

Automating Public Announcement Logic with Relativized Common Knowledge as a Fragment of HOL in LogiKEy

Christoph Benzmüller

University of Bamberg, AI Systems Engineering, Bamberg, Germany &
FU Berlin, Dep. of Mathematics and Computer Science, Berlin, Germany
c.benzmueller@fu-berlin.de

Sebastian Reiche*

FU Berlin, Dep. of Mathematics and Computer Science, Berlin, Germany
sebastian.reiche@fu-berlin.de

January 28, 2022

Abstract

A shallow semantical embedding for public announcement logic with relativized common knowledge is presented. This embedding enables the first-time automation of this logic with off-the-shelf theorem provers for classical higher-order logic. It is demonstrated (i) how meta-theoretical studies can be automated this way, and (ii) how non-trivial reasoning in the target logic (public announcement logic), required e.g. to obtain a convincing encoding and automation of the wise men puzzle, can be realized.

Key to the presented semantical embedding is that evaluation domains are modeled explicitly and treated as an additional parameter in the encodings of the constituents of the embedded target logic; in previous related works, e.g. on the embedding of normal modal logics, evaluation domains were implicitly shared between meta-logic and target logic.

The work presented in this article constitutes an important addition to the pluralist LOGIKEY knowledge engineering methodology, which enables experimentation with logics and their combinations, with general and domain knowledge, and with concrete use cases — all at the same time.

Keywords: Public announcement logic; Relativized common knowledge; Semantical embedding; Higher-order logic; Proof automation

*Corresponding Author

1 Introduction

Previous work has studied the application of a universal (meta-)logical reasoning approach [7, 8] for solving a prominent riddle in epistemic reasoning, known as the *wise men puzzle*, on the computer [8]. The solution presented there puts a particular emphasis on the adequate modeling of (ordinary) common knowledge and it also illustrates the elegance and the practical relevance of shallow¹ semantical embeddings (SSEs; cf. [15, 7]) of non-classical ‘object’ logics in classical higher-order logic (HOL, aka Church’s simple type theory; cf. [9]), when being utilized within modern proof assistant systems such as Isabelle/HOL [35]. However, this work nevertheless falls short, since it did not convincingly address the interaction dynamics between the involved agents. To this end, we extend and adapt in this article the universal (meta-)logical reasoning approach for *public announcement logic*, and we demonstrate how it can be utilized to achieve a convincing encoding and automation of the wise men puzzle in Isabelle/HOL, so that also the interaction dynamics as given in the scenario is adequately addressed. In more general terms, we present the first automation of public announcement logic (PAL) with relativized common knowledge, and we demonstrate that, and how, this logic can be seen and handled as a fragment of HOL. Key to the presented extension of the shallow semantical embedding approach is that the evaluation domains of the embedded target logic (PAL with relativized common knowledge) are no longer implicitly shared with the meta-logic HOL, and are instead now explicitly modeled as an additional parameter in the encoding of the embedded logics constituents. We expect this approach of dynamizing shallow semantic embeddings of object logics in HOL to be applicable and adaptable to a wide spectrum of related works; cf. [3, 36] and the references therein.

The work presented in this article constitutes an important addition to the pluralist LOGIKEY approach and methodology [13, 11]. LOGIKEY’s unifying formal framework is fundamentally based on SSEs of ‘object’ logics (and their combinations) in HOL, enabling the provision of powerful tool support [12]: off-the-shelf theorem provers and model finders for HOL (as provided in Isabelle/HOL) are assisting the LOGIKEY knowledge engineer to *flexibly experiment* with underlying logics and their combinations, with general and domain knowledge, and with concrete use cases—all at the same time. Continuous improvements of these off-the-shelf provers, without further ado, boost the reasoning performance in LOGIKEY. Of course, specific PAL theorem provers (e.g. [1]) are still expected to outperform the generic reasoning tools in LOGIKEY. However, both approaches can be merged in the future and specialist provers can/should be added, e.g., as oracles to the LogiKEY framework. Regarding flexibility, there is clearly an advantage on the side of the LOGIKEY approach, and it should actually be straightforward to adapt the SSE we present in this article (in future work) also for *arbitrary announcement operators* as provided in APAL [1] or to DEL with action models [3].

This article is structured as follows: §2 briefly recaps HOL (Church’s simple type theory), and §3 sketches PAL with relativized common knowledge. The main contribution of this article is presented in §4: a shallow semantical embedding of PAL with relativized

¹Shallow semantical embeddings are different from *deep embeddings* of an object logic. In the latter case the syntax of the object logic is represented using an inductive data structure (e.g., following the definition of the language). The semantics of a formula is then evaluated by recursively traversing the data structure, and additionally a proof theory for the logic may be encoded. Deep embeddings typically require technical inductive proofs, which hinder proof automation, that can be avoided when shallow semantical embeddings are used instead. For more information on shallow and deep embeddings we refer to the literature [24, 34].

common knowledge in HOL. The soundness of this embedding is subsequently proved in §5. Automation aspects are studied in §6. This includes meta-level reasoning and completeness studies (in §6.1), the exploration of failures of uniform substitution (in §6.2), and finally an application of our framework to obtain an adequate modeling and automation of the prominent wise men puzzle (in §6.3). §7 discusses related work and §8 concludes the article.

This article extends and improves our prior paper [33] in various respects. In addition to an overall improved presentation, we add a soundness proof, we show how completeness can be ensured, we modularize our embedding and its encoding in Isabelle/HOL, and we further improve it by parameterizing the notions of shared and distributed knowledge over arbitrary groups of agents; moreover, we now prove the wise men puzzle automatically also for four agents.

2 Classical Higher-Order Logic

We briefly recap classical higher-order logic (HOL), respectively Church’s *simple theory of types* [18, 9], which is a logic defined on top of the simply typed lambda calculus. The presentation is partly adapted from Benzmüller [6]. For further information on the syntax and semantics of HOL we refer to [10].

Syntax of HOL. We start out with defining the set \mathcal{T} of *simple types* by the following abstract grammar: $\alpha, \beta := o \mid i \mid (\alpha \rightarrow \beta)$. Type o denotes a bivalent set of truth values, containing *truth* and *falsehood*, and i denotes a non-empty set of individuals.² Further base types are optional. \rightarrow is the function type constructor, such that $(\alpha \rightarrow \beta) \in \mathcal{T}$ whenever $\alpha, \beta \in \mathcal{T}$. We may generally omit parentheses.

The *terms* of HOL are defined by the following abstract grammar:

$$s, t := p_\alpha \mid X_\alpha \mid (\lambda X_\alpha s_\beta)_{\alpha \rightarrow \beta} \mid (s_{\alpha \rightarrow \beta} t_\alpha)_\beta$$

where $\alpha, \beta, o \in \mathcal{T}$. The $p_\alpha \in C_\alpha$ are typed constants and the $X_\alpha \in V_\alpha$ are typed variables (distinct from the p_α). If $s_{\alpha \rightarrow \beta}$ and t_α are HOL terms of types $\alpha \rightarrow \beta$ and α , respectively, then $(s_{\alpha \rightarrow \beta} t_\alpha)_\beta$, called *application*, is an HOL term of type β . If $X_\alpha \in V_\alpha$ is a typed variable symbol and s_β is an HOL term of type β , then $(\lambda X_\alpha s_\beta)_{\alpha \rightarrow \beta}$, called *abstraction*, is an HOL term of type $\alpha \rightarrow \beta$. The type of each term is given as a subscript (type subscripts, however, are often omitted if they are obvious in context). We call terms of type o *formulas*.³ As *primitive logical connectives* we choose $\neg_{o \rightarrow o}, \vee_{o \rightarrow o \rightarrow o}, =_{\alpha \rightarrow \alpha \rightarrow \alpha}$ and $\Pi_{(\alpha \rightarrow o) \rightarrow o}$. Other logical connectives can be introduced as abbreviations; e.g. $\rightarrow_{o \rightarrow o \rightarrow o} = \lambda X_o \lambda Y_o \neg X \vee Y$.

Semantics of HOL. A *frame* \mathcal{D} for HOL is a collection $\{\mathcal{D}_\alpha\}_{\alpha \in \mathcal{T}}$ of nonempty sets \mathcal{D}_α , such that $\mathcal{D}_o = \{T, F\}$ (for true and false). \mathcal{D}_i is chosen freely and $\mathcal{D}_{\alpha \rightarrow \beta}$ are collections of functions mapping \mathcal{D}_α into \mathcal{D}_β .

A *model* for HOL is a tuple $\mathcal{M} = \langle \mathcal{D}, I \rangle$, where \mathcal{D} is a frame, and I is a family of typed interpretation functions mapping constant symbols $p_\alpha \in C_\alpha$ to appropriate elements of

²In this article, we will actually associate type i later on with the domain of possible worlds.

³HOL formulas should not be confused with the PAL formulas to be defined in §3; PAL formulas will later be identified in §4.1 with HOL predicates of type $(i \rightarrow o) \rightarrow i \rightarrow o$.

\mathcal{D}_α , called the *denotation* of p_α . The logical connectives \neg, \vee, Π and $=$ are always given their expected standard denotations:

$$\begin{aligned}
I(\neg_{o \rightarrow o}) &= \text{not} \in \mathcal{D}_{o \rightarrow o} && \text{s.t. } \text{not}(\text{T}) = \text{F} \text{ and } \text{not}(\text{F}) = \text{T} \\
I(\vee_{o \rightarrow o \rightarrow o}) &= \text{or} \in \mathcal{D}_{o \rightarrow o \rightarrow o} && \text{s.t. } \text{or}(a, b) = \text{T} \text{ iff } (a = \text{T} \text{ or } b = \text{T}) \\
I(=_\alpha \rightarrow \alpha \rightarrow o) &= \text{id} \in \mathcal{D}_{\alpha \rightarrow \alpha \rightarrow o} && \text{s.t. for all } a, b \in \mathcal{D}_\alpha, \text{id}(a, b) = \text{T} \\
&&& \text{iff } a \text{ is identical to } b \\
I(\Pi_{(\alpha \rightarrow o) \rightarrow o}) &= \text{all} \in \mathcal{D}_{(\alpha \rightarrow o) \rightarrow o} && \text{s.t. for all } s \in \mathcal{D}_{\alpha \rightarrow o}, \text{all}(s) = \text{T} \\
&&& \text{iff } s(a) = \text{T} \text{ for all } a \in \mathcal{D}_\alpha
\end{aligned}$$

A *variable assignment* g maps variables X_α to elements in \mathcal{D}_α . $g[d/W]$ denotes the assignment that is identical to g , except for variable W , which is now mapped to d .

The *denotation* $\llbracket s_\alpha \rrbracket^{\mathcal{M}, g}$ of an HOL term s_α on a model $\mathcal{M} = \langle \mathcal{D}, I \rangle$ under assignment g is an element $d \in \mathcal{D}_\alpha$ defined in the following way:

$$\begin{aligned}
\llbracket p_\alpha \rrbracket^{\mathcal{M}, g} &= I(p_\alpha) \\
\llbracket X_\alpha \rrbracket^{\mathcal{M}, g} &= g(X_\alpha) \\
\llbracket (s_{\alpha \rightarrow \beta} t_\alpha)_\beta \rrbracket^{\mathcal{M}, g} &= \llbracket s_{\alpha \rightarrow \beta} \rrbracket^{\mathcal{M}, g}(\llbracket t_\alpha \rrbracket^{\mathcal{M}, g}) \\
\llbracket (\lambda X_\alpha s_\beta)_{\alpha \rightarrow \beta} \rrbracket^{\mathcal{M}, g} &= \text{the function } f \text{ from } \mathcal{D}_\alpha \text{ to } \mathcal{D}_\beta \\
&&& \text{s.t. } f(d) = \llbracket s_\beta \rrbracket^{\mathcal{M}, g[d/X_\alpha]} \text{ for all } d \in \mathcal{D}_\alpha
\end{aligned}$$

In a *standard model* a domain $\mathcal{D}_{\alpha \rightarrow \beta}$ is defined as the set of all total functions from \mathcal{D}_α to \mathcal{D}_β , i.e. $\mathcal{D}_{\alpha \rightarrow \beta} = \{f \mid f : \mathcal{D}_\alpha \rightarrow \mathcal{D}_\beta\}$. In a *Henkin model* (or general model) [27] function spaces are not necessarily required to be the full set of functions: $\mathcal{D}_{\alpha \rightarrow \beta} \subseteq \{f \mid f : \mathcal{D}_\alpha \rightarrow \mathcal{D}_\beta\}$. However, we require that the valuation function remains total, so that every term denotes.

A HOL formula s_o is *true in a Henkin model* \mathcal{M} under assignment g if and only if $\llbracket s_o \rrbracket^{\mathcal{M}, g} = \text{T}$; also denoted by $\mathcal{M}, g \models^{\text{HOL}} s_o$. A HOL formula s_o is called *valid in* \mathcal{M} , denoted by $\mathcal{M} \models^{\text{HOL}} s_o$, iff $\mathcal{M}, g \models^{\text{HOL}} s_o$ for all assignments g . Moreover, a formula s_o is called *valid*, denoted by $\models^{\text{HOL}} s_o$, if and only if s_o is valid in all Henkin models \mathcal{M} .

Due to Gödel [26] a sound and complete mechanization of HOL with standard semantics cannot be achieved. For HOL with Henkin semantics sound and complete calculi exist; cf. e.g. [10, 12] and the references therein.

Each standard model is obviously also a Henkin model. Consequently, when a HOL formula is Henkin-valid, it is also valid in all standard models.

3 Public Announcement Logic

The most important concepts and definitions of a *public announcement logic (PAL) with relativized common knowledge* are depicted. For more details, we refer to the literature [36, 29].

Before exploring these definitions, some general descriptions of the modeling approach are in order. We use a graph-theoretical structure, called *epistemic models*, to represent knowledge. Epistemic models describe situations in terms of possible worlds. A world represents one possibility about how the current situation can be. At each world an agent considers all other reachable worlds (within the given equivalence class) as possible. Each world in this set of possibilities needs to be consistent with the information the agent has. Knowledge is described using an (binary) accessibility relation between worlds, rather than directly representing the agent's information. A relation between worlds expresses that the agent is unable to tell which world is the one that represents the "real" situation.

