How to Verify an Epistemic Protocol with DEL

Jan van Eijck
CWI, Amsterdam

(based on joint work [1] with Hans van Ditmarsch and William Wu)

Lorentz Workshop on Formal Theories of Communication, 22 Feb 2010

Abstract

Verifying an epistemic protocol involves creating a formalized version of the protocol in a suitable logical language, and next showing (i) that the steps of the protocol are in one to one correspondence with the steps in its formalized version, (ii) that the formalized version satisfies certain correctness conditions, and (iii) hence, that the original version also satisfies these conditions. We will show that DEL is a suitable medium for carrying out this program for an interesting example protocol.

A Riddle and A Protocol

100 Prisoners and a Lightbulb

A group of 100 prisoners, all together in the prison dining area, are told that they will be all put in isolation cells and then will be interrogated one by one in a room containing a light with an on/off switch. The prisoners may communicate with one another by toggling the light-switch (and in no other way). The light is initially switched off. There is no fixed order of interrogation. Every day one prisoner will get interrogated. At any stage every prisoner will be interrogated again sometime.

When interrogated, a prisoner can either 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, but if it is false, they will all be executed. Can the prisoners agree on a protocol that will set them free?

A Protocol for Solving the Riddle

The set of prisoners is $\{0, \ldots, n-1\}$, with $n \geq 2$.

The prisoners appoint one among them as the counter. We will assume prisoner 0 is appointed as counter.

All prisoners except the counter act as follows: the first time they enter the room when the light is off, they switch it on; on all next occasions, they do nothing.

The counter acts as follows: The first $n-2$ times that the light is on when he enters the interrogation room, he turns it off. Then the next time he enters the room when the light is on, he announces that everybody has been interrogated.

How to Prove the Protocol Correct?

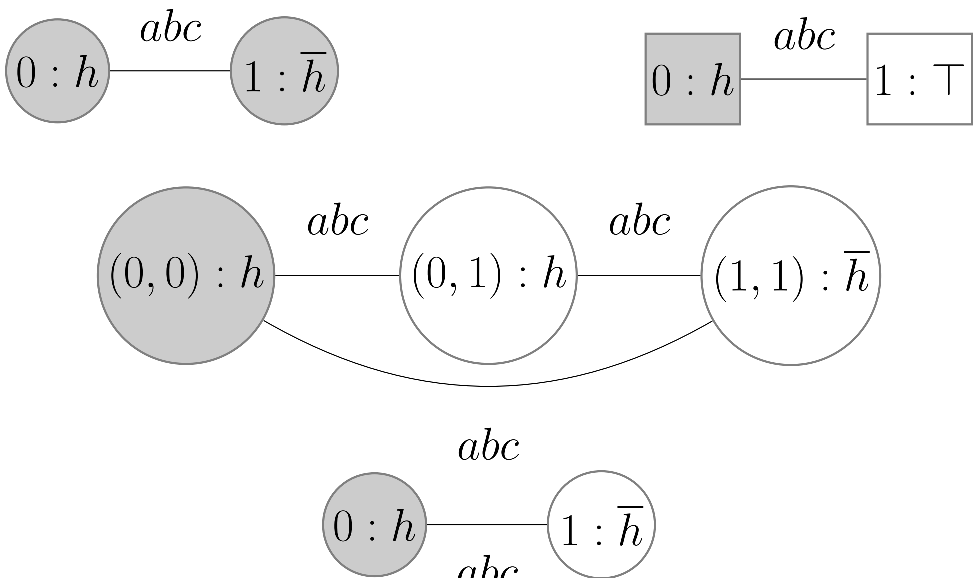
To formally prove that this protocol does indeed solve the problem, we have to first move to a formal version.

We will use DEL (with some minor variations) for formalizing the protocol solution.

