathbf{P} \to Bool(\mathbf{P})$.
$(a, p) \mapsto \phi$

- after executing action a, the propositional atom p is assigned the truth value of the proposition $\iota(a, p)$.

e.g.:
$\iota(a, p) = \top$; a: 'slam the door'. p: 'the door is closed'
$\iota(b, q) = \neg q$; b: 'toggling the switch' . q: 'the switch is on'

fact-changing actions

Definition

(***Fact-changing actions***). A set of fact-changing actions (*fc-actions*) is a tuple (Σ, ι) such that $\iota : \Sigma \times \mathbf{P} \to Bool(\mathbf{P})$.
$(a, p) \mapsto \phi$

- after executing action a, the propositional atom p is assigned the truth value of the proposition $\iota(a, p)$.

e.g.:
$\iota(a, p) = \top$; a: 'slam the door'. p: 'the door is closed'
$\iota(b, q) = \neg q$; b: 'toggling the switch' . q: 'the switch is on'

Definition

Def. 36 (***Factual change system***). A Σ-factual change system (fc-system) $\mathcal{F}$ is a tuple (Q, r) where $Q = \mathcal{P}(\mathbf{P})$ and $r : Q \times \Sigma \to Q$ is a function.

- Intuitively, a factual change system explicitly represents the post-conditions of actions that can change the facts on states.

Theorem

Prop. 37 *: sets of fact-changing actions can be seen as factual change systems and vice versa.*
(a) For each set of fc-actions (Σ, ι) there is an equivalent Σ-fc-system.
(b) For each Σ-fc-system there is an equivalent set of fc-actions (Σ, ι).

Definition

Def. 36 (*Factual change system*). A Σ-factual change system (fc-system) $\mathcal{F}$ is a tuple (Q, r) where $Q = \mathcal{P}(\mathbf{P})$ and $r : Q \times \Sigma \to Q$ is a function.

- Intuitively, a factual change system explicitly represents the post-conditions of actions that can change the facts on states.

Theorem

Prop. 37 : *sets of fact-changing actions can be seen as factual change systems and vice versa.*
(a) For each set of fc-actions (Σ, ι) *there is an equivalent* Σ*-fc-system.*
(b) For each Σ*-fc-system there is an equivalent set of fc-actions* (Σ, ι).

- $\mathcal{L}_g^{\mathcal{F}}(\eta)$
- $\mathcal{L}_g^{\mathcal{F}}(a) = \{\rho a \rho' \mid \rho \xrightarrow{a} \rho' \text{ in } \mathcal{F}\}$
- Prop. 41 ***Normal form with respect to*** $\mathcal{F}$ (**similar Prop. 18**)
  - Given an fc-system $\mathcal{F}$, every η has a normal form $\eta^{\mathcal{F}} = \Sigma_{\rho \subseteq \mathbf{P}}(?\varphi_\rho \cdot \pi_\rho)$ for some $\pi_\rho \in \mathcal{L}_{obs}$ s.t. $\mathcal{L}_g^{\mathcal{F}}(\eta) = \mathcal{L}_g^{\mathcal{F}}(\eta^{\mathcal{F}})$.
- *fact-changing epistemic expectation models* and *protocol models:*
  - $\mathcal{M}_{exp}^{\mathcal{F}} = (\mathcal{M}_{exp}, \mathcal{F})$
  - $\mathcal{A}^{\mathcal{F}} = (\mathcal{A}, \mathcal{F})$