Let \mathcal{A} be a set of agents and \mathcal{P} a set of atomic propositions. Atomic propositions are intended to describe ground facts. We use a set W to denote possible worlds and a valuation function $V : \mathcal{P} \rightarrow \wp(W)$ that assigns a set of worlds to each atomic proposition. Vice versa, we may identify each world with the set of propositions that are true in them.

Definition 3.1 (Epistemic Model). Let \mathcal{A} be a (finite) set of agents and \mathcal{P} a (finite or countable) set of atomic propositions. An *epistemic model* is a triple $\mathcal{M} = \langle W, \{R_i\}_{i \in \mathcal{A}}, V \rangle$ where $W \neq \emptyset$, $R_i \subseteq W \times W$ is an accessibility relation (for each $i \in \mathcal{A}$), and $V : \mathcal{P} \rightarrow \wp(W)$ is a valuation function ($\wp(W)$ is the powerset of W).

Information of agent i at world w can now be defined as: $R_i(w) = \{v \in W \mid wR_iv\}$. Having a separate (accessibility) relation for each agent enables them to have their own viewpoints.

Next, we introduce the syntax of our base epistemic logic as the set of sentences generated by the following grammar (where $p \in \mathcal{P}$ and $i \in \mathcal{A}$):

$$\varphi := p \mid \neg\varphi \mid \varphi \vee \psi \mid K_i\varphi$$

We also introduce the abbreviations $\varphi \wedge \psi := \neg(\neg\varphi \vee \neg\psi)$ and $\varphi \rightarrow \psi := \neg\varphi \vee \psi$.

Definition 3.2 (Truth at world w). Given an epistemic model $\mathcal{M} = \langle W, \{R_i\}_{i \in \mathcal{A}}, V \rangle$. For each $w \in W$, φ is true at world w , denoted $\mathcal{M}, w \models \varphi$, is defined inductively as follows:

$$\begin{aligned} \mathcal{M}, w \models p & \quad \text{iff} \quad w \in V(p) \\ \mathcal{M}, w \models \neg\varphi & \quad \text{iff} \quad \mathcal{M}, w \not\models \varphi \\ \mathcal{M}, w \models \varphi \vee \psi & \quad \text{iff} \quad \mathcal{M}, w \models \varphi \text{ or } \mathcal{M}, w \models \psi \\ \mathcal{M}, w \models K_i\varphi & \quad \text{iff} \quad \text{for all } v \in W, \text{ if } wR_iv \text{ then } \mathcal{M}, v \models \varphi \end{aligned}$$

The formula $K_i\varphi$ expresses that “Agent i knows φ ”. This describes knowledge as an all-or-nothing definition. If we postulate that agent i knows φ , we say that φ is true throughout all worlds in agents i ’s range of considerations (modeled as reachable worlds).

Truth of a formula φ for a model $\mathcal{M} = \langle W, \{R_i\}_{i \in \mathcal{A}}, V \rangle$ and a world $w \in W$ is expressed by writing that $\mathcal{M}, w \models \varphi$. We define $V^{\mathcal{M}}(\varphi) = \{w \in W \mid \mathcal{M}, w \models \varphi\}$. Formula φ is *valid* if and only if for all \mathcal{M} and for all worlds w we have $\mathcal{M}, w \models \varphi$.

Our (multi-)modal logic above – normal (multi-)modal logic K – is not yet sufficiently suited to encode epistemic reasoning. Therefore, additional conditions (reflexivity, transitivity and euclideaness) are imposed on the accessibility relations. In the remainder of this article we therefore assume the validity of the following S5 principles for the agent’s accessibility relations.⁴

	assumptions	axiom schemata	semantical properties
T	<i>truth</i>	$K_i\varphi \rightarrow \varphi$	reflexive
4	<i>positive introspection</i>	$K_i\varphi \rightarrow K_iK_i\varphi$	transitive
5	<i>negative introspection</i>	$\neg K_i \rightarrow K_i\neg K_i\varphi$	euclidean

⁴In the LOGIKEY approach this is achieved by simply postulating the listed semantical properties. Alternatively, the syntactic axiom schemata can be postulated in LOGIKEY, and it is also possible to automatically prove the correspondences between them [14]. Of course, the work we present in this article can be easily adapted to support also weaker versions of PAL.

We add public announcements [31] to our logic. The objective is to formulate an operation that informs all agents that some sentence φ is true. By the announcement, all agents will discard all worlds in which φ is false, and consider only those worlds in which φ is true. Because of the *publicity* of the announcement all agents are aware of the fact that all other agents know that φ is true.

Definition 3.3 (Public Announcement). Suppose that $\mathcal{M} = \langle W, \{R_i\}_{i \in \mathcal{A}}, V \rangle$ is an epistemic model and φ is a formula (in the language of our base logic). After φ is publicly announced, the resulting model is $\mathcal{M}^{!\varphi} = \langle W^{!\varphi}, \{R_i^{!\varphi}\}_{i \in \mathcal{A}}, V^{!\varphi} \rangle$ where $W^{!\varphi} = \{w \in W \mid \mathcal{M}, w \models \varphi\}$, $R_i^{!\varphi} = R_i \cap (W^{!\varphi} \times W^{!\varphi})$ for all $i \in \mathcal{A}$, and $V^{!\varphi}(p) = V(p) \cap W^{!\varphi}$ for all $p \in \mathcal{P}$.

To say that “ φ is true after the announcement of φ ” is represented as $![\varphi]\psi$. Truth for this new operator at world w in \mathcal{M} is defined as:

$$\mathcal{M}, w \models ![\varphi]\psi \text{ iff } \mathcal{M}, w \not\models \varphi \text{ or } \mathcal{M}^{!\varphi}, w \models \psi$$

We conclude this section with the introduction of notions for group knowledge.

Mutual knowledge, often stated as *everyone knows*, describes knowledge that each member of the group holds. Usually, it is defined for a group of agents $G \subseteq \mathcal{A}$ as $E_G\varphi := \bigwedge_{i \in G} K_i\varphi$. Equivalently, a new relation can be introduced to express mutual knowledge with the knowledge operator.

Definition 3.4 (Mutual Knowledge). Let $G \subseteq \mathcal{A}$ be a group of agents. Let $R_G = \bigcup_{i \in G} R_i$. The truth clause for mutual knowledge is:

$$\mathcal{M}, w \models E_G\psi \text{ iff for all } v \in W, \text{ if } wR_Gv \text{ then } \mathcal{M}, v \models \psi$$

To describe knowledge that is obtained when all agents put their individual knowledge together we introduce *distributed knowledge*.

Definition 3.5 (Distributed Knowledge). Let $G \subseteq \mathcal{A}$ be a group of agents. Let $R_D = \bigcap_{i \in G} R_i$. The truth clause for mutual knowledge is:

$$\mathcal{M}, w \models D_G\psi \text{ iff for all } v \in W, \text{ if } wR_Dv \text{ then } \mathcal{M}, v \models \psi$$

Still, there is a distinction to make between *everyone knows φ* and *it is common knowledge that φ* . A statement p is common knowledge when all agents know p , know that they all know p , know that they all know that they all know p , and so ad infinitum. Relativized common knowledge was introduced by van Benthem, van Eijck and Kooi [5] as a variant of common knowledge. As the name suggests, knowledge update is then treated as a *relativization*.

Definition 3.6 (Relativized Common Knowledge). Let $G \subseteq \mathcal{A}$ be a group of agents. Let $R_G = \bigcup_{i \in G} R_i$. The truth clause for relativized common knowledge is:

$$\mathcal{M}, w \models \mathcal{C}_G(\varphi|\psi) \text{ iff for all } v \in W, \text{ if } w(R_G^\varphi)^+v \text{ then } \mathcal{M}, v \models \psi$$

where $R_G^\varphi = R_G \cap (W \times V^{\mathcal{M}}(\varphi))$, and $(R_G^\varphi)^+$ denotes the transitive closure of R_G^φ .

Intuitively, $\mathcal{C}_G(\varphi|\psi)$ expresses that ψ is common knowledge among the agents in group G relative to the information that φ is true. This means, that every path from w , that is accessible using the agent’s relations through worlds in which φ is true, must end in

a world in which ψ is true. Ordinary unconditional common knowledge of φ can be abbreviated as $\mathcal{C}_G(\top|\varphi)$, where \top denotes an arbitrary tautology.

In the remainder we use PAL to refer to the depicted logic consisting of modal logic K, extended by the principles T45, public announcement and relativized common knowledge. The logic PAL, which we employ for the modeling of the wise men puzzle in the remainder, is now given as follows (where $p \in \mathcal{P}$, $i \in \mathcal{A}$, and $G \subseteq \mathcal{A}$):⁵

$$\varphi := p \mid \neg\varphi \mid \varphi \vee \varphi \mid K_i\varphi \mid \mathcal{C}_G(\varphi|\varphi) \mid [!\varphi]\varphi$$

4 Modeling PAL as a Fragment of HOL

A shallow semantical embedding (SSE) of a target logic into HOL provides a translation between the two logics in such a way that the former logic is identified and characterized as a proper fragment of the latter.⁶ Once such an SSE is obtained, all that is needed to prove (or refute) conjectures in the target logic is to provide the SSE, encoded in an input file, to the HOL prover in addition to the encoded conjecture. We can then use the HOL prover as-is, without making any changes to its source code, and use it to solve problems in our target logic.

4.1 Shallow Semantical Embedding

To define an SSE for target logic PAL, we lift the type of propositions in order to explicitly encode their dependency on possible worlds; this is analogous to prior work [7, 8]. In order to capture the model-changing behavior of PAL, we additionally introduce world domains (sets of worlds) as parameters/arguments in the encoding. The rationale thereby is to suitably constrain, and recursively pass-on, these domains after each model changing action.

PAL formulas are thus identified in our semantical embedding with certain HOL terms (predicates) of type $(i \rightarrow o) \rightarrow i \rightarrow o$. They can be applied to terms of type $i \rightarrow o$, which are assumed to denote evaluation domains, and subsequently to terms of type i , which are assumed to denote possible worlds. That is, the HOL type i is identified with a (non-empty) set of worlds, and the type $i \rightarrow o$, abbreviated by σ , is identified with a set of sets of worlds, i.e., a set of evaluation domains.

Type $(i \rightarrow o) \rightarrow i \rightarrow o$ is abbreviated as τ , α is an abbreviation for $i \rightarrow i \rightarrow o$, the type of accessibility relations between worlds, and ϱ abbreviates $\alpha \rightarrow o$, the type of sets of accessibility relations. Table 1 provides an overview of these abbreviations.

For each propositional symbol p^i of PAL, the associated HOL signature is assumed to contain a corresponding constant symbol p_σ^i , which is (rigidly) denoting the set of all those worlds in which p^i holds. We call the p_σ^i *σ -type-lifted propositions*. Moreover, for $k = 1, \dots, |\mathcal{A}|$ the HOL signature is assumed to contain the constant symbols $r_\alpha^1, \dots, r_\alpha^{|\mathcal{A}|}$. Without loss of generality, we assume that besides those constants symbols and the primitive logical connectives of HOL, no other constant symbols are given in the signature of HOL.

⁵The logic can be extended to include mutual knowledge (E_G) and distributed knowledge (D_G) if needed, and we also cover their embeddings in the remainder.

⁶The SSE technique is not be confused with higher-order abstract syntax [30], or with other forms of deep embeddings.

meaning	type	abbreviations
HOL formula	o	
worlds	i	
evaluation domains	$i \rightarrow o$	σ
PAL formulas	$(i \rightarrow o) \rightarrow i \rightarrow o$	τ
accessibility relations	$i \rightarrow i \rightarrow o$	α
set of relations	$\alpha \rightarrow o$	ϱ

Table 1: Type abbreviations as used in the remainder

Definition 4.1 (Mapping of PAL formulas φ into HOL terms $[\varphi]$). The mapping $[\cdot]$ translates a formula φ of PAL into a term $[\varphi]$ of HOL of type τ . The mapping is defined recursively:

$$\begin{aligned}
[p^j] &= ({}^A p_\sigma^j)_\tau \\
[\neg\varphi] &= \neg_{\tau \rightarrow \tau} [\varphi] \\
[\varphi \vee \psi] &= \vee_{\tau \rightarrow \tau \rightarrow \tau} [\varphi] [\psi] \\
[K r^k \varphi] &= K_{\alpha \rightarrow \tau \rightarrow \tau} r_\alpha^k [\varphi] \\
[![\varphi]\psi] &= [! \cdot]_{\cdot \tau \rightarrow \tau \rightarrow \tau} [\varphi] [\psi] \\
[\mathcal{C}_G(\varphi|\psi)] &= \mathcal{C}(\cdot|\cdot)_{\varrho \rightarrow \tau \rightarrow \tau \rightarrow \tau} G_\varrho [\varphi] [\psi]
\end{aligned}$$

A recursive definition is actually not needed in practice. By inspecting the above equations, it becomes clear that only the abbreviations for the logical connectives of PAL are required in combination with a type-lifting for the propositional symbols; cf. the non-recursive equations of the actual encoding in Lines 28-36 and 46-47 of Fig. 1 in Appendix A.

Operator ${}^A(\cdot)$, which evaluates atomic formulas, is defined as follows:

$${}^A_{\cdot \sigma \rightarrow \tau} = \lambda A_\sigma \lambda D_\sigma \lambda X_i (D X \wedge A X)$$

As a first argument, it accepts a σ -type-lifted proposition A_σ , which are rigidly interpreted. As a second argument, it accepts an evaluation domain D_σ , that is, an arbitrary subset of the domain associated with type σ . And as a third argument, it accepts a current world X_i . It then checks whether (i) the current world is a member of evaluation domain D_σ and (ii) whether the σ -type-lifted proposition A_σ holds in the current world.

