Noname manuscript No.
(will be inserted by the editor)

Reasoning about Agent Types
and the hardest logic puzzle ever

Fenrong Liu · Yanjing Wang

the date of receipt and acceptance should be inserted later

Abstract In this paper, we first propose a simple formal language to specify types of agents in terms of sufficient conditions for their announcements. Based on this language, types of agents are treated as 'first-class citizens' and studied extensively in various dynamic epistemic frameworks which are suitable for reasoning about knowledge and agent types via announcements and questions. To demonstrate our approach, we discuss various versions of Smullyan's *Knights and Knaves* puzzles, including the *Hardest Logic Puzzle Ever* (HLPE) proposed in (Boolos, 1996). In particular, we formalize HLPE and verify a classic solution to it. Moreover, we propose a spectrum of new puzzles based on HLPE by considering subjective (knowledge-based) agent types and relaxing the implicit epistemic assumptions in the original puzzle. The new puzzles are harder than the previously proposed ones in the literature, in the sense that they require deeper epistemic reasoning. Surprisingly, we also show that a version of HLPE in which the agents do not know the others' types does not have a solution at all. Our formalism paves the way for studying these new puzzles using automatic model checking techniques.

Keywords agent types, public announcement logic, questioning strategy, knight and knaves, the hardest logic puzzle ever

The authors are ordered alphabetically. Their contributions are equally important.

Fenrong Liu
Department of Philosophy
Tsinghua University, Beijing 100084, CHINA
E-mail: fenrong@tsinghua.edu.cn

Yanjing Wang
Department of Philosophy & Institute of Foreign Philosophy
Peking University, Beijing 100871, CHINA
E-mail: y.wang@pku.edu.cn

1 Introduction

In his popular book (Smullyan, 1978), Raymond Smullyan proposed a series of puzzles called *Knights and Knaves*, where the usual goal is to determine who are the knights (truth tellers) and who are the knaves (liars) by asking them questions. One variation of such puzzles is made famous by Boolos (1996), where it is called the *Hardest Logic Puzzle Ever* (HLPE):[1]

> Three gods A, B, and C are called, in some order, True, False, and Random. True always speaks truly, False always speaks falsely, but whether Random speaks truly or falsely is a completely random matter. Your task is to determine the identities of A, B, and C by asking three yes/no questions; each question must be put to exactly one god. The gods understand English, but will answer all questions in their own language, in which the words for *yes* and *no* are *da* and *ja*, in some order. You do not know which word means which.

Boolos (1996) gave a lengthy solution which makes use of solutions to three simpler puzzles. Rabern and Rabern (2008) noticed that the puzzle may be trivialized according to Boolos's original assumption on the behaviour of Random and thus proposed an amended version of HLPE. Uzquiano (2010) gave a two-question solution to the amended version of HLPE and proposed an even harder one which is proven to be not solvable in two questions by Wheeler and Barahona (2012). However, Wintein (2011) argues that the results in (Wheeler and Barahona, 2012) depend on a particular conception of answering self-referential questions truthfully or falsely, and propose a two-question solution to Uzquiano's puzzle based on a different conception. Except for the formal truth theory presented in (Wintein, 2011), existing discussions on HLPE are mostly informal to some extent, featuring Boolean reasoning in finding solutions expressed in natural language which often involve self-referential questions. A complete formalization of such puzzles should take care of many different aspects which are hard to put together, such as questions and answers, liars and truth tellers, epistemic reasoning, and solution concepts for puzzles.

In this paper, we will give a purely formal, yet intuitive account of HLPE-like scenarios, by introducing logical frameworks for reasoning about knowledge by communication under uncertainty of various *agent types*. As suggested in HLPE and other *Knights and Knaves* puzzles, people behave differently in their ways of information exchange. The same utterance may contain different intended information due to different types of the speakers. Here, by *'types'*, we mean the patterns that agents follow in communicating information. Knowledge of agent types is crucial in social communication, in particular for strategic settings where people have to interpret and predict the behaviours of their opponents. By developing our formal framework, our aim is not only to solve puzzles like HLPE, but also to deal with general epistemic reasoning under uncertainty about agent types.

As for HLPE itself, there are several advantages to going purely formal. First of all, some of the existing solutions can be verified formally. More importantly, by making everything precise, we will discover the implicit epistemic assumptions behind those puzzles about agent types. As we will show, modifying those assumptions may

[1] Boolos credits Raymond Smullyan as the originator of the Puzzle and John McCarthy for adding the twist of *ja* and *da*.

change the nature of the puzzles, which also leads to even harder puzzles involving interesting and complicated epistemic reasoning. On the other hand, the formal approach also limits the language of questions that we can use in solving these puzzles. For example, the self-referential questions and temporal-related questions as in (Wheeler and Barahona, 2012) are not expressible in our frameworks due to difficulties in defining their semantics. The good aspect of such limitations is that we can now prove *impossibility results*, e.g., non-existence of solutions to certain harder puzzles. The ultimate goal behind the development of our formal framework is to automate the reasoning process and thus handle the puzzles and other applications in an automatic fashion using computational tools, without tedious analysis of combinatorics hidden behind the scenes.

Related work Our logical framework is based on *Public Annoucement Logic* (PAL) (cf. (Plaza, 2007; Gerbrandy and Groeneveld, 1997)) where announcements update the knowledge of agents. The extra twist here is that *who* said what is important due to the different types of the speakers. Similar issues about agency have been considered in (Liu, 2004) and (Liu, 2009) where different revision policies of different agents towards new incoming information are studied. A particular type of agents, viz. the liar, has been studied in a dynamic epistemic framework similar to PAL in (van Ditmarsch et al, 2011) and (van Ditmarsch, 2011), where the focus is on epistemic effects of lying. The aim of the current paper, however, is to move further by considering general agent types and epistemic reasoning about these. The treatment of the type language is inspired by the analysis of protocols in (Wang, 2011b) where agent types are viewed as simple conditional protocol schemas.

There are a few points worth mentioning about our approach:

- We take agent types as first-class citizens in our logical framework by specifying them formally in a type language. Correspondingly, in the model we have type assignments for each agent. The interpretation of an announcement depends on its speaker's type.
- With both types and agents specified in our logical language, we can formulate complicated sentences and questions (e.g., 'What would be his answer if he were asked whether he is a liar?'). On the other hand, from a technical point of view of expressive power, such intriguing formulas with complex questions and answers can be reduced to formulas of a simple epistemic logic (with types).
- The puzzles are formalized in our framework as pairs consisting of a model and a goal formula. A solution is a questioning strategy that satisfies some conditions represented by model checking problems on the model.

In the rest of the paper, we will walk the readers through our technical developments step by step. Each step will be demonstrated by logic puzzles in the style of Knights and Knaves until we are ready to talk about HLPE and its variations. Section 2 looks at agent types in public announcements. We propose the basic logical framework $\mathtt{PALT}^{\mathbf{T}}$ and provide a complete axiomatization via a reduction to $\mathtt{EL}^{\mathbf{T}}$, epistemic logic with type formulas. In Section 3, we enrich $\mathtt{PALT}^{\mathbf{T}}$ with question and answer operators to obtain a new logic $\mathtt{PQLT}^{\mathbf{T}}$. To formally discuss HLPE, we replace announcement-like answers in $\mathtt{PQLT}^{\mathbf{T}}$ by arbitrary utterances and obtain $\mathtt{PQLT}^{\mathbf{T}}_{\mathbf{U}}$, which also allows us to define solutions to the puzzles formally. $\mathtt{PQLT}^{\mathbf{T}}_{\mathbf{U}}$ is used in Section 4 to verify an existing solution to HLPE. Moreover, a spectrum of

new, harder puzzles is proposed in 5, by considering subjective types instead of objective types and relaxing some of the epistemic assumptions in the original HLPE. We prove that a version of HLPE, where the agents do not know others' types, does not have any solution at all. Section 6 ends the paper with conclusions and further directions.

2 Agent types in public announcements

2.1 Language and semantics

In this work, an *agent type* specifies *necessary* condition for an agent to announce a proposition. For example, a *liar* is someone who *only* announces false propositions, i.e., if he announces ϕ then ϕ must be false, but he does not need to announce every false proposition. We introduce the following type language to specify agent types formally.

Definition 1 (Type language) Given a fixed agent variable $\boldsymbol{x}$ and a fixed formula variable $\boldsymbol{\varphi}$, the set **E** of agent types η is recursively defined as:

$$\begin{aligned}\eta &::= \psi \twoheadleftarrow !_{\boldsymbol{x}}\boldsymbol{\varphi}\\ \psi &::= \top \mid \boldsymbol{\varphi} \mid \neg\psi \mid \psi \wedge \psi \mid K_{\boldsymbol{x}}\psi\end{aligned}$$

where $\top$ stands for tautologies.

Note that $\boldsymbol{x}$ and $\boldsymbol{\varphi}$ are the only variables, thus $K_{\boldsymbol{x}}\boldsymbol{\varphi} \wedge K_{\boldsymbol{y}}\boldsymbol{\psi}$ is not a well-formed type. Each agent type η can also be viewed as a function assigning a precondition to each announcement made by an agent of this type.

We can use this type language to define many intuitive agent types.

Example 1 (objective truth teller, liar and bluffer)

- Type `TT` (truth teller): $\boldsymbol{\varphi} \twoheadleftarrow !_{\boldsymbol{x}}\boldsymbol{\varphi}$
- Type `LL` (liar): $\neg\boldsymbol{\varphi} \twoheadleftarrow !_{\boldsymbol{x}}\boldsymbol{\varphi}$
- Type `LT` (bluffer): $\top \twoheadleftarrow !_{\boldsymbol{x}}\boldsymbol{\varphi}$.

Next, if the knowledge of the speaker is taken into account, we can define more realistic *subjective* types: whether a proposition can be announced depends on the knowledge of the speaker.

Example 2 (subjective truth teller and liar)

- Type `STT` (subjective truth teller): $K_{\boldsymbol{x}}\boldsymbol{\varphi} \twoheadleftarrow !_{\boldsymbol{x}}\boldsymbol{\varphi}$
- Type `SLL` (subjective liar): $K_{\boldsymbol{x}}\neg\boldsymbol{\varphi} \twoheadleftarrow !_{\boldsymbol{x}}\boldsymbol{\varphi}$.

Remark 1 The above are just some examples of agent types. Other interesting types can be defined if we enrich the type language with other operators, e.g., $K_{\mathbf{G}}$ (everyone knows that ...) or $C_{\mathbf{G}}$ (it is common knowledge that ...). For example, a *progressive* speaker may only want to announce ϕ if ϕ is not known by all the audience. We may define the following types:

- Type `PSTT` (progressive subjective truth teller):
  $K_{\boldsymbol{x}}\boldsymbol{\varphi} \wedge K_{\boldsymbol{x}}\neg K_{\mathbf{G}}\boldsymbol{\varphi} \twoheadleftarrow !_{\boldsymbol{x}}\boldsymbol{\varphi}$

– Type CSLL (cautious subjective liar):
$K_{\boldsymbol{x}}\neg\boldsymbol{\varphi} \wedge K_{\boldsymbol{x}}\neg K_{\mathbf{G}}\neg\boldsymbol{\varphi} \twoheadleftarrow !_{\boldsymbol{x}}\boldsymbol{\varphi}$

Based on a finite set of agent types we can build our first logical language:

Definition 2 (Public announcement language with types) Given a finite set $\mathbf{T} \subseteq \mathbf{E}$ of agent types, a finite set $\mathbf{G}$ of agent names, a set $\mathbf{P}$ of basic proposition letters, the language $\mathtt{PALT}^{\mathbf{T}}$ is defined as:

$$\phi \ ::= \ \top \mid p \mid \eta(a) \mid \neg\phi \mid \phi \wedge \phi \mid K_a\phi \mid [!_a\phi]\phi$$

where $p \in \mathbf{P}$, $a \in \mathbf{G}$ and $\eta \in \mathbf{T}$.

We call the announcement-free fragment of $\mathtt{PALT}^{\mathbf{T}}$ the *epistemic language with type formulas* ($\mathtt{EL}^{\mathbf{T}}$) and sometimes denote $\mathtt{PALT}^{\mathbf{T}}$ by $\mathtt{EL}^{\mathbf{T}} + [!_a\phi]$.

The superscript $\mathbf{T}$ in $\mathtt{PALT}^{\mathbf{T}}$emphasises that the properties of $\mathtt{PALT}^{\mathbf{T}}$ may depend on the specific $\mathbf{T}$ that is selected. As usual, we have the following abbreviations: $\bot := \neg\top$, $\phi \vee \psi := \neg(\neg\phi \wedge \neg\psi), \phi \to \psi := \neg\phi \vee \psi, \langle !_a\psi\rangle\phi := \neg[!_a\psi]\neg\phi, \hat{K}_a\phi := \neg K_a\neg\phi$. We also write $K_a^W\phi$ for $K_a\phi \vee K_a\neg\phi$, meaning that a knows *whether* ϕ. $\eta(a)$ expresses that agent a is of the type η and $[!_a\psi]\phi$ says that if a can announce ψ then after the announcement, ϕ holds.

Recall that each η can be viewed as a function. Now given $\eta = \psi(\boldsymbol{\varphi}, \boldsymbol{x}) \twoheadleftarrow !_{\boldsymbol{x}}\boldsymbol{\varphi}$, let $\eta(\phi, a) = \psi[\phi/\boldsymbol{\varphi}, a/\boldsymbol{x}]$, i.e., replacing each occurrence of $\boldsymbol{\varphi}$ in $\psi(\boldsymbol{\varphi}, \boldsymbol{x})$ with ϕ and each occurrences of $\boldsymbol{x}$ with a. Intuitively, an agent a of a type η can announce a concrete proposition ϕ only when $\eta(\phi, a)$ holds. Although two agents may announce the same proposition ϕ, the actual information that it carries can be different due to different agent types.

Definition 3 (Semantics) A model for the language of $\mathtt{PALT}^{\mathbf{T}}$ is a tuple $\mathfrak{M} = (S, \{\sim_a \mid a \in \mathbf{G}\}, V, \lambda)$, where $(S, \{\sim_a \mid a \in \mathbf{G}\}, V)$ is a standard multi-agent **S5** Kripke model: S is a non-empty set of possible worlds, $\sim_a \subseteq S \times S$ is an equivalence relation over S, and $V : S \to 2^{\mathbf{P}}$ is a valuation function assigning to each world a set of basic propositions. The new component $\lambda : S \times \mathbf{G} \to \mathbf{T}$ assigns to each agent on each world a type in $\mathbf{T}$. The semantics of $\mathtt{PALT}^{\mathbf{T}}$ formulas is defined as follows:

$$\begin{array}{|r|}\hline \mathfrak{M}, s \vDash \top \Leftrightarrow \text{always} \\ \mathfrak{M}, s \vDash p \Leftrightarrow p \in V(s) \\ \mathfrak{M}, s \vDash \neg\phi \Leftrightarrow \mathfrak{M}, s \nvDash \phi \\ \mathfrak{M}, s \vDash \phi \wedge \psi \Leftrightarrow \mathfrak{M}, s \vDash \phi \text{ and } \mathfrak{M}, s \vDash \psi \\ \mathfrak{M}, s \vDash K_a\phi \Leftrightarrow \forall t : s \sim_a t \text{ implies } \mathfrak{M}, t \vDash \phi \\ \mathfrak{M}, s \vDash \eta(a) \Leftrightarrow \lambda(s, a) = \eta \\ \mathfrak{M}, s \vDash [!_a\psi]\phi \Leftrightarrow \mathfrak{M}, s \vDash \lambda(s, a)(\psi, a) \text{ implies } \mathfrak{M}|^a_\psi, s \vDash \phi \\ \hline \end{array}$$

where $\mathfrak{M}|^a_\psi$ is defined as $(S', \{\sim'_a \mid a \in \mathbf{G}\}, V', \lambda')$ where:

– $S' = \{t \mid t \in S \text{ and } \mathfrak{M}, t \vDash \lambda(t, a)(\psi, a)\}$
– For each $a \in \mathbf{G}$, $t \in S'$:$\sim'_a = \sim_a |_{S' \times S'}$, $V'(t) = V(t)$ and $\lambda'(t) = \lambda(t)$.

Note that $\mathfrak{M}|^a_\psi$ is well-defined if S' is not empty, and $\mathfrak{M}, s \vDash \lambda(s, a)(\psi, a)$ in the clause of $[!_a\psi]\phi$ guarantees that. We say ϕ is *valid on* $\mathfrak{M}$ ($\mathfrak{M} \vDash \phi$) if, for all s in $\mathfrak{M}$: $\mathfrak{M}, s \vDash \phi$. We say ϕ is *valid* ($\vDash \phi$) if for all the models $\mathfrak{M} : \mathfrak{M} \vDash \phi$.

Remark 2 For generality, we do not assume that the agents always know their types, i.e., $\eta(a) \to K_a\eta(a)$ is not valid, since in some cases an agent may not be aware of its own type although it behaves exactly according to this type.

The above semantics is similar to the one for the standard public announcement logic (PAL) (cf. (Plaza, 2007)), where after an announcement of ϕ, we simply delete all the worlds that do not satisfy ϕ, namely all the worlds where ϕ cannot be truthfully announced. In our setting, under the extra information of agent types, after a's announcing ϕ we delete all worlds where a would not have been able to announce ϕ *according to* a*'s type*.

To be more precise in later discussions, we define the language $\mathtt{PAL}^{\mathbf{T}}$ as $\mathtt{EL}^{\mathbf{T}}+[!\phi]$, public announcement logic with type formulas. Recall that $\mathtt{PALT}^{\mathbf{T}}$ is $\mathtt{EL}^{\mathbf{T}} + [!_a\phi]$, so the only difference between $\mathtt{PALT}^{\mathbf{T}}$ and $\mathtt{PAL}^{\mathbf{T}}$ is that announcements in $\mathtt{PAL}^{\mathbf{T}}$ are agent-less, i.e., announced by a single truth teller: 'the *god*'. Correspondingly, the semantics of $\mathtt{PAL}^{\mathbf{T}}$ differs from the semantics of $\mathtt{PALT}^{\mathbf{T}}$ only in the clause for announcement (we write the relevant satisfaction relation as $\Vdash$):

$$\boxed{\mathfrak{M}, s \Vdash [!\psi]\phi \Leftrightarrow \mathfrak{M}, s \Vdash \psi \text{ implies } \mathfrak{M}|_\psi, s \vDash \phi}$$

where $\mathfrak{M}|_\psi$ is defined as $(S', \{\sim'_a |\ a \in \mathbf{G}\}, V', \lambda')$ where:

- $S' = \{t \mid t \in S \text{ and } \mathfrak{M}, t \Vdash \psi\}$
- For each $a \in \mathbf{G}$, $t \in S'$:$\sim'_a = \sim_a |_{S' \times S'}$, $V'(t) = V(t)$ and $\lambda'(t) = \lambda(t)$.