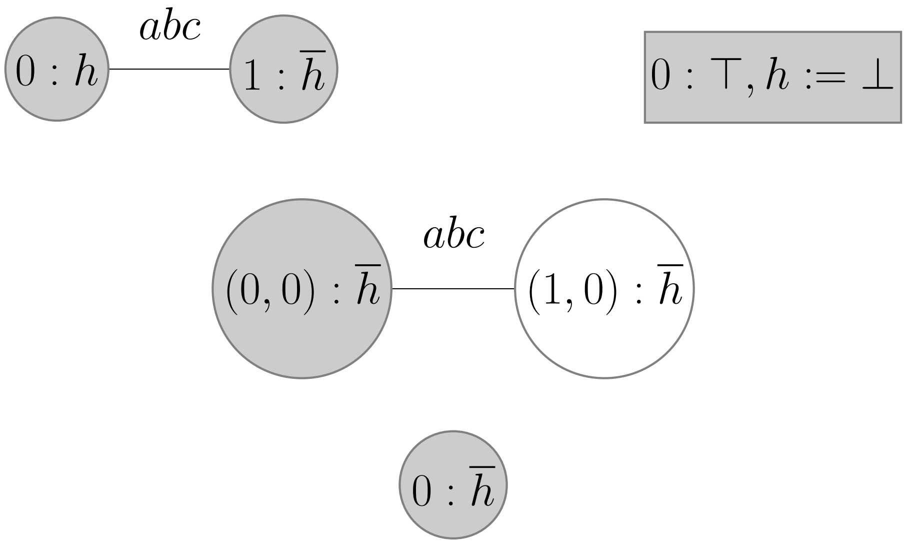
Next, we will give a formal proof that the solution is indeed correct by showing that the formal version of the protocol matches the informal version step-by-step.

DEL — Update Product

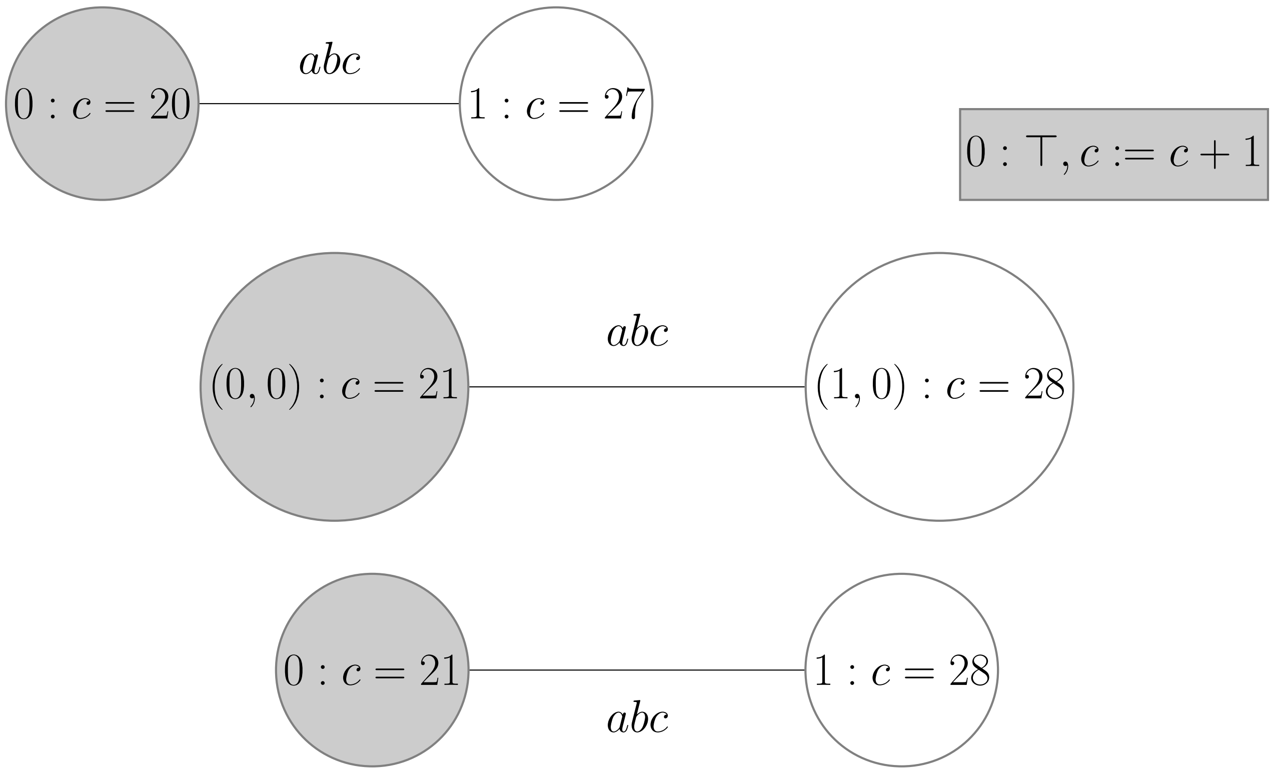
Use of update product to model the effects of communication (in a very broad sense):