The other logical connectives of PAL, except for $[! \cdot]_{\cdot \tau \rightarrow \tau \rightarrow \tau}$, are now defined in a way so that they simply pass on the evaluation domains as parameters to the atomic-level. Only $[! \cdot]_{\cdot \tau \rightarrow \tau \rightarrow \tau}$ is modifying, in fact, constraining, the evaluation domain it passes on, and it does this in the expected way (cf. Def. 3.3):

$$\begin{aligned}
\neg_{\tau \rightarrow \tau} &= \lambda A_\tau \lambda D_\sigma \lambda X_i \neg (A D X) \\
\vee_{\tau \rightarrow \tau \rightarrow \tau} &= \lambda A_\tau \lambda B_\tau \lambda D_\sigma \lambda X_i (A D X \vee B D X) \\
K_{\alpha \rightarrow \tau \rightarrow \tau} &= \lambda R_\alpha \lambda A_\tau \lambda D_\sigma \lambda X_i \forall Y_i ((D Y \wedge R X Y) \longrightarrow A D Y) \\
[! \cdot]_{\cdot \tau \rightarrow \tau \rightarrow \tau} &= \lambda A_\tau \lambda B_\tau \lambda D_\sigma \lambda X_i ((A D X) \longrightarrow (B (\lambda Y_i (D Y \wedge A D Y)) X))
\end{aligned}$$

To model $\mathcal{C}(\cdot|\cdot)_{\varrho \rightarrow \tau \rightarrow \tau \rightarrow \tau}$ we reuse the following operations on relations; cf. [7, 8].

$$\begin{aligned}
\text{transitive}_{\alpha \rightarrow o} &= \lambda R_\alpha \forall X_i \forall Y_i \forall Z_i ((R X Y \wedge R Y Z) \longrightarrow R X Z) \\
\text{intersection}_{\alpha \rightarrow \alpha \rightarrow \alpha} &= \lambda R_\alpha \lambda Q_\alpha \lambda X_i \lambda Y_i (R X Y \wedge Q X Y) \\
\text{sub}_{\alpha \rightarrow \alpha \rightarrow o} &= \lambda R_\alpha \lambda Q_\alpha \forall X_i \forall Y_i (R X Y \longrightarrow Q X Y)
\end{aligned}$$

The transitive closure tc of a relation can be elegantly defined in HOL as:

$$\text{tc}_{\alpha \rightarrow \alpha} = \lambda R_{\alpha} \lambda X_i \lambda Y_i \forall Q_{\alpha} (\text{transitive } Q \longrightarrow (\text{sub } R \ Q \longrightarrow Q \ X \ Y))$$

Additionally, we introduce higher-order definitions for the union and intersection of an arbitrary set of relations.

$$\begin{aligned} \text{big_union}_{\rho \rightarrow \alpha} &= \lambda G_{\rho} \lambda X_i \lambda Y_i \exists R_{\alpha} (G \ R \ \wedge \ R \ X \ Y) \\ \text{big_intersection}_{\rho \rightarrow \alpha} &= \lambda G_{\rho} \lambda X_i \lambda Y_i \forall R_{\alpha} (G \ R \longrightarrow R \ X \ Y) \end{aligned}$$

EVR , being applied to a set of accessibility relations G , returns a relation denoting the mutual knowledge of the set/group of agents G . EVR is subsequently used in the definition of relativized common knowledge to describe the mutual knowledge of multiple agents. Analogously, we introduce distributed knowledge DIS as the intersection of this set. In the case of three agents, we introduce a concrete set of relations of R consisting of r^1 , r^2 and r^3 of type α .

$$\begin{aligned} \mathbf{G}_{\rho} &= \lambda R_{\alpha} (R = r^1 \ \vee \ R = r^2 \ \vee \ R = r^3) \\ \text{EVR}_{\rho} &= \lambda G_{\rho} (\text{big_union } G) \\ \text{DIS}_{\rho} &= \lambda G_{\rho} (\text{big_intersection } G) \end{aligned}$$

One could also use a less verbose way of defining \mathbf{G} and leverage Isabelle's set notation (and associated theory): $\mathbf{G}_{\rho} = \{r^1, r^2, r^3\}$. However, we here deliberately choose a direct encoding in HOL in order to introduce as little unnecessary dependencies as possible.

The operator $\mathcal{C}(\cdot)_{\rho \rightarrow \tau \rightarrow \tau \rightarrow \tau}$ thus abbreviates the following HOL term:

$$\begin{aligned} \mathcal{C}(\cdot)_{\rho \rightarrow \tau \rightarrow \tau \rightarrow \tau} &= \lambda G_{\rho} \lambda A_{\tau} \lambda B_{\tau} \lambda D_{\sigma} \lambda X_i \forall Y_i \\ &(\text{tc } (\text{intersection } (\text{EVR } G) (\lambda U_i \lambda V_i (D \ V \ \wedge \ A \ D \ V))) \ X \ Y \\ &\longrightarrow B \ D \ Y) \end{aligned}$$

Analyzing the truth of a PAL formula φ , represented by the HOL term $[\varphi]$, in a particular domain d , represented by the term D_{σ} , and a world s , represented by the term S_i , corresponds to evaluating the application $([\varphi] \ D_{\sigma} \ S_i)$. We can verify whether S denotes a world in D by checking if $(D \ S)$ is true. If that is the case, we evaluate φ for this domain and world.

A formula φ is thus generally valid if and only if for all D_{σ} and all S_i we have $D \ S \rightarrow [\varphi] \ D \ S$. The validity function, therefore, is defined as follows:

$$\text{vld}_{\tau \rightarrow \circ} = \lambda A_{\tau} \forall D_{\sigma} \forall S_i (D \ S \longrightarrow A \ D \ S).$$

The necessity to quantify over all possible domains in this definition will be further illustrated below.

4.2 Encoding into Isabelle/HOL

What follows is a description of the concrete encoding of the presented SSE of PAL in HOL within the higher-order proof assistant Isabelle/HOL (cf. Fig. 1 in App. A).⁷

⁷The full sources of our encoding can be found at <http://logikey.org> in subfolder `Public-Announcement-Logic`.

All necessary types can be modeled in a straightforward way. We declare `i` to denote possible worlds and then introduce type aliases for σ , τ , α and ϱ . Type `bool` represents (the bivalent set of) truth values introduced as o before.

```

typedec1 i                (* Type of possible worlds *)
type_synonym  $\sigma$  = "i $\Rightarrow$ bool" (* Type of world domains *)
type_synonym  $\tau$  = " $\sigma\Rightarrow i\Rightarrow$ bool" (* Type of world depended formulas *)
type_synonym  $\alpha$  = "i $\Rightarrow i\Rightarrow$ bool" (* Type of accessibility relations *)
type_synonym  $\varrho$  = " $\alpha\Rightarrow$ bool" (* Type of group of agents *)

```

The agents are declared as accessibility relations, and the group of agents can then be denoted by a predicate of type ϱ . In order to obtain $S5$ (KT45) properties, we declare respective conditions on the accessibility relations in the group of agents A . Various Isabelle/HOL encodings from [7, 8] are reused here, including the encoding of transitive closure.

```

abbreviation S5Agent::" $\alpha\Rightarrow$ bool"
  where "S5Agent i  $\equiv$  reflexive i  $\wedge$  transitive i  $\wedge$  euclidean i"
abbreviation S5Agents::" $\varrho\Rightarrow$ bool"
  where "S5Agents A  $\equiv$   $\forall i. (A i \longrightarrow S5Agent i)"$ 

```

Each of the lifted unary and binary connectives of PAL accepts arguments of type τ , i.e. lifted PAL formulas, and returns such a lifted PAL formula.

A special case, as discussed before, is the new operator for atomic propositions $A(\cdot)$. When evaluating σ -type lifted atomic propositions p we need to check if p is true in the given world w , but we also need to check whether the given world w is still part of our evaluation domain W that has been recursively passed on. Operator $A(\cdot)$ is thus of type " $\sigma \Rightarrow \tau$ ". The need and effect of this distinction are addressed again in §6.2, where we present formulas that are only valid when restricted to the atomic case.

```

abbreviation patom::" $\sigma \Rightarrow \tau$ " (" $A\_$ ")
  where " $A_p \equiv \lambda W w. W w \wedge p w$ "
abbreviation ptop::" $\tau$ " (" $\top$ ")
  where " $\top \equiv \lambda W w. True$ "
abbreviation pneg::" $\tau \Rightarrow \tau$ " (" $\neg$ ")
  where " $\neg \varphi \equiv \lambda W w. \neg(\varphi W w)$ "
abbreviation pand::" $\tau \Rightarrow \tau \Rightarrow \tau$ " (" $\wedge$ ")
  where " $\varphi \wedge \psi \equiv \lambda W w. (\varphi W w) \wedge (\psi W w)$ "
abbreviation por::" $\tau \Rightarrow \tau \Rightarrow \tau$ " (" $\vee$ ")
  where " $\varphi \vee \psi \equiv \lambda W w. (\varphi W w) \vee (\psi W w)$ "
abbreviation pimp::" $\tau \Rightarrow \tau \Rightarrow \tau$ " (" $\longrightarrow$ ")
  where " $\varphi \longrightarrow \psi \equiv \lambda W w. (\varphi W w) \longrightarrow (\psi W w)$ "
abbreviation pequ::" $\tau \Rightarrow \tau \Rightarrow \tau$ " (" $\longleftrightarrow$ ")
  where " $\varphi \longleftrightarrow \psi \equiv \lambda W w. (\varphi W w) \longleftrightarrow (\psi W w)$ "

```

In the definition of the knowledge operator K , we have to make sure to add a domain check in the implication.

```

abbreviation pknow::" $\alpha \Rightarrow \tau \Rightarrow \tau$ " (" $K\_$ ")
  where " $K r \varphi \equiv \lambda W w. \forall v. (W v \wedge r w v) \longrightarrow (\varphi W v)$ "

```

Some additional abbreviations are introduced to improve readability. One is a more concise way to state knowledge, achieved by abbreviating `pknow` even further.

abbreviation agtknows::" $\alpha \Rightarrow \tau \Rightarrow \tau$ " ("K_ _")
 where " $K_r \varphi \equiv K \ r \ \varphi$ "

Additionally, operators for mutual and distributed knowledge are presented. To achieve this we introduce two additional encodings on relations. The union and intersection operators on a set of relations.

definition big_union_rel::" $\varrho \Rightarrow \alpha$ "
 where " $\text{big_union_rel } X \equiv \lambda u \ v. \exists R. (X \ R) \wedge (R \ u \ v)$ "
 definition big_intersection_rel::" $\varrho \Rightarrow \alpha$ "
 where " $\text{big_intersection_rel } X \equiv \lambda u \ v. \forall R. (X \ R) \longrightarrow (R \ u \ v)$ "

These encodings can then be applied to concrete groups of agents G of type ϱ , to define the relations (EVR G) and (DIS G).

abbreviation EVR::" $\varrho \Rightarrow \alpha$ " where " $\text{EVR } G \equiv \text{big_union_rel } G$ "
 abbreviation DIS::" $\varrho \Rightarrow \alpha$ " where " $\text{DIS } G \equiv \text{big_intersection_rel } G$ "

Now it is straightforward to abbreviate mutual and distributed knowledge.

abbreviation evrknows::" $\varrho \Rightarrow \tau \Rightarrow \tau$ " ("E_ _")
 where " $E_G \varphi \equiv K \ (\text{EVR } G) \ \varphi$ "
 abbreviation disknows::" $\varrho \Rightarrow \tau \Rightarrow \tau$ " ("D_ _")
 where " $D_G \varphi \equiv K \ (\text{DIS } G) \ \varphi$ "

We finally see the change of the evaluation domain in action, when introducing the public announcement operator. We already inserted domain checks in the definition of the operators K and $A(\cdot)$. Now, we need to constrain the domain after each public announcement. So far the evaluation domain, modeled by W , got passed on through all lifted operators without any change. In the public announcement operator, however, we modify the evaluation domain W into $(\lambda z. W \ z \ \wedge \ \varphi \ W \ z)$ (i.e., the set of all worlds z in W , such that φ holds for W and z), which is then recursively passed on. The public announcement operator is thus defined as:

abbreviation ppal :: " $\tau \Rightarrow \tau \Rightarrow \tau$ " ("[!_]_")
 where " $[!\varphi]\psi \equiv \lambda W \ w. (\varphi \ W \ w) \longrightarrow (\psi \ (\lambda z. W \ z \ \wedge \ \varphi \ W \ z) \ w)$ "

The following embedding of relativized common knowledge is a straightforward encoding of the semantic properties and definitions as proposed in Def. 5.

abbreviation prck :: " $\varrho \Rightarrow \tau \Rightarrow \tau \Rightarrow \tau$ " ("C_(|_|_)")
 where " $C_G(\varphi|\psi) \equiv \lambda W \ w. \forall v. (\text{tc} \ (\text{intersection_rel} \ (\text{EVR } G) \ (\lambda u \ v. W \ v \ \wedge \ \varphi \ W \ w)) \ w \ v) \longrightarrow (\psi \ W \ v)$ "

As described earlier we can abbreviate ordinary common knowledge as $C_G(\top|\varphi)$:

abbreviation pcmn :: " $\varrho \Rightarrow \tau \Rightarrow \tau$ " ("C_ _")
 where " $C_G \ \varphi \equiv C_G(\top|\varphi)$ "

Finally, an embedding for the notion of validity is needed. Generally, for a type-lifted formula φ to be valid, the application of φ has to hold true for all worlds w . In the context of PAL the evaluation domains also have to be incorporated in the definition. Originally, we were tempted to define PAL validity in such a way that we start with a “full evaluation domain”, a domain that evaluates to **True** for all possible worlds and gets restricted, whenever necessary after an announcement. Such a validity definition would look like this:

```

abbreviation tvalid::" $\tau \Rightarrow \text{bool}$ " (" $\_ ]^T$ ")
  where " $\_ ]^T \equiv \forall w. \varphi (\lambda x. \text{True}) w$ "

```

But this leads to undesired behavior, which we can easily see when using our reasoning tools to study e.g. the validity *announcement necessitation*: *from φ , infer $[!\psi]\varphi$* . If we check for a counterexample in Isabelle/HOL, the model finder Nitpick [17] reports the following:

```

lemma necessitation: assumes " $[\varphi]^T$ " shows " $[!\psi]\varphi]^T$ " nitpick oops

```

Nitpick found a counterexample for card i = 2:

Free variables:

```

 $\varphi = (\lambda x. \_)$ 
  (( $\lambda x. \_$ )( $i_1 := \text{True}, i_2 := \text{True}$ ),  $i_1$ ) := True,
  (( $\lambda x. \_$ )( $i_1 := \text{True}, i_2 := \text{True}$ ),  $i_2$ ) := True,
  (( $\lambda x. \_$ )( $i_1 := \text{True}, i_2 := \text{False}$ ),  $i_1$ ) := False,
  (( $\lambda x. \_$ )( $i_1 := \text{True}, i_2 := \text{False}$ ),  $i_2$ ) := False,
  (( $\lambda x. \_$ )( $i_1 := \text{False}, i_2 := \text{True}$ ),  $i_1$ ) := False,
  (( $\lambda x. \_$ )( $i_1 := \text{False}, i_2 := \text{True}$ ),  $i_2$ ) := False,
  (( $\lambda x. \_$ )( $i_1 := \text{False}, i_2 := \text{False}$ ),  $i_1$ ) := False,
  (( $\lambda x. \_$ )( $i_1 := \text{False}, i_2 := \text{False}$ ),  $i_2$ ) := False)