Note that types play no role in the semantics of announcements in $\mathtt{PAL}^{\mathbf{T}}$. Thus, $\mathtt{PAL}^{\mathbf{T}}$ behaves just like standard PAL equipped with a special set of basic propositions (the type formulas). In the rest of this paper, given a finite set of types $\mathbf{T}$, we let $\mathbf{P_T}$ be the set of *type propositions* i.e., $\{\eta(a) \mid \eta \in \mathbf{T}, a \in \mathbf{G}\}$.

It is a well-known result that public announcement logic can be translated back to epistemic logic qua expressiveness (cf. e.g., (van Ditmarsch et al, 2007)). This result clearly also holds in our setting with type formulas:

Proposition 1 *$\mathtt{PAL}^{\mathbf{T}}$ is equally expressive as $\mathtt{EL}^{\mathbf{T}}$ on **S5** models with type assignments.*

Proof (Sketch) We only define the relevant translation $f : \mathtt{PAL}^{\mathbf{T}} \to \mathtt{EL}^{\mathbf{T}}$ (where $p \in \mathbf{P} \cup \mathbf{P_T}$):

$$\begin{array}{llll}
f(\top) & = \top & f([!\psi]\top) & = f(\psi \to \top) \\
f(p) & = p & f([!\psi]p) & = f(\psi \to p) \\
f(\neg\phi) & = \neg f(\phi) & f([!\psi]\neg\phi) & = f(\psi \to \neg[!\psi]\phi) \\
f(\phi_1 \wedge \phi_2) & = f(\phi_1) \wedge f(\phi_2) & f([!\psi](\phi_1 \wedge \phi_2)) & = f([!\psi]\phi_1 \wedge [!\psi]\phi_2) \\
f(K_a\phi) & = K_a f(\phi) & f([!\psi]K_a\phi) & = f(\psi \to K_a(\psi \to [!\psi]\phi)) \\
 & & f([!\psi][!\chi]\phi) & = f([!\psi]f([!\chi]\phi))
\end{array}$$

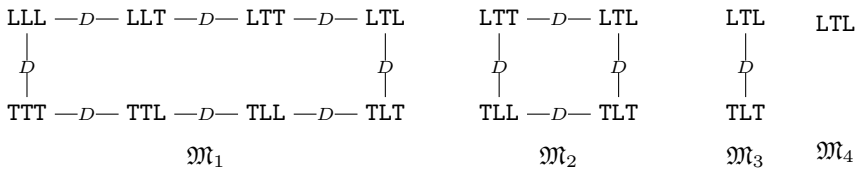
Based on a suitable definition of the complexity of formulas (cf. (van Ditmarsch et al, 2007)) we can show that the translation/rewriting always reduces the complexity. Hence, it will terminate at some point and eliminate all announcement operators in an inside-out fashion.

2.2 Knights and Knaves

Before moving on to technical results about $\texttt{PALT}^{\mathbf{T}}$, we demonstrate the use of this simple yet powerful framework by some examples. Consider the following Knights and Knaves puzzle first introduced by Smullyan (1978).

Example 3 (Three inhabitants) On a fictional island, the inhabitants are either knights, who always tell the truths, or knaves, who always lie. A visitor *D* from the outside world meets three inhabitants *A, B* and *C* on the island. *D* asks them to tell their types. *A* says: *B* is a knave. *B* says: *C* is a knave. *C* says: *A* and *B* are knaves. Now, is it possible for the visitor to find out the inhabitants' types from their statements?

Let us start with the following model $\mathfrak{M}_1$ where A, B, and C know their own types (either TT or LL) but D knows nothing about the types of A, B, and C. Note that we write LLT for a world s where $\lambda(s, A) = \texttt{LL}$, $\lambda(s, B) = \texttt{LL}$ and $\lambda(s, C) = \texttt{TT}$ (similarly for other abbreviations).[2] Following the usual convention in visualizing **S5** models, the actual relations are the reflexive transitive closures of the (bidirectional) ones denoted in the following graphs. $\mathfrak{M}_2$ is the model after A's announcement $!_A(\texttt{LL}(B))$, $\mathfrak{M}_3$ is the model after the second announcement $!_B\texttt{LL}(C)$ and $\mathfrak{M}_4$ is the model after the third announcement $!_C(\texttt{LL}(A) \wedge \texttt{LL}(B))$. Thus $\mathfrak{M}_2 = \mathfrak{M}_1|^A_{\texttt{LL}(B)}$, $\mathfrak{M}_3 = \mathfrak{M}_2|^B_{\texttt{LL}(C)}$ and $\mathfrak{M}_4 = \mathfrak{M}_3|^C_{\texttt{LL}(A)\wedge\texttt{LL}(B)}$.



Note that by the definition of the updated model, $\mathfrak{M}_2 = \mathfrak{M}_1|^A_{\texttt{LL}(B)}$ keeps the worlds s in $\mathfrak{M}_1$ where $\mathfrak{M}_1, s \vDash \lambda(s, A)(\texttt{LL}(B), A)$, that is: it keeps the worlds s satisfying one of the following conditions:

- $\lambda(s, A) = \texttt{TT}$ and $\mathfrak{M}_1, s \vDash \texttt{LL}(B)$,
- $\lambda(s, A) = \texttt{LL}$ and $\mathfrak{M}_1, s \vDash \neg\texttt{LL}(B)$.

Since $\mathbf{T} = \{\texttt{LL}, \texttt{TT}\}$, the above two conditions are equivalent to the following:

- $\lambda(s, A) = \texttt{TT}$ and $\lambda(s, B) = \texttt{LL}$ (i.e., the worlds in the shape of TL_)
- $\lambda(s, A) = \texttt{LL}$ and $\lambda(s, B) = \texttt{TT}$ (i.e., the worlds in the shape of LT_)

It is clear that $\mathfrak{M}_2$ only contains TL_ and LT_. A similar reasoning works for $\mathfrak{M}_3$ and $\mathfrak{M}_4$ by the definition of the updated model.

Note that according to the semantics, for $\langle!_a\psi\rangle\phi$ we have:

$$\mathfrak{M}, s \vDash \langle!_a\psi\rangle\phi \Leftrightarrow \mathfrak{M}, s \vDash \lambda(s, a)(\psi, a) \text{ and } \mathfrak{M}|^a_\psi, s \vDash \phi$$

Now it is easy to see that LTL is the only world s in $\mathfrak{M}_1$ such that all announcements in the story can be successfully announced in the given order:

$$\mathfrak{M}_1, s \vDash \langle!_A\texttt{LL}(B)\rangle\langle!_B\texttt{LL}(C)\rangle\langle!_C(\texttt{LL}(A) \wedge \texttt{LL}(B))\rangle\top$$

[2] Since D's type is irrelevant, we omit it in the model.

Moreover, since $\mathfrak{M}_4$ is a singleton model, it is clear that

$$\mathfrak{M}_1, \mathtt{LTL} \vDash \langle !_A \mathtt{LL}(B)\rangle\langle !_B \mathtt{LL}(C)\rangle\langle !_C(\mathtt{LL}(A) \wedge \mathtt{LL}(B))\rangle K_D(\mathtt{LL}(A) \wedge \mathtt{TT}(B) \wedge \mathtt{LL}(C))$$

Thus, after the three announcements, agent D knows that A and C are liars and B is a truth teller.

Now let us consider another variation of the Knights and Knaves:

Example 4 (Death or Freedom) A and B are standing at a fork in the road. Now comes C. C knows that one of them is a Knight and the other is a Knave, but C does not know who is who. C also knows that one road leads to Death, and the other leads to Freedom. Suppose A is the honest Knight, and he knows which way leads to Freedom, how can A let C know the right way to go?

Note that this puzzle is not trivial, since although A can tell the truth, C may not be sure that A is telling the truth. To solve the puzzle, let us first prove a simple proposition:

Proposition 2 *Given* $\mathbf{T} = \{TT, LL\}$, $a \in \mathbf{G}$ *and any* $PALT^{\mathbf{T}}$ *formula* ϕ, *let* $\phi_a^\circ = (TT(a) \to \phi) \wedge (LL(a) \to \neg\phi)$. *Now for any* $PALT^{\mathbf{T}}$ *formula* ϕ, *and any model* $\mathfrak{M}$, $\mathfrak{M}|^a_{\phi_a^\circ}$ *is the submodel of* $\mathfrak{M}$ *obtained by keeping all the worlds that satisfy* ϕ. *Moreover, for any* $a, b \in \mathbf{G}$ *and any modality-free* $PALT^{\mathbf{T}}$ *formula* ϕ *we have:* $\vDash [!_a(\phi_a^\circ)]K_b\phi$.

Proof For any model $\mathfrak{M}$, $\mathfrak{M}|^a_{\phi_a^\circ}$ only keeps the worlds s where $\mathfrak{M}, s \vDash \lambda(s,a)(\phi_a^\circ, a)$, that is: it keeps the worlds satisfying one of the following conditions:

- $\lambda(s,a) = \mathtt{TT}$ and $\mathfrak{M}, s \vDash \phi_a^\circ$
- $\lambda(s,a) = \mathtt{LL}$ and $\mathfrak{M}, s \vDash \neg\phi_a^\circ$

This can be stated equivalently as:

- $\mathfrak{M}, s \vDash \mathtt{TT}(a)$ and $\mathfrak{M}, s \vDash (\mathtt{TT}(a) \to \phi) \wedge (\mathtt{LL}(a) \to \neg\phi)$
- $\mathfrak{M}, s \vDash \mathtt{LL}(a)$ and $\mathfrak{M}, s \vDash \neg((\mathtt{TT}(a) \to \phi) \wedge (\mathtt{LL}(a) \to \neg\phi)$

which is equivalent to:

- $\mathfrak{M}, s \vDash \mathtt{TT}(a)$ and $\mathfrak{M}, s \vDash \mathtt{TT}(a) \to \phi$
- $\mathfrak{M}, s \vDash \mathtt{LL}(a)$ and $\mathfrak{M}, s \vDash \neg(\mathtt{LL}(a) \to \neg\phi)$

and this is again equivalent to:

- $\mathfrak{M}, s \vDash \mathtt{TT}(a)$ and $\mathfrak{M}, s \vDash \phi$
- $\mathfrak{M}, s \vDash \mathtt{LL}(a)$ and $\mathfrak{M}, s \vDash \phi$

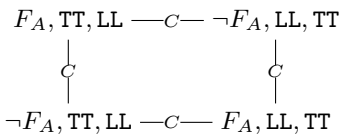
Since $\mathbf{T} = \{\mathtt{TT}, \mathtt{LL}\}$, $\mathfrak{M}|^a_{\phi_a^\circ}$ simply keeps all worlds where ϕ holds no matter what the type of a is. Since the updates do not change the truth values of Boolean formulas, the validity of $[!_a(\mathtt{TT}(a) \to \phi) \wedge (\mathtt{LL}(a) \to \neg\phi))]K_b\phi$ is immediate[3]. □

The preceding proposition says that given $\mathbf{T} = \{\mathtt{LL}, \mathtt{TT}\}$, an agent a can actually mimic a truthful announcement of ϕ, qua epistemic update effects, by $!_a\phi_a^\circ$, no matter what a's type actually is. Now let us come back to Example 4. Let F_A denote the proposition that the road behind A leads to Freedom, thus $\neg F_A$ says that the road behind A leads to Death. A solution to the puzzle of Example 4 is simply as follows:

[3] Truth values of epistemic formulas may not be preserved after announcement. For a study in the setting of PAL, we refer to (van Ditmarsch and Kooi, 2006) and (Holliday and Icard III, 2010).

- If the road behind the Knight is the one leading to Freedom (F_A) then he can say 'if I am a Knight, then the road behind me leads to Freedom, and if I am a Knave, then the road behind me leads to Death' ($!_A((\mathtt{TT}(A) \to F_A) \wedge (\mathtt{LL}(A) \to \neg F_A))$).
- On the other hand if $\neg F_A$ is true then $!_A((\mathtt{TT}(A) \to \neg F_A) \wedge (\mathtt{LL}(A) \to F_A))$ is enough.

To verify that the above solution indeed works, we first build the initial model $\mathfrak{M}$ as follows, where, for example, $(F_A, \mathtt{TT}, \mathtt{LL})$ denotes the world where F_A is true and A is assigned TT while B is assigned LL (similarly for other states).



Then based on Proposition 2, we have:

$$\mathfrak{M} \vDash \mathtt{TT}(A) \to \bigwedge_{\psi \in \{F_A, \neg F_A\}} (\psi \to \langle !_A((\mathtt{TT}(A) \to \psi) \wedge (\mathtt{LL}(A) \to \neg\psi))\rangle K_C^W F_A).$$

Since $\mathfrak{M} \vDash \mathtt{TT}(A) \leftrightarrow \mathtt{LL}(B)$ and $\mathfrak{M} \vDash \mathtt{LL}(A) \leftrightarrow \mathtt{TT}(B)$, we also have:

$$\mathfrak{M} \vDash \mathtt{TT}(A) \to \bigwedge_{\psi \in \{F_A, \neg F_A\}} (\psi \to \langle !_A((\mathtt{TT}(A) \to \psi) \wedge (\mathtt{TT}(B) \to \neg\psi))\rangle K_C^W F_A)$$

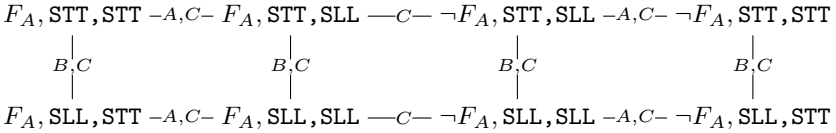
which gives an alternative solution. In words, it lets the Knight say 'The road behind the Knight leads to Freedom.' when F_A is true and 'The road behind the Knight leads to Death' when F_A is not true.

Yet another well-known solution is shorter in terms of announcements:

$$\mathfrak{M} \vDash \mathtt{TT}(A) \to (F_A \to \langle !_A \langle !_B \neg F_A\rangle \top\rangle K_C^W F_A) \wedge (\neg F_A \to \langle !_A \langle !_B F_A\rangle \top\rangle K_C^W F_A)$$

$!_A\langle !_B \neg F_A\rangle \top$ reads: A announces that 'The other guy would say that the road behind me leads to Death' (similarly for $!_A\langle !_B F_A\rangle \top$). The verification of this solution is left to the reader as a simple exercise.

However, the last solution does not work any more, if we make the puzzle harder by letting Knights and Knaves be ignorant of each other's types and replace objective types $\mathtt{TT}, \mathtt{LL}$ by subjective types (let $\mathbf{T} = \{\mathtt{STT}, \mathtt{SLL}\}$). Then the appropriate initial model $\mathfrak{M}'$ may look as follows:



To see what this model says, note the following validity:

$$\mathfrak{M}' \vDash \neg K_A^W \mathtt{STT}(B) \wedge K_C \neg K_A^W \mathtt{STT}(B) \wedge K_A^W F_A \wedge \neg K_C^W F_A \wedge \neg K_C^W \mathtt{STT}(A)$$

This says that A does not know B's type and C knows this, but A does know whether his road leads to Freedom while C does not know, as before, whether A's road leads to Freedom. (The case for B is similar.)

Now suppose the real situation is $(F_A, \mathtt{STT}, \mathtt{SLL})$. Let us verify the previous short solution 'The other guy would say that the road behind me leads to Death' in this state.

$$\begin{aligned}
&\mathfrak{M}', (F_A, \mathtt{STT}, \mathtt{SLL}) \vDash \langle !_A \langle !_B \neg F_A \rangle \top \rangle K_C^W F_A \\
\Longrightarrow\ &\mathfrak{M}', (F_A, \mathtt{STT}, \mathtt{SLL}) \vDash \langle !_A \langle !_B \neg F_A \rangle \top \rangle \top \\
\Longleftrightarrow\ &\mathfrak{M}', (F_A, \mathtt{STT}, \mathtt{SLL}) \vDash \mathtt{STT}(\langle !_B \neg F_A \rangle \top, A) \\
\Longleftrightarrow\ &\mathfrak{M}', (F_A, \mathtt{STT}, \mathtt{SLL}) \vDash K_A \langle !_B \neg F_A \rangle \top \\
\Longleftrightarrow\ &\mathfrak{M}', (F_A, \mathtt{STT}, \mathtt{SLL}) \vDash \langle !_B \neg F_A \rangle \top \text{ and } \mathfrak{M}', (F_A, \mathtt{STT}, \mathtt{STT}) \vDash \langle !_B \neg F_A \rangle \top \\
\Longleftrightarrow\ &\mathfrak{M}', (F_A, \mathtt{STT}, \mathtt{SLL}) \vDash K_B F_A \text{ and } \mathfrak{M}', (F_A, \mathtt{STT}, \mathtt{STT}) \vDash K_B \neg F_A
\end{aligned}$$

However, since $\mathfrak{M}', (F_A, \mathtt{STT}, \mathtt{STT}) \nvDash K_B \neg F_A$, we have

$$\mathfrak{M}', (F_A, \mathtt{STT}, \mathtt{SLL}) \nvDash \langle !_A \langle !_B \neg F_A \rangle \top \rangle K_C^W F_A$$

Therefore, A's announcing 'The other guy would say that the road behind me leads to Death' does not work any more (assuming F_A), since A does not know B's type and as a truth teller he can only say what he knows.

The above example demonstrates that subjective types and knowledge of the agents may make a difference. We will apply a similar modification to HLPE in the later part of the paper.

In the present example, we can overcome the difficulties caused by the ignorance of other players' types by modifying the previous short solution to (assuming F_A): 'I would say my path leads to Freedom (if I were asked)' ($!_A \langle !_A F_A \rangle \top$). Note that this is different from simply announcing F_A, for example:

$$\mathfrak{M}', (F_A, \mathtt{SLL}, \mathtt{STT}) \vDash \langle !_A \langle !_A F_A \rangle \top \rangle \top \text{ but } \mathfrak{M}', (F_A, \mathtt{SLL}, \mathtt{STT}) \nvDash \langle !_A F_A \rangle \top.$$

We can verify that this modified solution indeed works:

$$\mathfrak{M}' \vDash \mathtt{STT}(A) \to ((F_A \to \langle !_A \langle !_A F_A \rangle \top \rangle K_C^W F_A) \wedge (\neg F_A \to \langle !_A \langle !_A \neg F_A \rangle \top \rangle K_C^W F_A))$$

2.3 Axiomatization

Our language $\mathtt{PALT}^{\mathbf{T}}$ looks similar to $\mathtt{PAL}^{\mathbf{T}}$. In this section, we will make the link precise and use it to obtain a complete axiomatization of $\mathtt{PALT}^{\mathbf{T}}$. To ease the discussion, let us first define some useful notations.