DEL — Adding Factual Change



DEL — Adding Registers and Counting



Events

For $i \in \{0, \ldots, n-1\}$, let e_i be the event of the interrogation of prisoner i.

Let *light* express that the light is on.

If the light is on and if event e_0 (interrogation of the counter) takes place, then afterwards the light should be off, and the counter should know that it is off:

$$\textit{light} \rightarrow [e_0]K_0\neg\textit{light}$$

For $i \in \{1, \ldots, n-1\}$, let q_i express that prisoner i has toggled the light switch. Then for all $i \in \{1, \ldots, n-1\}$, the following should be true:

$$(\neg q_i \wedge \neg\textit{light}) \rightarrow [e_i](q_i \wedge \textit{light}).$$

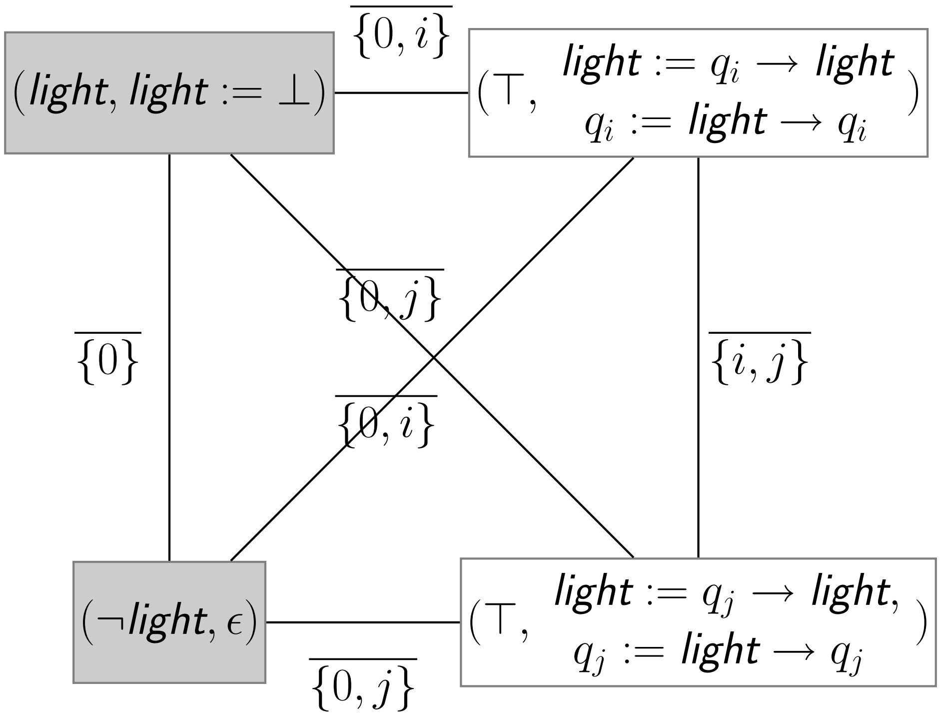
Actions of the prisoners according to the protocol

- prisoner 0, *light*: *light* $:= \bot$ (switch off light),
- prisoner 0, $\neg$*light*: ϵ (do nothing);
- prisoner $i \neq 0$, $\{$*light* $:= q_i \rightarrow$ *light*$, q_i :=$ *light* $\rightarrow q_i\}$.