```

```

 $\psi = (\lambda x. \_)$ 
  (( $\lambda x. \_$ )( $i_1 := \text{True}, i_2 := \text{True}$ ),  $i_1$ ) := False,
  (( $\lambda x. \_$ )( $i_1 := \text{True}, i_2 := \text{True}$ ),  $i_2$ ) := True,
  (( $\lambda x. \_$ )( $i_1 := \text{True}, i_2 := \text{False}$ ),  $i_1$ ) := False,
  (( $\lambda x. \_$ )( $i_1 := \text{True}, i_2 := \text{False}$ ),  $i_2$ ) := False,
  (( $\lambda x. \_$ )( $i_1 := \text{False}, i_2 := \text{True}$ ),  $i_1$ ) := False,
  (( $\lambda x. \_$ )( $i_1 := \text{False}, i_2 := \text{True}$ ),  $i_2$ ) := False,
  (( $\lambda x. \_$ )( $i_1 := \text{False}, i_2 := \text{False}$ ),  $i_1$ ) := False,
  (( $\lambda x. \_$ )( $i_1 := \text{False}, i_2 := \text{False}$ ),  $i_2$ ) := False)

```

Skolem constant:

```
w =  $i_2$ 
```

Here, for world i_2 the term $[\varphi]^T$ is true. Because we evaluate

```
(( $\lambda x. \_$ )( $i_1 := \text{True}, i_2 := \text{True}$ ),  $i_2$ ) := True
```

Yet, the evaluation of the term $[!\psi]\varphi]^T$ is false. This results from announcing ψ , where all worlds get discarded in which ψ does not hold. Such a world is i_1 , in which ψ does not hold initially (for our notion of validity). However, in world i_2 the formula ψ holds. We can see this if we have a look at the first two lines as presented above for ψ :

```
(( $\lambda x. \_$ )( $i_1 := \text{True}, i_2 := \text{True}$ ),  $i_1$ ) := False
(( $\lambda x. \_$ )( $i_1 := \text{True}, i_2 := \text{True}$ ),  $i_2$ ) := True
```

Thus, ultimately the term $[!\psi]\varphi]^T$ evaluates in the given context, where i_1 has been discarded, as follows:

```
(( $\lambda x. \_$ )( $i_1 := \text{False}, i_2 := \text{True}$ ),  $i_2$ ) := False
```

As a consequence, the validity function is defined in such a way that it checks validity not only for all worlds, but for all domains and worlds. Otherwise, the observed but undesired countermodel to necessitation may occur.

```

abbreviation pvalid :: " $\tau \Rightarrow \text{bool}$ " ("[_]")
  where "[_]"  $\equiv \forall W. \forall w. W \ w \longrightarrow \varphi \ W \ w$ 

```

The definitions as introduced in this section are intended to be hidden from the user, who can construct formulas directly in PAL syntax. The unfolding of these definitions is then handled automatically by Isabelle/HOL.

The SSE approach in LOGIKEY also supports the encoding and inspection of concrete models. For example, let $W = \{w1, w2, w3\}$ and let p be true at $w1$ and $w2$, but false at $w3$. Let the relations (given as partitions, i.e. lists of equivalence classes) be $R_a = [[w1, w2], [w3]]$ and $R_b = [[w1], [w2, w3]]$. Then we have $M, w1 \models p \wedge K_a p \wedge K_b p \wedge \neg K_a K_b p$. In Isabelle/HOL we can encode this as follows:

```

(* Concrete models can be defined and studied *)
lemma assumes: "W = ( $\lambda x. x = w1 \vee x = w2 \vee x = w3$ )"
  "w1  $\neq$  w2" "w1  $\neq$  w3" "w2  $\neq$  w3"
  "p W w1" "p W w2" " $\neg$ (p W w3)"
  "a w1 w1" "a w1 w2" "a w2 w1"
  "a w2 w2" " $\neg$ (a w1 w3)" " $\neg$ (a w3 w1)"
  " $\neg$ (a w2 w3)" " $\neg$ (a w3 w2)" "a w3 w3"
  "b w1 w1" " $\neg$ (b w1 w2)" " $\neg$ (b w2 w1)"
  "b w2 w2" " $\neg$ (b w1 w3)" " $\neg$ (b w3 w1)"
  "b w2 w3" "b w3 w2" "b w3 w3"
  shows " $((p \wedge (K_a \ p) \wedge (K_b \ p)) \wedge \neg(K_a \ (K_b \ p))) \ W \ w1$ "
unfolding Defs
  nitpick[satisfy, atoms=w1 w2 w3] (* model *)
  using assms(1) assms(5) assms(6) assms(7)
  assms(9) assms(12) assms(21) assms(23) by blast (* proof *)

```

Nitpick generates a model that satisfies these constraints, and the provers in Isabelle/HOL subsequently prove the validity of this claim as expected. We could of course deepen such experiments here and specify and inspect further models in full detail. This is however left for further work. Further work also includes experimentation with Isabelle/HOL as an educational tool, including e.g. the exploration and study of PAL models in classroom. Generally, the aspect of (counter-)model finding in the SSE approach deserves more attention in future work.

5 Soundness of the Embedding

To show that our embedding is sound, we exploit the following mapping of Kripke frames into Henkin models.

Definition 5.1 (Henkin Model $\mathcal{H}^{\mathcal{M}}$ for PAL model \mathcal{M}). Let \mathcal{A} be a group of agents. For any PAL model $\mathcal{M} = \langle W, \{R_i\}_{i \in \mathcal{A}}, V \rangle$, we define a corresponding Henkin model $\mathcal{H}^{\mathcal{M}}$. Thus, let a PAL model $\mathcal{M} = \langle W, \{R_i\}_{i \in \mathcal{A}}, V \rangle$ be given. Moreover, assume that $p^1, \dots, p^m \in \mathcal{P}$, for $m \geq 1$, are the only propositional symbols of PAL. Remember that our embedding requires the corresponding signature in HOL to provide constant symbols p_σ^j such that $[p^j] = p_\sigma^j$ for $j = 1, \dots, m$. Moreover, for each $R_i \in \mathcal{M}$ we require a corresponding constant symbol r_α^i in the signature of HOL (this is similar to what we do for the $p^i \in P$). A Henkin model $\mathcal{H}^{\mathcal{M}} = \langle \{\mathcal{D}_\alpha\}_{\alpha \in T}, I \rangle$ for \mathcal{M} is now defined as follows: \mathcal{D}_i is chosen as the set of possible worlds W ; all other sets $\mathcal{D}_{\alpha \rightarrow \beta}$ are chosen

as (not necessary full) sets of functions from \mathcal{D}_α to \mathcal{D}_β . For all $\mathcal{D}_{\alpha \rightarrow \beta}$ the rule that every term $t_{\alpha \rightarrow \beta}$ must have a denotation in $\mathcal{D}_{\alpha \rightarrow \beta}$ must be obeyed (Denotatpflicht). In particular, it is required that \mathcal{D}_σ and \mathcal{D}_α contain the elements $I p_\sigma^j$ and $I r_\alpha^i$, respectively. The interpretation function I of $\mathcal{H}^\mathcal{M}$ is defined as follows:

1. For $j = 1, \dots, m$, $I p_\sigma^j \in \mathcal{D}_\sigma$ chosen s.t. $I p_\sigma^j(d, s) = T$ iff $s \in V(p^j)$ in \mathcal{M} .
2. For $k = 1, \dots, |\mathcal{A}|$, $I r_\alpha^i \in \mathcal{D}_\alpha$ chosen s.t. $I r_\alpha^i(s, u) = T$ iff $u \in R_i(s)$ in \mathcal{M} .
3. For the logical connectives \neg, \vee, Π and $=$ of HOL the interpretation function I is defined as usual.

None of these choices is in conflict with the necessary requirements of a Henkin model.

Lemma 5.1. Let $\mathcal{H}^\mathcal{M}$ be a Henkin model for a PAL model $\mathcal{M} = \langle W, \{R_i\}_{i \in \mathcal{A}}, V \rangle$. For all PAL formulas δ , arbitrary variable assignments g , sets of worlds $d = W$ (the *evaluation domains*) and worlds s it holds:

$$\mathcal{M}, s \models \delta \text{ if and only if } \llbracket [\delta] D_\sigma S_i \rrbracket^{\mathcal{H}^\mathcal{M}, g[d/D_\sigma][s/S_i]} = T$$

Proof. We start with the case where δ is p^j . We have:

$$\begin{aligned} & \llbracket [p^j] D S \rrbracket^{\mathcal{H}^\mathcal{M}, g[d/D_\sigma][s/S_i]} = T \\ \Leftrightarrow & \llbracket [{}^A(p_\sigma^j)]_\tau D S \rrbracket^{\mathcal{H}^\mathcal{M}, g[d/D_\sigma][s/S_i]} = T \\ \Leftrightarrow & \llbracket D S \wedge p_\sigma^j S \rrbracket^{\mathcal{H}^\mathcal{M}, g[d/D_\sigma][s/S_i]} = T \\ \Leftrightarrow & s \in d = W \text{ and } I p_\sigma^j(s) = T && \text{(by definition of } \mathcal{H}^\mathcal{M}) \\ \Leftrightarrow & M, s \models p^j \end{aligned}$$

Induction hypothesis: For sentences δ' structurally smaller than δ we have: For all assignments g , domains d and worlds s , $\llbracket [\delta'] D S \rrbracket^{\mathcal{H}^\mathcal{M}, g[d/D_\sigma][s/S_i]} = T$ if and only if $M, s \models \delta'$.

We consider each inductive case in turn:

$$\begin{aligned} \delta &= \neg \varphi \\ \Leftrightarrow & \llbracket [\neg \varphi] D S \rrbracket^{\mathcal{H}^\mathcal{M}, g[d/D_\sigma][s/S_i]} = T \\ \Leftrightarrow & \llbracket [(\neg_{\tau \rightarrow \tau} \varphi)] D S \rrbracket^{\mathcal{H}^\mathcal{M}, g[d/D_\sigma][s/S_i]} = T \\ \Leftrightarrow & \llbracket [\neg([\varphi] D S)] \rrbracket^{\mathcal{H}^\mathcal{M}, g[d/D_\sigma][s/S_i]} = T && \text{(since } (\neg_{\tau \rightarrow \tau}([\varphi]) D S) =_{\beta\eta} \neg([\varphi] D S)) \\ \Leftrightarrow & \llbracket [\varphi] D S \rrbracket^{\mathcal{H}^\mathcal{M}, g[d/D_\sigma][s/S_i]} = F \\ \Leftrightarrow & M, s \not\models \varphi && \text{(by induction hypothesis)} \\ \Leftrightarrow & M, s \models \neg \varphi \end{aligned}$$

$$\begin{aligned} \delta &= \varphi \vee \psi \\ \Leftrightarrow & \llbracket [\varphi \vee \psi] D S \rrbracket^{\mathcal{H}^\mathcal{M}, g[d/D_\sigma][s/S_i]} = T \\ \Leftrightarrow & \llbracket [([\varphi] \vee_{\tau \rightarrow \tau} [\psi])] D S \rrbracket^{\mathcal{H}^\mathcal{M}, g[d/D_\sigma][s/S_i]} = T \\ & \hspace{15em} \text{(since } ([\varphi] \vee_{\tau \rightarrow \tau} [\psi]) D S =_{\beta\eta} ([\varphi] D S) \vee ([\psi] D S)) \\ \Leftrightarrow & \llbracket ([\varphi] D S) \vee ([\psi] D S) \rrbracket^{\mathcal{H}^\mathcal{M}, g[d/D_\sigma][s/S_i]} = T \\ \Leftrightarrow & \llbracket [\varphi] D S \rrbracket^{\mathcal{H}^\mathcal{M}, g[d/D_\sigma][s/S_i]} = T \text{ or } \llbracket [\psi] D S \rrbracket^{\mathcal{H}^\mathcal{M}, g[d/D_\sigma][s/S_i]} = T \\ \Leftrightarrow & M, s \models \varphi \text{ or } M, s \models \psi && \text{(by induction hypothesis)} \\ \Leftrightarrow & M, s \models \varphi \vee \psi \end{aligned}$$

$$\begin{aligned} \delta &= K r^i \varphi \\ \Leftrightarrow & \llbracket [K r^i \varphi] D S \rrbracket^{\mathcal{H}^\mathcal{M}, g[d/D_\sigma][s/S_i]} = T \end{aligned}$$