Given a finite set of types $\mathbf{T}$, let δ^a_ϕ be an abbreviation of $\bigvee_{\eta \in \mathbf{T}} (\eta(a) \wedge \eta(\phi, a))$ where $\eta(a)$ is a formula and $\eta(\phi, a)$ is the value (a formula) of the function η on the input (ϕ, a). Since in our models, an agent can have only one type at each state, each world can satisfy at most one disjunct of $\bigvee_{\eta \in \mathbf{T}} (\eta(a) \wedge \eta(\phi, a))$.

Now we can rewrite each $\mathtt{PALT}^{\mathbf{T}}$ formula into a $\mathtt{PAL}^{\mathbf{T}}$ formula by recursively replacing each $[!_a \psi]$ modality in $\mathtt{PALT}^{\mathbf{T}}$ formulas by an announcement modality in

$\texttt{PAL}^{\mathbf{T}}$. Formally, we define a translation $t : \texttt{PALT}^{\mathbf{T}} \to \texttt{PAL}^{\mathbf{T}}$ as follows:

$$t(\top) = \top \qquad t(p) = p \qquad t(\eta(a)) = \eta(a)$$
$$t(\neg\phi) = \neg t(\phi) \qquad t(\phi \wedge \psi) = t(\phi) \wedge t(\psi) \qquad t(K\phi) = Kt(\phi)$$
$$t([!_a\psi]\phi) = [!t(\delta^a_\psi)]t(\phi)$$

For example, given $\mathbf{T} = \{\texttt{TT}, \texttt{LL}\}$:

$$\begin{aligned} & t([!_a[!_b\texttt{TT}(a)]\bot]\bot) \\ =& [!((\texttt{TT}(a) \wedge t([!_b\texttt{TT}(a)]\bot)) \vee (\texttt{LL}(a) \wedge \neg t([!_b\texttt{TT}(a)]\bot)))]\bot \end{aligned}$$

where $t([!_b\texttt{TT}(a)]\bot) = [!((\texttt{TT}(b) \wedge \texttt{TT}(a)) \vee (\texttt{LL}(b) \wedge \neg\texttt{TT}(a)))]\bot$.

The result is a faithful $\texttt{PAL}^{\mathbf{T}}$ translation of $\texttt{PALT}^{\mathbf{T}}$ formulas.

Proposition 3 *For any $\texttt{PALT}^{\mathbf{T}}$ formula ϕ, and any pointed $\mathfrak{M}, s$: $\mathfrak{M}, s \vDash \phi \iff \mathfrak{M}, s \Vdash t(\phi)$.*

Proof We prove the proposition by induction on the structure of ϕ. The Boolean cases and the $K_a\phi$ case are trivial. Before we can approach the $[!_a\psi]\phi$ case, we need to prove the following claim within the induction for ϕ:

If $\mathfrak{M}, s \vDash \psi \iff \mathfrak{M}, s \Vdash t(\psi)$, then $\mathfrak{M}, s \vDash \lambda(s,a)(\psi, a) \iff \mathfrak{M}, s \Vdash t(\delta^a_\psi)$.
The argument goes by the following chain of equivalences:

$$\begin{aligned} & \mathfrak{M}, s \vDash \lambda(s,a)(\psi,a) \\ \iff & \mathfrak{M}, s \vDash \eta^*(a) \wedge \eta^*(\psi, a) \quad (\text{where } \eta^* = \lambda(s,a)) \\ \iff & \mathfrak{M}, s \Vdash \eta^*(a) \wedge t(\eta^*(\psi,a)) \quad (\text{see below}) \\ \iff & \mathfrak{M}, s \Vdash \bigvee_{\eta \in \mathbf{T}} (\eta(a) \wedge t(\eta(\psi,a))) \quad (\text{since } a \text{ has one and only one type on } s) \\ \iff & \mathfrak{M}, s \Vdash t(\bigvee_{\eta \in \mathbf{T}} (\eta(a) \wedge \eta(\psi,a))) \quad (\text{since } t \text{ commutes with } \wedge \text{ and } \neg) \\ \iff & \mathfrak{M}, s \Vdash t(\delta^a_\psi) \end{aligned}$$

Here the crucial second ' $\iff$ ' is due to the following: (i) $\mathfrak{M}, s \Vdash \eta^*(a) \iff \mathfrak{M}, s \vDash \eta^*(a)$; (ii) the assumption that $\mathfrak{M}, s \vDash \psi \iff \mathfrak{M}, s \Vdash t(\psi)$; (iii) the fact that $\eta^*(\psi, a)$ is constructed by Boolean connectives and epistemic operators based on ψ (by the definition of the type language); (iv) the Boolean cases and the $K_a\phi$ case in the main inductive proof.

Now based on the above claim, we know that $\mathfrak{M}|_{t(\delta^a_\psi)}$ is exactly the same as $\mathfrak{M}|^a_\psi$. Then the following reasoning for $[!_a\psi]\phi$ is immediate:

$$\begin{aligned} & \mathfrak{M}, s \vDash [!_a\psi]\phi \\ \text{iff } & \mathfrak{M}, s \vDash \lambda(s,a)(\psi,a) \text{ implies } \mathfrak{M}|^a_\psi, s \vDash \phi \\ \text{iff } & \mathfrak{M}, s \Vdash t(\delta^a_\psi) \text{ implies } \mathfrak{M}|_{t(\delta^a_\psi)}, s \Vdash t(\phi) \\ \text{iff } & \mathfrak{M}, s \Vdash t([!_a\psi]\phi). \end{aligned}$$

□

Note that the above proposition does not imply that we may just forget about $\texttt{PALT}^{\mathbf{T}}$: the translation that we defined clearly introduces an exponential blow-up in the length of formulas. For example, the executability of the announcements in Example 3 can be translated into the following formula with standard public announcements:[4]

$\langle !((\texttt{TT}(A) \wedge \texttt{LL}(B)) \vee (\texttt{LL}(A) \wedge \neg\texttt{LL}(B)))\rangle\langle !((\texttt{TT}(B) \wedge \texttt{LL}(C)) \vee (\texttt{LL}(B) \wedge \neg\texttt{LL}(C)))\rangle$
$\langle !(\texttt{TT}(C) \wedge \texttt{LL}(A) \wedge \texttt{LL}(B)) \vee (\texttt{TT}(C) \wedge \neg(\texttt{LL}(A) \wedge \texttt{LL}(B)))\rangle\top$

Based on Proposition 3 and the axiomatization of public announcement logic (cf. e.g., (Plaza, 2007)), we axiomatize $\texttt{PALT}^{\mathbf{T}}$ by the following Hilbert-style proof system **AT** where $\chi[\psi/\phi]$ denotes any formula obtained by replacing some occurrences of ϕ in χ by ψ.

Axiom Schemas	(for arbitrary $a, b \in \mathbf{G}$, $p \in \mathbf{P} \cup \mathbf{P_T}$)
TAUT	all the instances of tautologies
MU	$\bigwedge_{a\in\mathbf{G}}(\bigwedge_{\eta\in\mathbf{T}}(\eta(a) \leftrightarrow \bigwedge_{\eta'\neq\eta,\eta'\in\mathbf{T}} \neg\eta'(a)))$
DISTK	$K_a(\phi \to \psi) \to (K_a\phi \to K_a\psi)$
T	$K_a\phi \to \phi$
4	$K_a\phi \to K_aK_a\phi$
5	$\neg K_a\phi \to K_a\neg K_a\phi$
!ATOM	$[!_a\psi]p \leftrightarrow (\delta^a_\psi \to p)$
!NEG	$[!_a\psi]\neg\phi \leftrightarrow (\delta^a_\psi \to \neg[!_a\psi]\phi)$
!CON	$[!_a\psi](\phi \wedge \chi) \leftrightarrow ([!_a\psi]\phi \wedge [!_a\psi]\chi)$
!K	$[!_a\psi]K_b\phi \leftrightarrow (\delta^a_\psi \to K_b[!_a\psi]\phi)$
Rules	
GENK	$\dfrac{\phi}{K_a\phi}$
RE	$\dfrac{\phi \leftrightarrow \psi}{\chi[\psi/\phi] \leftrightarrow \chi}$
MP	$\dfrac{\phi, \phi \to \psi}{\psi}$

Theorem 1 **AT** *is sound and complete.*

Proof (Sketch) The soundness of MU is due to the fact that λ is a function, whence the basic type formulas of any agent are mutually exclusive and altogether exhaustive on each world of a model. The soundness of other axiom schemas and rules can be checked as for the standard axiomatization of PAL (cf. (Plaza, 2007)) based on Proposition 3. The completeness is proved by a reduction argument that makes use of the reduction axiom schemas (!ATOM, !NEG, !CON, !K), and the rule RE to eliminate $[!_a\psi]$ operators in an inside-out fashion (cf. (Wang, 2011a) for a detailed discussion). The only difficulty here is assigning 'announcement complexities' to $\texttt{PALT}^{\mathbf{T}}$ formulas in such a way that rewriting from the left-hand-side to the right-hand-side of !ATOM, !NEG, !CON, !K always reduces complexity. With a suitable complexity assignment, we can show that every $\texttt{PALT}^{\mathbf{T}}$ formula can be reduced to an equivalent $\texttt{EL}^{\mathbf{T}}$ formula by repeatedly applying the left-to-right rewriting rules specified by the reduction axiom schemas and the replacement of equals specified by the RE rule. It is not hard to

[4] We conjecture that $\texttt{PALT}^{\mathbf{T}}$ is at least exponentially more succinct than $\texttt{PAL}^{\mathbf{T}}$, but leave the proof for future work.

see that the system $\mathbf{AT}$ without !ATOM, !NEG, !CON, !K can completely axiomatize $\mathtt{EL}^{\mathbf{T}}$. Now, if $\models \phi$, then $\Vdash \phi'$ for some $\mathtt{EL}^{\mathbf{T}}$ formula ϕ' that can be obtained from ϕ by using the reduction axioms, and so $\phi \leftrightarrow \phi'$ can be derived in $\mathbf{AT}$. By the completeness of $\mathtt{EL}^{\mathbf{T}}$ we know that ϕ' can also be derived in $\mathbf{AT}$. Therefore ϕ can be derived in $\mathbf{AT}$. Hence $\mathbf{AT}$ is complete. □

The above proof shows that $\mathtt{PALT}^{\mathbf{T}}$ is equally expressive as $\mathtt{EL}^{\mathbf{T}}$. By Proposition 1, $\mathtt{PAL}^{\mathbf{T}}$ is equally expressive as $\mathtt{EL}^{\mathbf{T}}$. Therefore we have the following result:

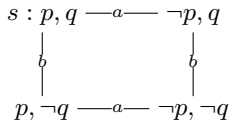
Proposition 4 *$\mathtt{EL}^{\mathbf{T}}$, $\mathtt{PAL}^{\mathbf{T}}$, and $\mathtt{PALT}^{\mathbf{T}}$ are equally expressive.*

In particular, $\mathtt{PALT}^{\mathbf{T}}$ formulas without knowledge operators or subjective (knowledge-based) types can be translated into propositional formulas based on $\mathbf{P} \cup \mathbf{P_T}$. This explains why solving puzzles like Example 3 normally only requires propositional reasoning. However, as we will show in the later part of the paper, knowledge-based subjective types make the story much more complicated and interesting, which will demonstrate the full power of our framework.

We end this subsection with a technical issue that has an interesting twist in the current context. In some axiomatizations of standard PAL, the following *composition axiom schema* is included instead of the inference rule RE (cf. e.g., (van Ditmarsch et al, 2007; Wang, 2011a)):

$$\mathtt{!COM} \quad [!\phi][!\psi]\chi \leftrightarrow [!(\phi \wedge [!\phi]\psi)]\chi$$

The idea is that one can always combine two announcements into one in PAL(and also in $\mathtt{PAL}^{\mathbf{T}}$). It is natural to ask whether some form of the composition axiom schema is valid in $\mathtt{PALT}^{\mathbf{T}}$. However, the answer is negative in general.[5] Suppose we only have one single subjective truth teller type: $\mathbf{T} = \{\mathtt{STT}\}$. Consider the following model, where a does not know if q and b does not know whether p:



Clearly $\mathfrak{M}, s \models \langle !_a p\rangle\langle !_b q\rangle(K_a(p \wedge q) \wedge K_b(p \wedge q))$. However, it is impossible to combine these two announcements into one announcement of the form of $\langle !_a \phi\rangle$ or $\langle !_b \phi\rangle$ after which both a and b know p and q. To see this, note that agents can only announce something that they know according to their type STT. Intuitively, you cannot let yourself know something new by just repeating things that you already know. Technically, a can only announce non-empty unions of equivalence classes w.r.t. $\sim_a$, which allows him only three different formulas (modulo logical equivalence): q, $\neg q$, or $\top$. None of these will let a know whether q.

On the other hand, for some special types $\mathbf{T}$, it is indeed possible to obtain a composition result.

[5] See (van Benthem and Minică, 2009) for a similar composition issue in dynamic-epistemic logics of questions and answers.

Proposition 5 *Given* $\mathbf{T} = \{TT, LL\}$*, the following is valid:*

$$[!_a\phi][!_b\psi]\chi \leftrightarrow [!_a\phi']\chi$$

where ϕ' *depends only on* ϕ*,* ψ*,* a*, and* b*.*

Proof Due to Proposition 3, $[!_a\phi][!_b\psi]\chi$ is equivalent to a $\mathtt{PAL}^{\mathbf{T}}$ formula of the shape $[!\phi^*][!\psi^*]t(\chi)$ for some $\mathtt{PAL}^{\mathbf{T}}$ formulas ϕ^* and ψ^*. Since $[!\phi^*][!\psi^*]t(\chi) \leftrightarrow [!(\phi^* \wedge [!\phi^*]\psi^*)]t(\chi)$ is valid in $\mathtt{PAL}^{\mathbf{T}}$ semantics, $[!_a\phi][!_b\psi]\chi$ is equivalent to the $\mathtt{PAL}^{\mathbf{T}}$ formula $[!(\phi^* \wedge [!\phi^*]\psi^*)]t(\chi)$. Now it is not hard to reduce $\phi^* \wedge [!\phi^*]\psi^*$ into an $\mathtt{EL}^{\mathbf{T}}$ formula θ using our translation f as in Proposition 1, and so $[!_a\phi][!_b\psi]\chi$ is equivalent to a $\mathtt{PAL}^{\mathbf{T}}$ formula $[!\theta]t(\chi)$. Now by Proposition 2, truthful announcement of θ can be mimicked by an announcement of a $\mathtt{PALT}^{\mathbf{T}}$ formula θ_a° by agent a. Hence it is easy to see that the $\mathtt{PAL}^{\mathbf{T}}$ formula $[!\theta]t(\chi)$ is equivalent to the $\mathtt{PALT}^{\mathbf{T}}$ formula $[!_a\theta_a^\circ]\chi$. Taking things together, $[!_a\phi][!_b\psi]\chi$ is equivalent to the $\mathtt{PALT}^{\mathbf{T}}$formula $[!_a\theta_a^\circ]\chi$. □

2.4 'I am a liar'

Careful readers may have found out that the language of $\mathtt{PALT}^{\mathbf{T}}$ allows us to express the following announcement: $!_a\mathtt{LL}(a)$ which may be roughly read as *'I am a liar.'*. It sounds like a *liar sentence*. However, a closer look should reveal that in our framework this is not a self-referential liar sentence such as *'This sentence is a lie.'*.

First note that $!_a\mathtt{LL}(a)$ is not even a well-formed formula in $\mathtt{PALT}^{\mathbf{T}}$. Therefore, it does not make sense to talk about its truth value. On the other hand, $!_a\mathtt{LL}(a)$ is viewed as an action in our framework and we may talk about its executability and update effects.

Now given $\mathbf{T} = \{\mathtt{TT}, \mathtt{LL}\}$, from Proposition 3, $!_a\mathtt{LL}(a)$ can be translated into a public announcement $!((\mathtt{TT}(a) \wedge \mathtt{LL}(a)) \vee (\mathtt{LL}(a) \wedge \neg\mathtt{LL}(a)))$ which amounts to the action of truthfully announcing $\bot$. It is impossible to truthfully announce $\bot$, so $!_a\mathtt{LL}(a)$ is not executable at all. According to the semantics, we can easily verify that $[!_a\mathtt{LL}(a)]\bot$ is valid, which is a formal way of saying $!_a\mathtt{LL}(a)$ is not executable. Since $!_a\mathtt{LL}(a)$ can never happen according to the types that govern the behaviours of agents, it has no non-trivial update effects.

On the other hand, if $\mathbf{T} = \{\mathtt{LL}, \mathtt{TT}, \mathtt{LT}\}$ then $[!_a\mathtt{LL}(a)]\bot$ is not valid any more, and instead, $[!_a\mathtt{LL}(a)]K_b\mathtt{LT}(a)$ becomes valid for any $a, b \in \mathbf{G}$. This is because only the bluffers can possibly execute $!_a\mathtt{LL}(a)$, by the definition of $\mathtt{LL}$, $\mathtt{TT}$, and $\mathtt{LT}$. This demonstrates that when bluffers are involved, successfully saying 'I am a liar' amounts to signalling that the speaker is a bluffer.

A final, related question is: Can a liar tell others that he is a liar in some way? It is rather easy when $\mathbf{T} = \{\mathtt{TT}, \mathtt{LL}\}$. The liar can just announce $\bot$. What about $\mathbf{T} = \{\mathtt{TT}, \mathtt{LL}, \mathtt{LT}\}$? Unfortunately, it is no easy task without signalling who are the bluffers first: whatever the truth teller and liar may say, the hearer just cannot rule out the possibility that the speaker is a bluffer.

3 Agent types in questions and answers

Question-answer situations are typical interactive scenarios in which agents exchange information with each other. In this section, we extend the language of $\mathtt{PALT}^{\mathbf{T}}$ to han-

dle questions and answers. Moreover, by formally defining *puzzles* and their *solutions* within our framework, we will apply our logic to HLPE-like puzzles.

3.1 A question-answer logic

First, we extend $\texttt{PALT}^{\mathbf{T}}$ with question modalities:

Definition 4 (Public question logic with types $\texttt{PQLT}^{\mathbf{T}}$) Given $\mathbf{T}$, $\mathbf{P}$ and $\mathbf{G}$ as before, the language $\texttt{PQLT}^{\mathbf{T}}$ extends $\texttt{PALT}^{\mathbf{T}}$ with question operators and arbitrary answer operators:

$$\phi \ ::= \ \top \mid p \mid \neg\phi \mid \phi \wedge \phi \mid K_a\phi \mid \eta(a) \ \mid [!_a\phi]\phi \mid [?_a\phi]\phi \mid [!_a]\phi$$

where $\eta \in \mathbf{T}$ and $a \in \mathbf{G}$.