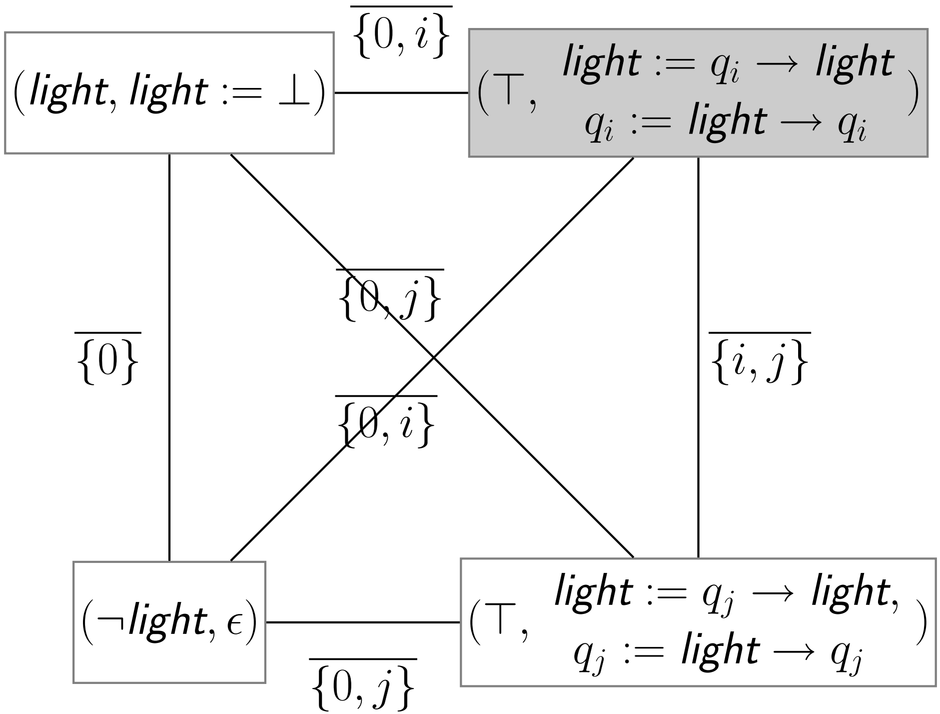
Effect of *light* $:= q_i \rightarrow$ *light*. If *light* is true, then *light* remains true, if *light* is false, then *light* will become true if q_i is false, and will remain false otherwise. This is in accordance with the informal version of the protocol.

Effect of $q_i :=$ *light* $\rightarrow q_i$. If q_i is true, then q_i will remain true. If q_i is false then q_i will become true if *light* is false, and will remain false otherwise. This is in accordance with the informal version of the protocol.