$\Leftrightarrow \llbracket K_{\alpha \rightarrow \tau \rightarrow \tau} r^i [\varphi] D S \rrbracket^{\mathcal{H}^{\mathcal{M}}, g[d/D_\sigma][s/S_i]} = T$
 $\Leftrightarrow \llbracket \forall Y_i (\neg(D Y \wedge r^i S Y) \vee [\varphi] D Y) \rrbracket^{\mathcal{H}^{\mathcal{M}}, g[d/D_\sigma][s/S_i]} = T$
 \Leftrightarrow For all $a \in \mathcal{D}_i$ we have $\llbracket (\neg(D Y \wedge r^i S Y) \vee [\varphi] D Y) \rrbracket^{\mathcal{H}^{\mathcal{M}}, g[d/D_\sigma][s/S_i][a/Y_i]} = T$
 \Leftrightarrow For all $a \in \mathcal{D}_i$ we have $\llbracket \neg(D Y \wedge r^i S Y) \rrbracket^{\mathcal{H}^{\mathcal{M}}, g[d/D_\sigma][s/S_i][a/Y_i]} = T$ or
 $\llbracket [\varphi] D Y \rrbracket^{\mathcal{H}^{\mathcal{M}}, g[d/D_\sigma][s/S_i][a/Y_i]} = T$
 \Leftrightarrow For all $a \in \mathcal{D}_i$ we have $\llbracket \neg D Y \vee \neg r^i S Y \rrbracket^{\mathcal{H}^{\mathcal{M}}, g[d/D_\sigma][s/S_i][a/Y_i]} = T$ or
 $\llbracket [\varphi] D Y \rrbracket^{\mathcal{H}^{\mathcal{M}}, g[d/D_\sigma][s/S_i][a/Y_i]} = T$ ($S \notin \text{free}([\varphi])$)
 \Leftrightarrow For all $a \in \mathcal{D}_i$ we have $\llbracket \neg D Y \rrbracket^{\mathcal{H}^{\mathcal{M}}, g[d/D_\sigma][s/S_i][a/Y_i]} = T$ or
 $\llbracket \neg r^i S Y \rrbracket^{\mathcal{H}^{\mathcal{M}}, g[d/D_\sigma][s/S_i][a/Y_i]} = T$ or $\llbracket [\varphi] D Y \rrbracket^{\mathcal{H}^{\mathcal{M}}, g[d/D_\sigma][s/S_i][a/Y_i]} = T$
 \Leftrightarrow For all $a \in \mathcal{D}_i$ we have $\llbracket D Y \rrbracket^{\mathcal{H}^{\mathcal{M}}, g[d/D_\sigma][s/S_i][a/Y_i]} = F$ or
 $\llbracket r^i S Y \rrbracket^{\mathcal{H}^{\mathcal{M}}, g[d/D_\sigma][s/S_i][a/Y_i]} = F$ or $\llbracket [\varphi] D Y \rrbracket^{\mathcal{H}^{\mathcal{M}}, g[d/D_\sigma][s/S_i][a/Y_i]} = T$ (by ind. hyp.)
 \Leftrightarrow For all $a \in \mathcal{D}_i$ we have $(a \notin d$ or $a \notin r^i(s))$ or $M, a \models \varphi$
 \Leftrightarrow For all $a \in \mathcal{D}_i$ we have $a \notin (d \cap r^i(s))$ or $M, a \models \varphi$
 $\Leftrightarrow M, s \models K r^i \varphi$

$\delta = \llbracket !\varphi \rrbracket \psi$
 $\Leftrightarrow \llbracket \llbracket !\varphi \rrbracket \psi \rrbracket D S \rrbracket^{\mathcal{H}^{\mathcal{M}}, g[d/D_\sigma][s/S_i]} = T$
 $\Leftrightarrow \llbracket (\llbracket ! \cdot \rrbracket \cdot)_{\tau \rightarrow \tau \rightarrow \tau} [\varphi] [\psi] \rrbracket D S \rrbracket^{\mathcal{H}^{\mathcal{M}}, g[d/D_\sigma][s/S_i]} = T$
 $\Leftrightarrow \llbracket (\neg[\varphi] D S) \vee (\llbracket !\psi \rrbracket (\lambda Y_i D Y \wedge [\varphi] D Y) S) \rrbracket^{\mathcal{H}^{\mathcal{M}}, g[d/D_\sigma][s/S_i]} = T$
 $\Leftrightarrow \llbracket (\neg[\varphi] D S) \rrbracket^{\mathcal{H}^{\mathcal{M}}, g[d/D_\sigma][s/S_i]} = T$ or $\llbracket (\llbracket !\psi \rrbracket (\lambda Y_i D Y \wedge [\varphi] D Y) S) \rrbracket^{\mathcal{H}^{\mathcal{M}}, g[d/D_\sigma][s/S_i]} = T$
 $\Leftrightarrow \llbracket ([\varphi] D S) \rrbracket^{\mathcal{H}^{\mathcal{M}}, g[d/D_\sigma][s/S_i]} = F$ or $\llbracket (\llbracket !\psi \rrbracket (\lambda Y_i D Y \wedge [\varphi] D Y) S) \rrbracket^{\mathcal{H}^{\mathcal{M}}, g[d/D_\sigma][s/S_i]} = T$
 $\Leftrightarrow M, s \not\models \varphi$ or $\llbracket (\llbracket !\psi \rrbracket (\lambda Y_i D Y \wedge [\varphi] D Y) S) \rrbracket^{\mathcal{H}^{\mathcal{M}}, g[d/D_\sigma][s/S_i]} = T$ (by ind. hyp.)
 $\Leftrightarrow M, s \not\models \varphi$ or $M^{! \varphi}, s \models \psi$ (**Justification**)
 $\Leftrightarrow M, s \models \llbracket !\varphi \rrbracket \psi$

Justification:

From the induction hypothesis follows that $\llbracket \llbracket !\psi \rrbracket D S \rrbracket^{\mathcal{H}^{\mathcal{M}}, g[d/D_\sigma][s/S_i]} = T$ if and only if $\mathcal{M}, s \models \psi$. In order to see how $\llbracket \llbracket !\psi \rrbracket (\lambda Y_i D Y \wedge [\varphi] D Y) S \rrbracket^{\mathcal{H}^{\mathcal{M}}, g[d/D_\sigma][s/S_i]} = T$ and $M^{! \varphi}, s \models \psi$ relate, we remind ourselves of the definition of the model \mathcal{M} after φ is publicly announced: $\mathcal{M}^{! \varphi} = \langle W^{! \varphi}, \{R_i^{! \varphi}\}_{i \in \mathcal{A}}, V^{! \varphi} \rangle$ where $W^{! \varphi} = \{w \in W \mid \mathcal{M}, w \models \varphi\}$, $R_i^{! \varphi} = R_i \cap (W^{! \varphi} \times W^{! \varphi})$ for all $i \in \mathcal{A}$, and $V^{! \varphi}(p) = V(p) \cap W^{! \varphi}$ for all $p \in \mathcal{P}$. For the embedding this means that $W^{! \varphi}$ retains only worlds in which φ is true, while the arrows/relations between worlds remain the same and get evaluated in the embedding as explained. By encoding the updated domain as $(\lambda Y_i D Y \wedge [\varphi] D Y)$, denoting $\{w \in W \mid \mathcal{M}, w \models \varphi\}$ in the given context, we ultimately restrict ourselves to worlds in $W^{! \varphi}$. Relations indirectly get restricted to this new domain ($R_i^{! \varphi} = R_i \cap (W^{! \varphi} \times W^{! \varphi})$), due to the recursively conducted domain checks (see e.g. the definitions of the public announcement operator), and an analogous argument applies for the evaluation of atomic propositions ($V^{! \varphi}(p) = V(p) \cap W^{! \varphi}$). We thus get: $\llbracket (\llbracket !\psi \rrbracket (\lambda Y_i D Y \wedge [\varphi] D Y) S) \rrbracket^{\mathcal{H}^{\mathcal{M}}, g[d/Y_i, s/S_i]} = T$ if and only if $M^{! \varphi}, s$. Our argument here is informal in order to avoid further technicalities. Formally, another (analogous) inductive argument is required (now on the structure of ψ).

$\delta = \mathcal{C}_G(\varphi|\psi)$

This case is similar to $K r^i \varphi$. The only difference is the construction of the accessibility relation, which now depends on φ (and D). When comparing the definition of relativized common knowledge⁸ with the proposed embedding of $K r^i \varphi$, the analogy

⁸ $\mathcal{M}, w \models \mathcal{C}^{! \varphi} \psi$ iff $(w, v) \in (R_{EG} \cap (W \times \llbracket \varphi \rrbracket^{\mathcal{M}}))^+$ implies $\mathcal{M}, v \models \psi$.

becomes apparent; the proof is technical and therefore omitted. \square

We can now prove the soundness of the embedding.

Theorem 5.2 (Soundness of the Embedding).

$$\text{If } \models^{\text{HOL}} \text{vld}([\varphi]) \text{ then } \models^{\text{PAL}} \varphi$$

Proof. The proof is by contraposition. Assume $\not\models^{\text{PAL}} \varphi$, i.e., there is a PAL model $\mathcal{M} = \langle W, \{R_i\}_{i \in \mathcal{A}}, V \rangle$ and a world $s \in W$, such that $\mathcal{M}, s \not\models \varphi$. By Lemma 5.1 it holds that $\llbracket [\varphi] D_\sigma S_i \rrbracket^{\mathcal{H}^{\mathcal{M}}, g[d/D_\sigma][s/S_i]} = F$ (for some g and $d = W$) in Henkin model $\mathcal{H}^{\mathcal{M}} = \langle \{\mathcal{D}_\alpha\}_{\alpha \in T}, I \rangle$ for M . Now, $\llbracket [\varphi] D_\sigma S_i \rrbracket^{\mathcal{H}^{\mathcal{M}}, g[d/D_\sigma][s/S_i]} = F$ implies that $\llbracket \forall D_\sigma \forall S_i (D S \rightarrow [\varphi] D S) \rrbracket^{\mathcal{H}^{\mathcal{M}}, g} = \llbracket \text{vld}[\varphi] \rrbracket^{\mathcal{H}^{\mathcal{M}}, g} = F$. Hence, $\mathcal{H}^{\mathcal{M}} \not\models^{\text{HOL}} \text{vld}[\varphi]$. \square

The completeness of our embedding of PAL in HOL is addressed in the next chapter. For this, we show that standard axioms and inference rules of PAL can be inferred from our embedding. Except for two axioms (which seem to require induction) all these meta-theoretical proofs were found fully automatically.

6 Experiments (including Completeness Aspects)

6.1 Proving Axioms and Rules of Inference of PAL in HOL

The presented SSE of PAL is able to prove the following axioms and rules of inference as presented for PAL by Baltag and Renne [3, Supplement F]:

System K

–	All substitutions instances of propositional tautologies
Axiom K	$K_i(\varphi \rightarrow \psi) \rightarrow (K_i\varphi \rightarrow K_i\psi)$
Modus ponens	From φ and $\varphi \rightarrow \psi$ infer ψ
Necessitation	From φ infer $K_i\varphi$

System S5

Axiom T	$K_i\varphi \rightarrow \varphi$
Axiom 4	$K_i\varphi \rightarrow K_iK_i\varphi$
Axiom 5	$\neg K_i\varphi \rightarrow K_i\neg K_i\varphi$

Reduction Axioms

Atomic Permanence	$[\! \varphi]p \leftrightarrow (\varphi \rightarrow p)$
Conjunction	$[\! \varphi](\psi \wedge \chi) \leftrightarrow ([\! \varphi]\psi \wedge [\! \varphi]\chi)$
Partial Functionality	$[\! \varphi]\neg\psi \leftrightarrow (\varphi \rightarrow \neg[\! \varphi]\psi)$
Action-Knowledge	$[\! \varphi]K_i\psi \leftrightarrow (\varphi \rightarrow K_i(\varphi \rightarrow K_i(\varphi \rightarrow [\! \varphi]\psi)))$
–	$[\! \varphi]\mathcal{C}(\chi \psi) \leftrightarrow (\varphi \rightarrow \mathcal{C}(\varphi \wedge [\! \varphi]\chi [\! \varphi]\psi))$

Axiom schemes for Relativized Common Knowledge (RCK)

\mathcal{C} -normality	$\mathcal{C}(\chi (\varphi \rightarrow \psi)) \rightarrow (\mathcal{C}(\chi \varphi) \rightarrow \mathcal{C}(\chi \psi))$
Mix axiom	$\mathcal{C}(\psi \varphi) \leftrightarrow E(\psi \rightarrow (\varphi \wedge \mathcal{C}(\psi \varphi)))$
Induction axiom	$(E(\psi \rightarrow \varphi) \wedge \mathcal{C}(\psi \varphi \rightarrow E(\psi \rightarrow \varphi))) \rightarrow \mathcal{C}(\psi \varphi)$

Rules of Inference

Announcement Nec.	from φ , infer $[\! \psi]\varphi$
RCK Necessitation	from φ , infer $\mathcal{C}(\psi \varphi)$

Automatic proofs in Isabelle/HOL can be found for all axioms except for the right-to-left direction of the mix axiom and the induction axiom schemata for relativized common knowledge (cf. Fig. 2 in App. A). While for these two open cases full proof automation (with the current tools) fails, some simple edge cases can nevertheless be proved automatically.⁹ However, to ensure completeness of our SSE of PAL in HOL, we can simply postulate the two axiom schemes for induction and mix and postpone proving that they are in fact already entailed. This, in fact, illustrates another interesting feature with respect to the rapid prototyping of new logical formalisms in the LOGIKEY approach.

The consistency of our embedding, resp. axiomatization, of PAL in Isabelle/HOL (and also of the additional axioms for induction and mix, and for the wise men puzzle) is confirmed by the model finder Nitpick.

6.2 Exploring Failures of Uniform Substitution

The following principles are examples of sentences that are *valid* for atomic propositions p , but not *schematically valid* for arbitrary formulas φ [28]. The results of our experiments are as expected; proofs can be found for the atomic cases p . For the schematic formulas φ , however, countermodels are reported by the model finder Nitpick (cf. Fig. 2 in App. A)

1. $p \rightarrow \neg[\!|p](\neg p)$
2. $p \rightarrow \neg[\!|p](\neg K_i p)$
3. $p \rightarrow \neg[\!|p](p \wedge \neg K_i p)$
4. $(p \wedge \neg K_i p) \rightarrow \neg[\!|p \wedge \neg K_i p](p \wedge \neg K_i p)$
5. $K_i p \rightarrow \neg[\!|p](\neg K_i p)$
6. $K_i p \rightarrow \neg[\!|p](p \wedge \neg K_i p)$

As an example, consider the schematic counterpart of (1) for which Nitpick reports the following countermodel:

```
lemma "[ $\varphi \rightarrow \neg[\!|\varphi](\neg\varphi)$ ]" nitpick oops
```

Nitpick found a counterexample for card i = 2:

Free variables:

```
 $\varphi = (\lambda x. \_)$   
  (( $\lambda x. \_$ )(i1 := True, i2 := True), i1) := True,  
  (( $\lambda x. \_$ )(i1 := True, i2 := True), i2) := False,  
  (( $\lambda x. \_$ )(i1 := True, i2 := False), i1) := False,  
  (( $\lambda x. \_$ )(i1 := True, i2 := False), i2) := False,
```

⁹Should induction proofs be needed to prove the general cases, this will lead to interesting further work: How to best handle structural induction over the shallowly embedded PAL formulas, while still avoiding a deep embedding of PAL in HOL?

```

((λx. _)(i1 := False, i2 := True), i1) := False,
((λx. _)(i1 := False, i2 := True), i2) := False,
((λx. _)(i1 := False, i2 := False), i1) := False,
((λx. _)(i1 := False, i2 := False), i2) := False

```

Skolem constant:

```

W = (λx. _)(i1 := True, i2 := True)
w = i1

```

The explanation for this model is similar to the one presented in §4.2. This output is expected, given that (1) has been shown not to be *schematically* valid [37].