Intuitively, $[?_a\psi]\phi$ expresses that 'After asking a whether ψ, ϕ holds', and $[!_a]\phi$ says that 'No matter what answer a gives (to the current question), afterwards ϕ holds'. Here we only focus on yes/no questions. Note that this language is expressive enough to express counterfactual questions. For example, $?_a([?_ap]\langle!_ap\rangle\top)$ expresses the question 'would you answer yes if you were asked whether p?'.

Definition 5 (Semantics for $\texttt{PQLT}^{\mathbf{T}}$) The semantics of $\texttt{PQLT}^{\mathbf{T}}$ formulas on a model $\mathfrak{M} = (S, \sim, V, \lambda)$ is defined as the following w.r.t. a *context* $\mu \in \{\#\} \cup \{\mathbf{G} \times Form(\texttt{PQLT}^{\mathbf{T}})\}$ where $Form(\texttt{PQLT}^{\mathbf{T}})$ is the set of $\texttt{PQLT}^{\mathbf{T}}$ formulas[6]. Intuitively, μ is used to record the current question: it can be of the form (a, ϕ) (a needs to answer whether ϕ) or simply $\#$ (there is currently no question to be answered).

$\mathfrak{M}, s \Vdash \phi$	$\Leftrightarrow \mathfrak{M}, s \Vdash_{\#} \phi$	
$\mathfrak{M}, s \Vdash_\mu \top$	$\Leftrightarrow$ always	
$\mathfrak{M}, s \Vdash_\mu p$	$\Leftrightarrow p \in V(s)$	
$\mathfrak{M}, s \Vdash_\mu \neg\phi$	$\Leftrightarrow \mathfrak{M}, s \nVdash_\mu \phi$	
$\mathfrak{M}, s \Vdash_\mu \phi \wedge \psi$	$\Leftrightarrow \mathfrak{M}, s \Vdash_\mu \phi$ and $\mathfrak{M}, s \Vdash_\mu \psi$	
$\mathfrak{M}, s \Vdash_\mu K_a\phi$	$\Leftrightarrow \forall t : s \sim_a t$ implies $\mathfrak{M}, t \Vdash_\mu \phi$	
$\mathfrak{M}, s \Vdash_\mu \eta(a)$	$\Leftrightarrow \lambda(s, a) = \eta(a)$	
$\mathfrak{M}, s \Vdash_\mu [?_a\psi]\phi$	$\Leftrightarrow \mathfrak{M}, s \Vdash_{(a,\psi)} \phi$	
$\mathfrak{M}, s \Vdash_\mu [!_a\psi]\phi$	$\Leftrightarrow \begin{array}{c}\mu = (a, \chi), \psi = \pm\chi \text{ and} \\ \mathfrak{M}, s \Vdash_{\#} \lambda(s, a)(\psi, a)\end{array}$	implies $\mathfrak{M}\vert^a_\psi, s \Vdash_{\#} \phi$
$\mathfrak{M}, s \Vdash_\mu [!_a]\phi$	$\Leftrightarrow$ for all $\psi : \mathfrak{M}, s \Vdash_\mu [!_a\psi]\phi$	

where $\psi = \pm\chi$ means $\psi = \chi$ or $\psi = \neg\chi$. $\mathfrak{M}\vert^a_\psi$ is defined like before as $(S', \{\sim'_a \mid a \in \mathbf{G}\}, V', \lambda')$ with:

- $S' = \{t \mid t \in S \text{ and } \mathfrak{M}, t \Vdash_{\#} \lambda(s, a)(\psi, a)\}$
- For each $a \in \mathbf{G}$, $t \in S'$: $\sim'_a = \sim_a \vert_{S' \times S'}$, $V'(t) = V(t)$, and $\lambda'(t) = \lambda(t)$.

We say $\mathfrak{M}\vert^a_\psi$ is defined if $\{t \mid t \in S \text{ and } \mathfrak{M}, t \Vdash_{\#} \lambda(s, a)(\psi, a)\}$ is not empty.

The ideas behind the above semantics can be summarized as follows:

- Initially no question is asked (the use of $\#$ in the first clause).

[6] See (Wang, 2011b) and (Wang, 2011a) for other applications of the context dependent semantics in DEL.

- When a question $?_a\psi$ is asked, the question ψ and its answerer a are recorded (see the use of (a,ψ) in the clause for $[?_a\psi]\phi$), replacing the previously unanswered one, if there is any.
- A proposition can be announced by a ($!_a\psi$) *only* if ψ is a proper answer to the current question for a (the clause for $[!_a\psi]\phi$). Thus no one can say anything before a question is raised.
- After an answer is given, the record is set to $\#$.
- Any question can be addressed to any one, and the arbitrary answer operator can be split into two answers, as demonstrated by the following two valid formulas:

$$[?_a\phi]\chi \leftrightarrow \langle ?_a\phi\rangle\chi \qquad\qquad [?_a\phi][!_a]\chi \leftrightarrow [?_a\phi]([!_a\phi]\chi \wedge [!_a\neg\phi]\chi)$$

Remark 3 Questions have been discussed in dynamic epistemic logic ((van Benthem and Minică, 2009; Minică, 2011)), where questions partition the set of possible worlds. Our treatment is simpler, due to our intended application in HLPE-like puzzles where a question is always answered before the next question is raised. Therefore we do not consider the effect of consecutive questions: a new question will simply replace the old one, thus there is at most just one question for exactly one of the agents. This limitation can be overcome by using more complicated records μ, which we leave for future work.

The language of $\texttt{PQLT}^{\mathbf{T}}$ extends $\texttt{PALT}^{\mathbf{T}}$. However, $\texttt{PQLT}^{\mathbf{T}}$ formulas can be translated into $\texttt{PALT}^{\mathbf{T}}$ by the following translation g:

$$\begin{array}{rcl}
g(\phi) & = & g_{\#}(\phi)\\
g_\mu(\top) & = & \top\\
g_\mu(p) & = & p\\
g_\mu(\eta(a)) & = & \eta(a)\\
g_\mu(\neg\phi) & = & \neg g_\mu(\phi)\\
g_\mu(\phi_1\wedge\phi_2) & = & g_\mu(\phi_1)\wedge g_\mu(\phi_2)\\
g_\mu([!_a\psi]\phi) & = & \begin{cases} [!_a g_{\#}(\psi)]g_{\#}(\phi) & \text{if } \mu=(a,\chi) \text{ and } \psi=\pm\chi\\ \top & \text{if otherwise}\end{cases}\\
g_\mu([!_a]\phi) & = & \begin{cases} g_\mu([!_a\chi]\phi\wedge[!_a\neg\chi]\phi) & \text{if } \mu=(a,\chi)\\ \top & \text{if otherwise}\end{cases}\\
g_\mu([?_a\psi]\phi) & = & g_{(a,\psi)}(\phi)
\end{array}$$

By this translation we show that $\texttt{PQLT}^{\mathbf{T}}$ is no more expressive than $\texttt{PALT}^{\mathbf{T}}$:

Proposition 6 *For any $\mathfrak{M}, s$ and any $\texttt{PQLT}^{\mathbf{T}}$ formula ϕ, the following holds: $\mathfrak{M}, s \Vdash \phi \iff \mathfrak{M}, s \vDash g(\phi)$*

Proof We can actually prove the following stronger claim by a straightforward induction on the structure of the formulas:

For any $\mathfrak{M}, s$, any $\texttt{PQLT}^{\mathbf{T}}$ formula ϕ, and any $\mu \in \{\#\} \cup \{\mathbf{G} \times Form(\texttt{PQLT}^{\mathbf{T}})\}$: $\mathfrak{M}, s \Vdash_\mu \phi \iff \mathfrak{M}, s \vDash g_\mu(\phi)$.

Note that although g translates $[!_a]\phi$ into a conjunction of two concrete formulas, we cannot eliminate the operator $[!_a]$ in $\texttt{PQLT}^{\mathbf{T}}$, since it depends on the previously asked question.

On the other hand, we can also translate $\texttt{PALT}^{\mathbf{T}}$ to $\texttt{PQLT}^{\mathbf{T}}$ by g':

$$\begin{array}{rcl} g'(\top) & = & \top \\ g'(p) & = & p \\ g'(\eta(a)) & = & \eta(a) \\ g'(\neg\phi) & = & \neg g'(\phi) \\ g'(\phi_1 \wedge \phi_2) & = & g'(\phi_1) \wedge g'(\phi_2) \\ g'([!_a\psi]\phi) & = & [?_a g'(\psi)][!_a g'(\psi)]g'(\phi) \end{array}$$

Again, by a straightforward induction, we can show:

Proposition 7 *For any $\mathfrak{M}, s$ and any $\texttt{PALT}^{\mathbf{T}}$ formula ϕ, the following holds: $\mathfrak{M}, s \Vdash g'(\phi) \iff \mathfrak{M}, s \vDash \phi$*

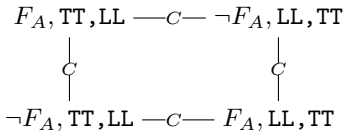
Therefore $\texttt{PQLT}^{\mathbf{T}}$ is equally expressive as $\texttt{PALT}^{\mathbf{T}}$, $\texttt{PAL}^{\mathbf{T}}$ and $\texttt{EL}^{\mathbf{T}}$, based on Proposition 4. Again, although $\texttt{PQLT}^{\mathbf{T}}$ does not increase the expressive power of the language, it eases the syntactic specification. Let us consider another (more popular) variation of the Knights and Knaves puzzle as follows.

Example 5 (Death or Freedom with questions) The setting is exactly the same as before in Example 4, but now C is allowed to ask a question to one of A and B. How should he ask his question in such a way that he will know the way to Freedom no matter what the answer is?

Again let $\mathbf{T} = \{\texttt{LL}, \texttt{TT}\}$. We can express the following questions:

- $?_A([?_B F_A]\langle !_B F_A\rangle\top)$: 'Will the other man tell me that your path leads to Freedom?'
- $?_A([?_A F_A]\langle !_A F_A\rangle\top)$: 'Will you say 'yes' if you are asked whether your path leads to Freedom?'

Recall the model $\mathfrak{M}$ of Example 4:



We can verify that

$$\mathfrak{M} \Vdash [?_A([?_B F_A]\langle !_B F_A\rangle\top)][!_A]K_C^W F_A \wedge [?_A([?_A F_A]\langle !_A F_A\rangle\top)][!_A]K_C^W F_A.$$

As an example, let us take the first conjunct and verify it at the world $(F_A, \texttt{TT}, \texttt{LL})$:

$$\begin{array}{rl} & \mathfrak{M}, (F_A, \texttt{TT}, \texttt{LL}) \Vdash [?_A([?_B F_A]\langle !_B F_A\rangle\top)][!_A]K_C^W F_A \\ \iff & \mathfrak{M}, (F_A, \texttt{TT}, \texttt{LL}) \Vdash_{\#} [?_A([?_B F_A]\langle !_B F_A\rangle\top)][!_A]K_C^W F_A \\ \iff & \mathfrak{M}, (F_A, \texttt{TT}, \texttt{LL}) \Vdash_{(A,[?_B F_A]\langle !_B F_A\rangle\top)} [!_A]K_C^W F_A \\ \iff & \mathfrak{M}, (F_A, \texttt{TT}, \texttt{LL}) \Vdash_{\#} [!_A([?_B F_A]\langle !_B F_A\rangle\top)]K_C^W F_A \text{ and} \\ & \mathfrak{M}, (F_A, \texttt{TT}, \texttt{LL}) \Vdash_{\#} [!_A(\neg[?_B F_A]\langle !_B F_A\rangle\top)]K_C^W F_A \end{array}$$

Now let us continue with the second conjunct of the final part (the first conjunct can be verified similarly):

$$\begin{aligned}
&\mathfrak{M}, (F_A, \mathtt{TT}, \mathtt{LL}) \Vdash_{\#} [!_A(\neg[?_B F_A]\langle !_B F_A\rangle\top)]K_C^W F_A\\
\iff &\mathfrak{M}, (F_A, \mathtt{TT}, \mathtt{LL}) \Vdash_{\#} \mathtt{TT}(\neg[?_B F_A]\langle !_B F_A\rangle\top, A)\\
&\text{implies } \mathfrak{M}|^A_{\neg[?_B F_A]\langle !_B F_A\rangle\top}, (F_A, \mathtt{TT}, \mathtt{LL}) \Vdash_{\#} K_C^W F_A\\
\iff &\mathfrak{M}, (F_A, \mathtt{TT}, \mathtt{LL}) \nVdash_{(B,F_A)} \langle !_B F_A\rangle\top\\
&\text{implies } \mathfrak{M}|^A_{\neg[?_B F_A]\langle !_B F_A\rangle\top}, (F_A, \mathtt{TT}, \mathtt{LL}) \Vdash_{\#} K_C^W F_A\\
\iff &\mathfrak{M}, (F_A, \mathtt{TT}, \mathtt{LL}) \nVdash_{\#} \mathtt{LL}(F_A, B)\\
&\text{implies } \mathfrak{M}|^A_{\neg[?_B F_A]\langle !_B F_A\rangle\top}, (F_A, \mathtt{TT}, \mathtt{LL}) \Vdash_{\#} K_C^W F_A
\end{aligned}$$

where $\mathfrak{M}|^A_{\neg[?_B F_A]\langle !_B F_A\rangle\top}$ keeps the worlds s in $\mathfrak{M}$ such that

$$\mathfrak{M}, s \Vdash \lambda(s, A)(\neg[?_B F_A]\langle !_B F_A\rangle\top, A).$$

Therefore the worlds satisfying one of the following conditions are kept:
$\mathfrak{M}, \_, \mathtt{TT}, \mathtt{LL} \Vdash \neg[?_B F_A]\langle !_B F_A\rangle\top$ or $\mathfrak{M}, \_, \mathtt{LL}, \mathtt{TT} \Vdash [?_B F_A]\langle !_B F_A\rangle\top$.
Equivalently: $\mathfrak{M}, \_, \mathtt{TT}, \mathtt{LL} \nVdash_{B,F_A} \langle !_B F_A\rangle\top$ or $\mathfrak{M}, \_, \mathtt{LL}, \mathtt{TT} \Vdash_{B,F_A} \langle !_B F_A\rangle\top$.
Then it is not hard to see that $\mathfrak{M}|^A_{\neg[?_B F_A]\langle !_B F_A\rangle\top}$ only keeps the worlds $(F_A, \mathtt{TT}, \mathtt{LL})$ and $(F_A, \mathtt{LL}, \mathtt{TT})$, thus $\mathfrak{M}|^A_{\neg[?_B F_A]\langle !_B F_A\rangle\top}, (F_A, \mathtt{TT}, \mathtt{LL}) \Vdash_{\#} K_C^W F_A$.

Alternatively, we can verify the above $\mathtt{PQLT}^{\mathbf{T}}$ formulas by using the translation g and the semantics for $\mathtt{PALT}^{\mathbf{T}}$, as we showed in Proposition 6:

$$\begin{aligned}
&g([?_A([?_B F_A]\langle !_B F_A\rangle\top)][!_A]K_C^W F_A)\\
=\ &g_{\#}([?_A([?_B F_A]\langle !_B F_A\rangle\top)][!_A]K_C^W F_A)\\
=\ &g_\mu([!_A]K_C^W F_A) \text{ where } \mu = (A, [?_B F_A]\langle !_B F_A\rangle\top)\\
=\ &g_\mu([!_A([?_B F_A]\langle !_B F_A\rangle\top)]K_C^W F_A) \wedge g_\mu([!_A(\neg[?_B F_A]\langle !_B F_A\rangle\top)]K_C^W F_A)\\
=\ &([!_A g_{\#}([?_B F_A]\langle !_B F_A\rangle\top)]g_{\#}(K_C^W F_A) \wedge ([!_A g_{\#}(\neg[?_B F_A]\langle !_B F_A\rangle\top)]g_{\#}(K_C^W F_A)\\
=\ &([!_A g_{(B,F_A)}(\langle !_B F_A\rangle\top)]K_C^W F_A \wedge ([!_A \neg g_{(B,F_A)}(\langle !_B F_A\rangle\top)]K_C^W F_A\\
=\ &[!_A\langle !_B F_A\rangle\top]K_C^W F_A \wedge [!_A\neg\langle !_B F_A\rangle\top]K_C^W F_A
\end{aligned}$$

The announcements in the last line may look familiar: actually, under the translation g, the solutions to Example 5 are translated into solutions to Example 4 without using questions.

3.2 Handling arbitrary utterances

To formally discuss the original HLPE, we still need one last technical preparation, since the gods in the story of HLPE answer questions in their own language. In this subsection, we also take this into consideration.

Definition 6 (Public question language with types and utterances) Let $\mathbf{U}$ be a finite set of *utterances*, the language $\mathtt{PQLT}^{\mathbf{T}}_{\mathbf{U}}$ replaces the announcements $!_a\phi$ in $\mathtt{PQLT}^{\mathbf{T}}$ by utterances $!_a u$:

$$\phi \ ::= \ \top \mid p \mid \neg\phi \mid \phi \wedge \phi \mid K_a\phi \mid \eta(a) \ \mid [!_a u]\phi \mid [?_a\phi]\phi \mid [!_a]\phi$$

where $\eta \in \mathbf{T}$, $u \in \mathbf{U}$ and $a \in \mathbf{G}$.

$[!_a u]\phi$ expresses that, if a says u, then ϕ is true.

A model $\mathfrak{M}$ for $\mathtt{PQLT}^{\mathbf{T}}_{\mathbf{U}}$ is a tuple: $(S, \{\sim_a |\ a \in \mathbf{G}\}, V, \lambda, I)$ where $I : S \times Form(\mathtt{PQLT}^{\mathbf{T}}_{\mathbf{U}}) \times \mathbf{U} \to Form(\mathtt{PQLT}^{\mathbf{T}}_{\mathbf{U}})$ is a function and $I(s, \phi, u)$ is the interpretation of an answer u on world s given the question ϕ. For example, if $u = \{yes, no\}$, we can define a function I corresponding to the usual interpretation of *yes* and *no* as answers to questions: $I(s, \phi, yes) = \phi$ and $I(s, \phi, no) = \neg\phi$ for each s and each ϕ.

The semantics of $\mathtt{PQLT}^{\mathbf{T}}_{\mathbf{U}}$ is mostly the same as that of $\mathtt{PQLT}^{\mathbf{T}}$, except for the formulas involving utterances, which depend on the interpretation function.