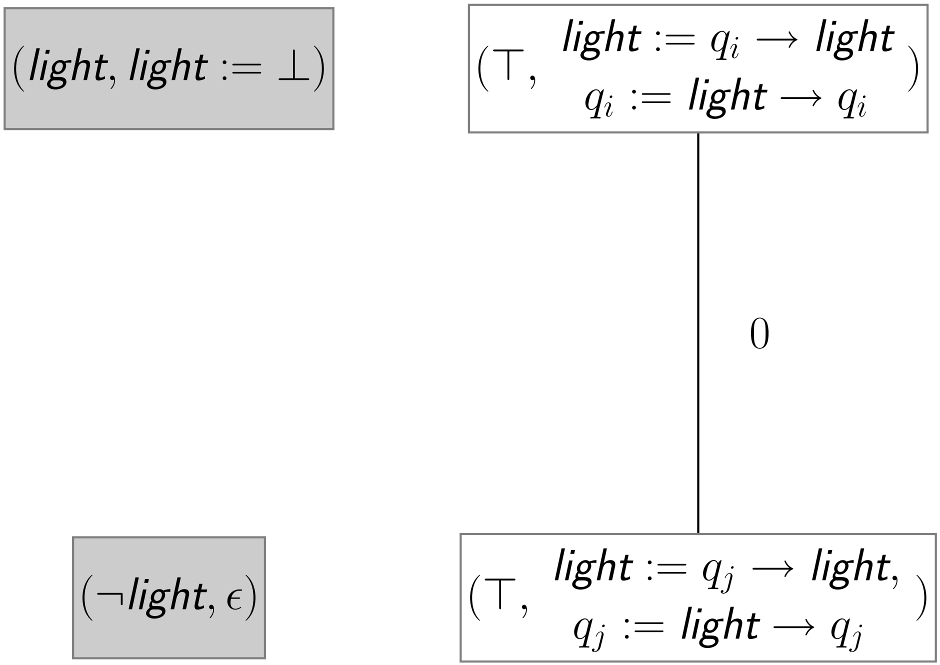
Perspective of Counter on Protocol



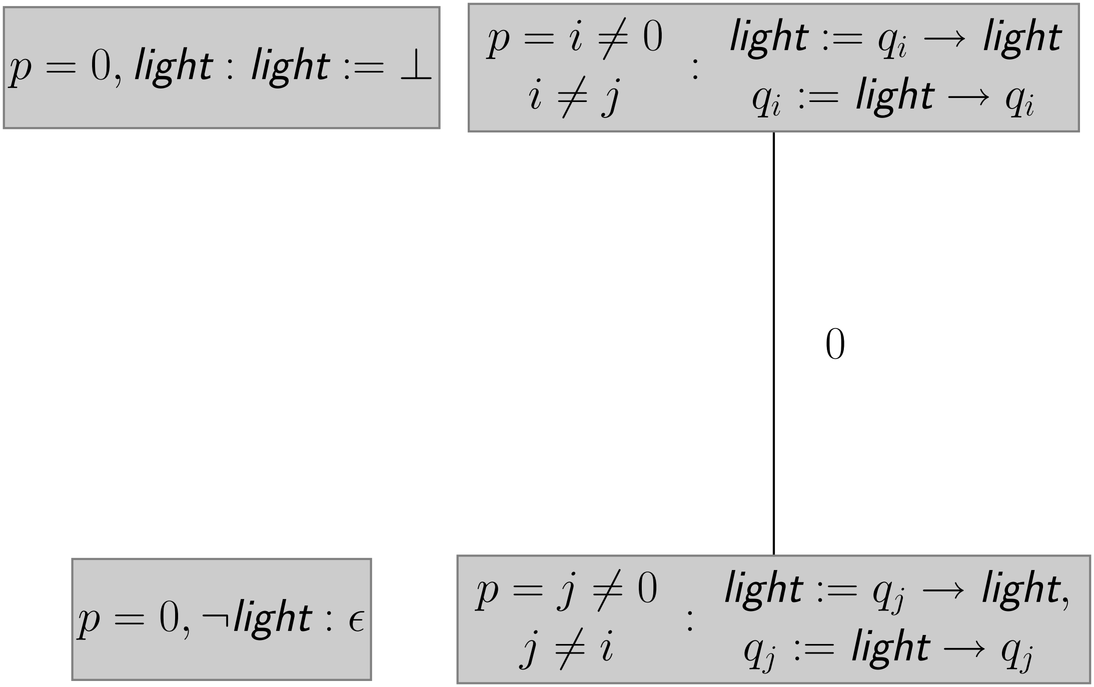
Perspective of Non-Counter on Protocol



Restriction to Epistemic Accessibilities of Counter



Putting Prisoner Number in Precondition



Letting the Counter Count: Formal Version of the Protocol

$$p = 0, \textit{light} : \begin{array}{c} \textit{light} := \bot, \\ c := c + 1 \end{array}$$

$$\begin{array}{c} p = i \neq 0 \\ i \neq j \end{array} : \begin{array}{c} \textit{light} := q_i \rightarrow \textit{light} \\ q_i := \textit{light} \rightarrow q_i \end{array}$$