6.3 Example Application: The Wise Men Puzzle

The wise men puzzle is an interesting riddle in epistemic reasoning. It is well suited to demonstrate epistemic actions in a multi-agent scenario. Baldoni [2] gave a formalization for this, which later got embedded into Isabelle/HOL by Benzmüller [7, 8]. In the following implementation, these results will be used as a stepping stone. Note that below we are not going to define a specific model for the wise men, but instead we analyze the puzzle using the semantic consequence relation.

First, the riddle is recited, and then we go into detail on how the uncertainties change.¹⁰

Once upon a time, a king wanted to find the wisest out of his three wisest men. He arranged them in a circle so that they can see and hear each other and told them that he would put a white or a black spot on their foreheads and that one of the three spots would certainly be white. The three wise men could see and hear each other but, of course, they could not see their faces reflected anywhere. The king, then, asked each of them [sequentially] to find out the color of his own spot. After a while, the wisest correctly answered that his spot was white.

The already existing encoding by Benzmüller puts a particular emphasis on the adequate modeling of common knowledge. In [33], this solution was enhanced by the public announcement operator. Consequently, common knowledge was no longer statically stated after each iteration, but a dynamic approach was used for this. Here, the modeling of this riddle has been further improved (e.g., by better parameterizing our notions over groups of agents) and we are automating the puzzle now for four agents instead of three.

Before we can evaluate the knowledge of the first wise man, we need to formulate the initial circumstances and background knowledge. Let **a**, **b**, **c** and **d** be the wise men (they are being encoded as relations of type α). It is common knowledge, that each wise man can see the foreheads of the other wise men. The only doubt a wise man has, is whether he has a white spot on his own forehead or not. Additionally, it is common knowledge that at least one of the four wise men has a white spot on his forehead. The rules of the riddle are encoded as follows:

```

(* Agents modeled as accessibility relations *)
consts a::"α" b::"α" c::"α" d::"α"
abbreviation Agent::"σ⇒bool" ("A")
where "A x ≡ x = a ∨ x = b ∨ x = c ∨ x = d"

```

¹⁰A very similar riddle, that is often presented in the literature is the *Muddy Children* puzzle [22]. A difference between these two riddles is that in the version presented here, the agents get asked sequentially, not synchronously.

axiomatization where group_S5: "S5Agents \mathcal{A} "

(* Common knowledge: at least one of a, b, c and d has a white spot *)

consts ws: " $\alpha \Rightarrow \sigma$ "

axiomatization where WM1: " $[C_{\mathcal{A}} (^{A_{ws}} a \vee ^{A_{ws}} b \vee ^{A_{ws}} c \vee ^{A_{ws}} d)]$ "

axiomatization where

(* Common knowledge: if x has not a white spot then y know this *)

WM2ab: " $[C_{\mathcal{A}} (\neg(^{A_{ws}} a) \rightarrow K_b(\neg(^{A_{ws}} a)))]$ " and

WM2ac: " $[C_{\mathcal{A}} (\neg(^{A_{ws}} a) \rightarrow K_c(\neg(^{A_{ws}} a)))]$ " and

WM2ad: " $[C_{\mathcal{A}} (\neg(^{A_{ws}} a) \rightarrow K_d(\neg(^{A_{ws}} a)))]$ " and

WM2ba: " $[C_{\mathcal{A}} (\neg(^{A_{ws}} b) \rightarrow K_a(\neg(^{A_{ws}} b)))]$ " and

WM2bc: " $[C_{\mathcal{A}} (\neg(^{A_{ws}} b) \rightarrow K_c(\neg(^{A_{ws}} b)))]$ " and

WM2bd: " $[C_{\mathcal{A}} (\neg(^{A_{ws}} b) \rightarrow K_d(\neg(^{A_{ws}} b)))]$ " and

WM2ca: " $[C_{\mathcal{A}} (\neg(^{A_{ws}} c) \rightarrow K_a(\neg(^{A_{ws}} c)))]$ " and

WM2cb: " $[C_{\mathcal{A}} (\neg(^{A_{ws}} c) \rightarrow K_b(\neg(^{A_{ws}} c)))]$ " and

WM2cd: " $[C_{\mathcal{A}} (\neg(^{A_{ws}} c) \rightarrow K_d(\neg(^{A_{ws}} c)))]$ " and

WM2da: " $[C_{\mathcal{A}} (\neg(^{A_{ws}} d) \rightarrow K_a(\neg(^{A_{ws}} d)))]$ " and

WM2db: " $[C_{\mathcal{A}} (\neg(^{A_{ws}} d) \rightarrow K_b(\neg(^{A_{ws}} d)))]$ " and

WM2dc: " $[C_{\mathcal{A}} (\neg(^{A_{ws}} d) \rightarrow K_c(\neg(^{A_{ws}} d)))]$ "

The positive counterparts $C_{\mathcal{A}} ((^{A_{ws}} x) \rightarrow K_y(^{A_{ws}} x))$ for $x, y \in \mathcal{A}$ of the above negative axioms are implied; this is quickly confirmed by the automated proof tools in Isabelle/HOL. For example, we have (where group_S5 is referring to the S5 properties of the epistemic operators K_y):

lemma WM2ab': " $[C_{\mathcal{A}} ((^{A_{ws}} a) \rightarrow K_b(^{A_{ws}} a))]$ "

using WM2ab group_S5 unfolding Defs by (smt (z3))

Now the king asks whether the first wise man, say a , knows if he has a white spot or not. Assume that a publicly answers that he does not. This is a public announcement of the form: $\neg(K_a(^{A_{ws}} a) \vee (K_a \neg(^{A_{ws}} a)))$. Again, a wise man gets asked by the king whether he knows if he has a white spot or not. Now it's b 's turn, and assume that b also announces that he does not know whether he has a white spot on his forehead.¹¹ The third wise man, c , is also unable to tell whether he has a white spot or not.

When asked, d is able to give the right answer, namely that he has a white spot on his forehead. We can prove this automatically in Isabelle/HOL:¹²

theorem whitespot_d: " $[![\neg K_a(^{A_{ws}} a)]([![\neg K_b(^{A_{ws}} b)]([![\neg K_c(^{A_{ws}} c)](K_d(^{A_{ws}} d)))]))]$ "

using WM1 WM2ba WM2ca WM2cb WM2da WM2db WM2dc

unfolding Defs by (smt (verit))

Alternatively we e.g. get:

theorem whitespot_d':

" $[![\neg((K_a(^{A_{ws}} a) \vee (K_a \neg(^{A_{ws}} a))))]([![\neg((K_b(^{A_{ws}} b) \vee (K_b \neg(^{A_{ws}} b)))]([![\neg((K_c(^{A_{ws}} c) \vee (K_c \neg(^{A_{ws}} c)))](K_d(^{A_{ws}} d)))]))]$ "

using whitespot_c

unfolding Defs sledgehammer[verbose]() (* finds proof *)

(* reconstruction timeout *)

¹¹The case where neither a nor b can correctly infer the color of their forehead when being asked by the king is the most challenging case; we only discuss this one here.

¹²The experiments have been carried out using Isabelle 2021 on a Lenovo ThinkPad T480s with Intel®Core i7-8550U QuadCore@1.8Ghz and 16GB RAM.

7 Comparison with Related Work

In related work [4], van Benthem, van Eijck and colleagues have studied a “*faithful representation of DEL [dynamic epistemic logic] models as so-called knowledge structures that allow for symbolic model checking*”. The authors show that such an approach enables efficient and effective reasoning in epistemic scenarios with state-of-the-art Binary Decision Diagram (BDD) reasoning technology, outperforming other existing methods [20, 21] to automate DEL reasoning. Further related work [19] demonstrates how dynamic epistemic terms can be formalized in temporal epistemic terms to apply the model checkers MCK [23] or MCMAS [32]. Our approach differs in various respects, including:

External vs. internal representation transformation: Instead of writing external (e.g. Haskell-)code to realize the required conversions from DEL into Boolean representations, we work with logic-internal conversions into HOL, provided in the form of a set of equations stated in HOL itself (thereby heavily exploiting the virtues of λ -conversion). Our encoding is concise (only about 50 lines in Isabelle/HOL) and human readable.

Meta-logical reasoning: Since our conversion “code” is provided within the (meta-)logic environment itself, the conversion becomes better controllable and even amenable to formal verification. Moreover, as we have also demonstrated in this article, meta-logical studies about the embedded logics and their embedding in HOL are well-supported in our approach.

Scalability beyond propositional reasoning: Real-world applications often require differentiation between entities/individuals, their properties and functions defined on them. Moreover, quantification over entities (or properties and functions) supports generic statements that are not supported in propositional DEL. In contrast to the related work, the shallow semantical embedding approach very naturally scales for first-order and higher-order extensions of the embedded logics; for more details on this we refer to [7, 8] and the references therein.

Reuse of automated theorem proving and model finding technology: Both the related work and our approach reuse state-of-the-art automated reasoning technology. In our case, this includes world-leading first-order and higher-order theorem provers and model finders already integrated with Isabelle/HOL [16]. These tools in turn internally collaborate with the latest SMT and SAT solving technology. The burden to organize and orchestrate the technical communication with and between these tools is taken away from us by reuse of respective solutions as already provided in Isabelle/HOL (and recursively also within the integrated theorem provers). Well established and robustly supported language formats (e.g. TPTP syntax, <http://www.tptp.org>) are reused in these nested transformations. These cascades of already supported logic transformations are one reason why our embedding approach readily scales for automating reasoning beyond just propositional DEL.

We are convinced, for reasons as discussed above, that our approach is particularly well suited for the exploration and rapid prototyping of new logics (and logic combinations) and their embeddings in HOL, and for the study of their meta-logical properties, in particular, when it comes to first-order and higher-order extensions of DEL. At the

same time, we share with the related work by van Benthem, van Eijck and colleagues a deep interest in practical (object-level) applications, and therefore practical reasoning performance is obviously also of high relevance. In this regard, however, we naturally assume a performance loss in comparison to hand-crafted, specialist solutions. Previous studies in the context of first-order modal logic theorem proving nevertheless have shown that this is not always the case [25].

8 Conclusion

A shallow semantical embedding of public announcement logic with relativized common knowledge in classical higher-order logic has been presented. Our implementation of this embedding in Isabelle/HOL delivers promising initial results, as evidenced by the effective automation of the prominent wise men puzzle. In particular, we have shown how model-changing behavior can be adequately and elegantly addressed in our embedding approach. With reference to uniform substitution, we saw that our embedding enables the study of meta-logical properties of public announcement logic, and object-level reasoning has been demonstrated by a first-time automation of the wise men puzzle encoded in public announcement logic with a relativized common knowledge operator.

Acknowledgments

We thank the anonymous reviewers of this article for their valuable feedback and comments that helped us improve this article. We also thank the reviewers of our related earlier paper presented at the 3rd DaLí Workshop on Dynamic Logic.

References

- [1] BALBIANI, P. ; DITMARSCH, H. P. ; HERZIG, A. ; LIMA, T. de: A Tableau Method for Public Announcement Logics. In: OLIVETTI, N. (Hrsg.): *Automated Reasoning with Analytic Tableaux and Related Methods, 16th International Conference, TABLEAUX 2007, Aix en Provence, France, July 3-6, 2007, Proceedings* Bd. 4548, Springer (Lecture Notes in Computer Science), 43–59
- [2] BALDONI, M. : *Normal multimodal logics: Automatic deduction and logic programming extension*, Università degli Studi di Torino, Dipartimento di Informatica, Diss., 1998
- [3] BALTAG, A. ; RENNE, B. : Dynamic Epistemic Logic. In: ZALTA, E. N. (Hrsg.): *The Stanford Encyclopedia of Philosophy*, Metaphysics Research Lab, Stanford University, 2016
- [4] BENTHEM, J. van ; EIJCK, J. van ; GATTINGER, M. ; SU, K. : Symbolic model checking for Dynamic Epistemic Logic - S5 and beyond. In: *J.Log.Comp.* 28 (2018), Nr. 2, 367–402. <https://doi.org/10.1093/logcom/exx038>
- [5] BENTHEM, J. van ; EIJCK, J. van ; KOOI, B. : Logics of communication and change. In: *Information and Computation* 204 (2006), Nr. 11, 1620 - 1662. <https://doi.org/10.1016/j.ic.2006.04.006>. – ISSN 0890–5401