Definition 7 (Semantics for $\mathtt{PQLT}^{\mathbf{T}}_{\mathbf{U}}$) The semantics of $\mathtt{PQLT}^{\mathbf{T}}_{\mathbf{U}}$ formulas on the model $\mathfrak{M} = (S, \{\sim_a |\ a \in \mathbf{G}\}, V, \lambda, I)$ is defined exactly as the semantics of $\mathtt{PQLT}^{\mathbf{T}}$ w.r.t. $\mu \in \{\#\} \cup \mathbf{G} \times Form(\mathtt{PQLT}^{\mathbf{T}}_{\mathbf{U}})$, except for the following clauses:

$$\begin{array}{|lcll|}
\hline
 & & \mu = (a, \chi) \text{ and} & \\
\mathfrak{M}, s \Vdash_\mu [!_a u]\phi & \Leftrightarrow & I(s, \chi, u) = \pm\chi \text{ and} & \text{implies } \mathfrak{M}|^a_{\chi,u},\ s \Vdash_\# \phi \\
 & & \mathfrak{M}, s \Vdash_\# \lambda(s, a)(I(s, \chi, u), a) & \\
\mathfrak{M}, s \Vdash_\mu [!_a]\phi & \Leftrightarrow & \text{for all } u \in \mathbf{U} : \mathfrak{M}, s \Vdash_\mu [!_a u]\phi & \\
\hline
\end{array}$$

where $\mathfrak{M}|^a_{\chi,u}$ is defined as $(S', \{\sim'_a |\ a \in \mathbf{G}\}, V', \lambda', I')$ where:

- $S' = \{t \mid t \in S \text{ and } \mathfrak{M}, t \Vdash_\# \lambda(t, a)(I(t, \chi, u), a)\}$
- For each $a \in \mathbf{G}$, $t \in S', u \in \mathbf{U}, \phi \in \mathtt{PQLT}^{\mathbf{T}}_{\mathbf{U}}$: $\sim'_a = \sim_a |_{S' \times S'}$, $V'(t) = V(t)$, $\lambda'(t) = \lambda(t)$, and $I'(t, \phi, u) = I(t, \phi, u)$.

We say that $\mathfrak{M}|^a_{\chi,u}$ is *defined* if the set $\{t \mid t \in S \text{ and } \mathfrak{M}, t \Vdash_\# \lambda(t, a)(I(t, \chi, u), a)\}$ is not empty.

It is easy to see that:

$$\mathfrak{M}, s \Vdash_\mu \langle!_a\rangle\phi \Leftrightarrow \mathfrak{M}, s \Vdash_\mu \neg[!_a]\neg\phi \Leftrightarrow \text{ there exists a } u \in \mathbf{U} : \mathfrak{M}, s \Vdash_\mu \langle!_a u\rangle\phi$$

Remark 4 It is important that we use $\Vdash_\#$ in the third condition of the clause for $[!_a u]\phi$. Replacing $\#$ by μ will cause circularity in the semantics. For instance, $?_a\langle!_a u\rangle\top$ may then expresses the self-referential question '*Will you answer u (to this question)?*'.

3.3 Questioning strategy

In the previous sections, we talked about the notions of *puzzles* and *solutions* in an rather informal manner. In this subsection, we attempt to formalize them precisely in the framework of $\mathtt{PQLT}^{\mathbf{T}}_{\mathbf{U}}$.

Definition 8 (Questioning strategy) A *questioning strategy* π w.r.t. $\mathtt{PQLT}^{\mathbf{T}}_{\mathbf{U}}$ is a tuple (Q, F, r, δ, L) where

- Q is a non-empty finite set of *question states* and $r \in Q$ is the *initial state*,
- F is a non-empty finite set of *final states* such that $F \cap Q = \emptyset$,
- $\delta : Q \times \mathbf{U} \to Q \cup F$ is a transition function,
- $L : Q \to \mathbf{G} \times Form(\mathtt{PQLT}^{\mathbf{T}}_{\mathbf{U}})$ essentially assigns to each question state a question $?_a\phi$ expressible in $\mathtt{PQLT}^{\mathbf{T}}_{\mathbf{U}}$ (formally represented as a pair (a, ϕ)).

In this work, we only consider the questioning strategies that are *trees*[7].

For any questioning strategy $\pi = (Q, F, r, \delta, L)$ and any $q \in Q$, let $L^{\mathbf{G}}(q)$ and $L^{\Phi}(q)$ be the first and the second element of $L(q)$, respectively. Note that every q node has one and only one u successor for each u in $\mathbf{U}$. Two different question states may be assigned the same question (a, ϕ). Given a questioning strategy π, an *execution* of π is a path $r \xrightarrow{u_1} q_1 \cdots \xrightarrow{u_n} q_n$ in π such that $q_i \in Q$ for $i < n$ and $q_n \in F$. Let $P(\pi)$ be the collection of all the executions in π. The length of a strategy ($|\pi|$) is defined as the length of the longest execution of π (a natural number or ω).

For example, given $\mathbf{G} = \{A, B, C\}$, $\mathbf{T} = \{\mathtt{TT}, \mathtt{LL}, \mathtt{LT}\}$ and $\mathbf{U} = \{ja, da\}$, a simple questioning strategy π: '*asking them one by one if they are bluffers*' is illustrated as follows:

$$r :?_A\mathtt{LT}(A) \overset{ja}{\underset{da}{\rightrightarrows}} q_1 :?_B\mathtt{LT}(B) \overset{ja}{\underset{da}{\rightrightarrows}} q_2 :?_C\mathtt{LT}(C) \overset{ja}{\underset{da}{\rightrightarrows}} f$$

where $r :?_A\mathtt{LT}(A)$ means $L(r) = (A, \mathtt{LT}(A))$, similarly for other nodes.

Let $Seq(\pi)$ be all the potential question-answer sequences of π, namely,

$$Seq(\pi) = \{?_{a_1}\phi_1!_{a_1}u_1 \ldots ?_{a_n}\phi_n!_{a_n}u_n \mid q_0 \xrightarrow{u_1} q_1 \cdots \xrightarrow{u_n} q_{n+1} \in P(\pi), \\ \forall i : a_i = L^{\mathbf{G}}(q_i), \phi_i = L^{\Phi}(q_i)\}.$$

A *puzzle* of $\mathtt{PQLT}^{\mathbf{T}}_{\mathbf{U}}$ is a pair consisting of a $\mathtt{PQLT}^{\mathbf{T}}_{\mathbf{U}}$ model and a $\mathtt{PQLT}^{\mathbf{T}}_{\mathbf{U}}$ formula as the goal: $(\mathfrak{M}, \phi)$. Intuitively, a puzzle asks for a questioning strategy π such that ϕ is guaranteed after executing π. A questioning strategy π is a *solution* to a puzzle $(\mathfrak{M}, \phi)$ if for all $?_{a_1}\phi_1!_{a_1}u_1 \cdots ?_{a_n}\phi_n!_{a_n}u_n \in Seq(\pi)$:

$$\mathfrak{M} \vDash [?_{a_1}\phi_1](\langle!_{a_1}\rangle\top \wedge [!_{a_1}u_1][?_{a_2}\phi_2](\langle!_{a_2}\rangle\top \wedge [!_{a_2}u_2][?_{a_3}\phi_3](\ldots [?_{a_n}\phi_n] \\ (\langle!_{a_n}\rangle\top \wedge [!_{a_n}u_n]\phi)..)))$$

Intuitively it says that for each execution $?_{a_1}\phi_1!_{a_1}u_1 \cdots ?_{a_n}\phi_n!_{a_n}u_n \in Seq(\pi)$, if the kth question $?_{a_k}\phi_k$ is asked then it must be *answerable* by some $u \in \mathbf{U}$, and if the answer is indeed $!_{a_k}u_k$ then we can proceed to the next question $?_{a_{k+1}}\phi_{k+1}$ and so on; eventually if the last question $a_n\phi_n$ is answered then ϕ holds. The idea behind the answerability condition $\langle!_{a_k}\rangle\top$ is that we need to ask sensible questions that always have answers, otherwise $[!_a]\psi$ may hold trivially. For example, if an agent is a subjective truth teller, he may not be able to answer $?\phi$ if he does not know whether ϕ. If *no answer* is also regarded as an answer, then the utterance 'I don't know' should be included in $\mathbf{U}$ as well. See Remark 5 at the end of the next section for further discussion.

The above formal requirement looks complicated, but it can be simplified under certain conditions. If we are sure that every question in π is *always answerable* w.r.t. any world in $\mathfrak{M}$, then π is a solution to $(\mathfrak{M}, \phi)$ iff every executable path of π leads to ϕ: for any $?_{a_1}\phi_1!_{a_1}u_1 \cdots ?_{a_n}\phi_n!_{a_n}u_n \in Seq(\pi)$:

$$\mathfrak{M} \Vdash [?_{a_0}\phi_0][!_{a_0}u_0] \cdots [?_{a_n}\phi_n][!_{a_n}u_n]\phi.$$

In the discussion of HLPE, we will only consider questions that are always answerable by *ja* or *da*, so the above simplified condition suffices.

[7] I.e., $(Q \cup F, \delta)$ is an acyclic graph where each node except r has one and only one u-predecessor for each $u \in \mathbf{U}$, and r can reach all other nodes.

4 Formalizing the hardest logic puzzle ever

In this section, we review one classic solution to the original HLPE in our formal framework.

Recall the story of HLPE mentioned at the beginning of this paper. Boolos provides the following guidelines in (Boolos, 1996):

B1 Each god may get asked more than one question;
B2 Later questions may depend on previous ones and their answers;
B3 Whether Random speaks truly or not depends on the flip of a coin in his mind: if the coin comes down heads, he speaks truly; if tails, falsely.
B4 Random will always answer '*da*' or '*ja*'.

Rabern and Rabern (2008) first noticed that B3 may trivialize the puzzle, and therefore proposed an alternative assumption B3' that we will follow in this work:

B3' Whether Random answers '*ja*' or '*da*' depends on the coin flip in his mind: if it comes down heads, he answers '*ja*'; if tails, he answers '*da*'.

Note that B1, B2 are already assumed implicitly in our formal definition of solutions to a puzzle, while B3' and B4 actually say that Random is indeed of the type LT that we have defined given any interpretation of *da* and *ja*.

However, to formalize the puzzle precisely, there is still a lot more left to be clarified about the knowledge of agents. Let us list the implicit (epistemic) assumptions as follows:

E0 A, B, and C are of the types in $\mathbf{T} = \{\mathtt{TT}, \mathtt{LL}, \mathtt{LT}\}$ and this is common knowledge (to all of the agents including the *questioner* D).
E1 A, B, and C are of different types and this is common knowledge.
E2 A, B, and C know each other's types and this is common knowledge.
E3 A, B, and C know the meaning of '*da*' and '*ja*' and this is common knowledge.
E4 D does not know the types of A, B, C and this is common knowledge.
E5 D does not know the exact meanings of 'da' and 'ja' but he knows that one means 'yes' and the other means 'no', and this is common knowledge.

Moreover, we assume the following:

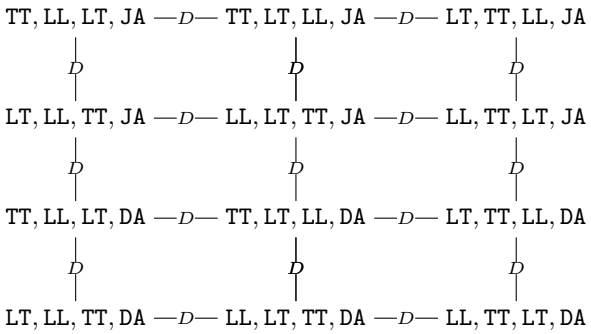
Q1 All questions are asked and answered publicly.
Q2 D does not mention himself in the questions.
LS We only consider solutions of length less than 4.

Q1 and Q2 may look unnecessary but they do play a role in the analysis of HLPE within our framework: we only consider public questions and answers in our technical preparations, and Q2 will simplify our discussion later on in the paper.

4.1 Formalizing HLPE

In the sequel, we fix $\mathbf{U} = \{ja, da\}$, $\mathbf{T} = \{\mathtt{TT}, \mathtt{LL}, \mathtt{LT}\}$ and $\mathbf{G} = \{A, B, C, D\}$. According to the assumptions E0-E5 we can build the following model $\mathfrak{M}_0$ (as usual we

omit the *reflexive transitive* arrows and also the type of D since it is irrelevant):



where JA at a world s denotes the interpretation that *ja* means *yes*, and *da* means *no* at world s, i.e., $I(s,\phi,ja)=\phi$ and $I(s,\phi,da)=\neg\phi$ for any $\mathtt{PQLT}^{\mathbf{T}}_{\mathbf{U}}$ formula ϕ. Similarly, DA at world s denotes that $I(s,\phi,ja)=\neg\phi$ and $I(s,\phi,da)=\phi$ for any ϕ.

Note that although we do not include a common knowledge operator $C_{\mathbf{G}}$ in our logical language, we can define common knowledge of ϕ ($C_{\mathbf{G}}\phi$) as a conjunction of all formulas of the form $K_{a_1}\dots K_{a_n}\phi$ where $a_i\in\mathbf{G}$. We may write $\mathfrak{M}\Vdash C_{\mathbf{G}}\phi$ if all the formulas in the collection are true at all worlds in $\mathfrak{M}$. With the help of $C_{\mathbf{G}}$, we can verify that $\mathfrak{M}_0$ indeed validates the formulas corresponding to the assumptions E0 to E5. Take E5 as a non-trivial example, and let

$$\phi_x^{\mathtt{JA}}=\mathtt{TT}(x)\to([?_x\mathtt{TT}(x)]\langle!_x ja\rangle\top\wedge[?_x\neg\mathtt{TT}(x)]\langle!_x da\rangle\top)$$
$$\phi_x^{\mathtt{DA}}=\mathtt{TT}(x)\to([?_x\mathtt{TT}(x)]\langle!_x da\rangle\top\wedge[?_x\neg\mathtt{TT}(x)]\langle!_x ja\rangle\top)$$

Intuitively, $\bigwedge_{x\in\mathbf{G}}\phi_x^{\mathtt{JA}}$ is a clumsy way of saying that *ja* means yes and *da* means no (we cannot express this directly in our language). Similarly for $\bigwedge_{x\in\mathbf{G}}\phi_x^{\mathtt{DA}}$. We can formalize E5 by the following formula (more precisely, an infinite set of formulas):

$$\phi_{E5}=C_{\mathbf{G}}(K_D(\bigwedge_{x\in\mathbf{G}}\phi_x^{\mathtt{JA}}\vee\bigwedge_{x\in\mathbf{G}}\phi_x^{\mathtt{DA}})\wedge\neg(K_D\bigwedge_{x\in\mathbf{G}}\phi_x^{\mathtt{JA}}\vee K_D\bigwedge_{x\in\mathbf{G}}\phi_x^{\mathtt{DA}}))$$

We can then verify that $\mathfrak{M}_0\vDash\phi_{E5}$.

All the other assumptions E0-E4 can also be formalized and checked on $\mathfrak{M}_0$, which we leave as an exercise for the interested reader.

This shows that the model $\mathfrak{M}_0$ complies with our assumptions. Now let $\chi(a)$ be the formula $K_D\mathtt{LL}(a)\vee K_D\mathtt{TT}(a)\vee K_D\mathtt{LT}(a)$, and let χ be $\chi(A)\wedge\chi(B)\wedge\chi(C)$. The HLPE puzzle can be formalized as $(\mathfrak{M}_0,\chi)$.

4.2 Verification of a classic solution

Before verifying an existing solution, let us formally prove the following crucial result from (Rabern and Rabern, 2008):

> *Let E^* be the function that takes a question q to the question 'If you were asked whether q would you say "ja?"'. When either True or False are asked $E^*(q)$, a response of 'ja' indicates that the correct answer to q is affirmative and a response of 'da' indicates that the correct answer to q is negative.*

Lemma 1 (Embedded question lemma[8]) *For any modality-free formula ϕ of $\mathtt{PQLT}^{\mathbf{T}}_{\mathbf{U}}$, any $a \in \{A, B, C\}$ and any submodel $\mathfrak{N}$ of $\mathfrak{M}_0$:*

$$\mathfrak{N} \Vdash [?_a[?_a\phi]\langle !_a ja\rangle\top](([!_a ja]K_D(\neg LT(a) \to \phi) \wedge [!_a da]K_D(\neg LT(a) \to \neg\phi))$$

where $[?_a\phi]\langle !_a ja\rangle\top$ expresses 'If I asked you ϕ would you say "ja?"'.

Proof Without loss of generality, let $a = A$. Let $\psi = [?_A\phi]\langle !_A ja\rangle\top$, $\phi_s^{\mathtt{JA}} = \lambda(s, A)(\psi, A)$ and $\phi_s^{\mathtt{DA}} = \lambda(s, A)(\neg\psi, A)$ for any s in $\mathfrak{N}$. Then we have the following chain of equivalences:

$$\begin{aligned}
&\mathfrak{N}, s \Vdash [?_A[?_A\phi]\langle !_A ja\rangle\top][!_A ja]K_D(\neg \mathtt{LT}(a) \to \phi)\\
\iff &\mathfrak{N}, s \Vdash_{(A,[?_A\phi]\langle !_A ja\rangle\top)} [!_A ja]K_D(\neg \mathtt{LT}(A) \to \phi)\\
\iff &\begin{cases} \mathfrak{N}, s \Vdash_{\#} \phi_s^{\mathtt{JA}} \text{ implies } \mathfrak{N}|^A_{(\psi,ja)}, s \Vdash_{\#} K_D(\neg \mathtt{LT}(A) \to \phi) & \text{if } s = \mathtt{___JA}\\ \mathfrak{N}, s \Vdash_{\#} \phi_s^{\mathtt{DA}} \text{ implies } \mathfrak{N}|^A_{(\psi,ja)}, s \Vdash_{\#} K_D(\neg \mathtt{LT}(A) \to \phi) & \text{if } s = \mathtt{___DA} \end{cases} (\star).
\end{aligned}$$

Now,

$$\begin{aligned}
&\mathfrak{N}, \mathtt{___JA} \Vdash_{\#} \phi_s^{\mathtt{JA}}\\
\iff &\mathfrak{N}, s \Vdash_{\#} \lambda(s, A)([?_A\phi]\langle !_A ja\rangle\top, A) \text{ if } s = \mathtt{___JA}\\
\iff &\begin{cases} \mathfrak{N}, s \Vdash_{\#} [?_A\phi]\langle !_A ja\rangle\top & \text{if } s = \mathtt{TT__JA}\\ \mathfrak{N}, s \Vdash_{\#} \neg[?_A\phi]\langle !_A ja\rangle\top & \text{if } s = \mathtt{LL__JA}\\ \mathfrak{M}, s \Vdash_{\#} \top & \text{if } s = \mathtt{LT__JA} \end{cases}\\
\iff &\begin{cases} \mathfrak{N}, s \Vdash_{\#} \phi & \text{if } s = \mathtt{TT__JA}\\ \mathfrak{N}, s \nVdash_{\#} \neg\phi & \text{if } s = \mathtt{LL__JA}\\ \mathfrak{N}, s \Vdash_{\#} \top & \text{if } s = \mathtt{LT__JA} \end{cases}\\
\iff &\begin{cases} \mathfrak{N}, s \Vdash_{\#} \phi & \text{if } s \neq \mathtt{LT__JA}\\ \mathfrak{N}, s \Vdash_{\#} \top & \text{if } s = \mathtt{LT__JA} \end{cases}
\end{aligned}$$

Similarly,

$$\begin{aligned}
&\mathfrak{N}, \mathtt{___DA} \Vdash_{\#} \phi_s^{\mathtt{DA}}\\
\iff &\begin{cases} \mathfrak{N}, s \Vdash_{\#} \neg[?_A\phi]\langle !_A ja\rangle\top & \text{if } s = \mathtt{TT__DA}\\ \mathfrak{N}, s \Vdash_{\#} \neg\neg[?_A\phi]\langle !_A ja\rangle\top & \text{if } s = \mathtt{LL__DA}\\ \mathfrak{N}, s \Vdash_{\#} \top & \text{if } s = \mathtt{LT__DA} \end{cases}\\
\iff &\begin{cases} \mathfrak{N}, s \nVdash_{\#} \neg\phi & \text{if } s = \mathtt{TT__DA}\\ \mathfrak{N}, s \Vdash_{\#} \phi & \text{if } s = \mathtt{LL__DA}\\ \mathfrak{N}, s \Vdash_{\#} \top & \text{if } s = \mathtt{LT__DA} \end{cases}\\
\iff &\begin{cases} \mathfrak{N}, s \Vdash_{\#} \phi & \text{if } s \neq \mathtt{LT__DA}\\ \mathfrak{N}, s \Vdash_{\#} \top & \text{if } s = \mathtt{LT__DA} \end{cases}
\end{aligned}$$

According to the semantics, $\mathfrak{N}|^A_{\psi,ja}$ retains the worlds t where:

$$\begin{cases} \mathfrak{N}, t \Vdash_{\#} \phi_t^{\mathtt{JA}} & \text{if } t = \mathtt{\_\_\_JA}\\ \mathfrak{N}, t \Vdash_{\#} \phi_t^{\mathtt{DA}} & \text{if } t = \mathtt{\_\_\_DA} \end{cases}$$

[8] We adopt the name of the lemma from (Rabern and Rabern, 2008).