0

$$p = 0, \neg\textit{light} : \epsilon$$

$$\begin{array}{c} p = j \neq 0 \\ j \neq i \end{array} : \begin{array}{c} \textit{light} := q_j \rightarrow \textit{light}, \\ q_j := \textit{light} \rightarrow q_j \end{array}$$

Initial Model

$$p = 0, c = 0,$$
$$\neg \mathit{light}$$
$$\neg q_1, \ldots, \neg q_{n-1}$$

Representing updates

e_i gets represented by update with

$\top : p := i$ followed by P

where P is the formal version of the protocol.

Update Effects

Effect of update with event e_0 in the initial situation: nothing happens. Why?

Effect of update with event e_i for $i \neq 0$ in the initial situation: the light gets turned on, but from the point of view of the counter, anyone could have done it.

For the case of 100 prisoners, this gives 99 different possibilities, all indistinguishable for the counter.

For the next event e_j where a prisoner switches the light on, there are 98 possibilities, all indistinguishable for the counter, and so on.

Fortunately, we can do much better.

DEL with Awareness Restrictions

The awareness restriction of an agent is a subset of the set of propositional variables and registers (integer variables). These are the variables and registers that the agent is aware of.

Awareness of counter: $p = 0$ versus $p \neq 0$, c, *light*.

Awareness equivalence on possible worlds: $w \approx_i w'$ iff the valuations of the worlds restricted to i-awareness are the same.

Awareness equivalence on possible actions: $s \approx_i s'$ if preconditions and substitutions of the actions are invariant for i-awareness.

ϕ is invariant for i-awareness if $w \approx_i w'$ implies $(M, w \models \phi$ iff $M, w' \models \phi)$.

γ is invariant for i-awareness if $w \approx_i w'$ implies $w^\gamma \approx w'^\gamma$.

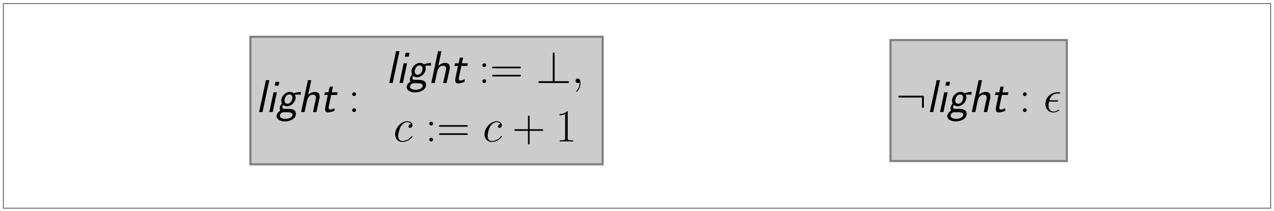
Actions Modulo Awareness Restriction

$$\boxed{\begin{array}{c} p = i \neq 0 \\ \neg \textit{light}, \neg q_i \end{array} : \begin{array}{c} \textit{light} := \top \\ q_i := \top \end{array}} \approx_0 \boxed{\begin{array}{c} p = j \neq 0 \\ \neg \textit{light}, \neg q_j \end{array} : \begin{array}{c} \textit{light} := \top \\ q_j := \top \end{array}}$$