- $\mathcal{L}_g^{\mathcal{F}}(\eta)$
- $\mathcal{L}_g^{\mathcal{F}}(a) = \{\rho a \rho' \mid \rho \xrightarrow{a} \rho' \text{ in } \mathcal{F}\}$
- Prop. 41 ***Normal form with respect to $\mathcal{F}$*** (**similar Prop. 18**)
  - Given an fc-system $\mathcal{F}$, every η has a normal form $\eta^{\mathcal{F}} = \Sigma_{\rho \subseteq \mathbf{P}}(?\varphi_\rho \cdot \pi_\rho)$ for some $\pi_\rho \in \mathcal{L}_{obs}$ s.t. $\mathcal{L}_g^{\mathcal{F}}(\eta) = \mathcal{L}_g^{\mathcal{F}}(\eta^{\mathcal{F}})$.
- *fact-changing epistemic expectation models* and *protocol models:*
  - $\mathcal{M}_{exp}^{\mathcal{F}} = (\mathcal{M}_{exp}, \mathcal{F})$
  - $\mathcal{A}^{\mathcal{F}} = (\mathcal{A}, \mathcal{F})$

- $\mathcal{L}_g^{\mathcal{F}}(\eta)$
- $\mathcal{L}_g^{\mathcal{F}}(a) = \{\rho a \rho' \mid \rho \xrightarrow{a} \rho' \text{ in } \mathcal{F}\}$
- Prop. 41 ***Normal form with respect to*** $\mathcal{F}$ (**similar Prop. 18**)
  - Given an fc-system $\mathcal{F}$, every η has a normal form $\eta^{\mathcal{F}} = \Sigma_{\rho \subseteq \mathbf{P}}(?\varphi_\rho \cdot \pi_\rho)$ for some $\pi_\rho \in \mathcal{L}_{obs}$ s.t. $\mathcal{L}_g^{\mathcal{F}}(\eta) = \mathcal{L}_g^{\mathcal{F}}(\eta^{\mathcal{F}})$.
- *fact-changing epistemic expectation models* and *protocol models:*
  - $\mathcal{M}_{exp}^{\mathcal{F}} = (\mathcal{M}_{exp}, \mathcal{F})$
  - $\mathcal{A}^{\mathcal{F}} = (\mathcal{A}, \mathcal{F})$

update

Definition

(***Protocol update with factual changes***)
Given $\mathcal{M}^{\mathcal{F}}_{exp} = (S, \sim, V, Exp, \mathcal{F})$, $\mathcal{A}^{\mathcal{G}} = (T, \sim, Prot, \mathcal{G})$, define the product:
$\mathcal{M}^{\mathcal{F}}_{exp} \otimes \mathcal{A}^{\mathcal{F}} = (S', \sim', V', Exp', \mathcal{F}')$

1. $S' = \{(s,t) \in S \times T \mid \mathcal{L}(f_{V_{\mathcal{M}(s)}}(Prot^{\mathcal{G}}(t))) \neq \emptyset\}$
2. $(s,t) \sim_i (s',t')$ iff $s \sim_i s'$ in $\mathcal{M}_{exp}$ and $t \sim_i t'$ in $\mathcal{A}$
3. $V'(s,t) = V(s)$
4. $Exp'((s,t)) = f_{V_{\mathcal{M}(s)}}(Prot^{\mathcal{G}}(t))$
5. $\mathcal{F}' = \mathcal{G}$

where $Prot^{\mathcal{G}}(t)$ is the normal form of $Prot(t)$ with respect to G.

Truth

Definition

Truth:

- $\mathcal{M}^{\mathcal{F}}_{exp}, s \models [!\mathcal{A}^{\mathcal{G}}_{e}]\varphi$ iff $\mathcal{L}(f_{V(s)}(Prot^{\mathcal{G}}(e))) \neq \emptyset \Longrightarrow \mathcal{M}^{\mathcal{F}}_{exp} \otimes \mathcal{A}^{\mathcal{F}}, (s, e) \models \varphi$
- $\mathcal{M}^{\mathcal{F}}_{exp}, s \models [\pi]\varphi$ iff $\forall w \in \mathcal{L}(\pi), w \in init(Exp(s)) \Longrightarrow \mathcal{M}^{\mathcal{F}}_{exp}|_{w}, s \models \varphi$
- $w \in init(\pi) iff \exists v \in \Sigma^*, wv \in \mathcal{L}(\pi)$ (i.e.$\mathcal{L}(\pi \backslash w) \neq \emptyset$)