- [6] BENZMÜLLER, C. : Cut-Elimination for Quantified Conditional Logic. In: *Journal of Philosophical Logic* 46 (2017), Nr. 3, 333-353. <https://doi.org/10.1007/s10992-016-9403-0>
- [7] BENZMÜLLER, C. : Universal (Meta-)Logical Reasoning: Recent Successes. In: *Science of Computer Programming* 172 (2019), 48-62. <https://doi.org/10.1016/j.scico.2018.10.008>
- [8] BENZMÜLLER, C. : Universal (Meta-)Logical Reasoning: The Wise Men Puzzle (Isabelle/HOL Dataset). In: *Data in Brief* 24 (2019), Nr. 103823, 1–5. <https://doi.org/10.1016/j.dib.2019.103823>
- [9] BENZMÜLLER, C. ; ANDREWS, P. : Church’s Type Theory. Version: Summer 2019, 2019. <https://plato.stanford.edu/entries/type-theory-church/>. In: ZALTA, E. N. (Hrsg.): *The Stanford Encyclopedia of Philosophy*. Summer 2019. Metaphysics Research Lab, Stanford University, 1–62 (in pdf version)
- [10] BENZMÜLLER, C. ; BROWN, C. ; KOHLHASE, M. : Higher-Order Semantics and Extensionality. In: *Journal of Symbolic Logic* 69 (2004), Nr. 4, 1027-1088. <https://doi.org/10.2178/jsl/1102022211>
- [11] BENZMÜLLER, C. ; FARJAMI, A. ; FUENMAYOR, D. ; MEDER, P. ; PARENT, X. ; STEEN, A. ; TORRE, L. van d. ; ZAHORANSKY, V. : LogiKEy Workbench: Deontic Logics, Logic Combinations and Expressive Ethical and Legal Reasoning (Isabelle/HOL Dataset). In: *Data in Brief* 33 (2020), Nr. 106409, 1-10. <https://doi.org/10.1016/j.dib.2020.106409>
- [12] BENZMÜLLER, C. ; MILLER, D. : Automation of Higher-Order Logic. Version: 2014. <https://doi.org/10.1016/B978-0-444-51624-4.50005-8>. In: GABBAY, D. M. (Hrsg.) ; SIEKMANN, J. H. (Hrsg.) ; WOODS, J. (Hrsg.): *Handbook of the History of Logic, Volume 9 — Computational Logic*. North Holland, Elsevier. – ISBN 978–0–444–51624–4, 215-254
- [13] BENZMÜLLER, C. ; PARENT, X. ; TORRE, L. van d.: Designing Normative Theories for Ethical and Legal Reasoning: LogiKEy Framework, Methodology, and Tool Support. In: *Artificial Intelligence* 287 (2020), 103348. <https://doi.org/10.1016/j.artint.2020.103348>
- [14] BENZMÜLLER, C. ; PAULSON, L. C.: Multimodal and Intuitionistic Logics in Simple Type Theory. In: *The Logic Journal of the IGPL* 18 (2010), Nr. 6, 881-892. <https://doi.org/10.1093/jigpal/jzp080>
- [15] BENZMÜLLER, C. ; PAULSON, L. C.: Quantified Multimodal Logics in Simple Type Theory. In: *Logica Universalis (Special Issue on Multimodal Logics)* 7 (2013), Nr. 1, 7-20. <https://doi.org/10.1007/s11787-012-0052-y>
- [16] BLANCHETTE, J. ; BÖHME, S. ; PAULSON, L. : Extending Sledgehammer with SMT Solvers. In: *Journal of Automated Reasoning* 51 (2011), 10, 116-130. https://doi.org/10.1007/978-3-642-22438-6_11

- [17] BLANCHETTE, J. C. ; NIPKOW, T. : Nitpick: A Counterexample Generator for Higher-Order Logic Based on a Relational Model Finder. In: KAUFMANN, M. (Hrsg.) ; PAULSON, L. C. (Hrsg.): *Interactive Theorem Proving, First International Conference, ITP 2010, Edinburgh, UK, July 11-14, 2010. Proceedings* Bd. 6172, Springer (Lecture Notes in Computer Science), 131–146
- [18] CHURCH, A. : A formulation of the simple theory of types. In: *Journal of Symbolic Logic* 5 (1940), Nr. 2, 56–68. <https://doi.org/10.2307/2266170>
- [19] DITMARSCH, H. van ; HOEK, W. van d. ; MEYDEN, R. van d. ; RUAN, J. : Model checking russian cards. In: *ENTCS* 149 (2006), Nr. 2, S. 105–123
- [20] EIJCK, J. van: *DEMO—a demo of epistemic modelling*. Interactive Logic. Selected Papers from the 7th Augustus de Morgan Workshop, London, http://homepages.cwi.nl/~jve/papers/07/pdfs/DEMO_IL.pdf. http://homepages.cwi.nl/~jve/papers/07/pdfs/DEMO_IL.pdf. Version: 2007
- [21] EIJCK, J. van: *DEMO-S5*. Tech. rep., CWI, http://homepages.cwi.nl/~jve/software/demo_s5. http://homepages.cwi.nl/~jve/software/demo_s5. Version: 2014
- [22] FAGIN, R. ; HALPERN, J. Y. ; MOSES, Y. ; VARDI, M. Y.: Common Knowledge Revisited. In: *Ann. Pure Appl. Log.* 96 (1999), Nr. 1-3, 89–105. [https://doi.org/10.1016/S0168-0072\(98\)00033-5](https://doi.org/10.1016/S0168-0072(98)00033-5)
- [23] GAMMIE, P. ; MEYDEN, R. van d.: MCK: Model Checking the Logic of Knowledge. In: ALUR, R. (Hrsg.) ; PELED, D. A. (Hrsg.): *Computer Aided Verification, 16th International Conference, CAV 2004, Boston, MA, USA, July 13-17, 2004, Proceedings* Bd. 3114, Springer (Lecture Notes in Computer Science), 479–483
- [24] GIBBONS, J. ; WU, N. : Folding domain-specific languages: deep and shallow embeddings (functional Pearl). In: JEURING, J. (Hrsg.) ; CHAKRAVARTY, M. M. T. (Hrsg.): *Proceedings of the 19th ACM SIGPLAN international conference on Functional programming, Gothenburg, Sweden, September 1-3, 2014*, ACM, 339–347
- [25] GLEISSNER, T. ; STEEN, A. ; BENZMÜLLER, C. : Theorem Provers for Every Normal Modal Logic. In: EITER, T. (Hrsg.) ; SANDS, D. (Hrsg.): *LPAR-21. 21st International Conference on Logic for Programming, Artificial Intelligence and Reasoning* Bd. 46. EasyChair (EPiC Series in Computing). – ISSN 2398–7340, 14-30
- [26] GÖDEL, K. : Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I. In: *Monatshefte für Mathematik und Physik* 38 (1931), Nr. 1, 173–198. <https://doi.org/10.1007/BF01700692>
- [27] HENKIN, L. : Completeness in the theory of types. In: *The Journal of Symbolic Logic* 15 (1950), Nr. 2, 81–91. <https://doi.org/10.2307/2268698>
- [28] HOLLIDAY, W. H. ; HOSHI, T. ; ICARD, T. F.: Information dynamics and uniform substitution. In: *Synthese* 190 (2013), 31–55. <https://doi.org/10.1007/s11229-013-0278-0>. – ISSN 00397857, 15730964

- [29] PACUIT, E. : Dynamic Epistemic Logic I: Modeling Knowledge and Belief. In: *Philosophy Compass* 8 (2013), Nr. 9, 798-814. <https://doi.org/10.1111/phc3.12059>
- [30] PFENNING, F. ; ELLIOTT, C. : Higher-Order Abstract Syntax. In: *Proc. of the ACM SIGPLAN'88 Conference on Programming Language Design and Implementation (PLDI), Atlanta, Georgia, USA, June 22-24, 1988*, ACM, 199–208
- [31] PLAZA, J. : Logics of public communications. In: *Synthese* 158 (2007), Nr. 2, 165–179. <https://doi.org/10.1007/s11229-007-9168-7>
- [32] RAIMONDI, F. ; LOMUSCIO, A. : Verification of multiagent systems via ordered binary decision diagrams: an algorithm and its implementation. In: *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems, 2004. AAMAS 2004* IEEE, 2004, S. 630–637
- [33] REICHE, S. ; BENZMÜLLER, C. : Public Announcement Logic in HOL. In: MARTINS, M. A. (Hrsg.) ; IGOR, S. (Hrsg.): *Dynamic Logic. New Trends and Applications. DaLi 2020* Bd. 12569, Springer, Cham (Lecture Notes in Computer Science). – ISBN 978–3–030–65839–7, 222–238
- [34] SVENNINGSSON, J. ; AXELSSON, E. : Combining deep and shallow embedding of domain-specific languages. In: *Comput. Lang. Syst. Struct.* 44 (2015), 143–165. <https://doi.org/10.1016/j.cl.2015.07.003>
- [35] TOBIAS NIPKOW, M. W. Lawrence C. Paulson P. Lawrence C. Paulson: *Isabelle/HOL: A Proof Assistant for Higher-Order Logic*. Springer-Verlag Berlin Heidelberg <https://doi.org/10.1007/3-540-45949-9>
- [36] VAN DITMARSCH, H. ; DER HOEK, W. van ; KOOI, B. : *Dynamic epistemic logic*. Bd. 337. Springer Science & Business Media <https://doi.org/10.1007/978-1-4020-5839-4>
- [37] VAN DITMARSCH, H. ; HOEK, W. Van d. ; ILIEV, P. : Everything is knowable—how to get to know whether a proposition is true. In: *Theoria* 78 (2012), Nr. 2, 93–114. <http://doi.org/10.1111/j.1755-2567.2011.01119.x>

A Isabelle/HOL sources

The sources of our modeling and experiments in Isabelle/HOL are presented in Figs. 1-4.

```

1 (* Sebastian Reiche and Christoph Benzmüller, 2021 *)
2 theory PAL_definitions imports Main
3
4 begin
5 typedef i (* Type of possible worlds *)
6 type_synonym  $\sigma$  = "i $\Rightarrow$ bool" (*Type of world domains *)
7 type_synonym  $\tau$  = " $\sigma \Rightarrow i \Rightarrow$ bool" (* Type of world depended formulas (truth sets) *)
8 type_synonym  $\alpha$  = "i $\Rightarrow i \Rightarrow$ bool" (* Type of accessibility relations between world *)
9 type_synonym  $\varrho$  = " $\alpha \Rightarrow$ bool" (* Type of groups of agents *)
10
11 (* Some useful relations (for constraining accessibility relations) *)
12 definition reflexive::" $\alpha \Rightarrow$ bool" where "reflexive R  $\equiv \forall x. R x x$ "
13 definition symmetric::" $\alpha \Rightarrow$ bool" where "symmetric R  $\equiv \forall x y. R x y \longrightarrow R y x$ "
14 definition transitive::" $\alpha \Rightarrow$ bool" where "transitive R  $\equiv \forall x y z. R x y \wedge R y z \longrightarrow R x z$ "
15 definition euclidean::" $\alpha \Rightarrow$ bool" where "euclidean R  $\equiv \forall x y z. R x y \wedge R x z \longrightarrow R y z$ "
16 definition intersection_rel::" $\alpha \Rightarrow \alpha \Rightarrow \alpha$ " where "intersection_rel R Q  $\equiv \lambda u v. R u v \wedge Q u v$ "
17 definition union_rel::" $\alpha \Rightarrow \alpha \Rightarrow \alpha$ " where "union_rel R Q  $\equiv \lambda u v. R u v \vee Q u v$ "
18 definition sub_rel::" $\alpha \Rightarrow \alpha \Rightarrow \alpha$ " where "sub_rel R Q  $\equiv \forall u v. R u v \longrightarrow Q u v$ "
19 definition inverse_rel::" $\alpha \Rightarrow \alpha$ " where "inverse_rel R  $\equiv \lambda u v. R v u$ "
20 definition big_union_rel::" $\varrho \Rightarrow \alpha$ " where "big_union_rel X  $\equiv \lambda u v. \exists R. (X R) \wedge (R u v)$ "
21 definition big_intersection_rel::" $\varrho \Rightarrow \alpha$ "
22   where "big_intersection_rel X  $\equiv \lambda u v. \forall R. (X R) \longrightarrow (R u v)$ "
23
24 (*In HOL the transitive closure of a relation can be defined in a single line.*)
25 definition tc::" $\alpha \Rightarrow \alpha$ " where "tc R  $\equiv \lambda x y. \forall Q. \text{transitive } Q \longrightarrow (\text{sub\_rel } R Q \longrightarrow Q x y)$ "
26
27 (* Lifted HOMML connectives for PAL *)
28 abbreviation patom::" $\sigma \Rightarrow \tau$ " ("A_"[79]80) where "A p  $\equiv \lambda W w. W w \wedge p w$ "
29 abbreviation ptop::" $\tau$ " ("T") where "T  $\equiv \lambda W w. \text{True}$ "
30 abbreviation pneg::" $\tau \Rightarrow \tau$ " ("¬_"[52]53) where "¬  $\varphi \equiv \lambda W w. \neg(\varphi W w)$ "
31 abbreviation pand::" $\tau \Rightarrow \tau \Rightarrow \tau$ " (infixr"∧"51) where " $\varphi \wedge \psi \equiv \lambda W w. (\varphi W w) \wedge (\psi W w)$ "
32 abbreviation por::" $\tau \Rightarrow \tau \Rightarrow \tau$ " (infixr"∨"50) where " $\varphi \vee \psi \equiv \lambda W w. (\varphi W w) \vee (\psi W w)$ "
33 abbreviation pimp::" $\tau \Rightarrow \tau \Rightarrow \tau$ " (infixr"→"49) where " $\varphi \rightarrow \psi \equiv \lambda W w. (\varphi W w) \longrightarrow (\psi W w)$ "
34 abbreviation pequ::" $\tau \Rightarrow \tau \Rightarrow \tau$ " (infixr"↔"48) where " $\varphi \leftrightarrow \psi \equiv \lambda W w. (\varphi W w) \longleftrightarrow (\psi W w)$ "
35 abbreviation pknow::" $\alpha \Rightarrow \tau \Rightarrow \tau$ " ("K_" ) where "K r  $\varphi \equiv \lambda W w. \forall v. (W v \wedge r w v) \longrightarrow (\varphi W v)$ "
36 abbreviation ppa::" $\tau \Rightarrow \tau \Rightarrow \tau$ " ("!_" ) where "!"  $\varphi \equiv \lambda W w. (\varphi W w) \longrightarrow (\psi (\lambda z. W z \wedge \varphi W z) w)$ "
37
38 (* Validity of  $\tau$ -type lifted PAL formulas *)
39 abbreviation pvalid::" $\tau \Rightarrow$  bool" ("|" [7]8) where "|  $\varphi \equiv \forall W. \forall w. W w \longrightarrow \varphi W w$ "
40
41 (* Agent Knowledge, Mutual Knowledge, Common Knowledge *)
42 abbreviation EVR::" $\varrho \Rightarrow \alpha$ " where "EVR G  $\equiv \text{big\_union\_rel } G$ "
43 abbreviation DIS::" $\varrho \Rightarrow \alpha$ " where "DIS G  $\equiv \text{big\_intersection\_rel } G$ "
44 abbreviation agtknows::" $\alpha \Rightarrow \tau \Rightarrow \tau$ " ("K_" ) where "K r  $\varphi \equiv K r \varphi$ "
45 abbreviation evrknows::" $\varrho \Rightarrow \tau \Rightarrow \tau$ " ("E_" ) where "EG  $\varphi \equiv K (\text{EVR } G) \varphi$ "
46 abbreviation disknows :: " $\varrho \Rightarrow \tau \Rightarrow \tau$ " ("D_" ) where "DG  $\varphi \equiv K (\text{DIS } G) \varphi$ "
47 abbreviation prck::" $\varrho \Rightarrow \tau \Rightarrow \tau$ " ("C_" )
48   where "CG( $\varphi | \psi$ )  $\equiv \lambda W w. \forall v. (\text{tc } (\text{intersection\_rel } (\text{EVR } G) (\lambda u v. W v \wedge \varphi W v)) w v) \longrightarrow (\psi W v)$ "
49 abbreviation pcmn::" $\varrho \Rightarrow \tau \Rightarrow \tau$ " ("C_" ) where "CG  $\varphi \equiv C_G(T | \varphi)$ "
50
51 (* S5 principles for the agent's accessibility relations *)
52 abbreviation S5Agent::" $\alpha \Rightarrow$ bool"
53   where "S5Agent i  $\equiv \text{reflexive } i \wedge \text{transitive } i \wedge \text{euclidean } i$ "
54 abbreviation S5Agents::" $\varrho \Rightarrow$ bool"
55   where "S5Agents A  $\equiv \forall i. (A i \longrightarrow \text{S5Agent } i)$ "
56
57 (* Introducing "Defs" as the set of the above definitions; useful for convenient unfolding *)
58 named_theorems Defs
59 declare reflexive_def[Defs] symmetric_def[Defs] transitive_def[Defs] euclidean_def[Defs]
60   intersection_rel_def[Defs] union_rel_def[Defs] sub_rel_def[Defs] inverse_rel_def[Defs]
61   big_union_rel_def[Defs] big_intersection_rel_def[Defs] tc_def[Defs]
62
63 (* Consistency confirmed by nitpick *)
64 lemma True nitpick [satisfy,show_all] oops (* model found *)
65 end