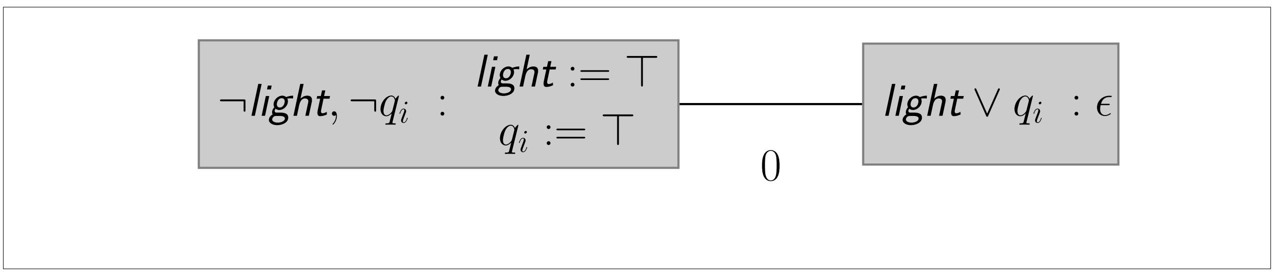
$$\boxed{\begin{array}{c} p = i \neq 0 \\ \textit{light} \vee q_i \end{array} : \epsilon} \approx_0 \boxed{\begin{array}{c} p = j \neq 0 \\ \textit{light} \vee q_j \end{array} : \epsilon}$$

New Version of Protocol: Implementation of Possible Events

e_0: $\boxed{\top, p := 0}$ followed by



e_i, $i \neq 0$: $\boxed{\top, p := i}$ followed by



Adjustment of product update: use match of precondition modulo $\approx_0$.

Interrogation Sequences

An interrogation sequence for n prisoners numbered $0, \ldots, n-1$ is an infinite list of natural numbers, with each number less than n.

Example:

$$\sigma = 0 : 1 : 2 : 3 : 4 : 5 : \sigma$$

Other way to write the same σ:

$$\sigma = [0,1,2,3,4,5] \mathbin{+\!\!+} \sigma$$

where $\mathbin{+\!\!+}$ is the operation that concats a finite list and a (finite or infinite) list.

σ_i is i-th member of σ, counting from 0.

Fairness of Sequences

σ is a fair interrogation sequence for n prisoners if

- for each i, $0 \leq s_i < n$ (σ is a sequence for n prisoners), and
- for each $i \in \mathbb{N}$ and each $j \in \{0, \ldots, n-1\}$ there is a $k \in \mathbb{N}$ with $\sigma_{i+k} = j$ (at each point i, each prisoner j will be interrogated at some future point $i + k$).

Input-Output Format

Here is one way to do it:

Input for the case where there are n prisoners: an infinite stream over $\{0, \ldots, n-1\}$, i.e., a member of the set $\{0, \ldots, n-1\}^\infty$.

Output is a natural number (or the protocol runs forever).

View of the informal protocol PROT_n for the case of n prisoners as a function

$$\text{PROT}_n :: \{0, \ldots, n-1\}^\infty \to \mathbb{N} \cup \{\infty\}$$

Correctness Statement

If σ is a fair interrogation sequence for n prisoners, then protocol PROT_n will output a natural number k with the property that

$$\{0, \ldots, n-1\} \subseteq \{\sigma_i \mid i < k\}.$$

This is a formal version of the informal statement that after the k-th interrogation, all of the n prisoners have been interrogated.

Update and Evaluation

Let M_0 be the initial epistemic model given before. Let $\mathbf{M}$ be the set of all Kripke models with valuations over the signature. Let E be the set of update events.

Let U be the function $\mathbf{M} \to E^\infty \to \mathbf{M}^\infty$ given by:

$$U\ M\ (e : \mathsf{es}) = M \circ e\ :\ U\ (M \circ e)\ (\mathsf{es}).$$

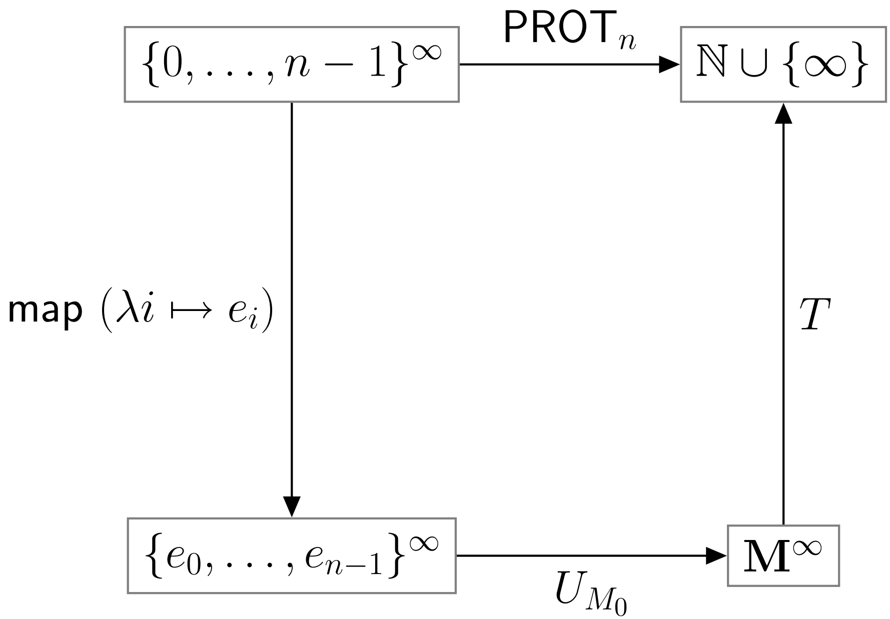
Then if the sequence of events starts $e_0, e_1, e_2, \ldots$, the image of U_{M_0} starts

$$M_0 \circ e_0, M_0 \circ e_0 \circ e_1, M_0 \circ e_0 \circ e_1 \circ e_2, \ldots$$

Let T be the function $\mathbf{M}^\infty \to \mathbb{N} \cup \{\infty\}$ given by

$$\begin{aligned} T(M : \mathsf{ms}) &= T_0(M : \mathsf{ms}) \\ T_i(M : \mathsf{ms}) &= \begin{cases} i & \text{if } M \models K_0(\mathit{light} \wedge c = n-2), \\ T_{i+1}(\mathsf{ms}) & \text{otherwise.} \end{cases} \end{aligned}$$

Diagram



Correctness statement: for all fair streams σ this diagram commutes on an natural number.

Correctness Proof

Induction on the number of prisoners n.

Case $n = 2$: PROT_2 ends after the first occurrence of 10 in the input stream. By fairness, 10 must occur in the stream. After $e_1 e_0$ occurs in the event stream, $K_0(\mathit{light} \wedge c = 0)$ is true in the resulting epistemic model, and T halts at the position of that model.

Induction step: assume the diagram commutes for all fair streams σ for PROT_n. We have to show that it also commutes for all fair streams for PROT_{n+1}. Let n be the last prisoner that has not been counted (rename prisoners if necessary). From the induction hypothesis we get that there is some k with $M_k \models K_0(\mathit{light} \wedge c = n - 2)$. Since σ is fair, the pattern $n \cdots 0$ has to occur after position k. Execution of e_n followed by $\cdots$ followed by execution of e_0 will create a model M with $M \models K_0(\mathit{light} \wedge c = n - 1)$.

Conclusions, Discussion

- Protocol modeling is an art, proving correctness statements is a science.
- DEL is a suitable medium for reasoning about communication protocols; practicing the art of DEL modeling is useful for finding interesting DEL extensions.
- An implementation of epistemic model checking for this example is available from the author upon request [2].
- Meta question: Why should (rational) prisoners agree on this protocol in the first place?
- Intuitive answer to meta question: because it is common knowledge that if the interrogation sequence is fair, the protocol will give a correct solution. (Motivates move from DEL to DEL+LTL [3].)

References

[1] Hans van Ditmarsch, Jan van Eijck, and William Wu. One hundred prisoners and a lightbulb — logic and computation. Twelfth International Conference on the Principles of Knowledge Representation and Reasoning, Toronto, Canada, May 2010.

[2] Jan van Eijck. DEMO — a demo of epistemic modelling. In Johan van Benthem, Dov Gabbay, and Benedikt Löwe, editors, Interactive Logic — Proceedings of the 7th Augustus de Morgan Workshop, number 1 in Texts in Logic and Games, pages 305–363. Amsterdam University Press, 2007.

[3] A. Pnueli. A temporal logic of programs. Theoretical Computer Science, 13:45–60, 1981.