Example

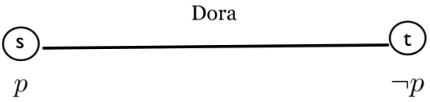
- a room where a child is playing with a seat, and Dora standing outside the room. Before Dora enters, she does not have any idea whether the seat is in an upright position.
- p :the seat is in an upright position; a : pulling the seat down; b : pulling the seat up.

Example

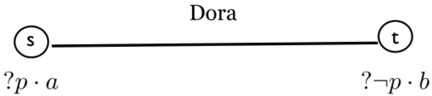
- a room where a child is playing with a seat, and Dora standing outside the room. Before Dora enters, she does not have any idea whether the seat is in an upright position.
- p :the seat is in an upright position; a : pulling the seat down; b : pulling the seat up.

Example

- $\mathcal{M}$:

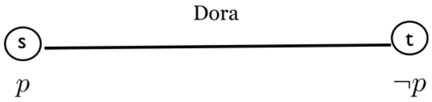


- $\mathcal{A}^{\mathcal{F}}$:

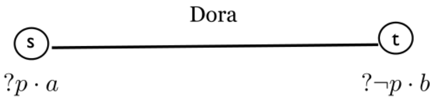


Example

- $\mathcal{M}$:

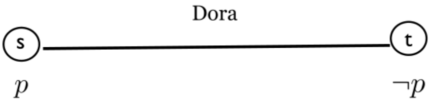


- $\mathcal{A}^{\mathcal{F}}$:

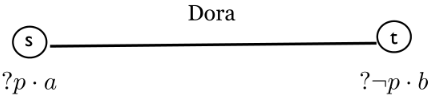


Example

- $\mathcal{M}$:

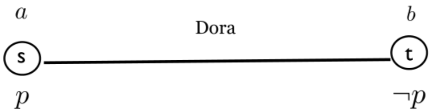


- $\mathcal{A}^{\mathcal{F}}$:



updated product model:

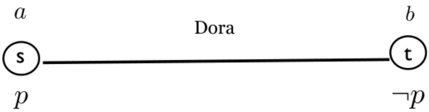
- updated product model:



- $\mathcal{M}, s \models [!\mathcal{A}_e^{\mathcal{F}}][a]K_{Dora}\neg p$

updated product model:

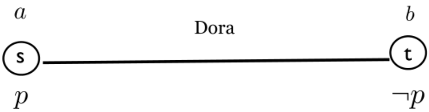
- updated product model:



- $\mathcal{M}, s \models [!\mathcal{A}^{\mathcal{F}}_{e}][a]K_{Dora}\neg p$

updated product model:

- updated product model:



- $\mathcal{M}, s \models [!\mathcal{A}_e^{\mathcal{F}}][a]K_{Dora}\neg p$

100 prisoners & 1 lightbulb

- 100 prisoners will be interrogated in **a room** containing **a light** with an on/off switch.
- common knowledge:
  - the light is initially switched **off**.
  - There is no fixed order of interrogation, or fixed interval between interrogations.
- When interrogated, a prisoner can
  - do nothing, or
  - toggle the light-switch,
  - or announce that all prisoners have been interrogated.
- If that announcement is true, the prisoners will (all) be set free, otherwise, they will all be executed.
- $\Rightarrow$ can the prisoners agree on a protocol that will set them free?

100 prisoners & 1 lightbulb

- 100 prisoners will be interrogated in **a room** containing **a light** with an on/off switch.
- common knowledge:
  - the light is initially switched **off**.
  - There is no fixed order of interrogation, or fixed interval between interrogations.
- When interrogated, a prisoner can
  - do nothing, or
  - toggle the light-switch,
  - or announce that all prisoners have been interrogated.
- If that announcement is true, the prisoners will (all) be set free, otherwise, they will all be executed.
- $\Rightarrow$ can the prisoners agree on a protocol that will set them free?