Based on the above observations, $\mathfrak{N}|^A_{\psi,ja}$ retains the worlds satisfying $\mathtt{LT}(A) \vee \phi$, independent from the interpretation of *ja* and *da*. Now since ϕ is modality-free, all the worlds in $\mathfrak{N}|^A_{\psi,ja}$ satisfy $\mathtt{LT}(A) \vee \phi$. Therefore $(\star)$ is indeed true, and hence $\mathfrak{N}, s \Vdash [?_A[?_A\phi]\langle !_A ja\rangle\top][!_A ja]K_D(\neg\mathtt{LT}(A) \to \phi)$ for an arbitrary s in $\mathfrak{N}$. Similarly we can show that

$$\mathfrak{N} \Vdash [?_A[?_A\phi]\langle !_A ja\rangle\top][!_A da]K_D(\neg\mathtt{LT}(A) \to \neg\phi).$$

Since the selection of A is arbitrary, the proof can be completed easily. □

Let $\phi = \mathtt{LT}(A)$. Based on the above lemma, we have:

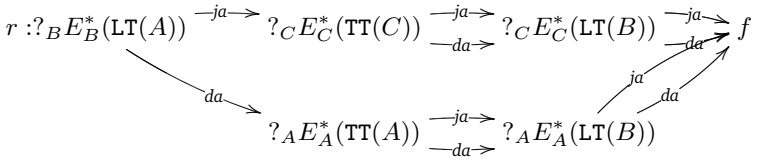
$$\mathfrak{M}_0 \Vdash [?_B[?_B\mathtt{LT}(A)]\langle !_B ja\rangle\top]([!_B ja]K_D(\neg\mathtt{LT}(B) \to \mathtt{LT}(A)) \wedge [!_B da]K_D(\neg\mathtt{LT}(B) \to \neg\mathtt{LT}(A)))$$

Note that $\neg\mathtt{LT}(B) \to \mathtt{LT}(A)$ is equivalent to $\mathtt{LT}(B) \vee \mathtt{LT}(A)$, and $\neg\mathtt{LT}(B) \to \neg\mathtt{LT}(A)$ is equivalent to $\mathtt{LT}(B) \vee \neg\mathtt{LT}(A)$. Since it is commonly known that there is only one bluffer, the above result implies the following:

$$\mathfrak{M}_0 \Vdash [?_B[?_B\mathtt{LT}(A)]\langle !_B ja\rangle\top]([!_B ja]K_D(\neg\mathtt{LT}(C)) \wedge [!_B da]K_D(\neg\mathtt{LT}(A)))$$

In words, this says that D will know that one of the agents is *not* a bluffer.

Based on this result, Rabern and Rabern (2008) proposed a three-step solution as follows (following (Rabern and Rabern, 2008) we use $E^*_a(\phi)$ as the short hand for formula $[?_a\phi]\langle !_a ja\rangle\top$):



In words, D first asks B whether A is a bluffer. Then depending on the answer, either A or C must be a non-bluffer. Thus D can then ask the non-bluffer about his own type and others' types.

Call the above questioning strategy π. We can verify π formally. Note that all the questions in π can be answered by at least one of *ja* and *da*, thus we only need to check that for all $?_{a_1}\phi_1!_{a_1}u_1 \cdots ?_{a_n}\phi_n!_{a_n}u_n \in Seq(\pi)$:

$$\mathfrak{M}_0 \vDash [?_{a_0}\phi_0][!_{a_0}u_0] \cdots [?_{a_n}\phi_n][!_{a_n}u_n]\chi.$$

Based on Lemma 1, the verification is immediate, and thus D will know the types of all the three agents.

5 New puzzles with epistemic twists

In the previous sections, we developed epistemic frameworks to handle various puzzles about agent types in question-answer scenarios such as the original HLPE. However, the power of our frameworks has not yet been fully demonstrated, since most of the previous examples can be treated as puzzles of Boolean algebra in the informal discussion style of the literature. This phenomenon has a technical explanation as we mentioned before: as long as we talk about objective types, the knowledge of agents is not really relevant and apparently complicated formulas can be translated back to Boolean formulas or simple epistemic formulas with no higher-order knowledge. Thus, existing puzzles are just too *easy* to require the full power of our $\mathtt{PQLT}^{\mathbf{T}}_{\mathbf{U}}$ framework. In this section, let us go a little bit further and consider some significantly harder puzzles where deeper epistemic reasoning is required.

One important underlying assumption in the original puzzle and its existing variations is that A, B, and C are *gods*. Intuitively, being gods, A, B, and C should know everything. Therefore their knowledge does not play a role in reasoning about their types. However, what if they are not gods but human beings? Being ordinary people, A, B, and C may not know everything and they will then behave according to their own knowledge.

In such a scenario, agents may not know each other's types, and they should have subjective, instead of objective types. Correspondingly, we should replace $\mathbf{T}$ in the assumption E0 by $\mathbf{T}' = \{\mathtt{STT}, \mathtt{SLL}, \mathtt{LT}\}$.[9] Since we do not require the agents to know each others' types, E2 should be abandoned. What would be an alternative to E2? Actually there are many possible assumptions. We just list a few examples:

- It is commonly known (to A, B, C, and D) that agents A, B and C only know their own types.
- It is commonly known that A knows everyone's type, but B and C only know their own types.
- It is commonly known that a bluffer knows everyone's type, but truth tellers and liars only know their own types.
- A knows everyone's type, but B and C only know their own types and doubt whether A indeed knows their types. D is not sure whether any of the three know all the types of each other.

To see that such epistemic assumptions can really make a difference, let us look at the following simple example $\mathfrak{N}$:

$$\mathtt{STT}, \mathtt{SLL}, \mathtt{LT}, \mathtt{JA} \; -_{A,D}- \; \mathtt{STT}, \mathtt{LT}, \mathtt{SLL}, \mathtt{JA}$$

From the model we can read off that it is commonly known that A does not know the types of B or C, but both B and C know the type of A. Moreover, it is commonly known that *ja* means yes and *da* means no. Now, can D determine A, B, and C's type by asking questions?

Surprisingly, the answer is negative. To prove it formally, we need Proposition 9 which will be proved later on. The intuition is this: first of all, asking A does not bring any new information since D knows everything that A knows. However, whatever D asks B or C, there is always a possibility that the answerer is a bluffer and

[9] A subjective bluffer is the same as an objective one.

thus at least one of the answers does not give any useful information. For example, suppose D asks B 'Are you a liar?' If the answer is *ja*, we know B must be LT, since a (subjective) liar cannot answer *ja*. However, if the answer is *da*, we cannot learn anything since both the liar and the bluffer can answer *da*. Note that in case A does not have any uncertainties between the two worlds, then D can simply ask A about the types of B and C.

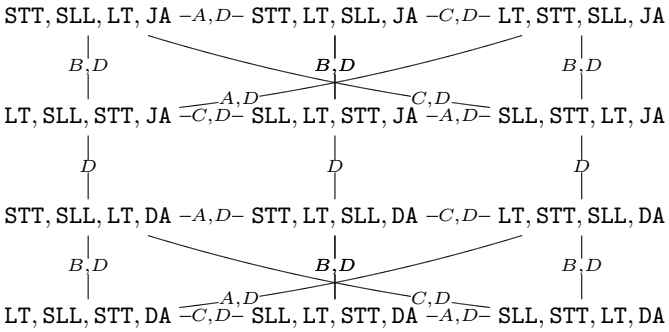
Now we are ready to consider a particular variation of the HLPE:

Example 6 (HLPE with ignorance) A (subjective) liar, a (subjective) truth teller and a bluffer are living on an island. They know their own types but do not know others' types. Moreover, it is commonly known that they are of different types. They understand English but can only answer questions in their own language, in which the words for *yes* and *no* are *da* and *ja*, in some order. Now the question is: can you determine their types by asking questions such that they are always able to answer *ja* or *da*.

Let us first list the new assumptions:

E0' A, B, and C are of types in $\mathbf{T} = \{\mathtt{STT}, \mathtt{SLL}, \mathtt{LT}\}$ and this is common knowledge (to all of the agents including the questioner D).
E1 A, B, and C are of different types and this is common knowledge.
E2' A, B, and C know their own types but do not know others' types, and this is also common knowledge.
E3 - E5, Q1, and Q2 are as before, but we do not constrain ourselves to 3-step solutions, thus giving up constraint LS.

Based on the above assumption, we can build the following model $\mathfrak{M}_1$:



It is not hard to check that E0', E3-E5 hold on $\mathfrak{M}_1$. For E2', note that for any agent $a \in \{A, B, C\}$, at each world s, agent a cannot distinguish s from another world t where his own type and the interpretation function are the same as in s. For example, agent B cannot distinguish STT, SLL, LT, JA from LT, SLL, STT, JA.

Let $\theta(a)$ be the formula $K_D\mathtt{SLL}(a) \vee K_D\mathtt{STT}(a) \vee K_D\mathtt{LT}(a)$ and $\theta = \theta(A) \wedge \theta(B) \wedge \theta(C)$. The puzzle is then formalized as $(\mathfrak{M}_1, \theta)$.

First note that Lemma 1 *does not* hold any more if we consider the submodels of $\mathfrak{M}_1$ instead of the submodels of $\mathfrak{M}_0$. For example, we have:

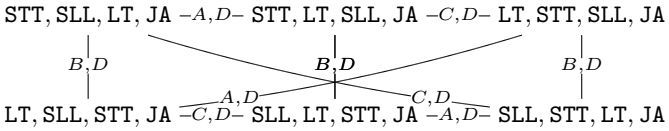
$$\mathfrak{M}_1, (\mathtt{STT}, \mathtt{SLL}, \mathtt{LT}, \mathtt{JA}) \not\Vdash [?_A[?_A\mathtt{LT}(B)]\langle !_A ja\rangle\top][!_A da](K_D(\neg\mathtt{LT}(A) \to \neg\mathtt{LT}(B)))$$

To see this, observe that $\mathfrak{M}_1|^{A}_{[?_A\mathtt{LT}(B)]\langle !_A ja\rangle\top, da}$ keeps the world $(\mathtt{STT}, \mathtt{LT}, \mathtt{SLL}, \mathtt{JA})$ where $\neg\mathtt{LT}(A) \to \neg\mathtt{LT}(B)$ does not hold:

$$
\begin{aligned}
& \mathfrak{M}_1, (\mathtt{STT}, \mathtt{LT}, \mathtt{SLL}, \mathtt{JA}) \Vdash \mathtt{STT}(\neg[?_A\mathtt{LT}(B)]\langle !_A ja\rangle\top, A) \\
\Longleftrightarrow\ & \mathfrak{M}_1, (\mathtt{STT}, \mathtt{LT}, \mathtt{SLL}, \mathtt{JA}) \Vdash_{\#} K_A\neg[?_A\mathtt{LT}(B)]\langle !_A ja\rangle\top \\
\Longleftrightarrow\ & \mathfrak{M}_1, (\mathtt{STT}, \mathtt{LT}, \mathtt{SLL}, \mathtt{JA}) \nVdash_{\#} [?_A\mathtt{LT}(B)]\langle !_A ja\rangle\top \\
& \quad \text{and } \mathfrak{M}_1, (\mathtt{STT}, \mathtt{SLL}, \mathtt{LT}, \mathtt{JA}) \nVdash_{\#} [?_A\mathtt{LT}(B)]\langle !_A ja\rangle\top \\
\Longleftrightarrow\ & \mathfrak{M}_1, (\mathtt{STT}, \mathtt{LT}, \mathtt{SLL}, \mathtt{JA}) \nVdash_{(A,\mathtt{LT}(B))} \langle !_A ja\rangle\top \\
& \quad \text{and } \mathfrak{M}_1, (\mathtt{STT}, \mathtt{SLL}, \mathtt{LT}, \mathtt{JA}) \nVdash_{(A,\mathtt{LT}(B))} \langle !_A ja\rangle\top \\
\Longleftrightarrow\ & \mathfrak{M}_1, (\mathtt{STT}, \mathtt{LT}, \mathtt{SLL}, \mathtt{JA}) \nVdash K_A\mathtt{LT}(B) \\
& \quad \text{and } \mathfrak{M}_1, (\mathtt{STT}, \mathtt{SLL}, \mathtt{LT}, \mathtt{JA}) \nVdash K_A\mathtt{LT}(B) \\
\Longleftarrow\ & \mathfrak{M}_1, (\mathtt{STT}, \mathtt{SLL}, \mathtt{LT}, \mathtt{JA}) \nVdash \mathtt{LT}(B)
\end{aligned}
$$

The essential problem is that when a subjective truth teller answers 'no' to a question 'will you be able to answer "yes" to a question $?\psi$', it does not mean that he will answer 'no' when he is actually asked whether ψ, because he might be not able to answer anything according to his type. Note that the question 'will you answer "yes" to a question $?\psi$' is always answerable, but the question $?\psi$ might not be answerable. The classic solution to the original HLPE involves asking questions about other's types. However, with the subjective liar and truth teller, such questions might not be answerable any more, since the agents may be ignorant about others' types.

Working toward solving the puzzle, we need a few new insights. Let $\mathfrak{M}_1'$ be the model just like $\mathfrak{M}_1$ but without D links between the JA zone and DA zone (that is, D knows the meanings of *ja* and *da*). Let $\mathfrak{M}_2$ be the upper part of $\mathfrak{M}_1$, being the following model:



Clearly, in the above model D also knows the exact meanings of *da* and *ja* and this is common knowledge.

Before we prove the following proposition, let us be more precise about answerable questions. We say that a question $?_a\phi$ is *answerable* on a model $\mathfrak{M}$ if $\mathfrak{M} \Vdash [?_a\phi]\langle !_a\rangle\top$. Thus, for any world in $\mathfrak{M}$, a has at least one possible answer to the question $?_a\phi$. It is not hard to see that if $?_a\phi$ is answerable on a submodel $\mathfrak{N}$ of $\mathfrak{M}_1$, then $\mathfrak{N} \Vdash \neg\mathtt{LT}(a) \to (K_a\phi \vee K_a\neg\phi)$.

Proposition 8 *There is a solution to* $(\mathfrak{M}_1, \theta)$ *iff there is a solution to* $(\mathfrak{M}_2, \theta)$.

Proof Proofs for this proposition and other results presented in this section are provided in the Appendix.

This proposition says that we can actually ignore uncertainties about *ja* and *da* when searching for solutions to $(\mathfrak{M}_1, \theta)$.

We say that a question $?_a\phi$ is *effective* on a model $\mathfrak{M}$ if for any $u \in \{ja, da\}$: $\mathfrak{M}|^a_{\phi,u}$ is defined (i.e., the domain is not empty), and $\mathfrak{M}|^a_{\phi,u} \neq \mathfrak{M}$: that is, answers to the question will always update the model by deleting some worlds. Now we make one crucial observation before proving our main impossibility result.

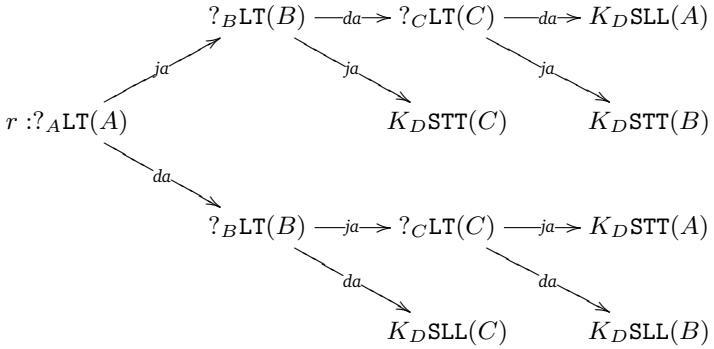
Proposition 9 *For any submodel $\mathfrak{N}$ of $\mathfrak{M}_2$ and any $a \in \{A, B, C\}$: $\mathfrak{N} \Vdash \neg SLL(a)$ or $\mathfrak{N} \Vdash \neg STT(a)$ implies that there is no effective question for a in model $\mathfrak{N}$.*

Based on Proposition 9, we have the following theorem.

Theorem 2 *There is no solution to $(\mathfrak{M}_2, \theta)$, and therefore, there is no solution to $(\mathfrak{M}_1, \theta)$.*

The proof of Theorem 2 gives a further interesting result: Although we cannot guarantee that D knows all the types of the agents, we can guarantee that D always knows the type of one of the non-bluffers (but he cannot make sure which one)!

Now let $\theta'(a)$ be $K_D\mathtt{SLL}(a) \vee K_D\mathtt{STT}(a)$ and let θ' be $\theta'(A) \vee \theta'(B) \vee \theta'(C)$. Although there is no solution to the original puzzle, we do have solutions to the puzzle $(\mathfrak{M}_2, \theta')$. A simple solution is to ask each of $A, B,$ and C 'are you a bluffer?' The questioning strategy (and the outcomes at final states) can be illustrated as follows:



Note that D cannot guarantee where he ends up in the questioning strategy tree, since the answer from a bluffer is essentially non-deterministic. Moreover, repeatedly using the strategy after D reaches one of the final states will not work, since we assume (Q1) that all the questions are asked and answered publicly. Thus when A and B get to know more, one cannot eliminate their knowledge[10]. We can also turn the above solution into a solution for $(\mathfrak{M}_1, \theta')$ by replacing each $?_a\psi$ in the above solution with $?_a([?_a((\mathtt{STT}(a) \to \psi) \wedge (\mathtt{SLL}(a) \to \neg\psi))]\langle !_a ja\rangle\top)$ as used in the proof of Proposition 8.

Remark 5 Note that in the above discussions, we only consider *ja* and *da* as well-formed answers and consider solutions with answerable questions only. In more realistic cases, agents should be able to answer 'I don't know' (or keep silent as in (Uzquiano, 2010)). However, the definition of types will be much more complicated, and there may be different options in redefining the liar and the bluffer. E.g.,

[10] We conjecture that even when D can ask questions privately, the puzzle $(\mathfrak{M}_1, \theta)$ still does not have any solution.

can a liar truthfully answer 'I don't know' or just say a random 'yes' or 'no' instead? Moreover, can a bluffer also announce 'I don't know' randomly? Given some acceptable new definitions of types involving 'I don't know', is it possible for D to know the types of A, B, and C? We suspect that the answer is still negative, but leave this for future exploration.

6 Conclusion and discussion

In this paper, we first proposed a simple type language to define agent types in terms of preconditions of announcements. Based on a finite set of types $\mathbf{T}$ defined in the type language, we introduced the following five logical languages:

- $\mathtt{EL}^{\mathbf{T}}$ Epistemic language (with type formulas),
- $\mathtt{PAL}^{\mathbf{T}} = \mathtt{EL}^{\mathbf{T}} + [!\phi]$ Public announcement language (with type formulas),
- $\mathtt{PALT}^{\mathbf{T}} = \mathtt{EL}^{\mathbf{T}} + [!_a\phi]$ Public announcement language with types,
- $\mathtt{PQLT}^{\mathbf{T}} = \mathtt{EL}^{\mathbf{T}} + [!_a\phi] + [?_a\phi] + [!_a]$ Public question language with types,
- $\mathtt{PQLT}^{\mathbf{T}}_{\mathbf{U}} = \mathtt{EL}^{\mathbf{T}} + [!_a u] + [?_a\phi] + [!_a]$ Public question language with types and arbitrary utterances.

In $\mathtt{PALT}^{\mathbf{T}}$, $\mathtt{PQLT}^{\mathbf{T}}$, and $\mathtt{PQLT}^{\mathbf{T}}_{\mathbf{U}}$, *who* says what is important due to the types of speakers. These languages are very powerful in expressing complicated announcements and questions such as the apparently paradoxical 'I am a liar' announcement and counterfactual questions like 'would he answer "yes " if he were asked ϕ?'.

The first four languages are interpreted on epistemic models with type assignments, while the last language $\mathtt{PQLT}^{\mathbf{T}}_{\mathbf{U}}$ is interpreted on epistemic models with type assignments and utterance interpretations. We have shown that the first four languages are equally expressive. This does not mean that we do not need $\mathtt{PAL}^{\mathbf{T}}$, $\mathtt{PALT}^{\mathbf{T}}$, and $\mathtt{PQLT}^{\mathbf{T}}$ any more: on the contrary, they allow us to express things more naturally. As with standard public announcement logic (cf. (Lutz, 2006) and (French et al, 2011)), we conjectured that $\mathtt{PALT}^{\mathbf{T}}$ enjoys an exponential gain in succinctness than $\mathtt{EL}^{\mathbf{T}}$. Moreover, the expressiveness results do not tell us everything about those logics, e.g., in $\mathtt{PALT}^{\mathbf{T}}$, two announcements cannot be composed into one in general, but only for special cases with certain $\mathbf{T}$. We also showed that the public announcements in $\mathtt{PAL}^{\mathbf{T}}$ can be mimicked by typed announcements with $\mathbf{T}$ containing $\mathtt{LL}$ and $\mathtt{TT}$. There is a lot more to be explored about these logics.

We studied several variations of the Knight and Knave puzzles within the logical frameworks that we developed. In particular, we formalized HLPE and verified a classic solution. It was also shown that puzzles involving only objective truth tellers and liars are usually simpler than those with subjective types and epistemic uncertainties. Following this insight, we proposed new harder puzzles based on the original HLPE with complicated epistemic reasoning involved. In particular, we showed that there is *no* solution to a variation of HLPE, when the gods in the original HLPE are replaced by humans who do not know each other's types. However, there is a questioning strategy that can let the questioner know the type of one non-bluffer.

The discussion of HLPE has demonstrated the power of our formal approach in handling complicated epistemic reasoning based on types of agents. However, the proofs for most of our results about HLPE boil down to tedious combinatorial analysis. Actually, we can save effort here by using automatic model checking methods

based on our logical frameworks (cf. e.g., (Clarke et al, 1999)). In this paper, we have formally defined what is a puzzle and what is a solution to a puzzle. The verification of a solution is then transformed into model checking problems for certain modal formulas. In principle, we can then use techniques from model checking in our setting. Our translations between languages allow us to do model checking of the complex languages by model checking the translated formulas in the simpler languages. Moreover, solutions to the puzzles can be found by a bounded search over the possible sequences of questions. Thus the spectrum of new puzzles should not be solved by hand but in an automatic manner. A detailed discussion of computational issues of the model checking problem is beyond the scope of this paper, and is left for future work.

Finally, we end our paper with a list of important further issues:

- *The boundary between the solvable and the unsolvable* We have shown that, if $A, B,$ and C do not know each other's types, then there is no solution to the revised HLPE puzzle. Since there are indeed solutions to the original HLPE puzzle, the natural question to ask is: Can we find a 'minimal' assumption on the knowledge of $A, B,$ and C such that there is a solution? On the other hand, we can also keep the assumption of ignorance but allow agents to say 'I do not know' in some way to see whether this will lead to a solvable puzzle. As we discussed in Remark 5, this may raise several new possibilities for defining the types of subjective liars and bluffers.
- *From knowledge to belief and more* In this paper we defined 'subjective' agent types by conditioning on *knowledge* of agents. However, realistic agents often rely on their *beliefs* to make announcements or answer questions. We can certainly replace knowledge operators with belief operators in types. An even further way to go is to consider probability distributions of propositions as preconditions of agent types, e.g., a liar is some one who tells lies 80% of the time.
- *Richer agent types* In this work, we focused on agent types in terms of *what* agents deliver by their announcements. There are definitely richer types in real life. For example, agent types may be reflected in *how much* information they would like to deliver w.r.t. what they *know*. A conservative agent may only announce $\phi \vee \psi$ even when he knows ϕ. We will leave those richer types for other occasions.

Acknowledgements The authors would like to thank Hans van Ditmarsch and Johan van Benthem for their detailed comments on earlier versions of this paper, and thank Gregory Wheeler for pointing out the literature on the HLPE, which helped to shape the development of this work. We are also grateful to two anonymous referees of this journal for their very valuable comments. Both authors are partially supported by the Major Program of National Social Science Foundation of China (NO.11&ZD088). Yanjing Wang is also supported by the MOE Project of Key Research Institute of Humanities and Social Sciences in Universities (No.12JJD720011).

References

van Benthem J, Minică S (2009) Toward a dynamic logic of questions. In: He X, Horty JF, Pacuit E (eds) Proceedings of the 2nd International Workshop on Logic, Rationality and Interaction (LORI-II), Springer, FoLLI-LNAI, vol 5834, pp 27–41

Blackburn P, de Rijke M, Venema Y (2002) Modal Logic. Cambridge University Press

Boolos G (1996) The hardest logic puzzle ever. The Harvard Review of Philosophy 6:62–65
Clarke EM, Grumberg O, Peled DA (1999) Model Checking. The MIT Press
van Ditmarsch H (2011) The Ditmarsch tale of wonders—dynamics of lying. Manuscript
van Ditmarsch H, Kooi B (2006) The secret of my success. Synthese 153(2):339
van Ditmarsch H, van der Hoek W, Kooi B (2007) Dynamic Epistemic Logic. Berlin: Springer
van Ditmarsch H, van Eijck J, Sietsma F, Wang Y (2011) On the logic of lying. In: van Eijck J, Verbrugge R (eds) Games, Actions and Social Software, Springer, pp 41–72
French T, van der Hoek W, Iliev P, Kooi BP (2011) Succinctness of epistemic languages. In: Walsh T (ed) Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence(IJCAI), pp 881–886
Gerbrandy J, Groeneveld W (1997) Reasoning about information change. Journal of Logic, Language and Information 6(2):147–169
Holliday W, Icard III T (2010) Moorean phenomena in epistemic logic. In: Advances in Modal Logic, pp 178–199
Liu F (2004) Dynamic variations: Update and revision for diverse agents. Master's thesis, MoL-2004-05. ILLC, University of Amsterdam
Liu F (2009) Diversity of agents and their interaction. Journal of Logic, Language and Information 18(1):23–53
Lutz C (2006) Complexity and succinctness of public announcement logic. In: Stone P, Weiss G (eds) Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS '06), ACM, New York, NY, USA, pp 137–143
Minică S (2011) Dynamic logic of questions. PhD thesis, Universiteit van Amsterdam
Plaza J (2007) Logics of public communications. Synthese 158(2):165–179
Rabern B, Rabern L (2008) A simple solution to the hardest logic puzzle ever. Analysis 68:105–112
Smullyan R (1978) What is the name of this book. Prentice-Hall
Uzquiano G (2010) How to solve the hardest logic puzzle ever in two questions. Analysis 70:39–44
Wang Y (2011a) On axiomatizations of PAL. In: van Ditmarsch H, Lang J, Ju S (eds) Proceedings of the 3nd International Workshop on Logic, Rationality and Interaction (LORI-III), Springer-Verlag, FoLLI-LNAI, vol 6953, pp 314–327
Wang Y (2011b) Reasoning about protocol change and knowledge. In: Banerjee M, Seth A (eds) Proceedings of the 4th Indian Conference on Logic and its Applications (ICLA), Springer-Verlag, LNCS, vol 6521, pp 189–203
Wheeler GR, Barahona P (2012) Why the hardest logic puzzle ever cannot be solved in less than three questions. Journal of Philosophical Logic 41(2):493–503
Wintein S (2011) On the behavior of true and false. Minds and Machines 22(1):1–24

Appendix

Proof of Proposition 8

Proof $\Rightarrow$: Intuitively, if D can find out the types of A, B, and C without knowing the meaning of *da* and *ja*, he should be able to find it out when he knows the meaning

of *ja* and *da*. We now prove it formally. Suppose there is a solution π for $(\mathfrak{M}_1, \theta)$. Now consider an arbitrary sequence in $Seq(\pi)$:

$$?_{a_1}\psi_1!_{a_1}u_1 \cdots ?_{a_n}\psi_n!_{a_n}u_n \in Seq(\pi).$$

Let γ be the following formula:

$$[?_{a_1}\psi_1](\langle!_{a_1}\rangle\top \wedge [!_{a_1}u_1][?_{a_2}\psi_2](\langle!_{a_2}\rangle\top \wedge [!_{a_2}u_2][?_{a_3}\psi_3] \\ (\ldots [?_{a_n}\psi_n](\langle!_n\rangle\top \wedge [!_{a_n}u_n]\theta)..)))$$

By the definition of solutions we have: $\mathfrak{M}_1 \Vdash \gamma$ for all such γ. Due to Q2, ψ_i is D-free for $i \leq n$, thus the updates introduced by the answers are not relevant to the D-links in the model. Moreover, since θ is a positive formula w.r.t. K_D and the answers are essentially submodel operations, it is not hard to show that θ is preserved under models with less D-links compared to $\mathfrak{M}_1$.[11] Therefore it is easy to see that $\mathfrak{M}_1' \Vdash \gamma$ for each γ. Now since $\mathfrak{M}_2$ is a generated submodel of $\mathfrak{M}_1'$ it is clear that: $\mathfrak{M}_2 \Vdash \gamma$ for each γ. This means that π is also a solution for $(\mathfrak{M}_2, \theta)$.

For the other direction: suppose there is a solution π for $(\mathfrak{M}_2, \theta)$, then we can assume that the questions in π are *ja*- and *da*-free. To see this, note that given a model, each formula can be viewed as a set of possible worlds in this model. However, any subset of the worlds in $\mathfrak{M}_2$ can be defined by a Boolean combination of type formulas without using any modalities. Therefore we can always replace questions involving modalities with a question without such modalities. Now we obtain π^* by replacing each question $?_a\psi$ in π by the question $?_aE_a^*((\mathtt{STT}(a) \to \psi) \wedge (\mathtt{SLL}(a) \to \neg\psi))$, namely:

$$?_a([?_a((\mathtt{STT}(a) \to \psi) \wedge (\mathtt{SLL}(a) \to \neg\psi))]\langle!_a ja\rangle\top).$$

We claim that π^* is a solution to $(\mathfrak{M}_1, \theta)$. To prove this, we will use the idea in the proof of Lemma 1. Recall that Lemma 1 does not work any more, if we replace $\mathfrak{M}_0$ by $\mathfrak{M}_1$. However, we know more about those ψ appeared in π^*: they are from the solution π to the puzzle $(\mathfrak{M}_2, \theta)$, thus they should be always answerable when asked. Namely when $?_a\psi$ is asked on a submodel $\mathfrak{N}$ of $\mathfrak{M}_2$ we have $\mathfrak{N} \Vdash \neg\mathtt{LT}(a) \to (K_a\psi \vee K_a\neg\psi)$, thus

$$\mathfrak{N} \Vdash \neg\mathtt{LT}(a) \to ((\psi \leftrightarrow K_a\psi) \wedge (\neg\psi \leftrightarrow K_a\neg\psi)) \qquad \text{(i)}.$$

Let $\xi(a) = (\mathtt{STT}(a) \to \psi) \wedge (\mathtt{SLL}(a) \to \neg\psi)$. Clearly, if $\mathfrak{N}, s \Vdash \mathtt{STT}(a) \wedge K_a\psi$ then $\mathfrak{N}, s \Vdash K_a\xi(a)$; if $\mathfrak{N}, s \Vdash \mathtt{STT}(a) \wedge K_a\neg\psi$ then $\mathfrak{N}, s \Vdash K_a\neg\xi(a)$; if $\mathfrak{N}, s \Vdash \mathtt{SLL}(a) \wedge K_a\psi$ then $\mathfrak{N}, s \Vdash K_a\neg\xi(a)$; if $\mathfrak{N}, s \Vdash \mathtt{SLL}(a) \wedge K_a\neg\psi$ then $\mathfrak{N}, s \Vdash K_a\xi(a)$. In sum, $\mathfrak{N} \Vdash \neg\mathtt{LT}(a) \to (K_a\xi \vee K_a\neg\xi)$, thus

$$\mathfrak{N} \Vdash \neg\mathtt{LT}(a) \to ((\xi(a) \leftrightarrow K_a\xi(a)) \wedge (\neg\xi(a) \leftrightarrow K_a\neg\xi(a))) \qquad \text{(ii)}.$$

Namely, $?_a\xi$ is always answerable in $\mathfrak{N}$. Therefore it is easy to see that the following holds:

$$\mathfrak{N} \Vdash \neg\mathtt{LT}(a) \to ((K_a\neg[?_a\xi(a)]\langle!_a ja\rangle\top) \leftrightarrow (K_a[?_a\xi(a)]\langle!_a da\rangle\top)) \qquad \text{(iii)}$$

[11] Interested readers may consult (Blackburn et al, 2002) for the preservation result of positive formulas in the standard setting of modal logic.