100 prisoners & 1 lightbulb

- 100 prisoners will be interrogated in **a room** containing **a light** with an on/off switch.
- common knowledge:
  - the light is initially switched **off**.
  - There is no fixed order of interrogation, or fixed interval between interrogations.
- When interrogated, a prisoner can
  - do nothing, or
  - toggle the light-switch,
  - or announce that all prisoners have been interrogated.
- If that announcement is true, the prisoners will (all) be set free, otherwise, they will all be executed.
- $\Rightarrow$ can the prisoners agree on a protocol that will set them free?

100 prisoners & 1 lightbulb

- 100 prisoners will be interrogated in **a room** containing **a light** with an on/off switch.
- common knowledge:
  - the light is initially switched **off**.
  - There is no fixed order of interrogation, or fixed interval between interrogations.
- When interrogated, a prisoner can
  - do nothing, or
  - toggle the light-switch,
  - or announce that all prisoners have been interrogated.
- If that announcement is true, the prisoners will (all) be set free, otherwise, they will all be executed.
- $\Rightarrow$ can the prisoners agree on a protocol that will set them free?

Protocol 1

$n+1$ prisoners, where $n \geq 2$:

- **Protocol 1**

1. 1*: leader, n:followers*.
2. *n* followers:
   1. first time they enter the room
      1. when the light is off $\longrightarrow$ turn the light on
      2. on other occasions $\longrightarrow$ they do not toggle the switch.
3. leader:
   1. the first *n* times that the light is on when he enters the interrogation room $\longrightarrow$ turns off ;
   2. on other occasions, he does not toggle the switch.
4. After turning the light off for the *n*th time, the leader announces.....

Protocol 2

- **Protocol 2**

1. The leader: does exactly as in **Protocol 1**.
2. The followers do all they do in Protocol 1,
   1. Each follower counts the number of times the state of the light has changed from off to on according to his own observation.
   2. If a follower has observed n such changes, he announces.....

formalization

leader: 0; *followers*:$1, \ldots, n$ ($n \geqslant 2$).
Σ: the set of possible actions for the $n+1$ prisoners, $i = 0, \ldots, n$ is as follows:

- t_i: i toggles
- a_i: i announces
- e_i: i enters
- x_i: i exits

formalization

P (atomic propositions):

- l: light is on
- fin: protocol terminates
- q_i: i has toggled the switch
- m_i: the light was on, last time when i left the room (where $i \neq 0$)
- p_0^j: 0 has toggled the light for at least j times (where $0 \leqslant j \leqslant n$)
- p_i^j: i has counted off-on changes for at least j times (where $i \neq 0$)

formalization

The post-conditions are given by the following table (where the remaining post-conditions are the identity).

(1)	$\iota(a_i, fin) = \top$	$i \geqslant 0$
(2)	$\iota(x_i, m_i) = l$	$i \geqslant 0$
(3)	$\iota(t_i, q_i) = \top$	$i \geqslant 0$
(4)	$\iota(t_i, l) = \neg l$	$i \geqslant 0$
(5)	$\iota(t_0, p_0^j) = p_0^j \vee (p_0^{j-1} \wedge l)$	$j > 0$
(6)	$\iota(e_i, p_i^j) = p_i^j \vee (p_i^{j-1} \wedge ((\neg m_i \wedge l) \vee (\neg q_i \wedge \neg l)))$	$i > 0$

Protocol 1 $\eta_1 = (?\neg fin \cdot \sum_{i=0}^{n}(e_i \cdot \theta_i \cdot x_i))^*$, where:

- $\theta_0 := ?l \cdot t_0 \cdot (?p_0^n \cdot a_0 + ?\neg p_0^n) + ?\neg l$
- $\theta_i := ?(\neg l \wedge \neg q_i) \cdot t_i + ?\neg(\neg l \wedge \neg q_i)$

Protocol 2 $\eta_2 = (?\neg fin \cdot \sum_{i=0}^{n}(e_i \cdot \theta_i' \cdot x_i))^*$, where:

- $\theta_0' := \theta_0$
- $\theta_i' := ?p_i^n \cdot a_i + ?\neg p_i^n \cdot \theta_i$

Example (Protocol 1)

- three prisoners $\{0, 1, 2\}$
- the sequence of interrogations is 1020.
- The initial situation:
  - a singleton expectation model $\mathcal{M}, s$ with the universal protocol Σ^*_{LB} and the valuation assigning $\top$ only to p^0_i for all $i \geqslant 0$.

Example (Protocol 1)

- three prisoners $\{0, 1, 2\}$
- the sequence of interrogations is 1020.
- The initial situation:
  - a singleton expectation model $\mathcal{M}, s$ with the universal protocol Σ_{LB}^* and the valuation assigning $\top$ only to p_i^0 for all $i \geqslant 0$.

Example (Protocol 1)

	l	fin	q_0	q_1	q_2	p_0^0	p_0^1	p_0^2
$\mathcal{M}$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\top$	$\bot$	$\bot$
e_1	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\top$	$\bot$	$\bot$
$t_1 \cdot x_1 \cdot e_0$	$\top$	$\bot$	$\bot$	$\top$	$\bot$	$\top$	$\bot$	$\bot$
$t_0 \cdot x_0 \cdot e_2$	$\bot$	$\bot$	$\top$	$\top$	$\bot$	$\top$	$\top$	$\bot$
$t_2 \cdot x_2 \cdot e_0$	$\top$	$\bot$	$\top$	$\top$	$\top$	$\top$	$\top$	$\bot$
t_0	$\bot$	$\bot$	$\top$	$\top$	$\top$	$\top$	$\top$	$\top$
a_0	$\bot$	$\top$	$\top$	$\top$	$\top$	$\top$	$\top$	$\top$

$\mathcal{M}, s \models [!\eta_1^{\mathcal{F}}] < e_1 \cdot t_1 \cdot x_1 \cdot e_0 \cdot t_0 > (\neg < a_0 > \top \wedge < x_0 \cdot e_2 \cdot t_2 \cdot x_2 \cdot e_0 \cdot t_0 >< a_0 > \top)$

Example (Protocol 1)

	l	fin	q_0	q_1	q_2	p_0^0	p_0^1	p_0^2
$\mathcal{M}$	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\top$	$\bot$	$\bot$
e_1	$\bot$	$\bot$	$\bot$	$\bot$	$\bot$	$\top$	$\bot$	$\bot$
$t_1 \cdot x_1 \cdot e_0$	$\top$	$\bot$	$\bot$	$\top$	$\bot$	$\top$	$\bot$	$\bot$
$t_0 \cdot x_0 \cdot e_2$	$\bot$	$\bot$	$\top$	$\top$	$\bot$	$\top$	$\top$	$\bot$
$t_2 \cdot x_2 \cdot e_0$	$\top$	$\bot$	$\top$	$\top$	$\top$	$\top$	$\top$	$\bot$
t_0	$\bot$	$\bot$	$\top$	$\top$	$\top$	$\top$	$\top$	$\top$
a_0	$\bot$	$\top$	$\top$	$\top$	$\top$	$\top$	$\top$	$\top$

$$\mathcal{M}, s \models [!\eta_1^{\mathcal{F}}] < e_1 \cdot t_1 \cdot x_1 \cdot e_0 \cdot t_0 > (\neg < a_0 > \top \wedge < x_0 \cdot e_2 \cdot t_2 \cdot x_2 \cdot e_0 \cdot t_0 >< a_0 > \top)$$

Thanks