Given a submodel $\mathfrak{N}$ of $\mathfrak{M}_1$, we want to know what do $\mathfrak{N}|^a_{?_a E^*_a(\xi(a)),ja}$ and $\mathfrak{N}|^a_{?_a E^*_a(\xi(a)),da}$ look like. Now let us follow the reasoning in the proof of Lemma 1 (assuming WLOG that $a = A$):

$$
\begin{aligned}
&\mathfrak{N}, \_\_\_\mathtt{JA} \Vdash [?_A[?_A\xi(A)]\langle !_A ja\rangle\top]\langle !_A ja\rangle\top \\
\iff &\mathfrak{N}, s \Vdash_{\#} \lambda(s, A)([?_A\xi(A)]\langle !_A ja\rangle\top, A) \text{ if } s = \_\_\_\mathtt{JA} \\
\iff &\begin{cases} \mathfrak{N}, s \Vdash_{\#} K_A[?_A\xi(A)]\langle !_A ja\rangle\top & \text{if } s = \mathtt{TT}\_\_\mathtt{JA} \\ \mathfrak{N}, s \Vdash_{\#} K_A\neg[?_A\xi(A)]\langle !_A ja\rangle\top & \text{if } s = \mathtt{LL}\_\_\mathtt{JA} \\ \mathfrak{N}, s \Vdash_{\#} \top & \text{if } s = \mathtt{LT}\_\_\mathtt{JA} \end{cases} \\
\iff &\begin{cases} \mathfrak{N}, s \Vdash_{\#} K_A[?_A\xi(A)]\langle !_A ja\rangle\top & \text{if } s = \mathtt{TT}\_\_\mathtt{JA} \\ \mathfrak{N}, s \Vdash_{\#} K_A[?_A\xi(A)]\langle !_A da\rangle\top & \text{if } s = \mathtt{LL}\_\_\mathtt{JA} \text{ (due to (iii))} \\ \mathfrak{N}, s \Vdash_{\#} \top & \text{if } s = \mathtt{LT}\_\_\mathtt{JA} \end{cases} \\
\iff &\begin{cases} \mathfrak{N}, s \Vdash_{\#} K_A K_A\xi(A) & \text{if } s = \mathtt{TT}\_\_\mathtt{JA} \\ \mathfrak{N}, s \Vdash_{\#} K_A K_A\neg\neg\xi(A) & \text{if } s = \mathtt{LL}\_\_\mathtt{JA} \\ \mathfrak{N}, s \Vdash_{\#} \top & \text{if } s = \mathtt{LT}\_\_\mathtt{JA} \end{cases} \\
\iff &\begin{cases} \mathfrak{N}, s \Vdash_{\#} \xi(A) & \text{if } s = \mathtt{TT}\_\_\mathtt{JA} \\ \mathfrak{N}, s \Vdash_{\#} \neg\neg\xi(A) & \text{if } s = \mathtt{LL}\_\_\mathtt{JA} \text{ (due to (ii))} \\ \mathfrak{N}, s \Vdash_{\#} \top & \text{if } s = \mathtt{LT}\_\_\mathtt{JA} \end{cases} \\
\iff &\begin{cases} \mathfrak{N}, s \Vdash_{\#} \xi(A) & \text{if } s \neq \mathtt{LT}\_\_\mathtt{JA} \\ \mathfrak{N}, s \Vdash_{\#} \top & \text{if } s = \mathtt{LT}\_\_\mathtt{JA} \end{cases}
\end{aligned}
$$

Similarly, we have:

$$
\begin{aligned}
&\mathfrak{N}, \_\_\_\mathtt{DA} \Vdash [?_A[?_A\xi(A)]\langle !_A ja\rangle\top]\langle !_A ja\rangle\top \\
\iff &\begin{cases} \mathfrak{N}, s \Vdash_{\#} \xi(A) & \text{if } s \neq \mathtt{LT}\_\_\mathtt{DA} \\ \mathfrak{N}, s \Vdash_{\#} \top & \text{if } s = \mathtt{LT}\_\_\mathtt{DA} \end{cases} \\
&\mathfrak{N}, \_\_\_\_ \Vdash [?_A[?_A\xi(A)]\langle !_A ja\rangle\top]\langle !_A da\rangle\top \\
\iff &\begin{cases} \mathfrak{N}, s \Vdash_{\#} \neg\xi(A) & \text{if } s \neq \mathtt{LT}\_\_\_ \\ \mathfrak{N}, s \Vdash_{\#} \top & \text{if } s = \mathtt{LT}\_\_\_ \end{cases}
\end{aligned}
$$

In sum, we have:

$$
?_a E^*_a(\xi(a)) \begin{cases} \text{answer } ja\text{, then } \ \mathfrak{N}|^a_{E^*_a(\xi(a)),ja} \text{ keeps the worlds } \begin{cases} \xi(a) \wedge \mathtt{STT}(a) \\ \xi(a) \wedge \mathtt{SLL}(a) \\ \top \wedge \mathtt{LT}(a) \end{cases} \\ \text{answer } da\text{, then } \ \mathfrak{N}|^a_{E^*_a(\xi(a)),da} \text{ keeps the worlds } \begin{cases} \neg\xi(a) \wedge \mathtt{STT}(a) \\ \neg\xi(a) \wedge \mathtt{SLL}(a) \\ \top \wedge \mathtt{LT}(a) \end{cases} \end{cases}
$$

Instantiate $\xi(a) = (\mathtt{STT}(a) \to \psi) \wedge (\mathtt{SLL}(a) \to \neg\psi)$, we have:

$$
?_a E^*_a(\xi(a)) \begin{cases} \text{answer } ja\text{, then } \ \mathfrak{N}|^a_{E^*_a(\xi(a)),ja} \text{ keeps the worlds } \begin{cases} \psi \wedge \mathtt{STT}(a) \\ \neg\psi \wedge \mathtt{SLL}(a) \\ \top \wedge \mathtt{LT}(a) \end{cases} \\ \text{answer } da\text{, then } \ \mathfrak{N}|^a_{E^*_a(\xi(a)),da} \text{ keeps the worlds } \begin{cases} \neg\psi \wedge \mathtt{STT}(a) \\ \psi \wedge \mathtt{SLL}(a) \\ \top \wedge \mathtt{LT}(a) \end{cases} \end{cases}
$$

On the other hand, for any submodel $\mathfrak{N}'$ of $\mathfrak{M}_2$:

$$?_a\psi \begin{cases} \text{answer } \textit{ja}, \text{ then } \ \mathfrak{N}'|^a_{\psi,ja} \text{ keeps the worlds } \begin{cases} \psi \wedge \mathtt{STT}(a) \\ \neg\psi \wedge \mathtt{SLL}(a) \quad \text{(due to (i))} \\ \top \wedge \mathtt{LT}(a) \end{cases} \\ \text{answer } \textit{da}, \text{ then } \ \mathfrak{N}'|^a_{\psi,da} \text{ keeps the worlds } \begin{cases} \neg\psi \wedge \mathtt{STT}(a) \\ \psi \wedge \mathtt{SLL}(a) \quad \text{(due to (i))} \\ \top \wedge \mathtt{LT}(a) \end{cases} \end{cases}$$

Clearly, the answers to $?_a E^*((\mathtt{STT}(a) \to \psi) \wedge (\mathtt{SLL}(a) \to \neg\psi))$ have exactly the same update effects on submodels of $\mathfrak{M}_1$ (modulo JA and DA) as the answers to $?_a\psi$ on the corresponding submodels of $\mathfrak{M}_2$ where *ja* and *da* stand for 'yes' and 'no' respectively. Since π can guarantee we reach a singleton model in the end, the π^* can make sure we reach a model with at most two worlds, which differ from each other only in the interpretations. Therefore π^* is a solution to $(\mathfrak{M}_1, \theta)$. □

Proof of Proposition 9

Proof WLOG let $a = A$. Suppose $\mathfrak{N} \Vdash \neg\mathtt{SLL}(A)$ or $\mathfrak{N} \Vdash \neg\mathtt{STT}(A)$, namely, either $\mathfrak{N}$ does not have any SLL__JA world or $\mathfrak{N}$ does not have any STT__JA world. In the sequel, we only consider the first case. For the other case, similar proof works.

If $\mathfrak{N}$ does not have any SLL__JA world, there are three subcases:

1. $\mathfrak{N}$ only has STT__JA worlds. Since $\mathfrak{N}$ is the submodel of $\mathfrak{M}_2$ then there are at most two STT__JA worlds in $\mathfrak{N}$. If there is only one world then no question can be effective since the model is already minimal. If there are two worlds (call them s and t), then these two worlds are clearly linked by indistinguishability relations of A and D, since $\mathfrak{N}$ is a submodel of $\mathfrak{M}_2$. Now whatever A answers to the question $?_A\phi$, the answer must be known to A, thus it holds on both worlds. More formally, given a question $?_A\phi$, the update effects of its answers can be analysed as follows:[12]

$$?_A\phi \begin{cases} \text{answer } \textit{ja}, \text{ then } \ \mathfrak{N}|^A_{\phi,ja} \text{ keeps the worlds where } K_A\phi \wedge \mathtt{STT}(A) \text{ is true} \\ \text{answer } \textit{da}, \text{ then } \ \mathfrak{N}|^A_{\phi,da} \text{ keeps the worlds where } K_A\neg\phi \wedge \mathtt{STT}(A) \text{ is true} \end{cases}$$

Clearly these answers, if executable, will not change the model.

2. $\mathfrak{N}$ only has LT__JA worlds. Whatever the bluffer answers, the model will not be changed at all due to the definition of the bluffer type.

3. $\mathfrak{N}$ has at least one STT__JA world and at least one LT__JA world but does not have any SLL__JA world. Given any question $?_A\phi$, the update effects of its answers can be analysed as follows:

$$?_A\phi \begin{cases} \text{answer } \textit{ja}, \text{ then } \ \mathfrak{N}|^A_{\phi,ja} \text{ keeps the worlds } \begin{cases} K_A\phi \wedge \mathtt{STT}(A) \\ \top \wedge \mathtt{LT}(A) \end{cases} \\ \text{answer } \textit{da}, \text{ then } \ \mathfrak{N}|^A_{\phi,da} \text{ keeps the worlds } \begin{cases} K_A\neg\phi \wedge \mathtt{STT}(A) \\ \top \wedge \mathtt{LT}(A) \end{cases} \end{cases}$$

Note that both $\mathfrak{N}|^A_{\phi,ja}$ and $\mathfrak{N}|^A_{\phi,da}$ are defined since there is at least one LT__JA world. Now suppose ϕ is effective, then both answers should eliminate some worlds in $\mathfrak{N}$.

[12] For instance, according to the semantics when s is in the shape of STT__JA, $\lambda(A,s)(I(s,\phi,ja),A) = K_A\phi$. Therefore when answering *ja*, the updated model keeps the worlds satisfying $K_A\phi \wedge \mathtt{STT}(A)$.

Clearly the condition $\top \wedge \texttt{LT}(A)$ keeps all the LT__JA worlds in $\mathfrak{N}$, thus there must be some STT__JA world t which does not satisfy $K_A\phi$ and some STT__JA world t' which does not satisfy $K_A\neg\phi$. However this is impossible, since all the STT__JA worlds are indistinguishable for A thus t and t' satisfy the same K_A formulas.

In sum, $?_A\phi$ cannot be effective on $\mathfrak{N}$. □

Proof of Theorem 2

Proof According to Proposition 8, if there is no solution to $(\mathfrak{M}_2, \theta)$, then there is no solution to $(\mathfrak{M}_1, \theta)$.

Towards a contradiction, suppose that $(\mathfrak{M}_2, \theta)$ has a solution π. Since D cannot distinguish any two worlds in $\mathfrak{M}_2$ and the effects of answers are taking submodels, it is not hard to see that θ is satisfiable in a submodel $\mathfrak{N}$ of $\mathfrak{M}_2$ iff $\mathfrak{N}$ has only one world. According to the definition of solutions, for all $?_{a_1}\phi_1!_{a_1}u_1 \cdots ?_{a_n}\phi_n!_{a_n}u_n \in Seq(\pi)$, the following model (if defined) must be singleton:

$$(\dots(\mathfrak{M}_2|^{a_1}_{\phi_1,u_1})|^{a_2}_{\phi_2,u_2}\dots)|^{a_n}_{\phi_n,u_n}$$

WLOG we may assume that the solution starts with a question to A. We claim the following:

$(\mathfrak{M}_2, \theta)$ has a solution whose initial question $?_A\phi$ is effective on $\mathfrak{M}_2$ $(\star)$.

To see this, first note that both $\mathfrak{M}_2|^A_{\phi,ja}$ and $\mathfrak{M}_2|^A_{\phi,da}$ are well-defined, since *da* and *ja* can be answered for the worlds LT__JA in $\mathfrak{M}_2$. Now if ϕ is not effective, then either $\mathfrak{M}_2|^A_{\phi,ja} = \mathfrak{M}_2$ or $\mathfrak{M}_2|^A_{\phi,da} = \mathfrak{M}_2$. In either case, the initial question is useless, e.g., if $\mathfrak{M}_2|^A_{\phi,ja} = \mathfrak{M}_2$ then answering *ja* will not bring any new information, thus we may well ignore the first question and let the question which was previously after answering *ja* be the new initial question.

Based on the claim $(\star)$, let us focus on the effective questions and their update effects by analysing the updated models:

$$?_A\phi \begin{cases} \text{answer } ja\text{, then } \ \mathfrak{M}_2|^A_{\phi,ja} \text{ keeps the worlds } \begin{cases} K_A\phi \wedge \texttt{STT}(A) \\ K_A\neg\phi \wedge \texttt{SLL}(A) \\ \top \wedge \texttt{LT}(A) \end{cases} \\ \text{answer } da\text{, then } \ \mathfrak{M}_2|^A_{\phi,da} \text{ keeps the worlds } \begin{cases} K_A\neg\phi \wedge \texttt{STT}(A) \\ K_A\phi \wedge \texttt{SLL}(A) \\ \top \wedge \texttt{LT}(A) \end{cases} \end{cases}$$

It is easy to see that all the LT__JA worlds will be kept in the updated models. If the question is effective then answering *ja* or *da* should both change the model by eliminating some worlds. For the case of *ja* this means either there is some STT__JA world t which does not satisfy $K_A\phi$ or there is some SLL__JA world t' which does not satisfy $K_A\neg\phi$.

(i) Suppose it is the first case. Since $?_A\phi$ should be answerable, then $K_A\phi \vee K_A\neg\phi$ holds at t, thus $K_A\neg\phi$ holds at t. However, this also means that $K_A\neg\phi$ holds at all the worlds in the shape of STT__JA (1). Thus in the clause for *da* the condition $K_A\neg\phi \wedge \texttt{STT}(A)$ will be satisfied by all the STT__JA worlds. Since the question is effective, there must be some SLL__JA world which does not satisfy $K_A\phi$. Again since ϕ is answerable, there must be some SLL__JA world which satisfy $K_A\neg\phi$. Then

it means that all the $\texttt{SLL\_\_JA}$ worlds satisfy $K_A\neg\phi$ (2). Together with (1), we know that $\texttt{SLL}(A)\vee\texttt{STT}(A)\to K_A\neg\phi$ is valid in $\mathfrak{M}_2$. Therefore:

$$?_A\phi\begin{cases}\text{answer } \textit{ja}, \text{ then } & \mathfrak{M}_2|^A_{\phi,ja} \text{ keeps the worlds } \begin{cases}\texttt{SLL\_\_JA}\\ \texttt{LT\_\_JA}\end{cases}\\ \text{answer } \textit{da}, \text{ then } & \mathfrak{M}_2|^A_{\phi,da} \text{ keeps the worlds } \begin{cases}\texttt{STT\_\_JA}\\ \texttt{LT\_\_JA}\end{cases}\end{cases}$$

(ii) Suppose there is some $\texttt{SLL\_\_JA}$ world t' which does not satisfy $K_A\neg\phi$. With similar analysis we conclude that: $\texttt{SLL}(A)\vee\texttt{STT}(A)\to K_A\phi$ is valid in $\mathfrak{M}_2$. Therefore:

$$?_A\phi\begin{cases}\text{answer } \textit{ja}, \text{ then } & \mathfrak{M}_2|^A_{\phi,ja} \text{ keeps the worlds } \begin{cases}\texttt{STT\_\_JA}\\ \texttt{LT\_\_JA}\end{cases}\\ \text{answer } \textit{da}, \text{ then } & \mathfrak{M}_2|^A_{\phi,da} \text{ keeps the worlds } \begin{cases}\texttt{SLL\_\_JA}\\ \texttt{LT\_\_JA}\end{cases}\end{cases}$$

Based on (i) and (ii), we know that one of the answers to an effective question eliminates $\texttt{STT\_\_JA}$ worlds and the other eliminates $\texttt{SLL\_\_JA}$ worlds. Note that such effective questions do exist, e.g., $?_A\texttt{LT}(A)$ and $?_A\neg\texttt{LT}(A)$.

Now by Proposition 9, asking A again cannot be effective any more thus the next effective questions must be asked to B or C. Note that we can again ignore the ineffective questions for the reasons mentioned earlier. Suppose WLOG that after A's answering *ja* we are left with $\texttt{SLL\_\_JA}$ and $\texttt{LT\_\_JA}$ worlds as in case (i), and B is then asked. The effects of B's answers are as follows:

$$?_B\phi'\begin{cases}(\mathfrak{M}_2|^A_{\phi,ja})|^B_{\phi',ja} \text{ keeps the worlds } \begin{cases}K_B\phi'\wedge\texttt{STT}(B)\\ K_B\neg\phi'\wedge\texttt{SLL}(B)\\ \top\wedge\texttt{LT}(B)\end{cases} \text{ out of } \begin{cases}\texttt{SLL\_\_JA}\\ \texttt{LT\_\_JA}\end{cases}\\ (\mathfrak{M}_2|^A_{\phi,ja})|^B_{\phi',da} \text{ keeps the worlds } \begin{cases}K_B\neg\phi'\wedge\texttt{STT}(B)\\ K_B\phi'\wedge\texttt{SLL}(B)\\ \top\wedge\texttt{LT}(B)\end{cases} \text{ out of } \begin{cases}\texttt{SLL\_\_JA}\\ \texttt{LT\_\_JA}\end{cases}\end{cases}$$

With exactly the same argument as in the case of $?_A\phi$ above, to make ϕ' effective, $\texttt{SLL}(B)\vee\texttt{STT}(B)\to K_B\phi'$ or $\texttt{SLL}(B)\vee\texttt{STT}(B)\to K_B\neg\phi'$ should be valid in $\mathfrak{M}_2|^A_{\phi,ja}$ and the update effects of the answers are to eliminate one of the possibilities of $\_\texttt{STT}(B)\_\texttt{JA}$ and $\_\texttt{SLL}(B)\_\texttt{JA}$. Note that $\mathfrak{M}_2|^A_{\phi,ja}$ keeps $\texttt{SLL\_\_JA}$ and $\texttt{LT\_\_JA}$ worlds, thus $\mathfrak{M}_2|^A_{\phi,ja}$ can be depicted as below:

$$\begin{array}{ccc}\texttt{SLL},\texttt{STT},\texttt{LT},\texttt{JA} & -B,D- & \texttt{LT},\texttt{STT},\texttt{SLL},\texttt{JA}\\ \Big|{\scriptstyle A,D} & & \Big|{\scriptstyle A,D}\\ \texttt{SLL},\texttt{LT},\texttt{STT},\texttt{JA} & -C,D- & \texttt{LT},\texttt{SLL},\texttt{STT},\texttt{JA}\end{array}$$

Then $(\mathfrak{M}_2|^A_{\phi,ja})|^B_{\phi',ja}$ is one of the following two models and $(\mathfrak{M}_2|^A_{\phi,ja})|^B_{\phi',da}$ is the other one:

$$\begin{array}{ccc}\texttt{SLL},\texttt{STT},\texttt{LT},\texttt{JA} & -B,D- & \texttt{LT},\texttt{STT},\texttt{SLL},\texttt{JA}\\ & & \Big|{\scriptstyle A,D}\\ \texttt{SLL},\texttt{LT},\texttt{STT},\texttt{JA} & -C,D- & \texttt{LT},\texttt{SLL},\texttt{STT},\texttt{JA}\end{array}\qquad\qquad \texttt{SLL},\texttt{LT},\texttt{STT},\texttt{JA}$$

For the left-hand-side case, there is no effective questions to ask any more due to Proposition 9 and the fact that $\neg\mathtt{STT}(A) \wedge \neg\mathtt{STT}(B) \wedge \neg\mathtt{SLL}(C)$ is valid in the model. Thus it is easy to see that we cannot guarantee one of the two worlds will be eliminated by an answer. It is interesting that although D now knows C's type, he cannot know A and B's type. Moreover, C cannot help him since he also does not know the others' types. On the other hand, although A and B now know the types of everyone, they cannot help D either, since D cannot distinguish who is bluffing. For the right-hand model, there are still effective questions for C since all the possibilities $\mathtt{STT}(C), \mathtt{SLL}(C)$ and $\mathtt{LT}(C)$ are still there in the model. However, with a similar analysis as in the first two steps, after C answers an effective question, the situation will be similar to the above left-hand-side model: we are left with two worlds and cannot guarantee that every answer will make a difference.

In sum, we cannot guarantee that we will reach a singleton model in the end, thus it is contradictory to the assumption that there is a solution. □