```

Figure 1: Embedding of PAL in HOL

```

1 (* Sebastian Reiche and Christoph Benzmüller, 2021 *)
2 theory PAL_experiments imports PAL_definitions
3
4 begin
5 (* Parameter settings *)
6 nitpick_params[user_axioms=true, format=4, show_all]
7 declare [[smt_solver=cvc4,smt_oracle]]
8
9 (*Some useful lemmata *)
10 lemma trans_tc: "transitive (tc R)" unfolding Defs by metis
11 lemma trans_inv_tc: "transitive (inverse_rel (tc R))" unfolding Defs by metis
12 lemma sub_rel_tc: "symmetric R  $\implies$  (sub_rel R (inverse_rel (tc R)))" unfolding Defs by smt
13 lemma sub_rel_tc_tc: "symmetric R  $\implies$  (sub_rel (tc R) (inverse_rel (tc R)))"
14   using sub_rel_def sub_rel_tc tc_def trans_inv_tc by fastforce
15 lemma symm_tc: "symmetric R  $\implies$  symmetric (tc R)"
16   using inverse_rel_def sub_rel_def sub_rel_tc_tc symmetric_def by auto
17
18 (* System K: is implied by the semantical embedding *)
19 lemma tautologies: "[T]" by auto
20 lemma axiom_K: " $A \text{ i} \implies [(K_i (\varphi \rightarrow \psi)) \rightarrow ((K_i \varphi) \rightarrow (K_i \psi))]$ " by auto
21 lemma modusponens: assumes 1: " $[\varphi \rightarrow \psi]$ " and 2: " $[\varphi]$ " shows " $[\psi]$ " using 1 2 by auto
22 lemma necessitation: assumes 1: " $[\varphi]$ " shows " $A \text{ i} \implies [K_i \varphi]$ " using 1 by auto
23 (* More axioms: implied by the semantical embedding *)
24 lemma axiom_T: "reflexive i  $\implies [(K_i \varphi) \rightarrow \varphi]$ " using reflexive_def by auto
25 lemma axiom_4: "transitive i  $\implies [(K_i \varphi) \rightarrow (K_i (K_i \varphi))]$ " using transitive_def by meson
26 lemma axiom_5: "euclidean i  $\implies [(\neg K_i \varphi) \rightarrow (K_i (\neg K_i \varphi))]$ " using euclidean_def by meson
27 (*Reduction axioms: implied by the semantical embedding *)
28 lemma atomic_permanence: " $[(\![\varphi]\!]^A p) \leftrightarrow (\varphi \rightarrow \!^A p)]$ " by auto
29 lemma conjunction: " $[(\![\varphi]\!](\psi \wedge \chi)) \leftrightarrow ((\![\varphi]\!]\psi) \wedge (\![\varphi]\!]\chi)]$ " by auto
30 lemma part_func: " $[(\![\varphi]\!]\neg\psi) \leftrightarrow (\varphi \rightarrow (\neg(\![\varphi]\!]\psi)))]$ " by auto
31 lemma action_knowledge: " $[(\![\varphi]\!](K_i \psi)) \leftrightarrow (\varphi \rightarrow (K_i (\varphi \rightarrow (\![\varphi]\!]\psi))))]$ " by auto
32 lemma " $[(\![\varphi]\!](C_A(\chi|\psi)) \leftrightarrow (\varphi \rightarrow (C_A(\varphi \wedge (\![\varphi]\!]\chi) | \![\varphi]\!]\psi)))]$ "
33   by (smt (z3) intersection_rel_def sub_rel_def tc_def transitive_def)
34
35
36 (* Axiom schemes for RCK: implied by the semantical embedding *)
37 lemma C_normality: " $[C_A(\chi|\varphi \rightarrow \psi) \rightarrow (C_A(\chi|\varphi) \rightarrow C_A(\chi|\psi))]$ " unfolding Defs by blast
38
39 lemma mix_axiom1: " $[C_A(\chi|\varphi) \rightarrow E_A(\chi \rightarrow (\varphi \wedge (C_A(\chi|\varphi))))]$ " unfolding Defs by smt
40
41 lemma mix_axiom2': " $A = (\lambda x. \text{False}) \implies [(E_A(\chi \rightarrow (\varphi \wedge (C_A(\chi|\varphi)))) \rightarrow C_A(\chi|\varphi)]$ "
42   unfolding Defs by (metis (full_types))
43 lemma mix_axiom2'': " $A = (\lambda x. \text{True}) \implies [(E_A(\chi \rightarrow (\varphi \wedge (C_A(\chi|\varphi)))) \rightarrow C_A(\chi|\varphi)]$ "
44   unfolding Defs by (metis (full_types)) (* takes long *)
45 lemma mix_axiom2''': " $A = (\lambda x. x = a) \wedge S5Agent\ a \implies [(E_A(\chi \rightarrow (\varphi \wedge (C_A(\chi|\varphi)))) \rightarrow C_A(\chi|\varphi)]$ "
46   unfolding Defs (* sledgehammer finds proof, but reconstruction times out *) oops
47 lemma mix_axiom2''''':
48   " $A = (\lambda x. x = a \vee x = b) \wedge S5Agent\ a \wedge S5Agent\ b \implies [(E_A(\chi \rightarrow (\varphi \wedge (C_A(\chi|\varphi)))) \rightarrow C_A(\chi|\varphi)]$ "
49   unfolding Defs (* sledgehammer and nitpick timeout *) oops
50 lemma mix_axiom2_general: " $[(E_A(\chi \rightarrow (\varphi \wedge (C_A(\chi|\varphi)))) \rightarrow C_A(\chi|\varphi)]$ "
51   unfolding Defs (* timeout *) oops
52
53 lemma induction_axiom':
54   " $A = (\lambda x. \text{False}) \implies [((E_A(\chi \rightarrow \varphi)) \wedge C_A(\chi|\varphi \rightarrow (E_A(\chi \rightarrow \varphi)))) \rightarrow (C_A(\chi|\varphi))]$ "
55   unfolding Defs by (metis (full_types))
56 lemma induction_axiom'':
57   " $A = (\lambda x. \text{True}) \implies [((E_A(\chi \rightarrow \varphi)) \wedge C_A(\chi|\varphi \rightarrow (E_A(\chi \rightarrow \varphi)))) \rightarrow (C_A(\chi|\varphi))]$ "
58   unfolding Defs (* sledgehammer finds proof, but reconstruction times out *) oops
59 lemma induction_axiom''':
60   " $A = (\lambda x. x = a) \wedge S5Agent\ a \implies [((E_A(\chi \rightarrow \varphi)) \wedge C_A(\chi|\varphi \rightarrow (E_A(\chi \rightarrow \varphi)))) \rightarrow (C_A(\chi|\varphi))]$ "
61   unfolding Defs (* sledgehammer finds proof, but reconstruction times out *) oops
62 lemma induction_axiom''''':
63   " $A = (\lambda x. x = a \vee x = b) \wedge S5Agent\ a \wedge S5Agent\ b$ 
64      $\implies [((E_A(\chi \rightarrow \varphi)) \wedge C_A(\chi|\varphi \rightarrow (E_A(\chi \rightarrow \varphi)))) \rightarrow (C_A(\chi|\varphi))]$ "
65   unfolding Defs (* sledgehammer and nitpick timeout *) oops
66 lemma induction_axiom_general: " $[(E_A(\chi \rightarrow \varphi)) \wedge C_A(\chi|\varphi \rightarrow (E_A(\chi \rightarrow \varphi)))) \rightarrow (C_A(\chi|\varphi))]$ "
67   unfolding Defs (* sledgehammer and nitpick timeout *) oops
68

```

Figure 2: Testing the automation of PAL in HOL

```

69 (* To ensure completeness we may also simply postulate axiom schemata in the LogiKey approach *)
70 axiomatization where
71   mix_axiom_general: "[Ca( $\chi$  |  $\varphi$ )  $\rightarrow$  Ea( $\chi$   $\rightarrow$  ( $\varphi$   $\wedge$  (Ca( $\chi$  |  $\varphi$ ))))]" and
72   induction_axiom_general: "[((Ea( $\chi$   $\rightarrow$   $\varphi$ ))  $\wedge$  Ca( $\chi$  |  $\varphi$   $\rightarrow$  (Ea( $\chi$   $\rightarrow$   $\varphi$ )))  $\rightarrow$  (Ca( $\chi$  |  $\varphi$ ))]"
73
74 (* Necessitation rules: implied by the semantical embedding *)
75 lemma announcement_nec: assumes "[ $\varphi$ ]" shows "[[! $\psi$ ] $\varphi$ ]" by (simp add: assms)
76 lemma rkc_necessitation: assumes "[ $\varphi$ ]" shows "[Ca( $\chi$  |  $\varphi$ )]"
77   by (smt assms intersection_rel_def sub_rel_def tc_def transitive_def)
78
79 (* Checking for consistency (again) *)
80 lemma True nitpick[satisfy,user_axioms] oops (* model found *)
81 lemma False sledgehammer oops (* provers time out, i.e. fail to prove falsity *)
82
83 (* Further axioms: implied for atomic formulas, but not implied in general *)
84 lemma "[ $\wedge p \rightarrow \neg[!p](\neg \wedge p)$ ]" by simp
85 lemma "[ $\varphi \rightarrow \neg[!p](\neg \varphi)$ ]" nitpick oops (* countermodel found *)
86 lemma "[ $\wedge p \rightarrow \neg[!p](\neg K_a \wedge p)$ ]" by simp
87 lemma "[ $\varphi \rightarrow \neg[!p](\neg K_a \varphi)$ ]" nitpick oops (* countermodel found *)
88 lemma "[ $\wedge p \rightarrow \neg[!p](\neg K_r \wedge p)$ ]" by simp
89 lemma "[ $\varphi \rightarrow \neg[!p](\neg K_r \varphi)$ ]" nitpick oops (* countermodel found *)
90 lemma "[ $\wedge p \rightarrow \neg[!p](\wedge p \wedge \neg K_r \wedge p)$ ]" by simp
91 lemma "[ $\varphi \rightarrow \neg[!p](\varphi \wedge \neg K_r \varphi)$ ]" nitpick oops (* countermodel found *)
92 lemma "[ $(\wedge p \wedge \neg K_r \wedge p) \rightarrow \neg[!p \wedge \neg K_r \wedge p](\wedge p \wedge \neg K_r \wedge p)$ ]" by blast
93 lemma "[ $(\varphi \wedge \neg K_r \varphi) \rightarrow \neg[!p \wedge \neg K_r \varphi](\varphi \wedge \neg K_r \varphi)$ ]" nitpick oops (* countermodel found *)
94 lemma "S5Agent r  $\implies$  [( $K_r \wedge p$ )  $\rightarrow$   $\neg[!p](\neg K_r \wedge p)$ ]" using reflexive_def by auto
95 lemma "S5Agent r  $\implies$  [( $K_r \varphi$ )  $\rightarrow$   $\neg[!p](\neg K_r \varphi)$ ]" nitpick oops (* countermodel found *)
96 lemma "S5Agent r  $\implies$  [( $K_r \wedge p$ )  $\rightarrow$   $\neg[!p](\wedge p \wedge \neg K_r \wedge p)$ ]" using reflexive_def by auto
97 lemma "S5Agent r  $\implies$  [( $K_r \varphi$ )  $\rightarrow$   $\neg[!p](\varphi \wedge \neg K_r \varphi)$ ]" nitpick oops (* countermodel found *)
98
99 (* Further checks on the atomic versus general validity. *)
100 lemma "[ $p \leftrightarrow q$ ]  $\implies$   $\forall W v. (p W v \leftrightarrow q W v)$ " unfolding Defs nitpick oops (* countermodel *)
101 lemma " $\forall W v. (p W v \leftrightarrow q W v) \implies [p \leftrightarrow q]$ " unfolding Defs by simp
102 lemma "[ $\wedge p \leftrightarrow \wedge q$ ]  $\implies$   $\forall v. (p v \leftrightarrow q v)$ " unfolding Defs by simp
103 lemma " $\forall v. (p v \leftrightarrow q v) \implies [\wedge p \leftrightarrow \wedge q]$ " unfolding Defs by simp
104
105 (* Concrete models can be defined and studied. *)
106 lemma assumes "W = ( $\lambda x. x = w1 \vee x = w2 \vee x = w3$ )"
107   "w1  $\neq$  w2" "w1  $\neq$  w3" "w2  $\neq$  w3"
108   "p W w1" "p W w2" " $\neg$ (p W w3)"
109   "a w1 w1" "a w1 w2" "a w2 w1"
110   "a w2 w2" " $\neg$ (a w1 w3)" " $\neg$ (a w3 w1)"
111   " $\neg$ (a w2 w3)" " $\neg$ (a w3 w2)" "a w3 w3"
112   "b w1 w1" " $\neg$ (b w1 w2)" " $\neg$ (b w2 w1)"
113   "b w2 w2" " $\neg$ (b w1 w3)" " $\neg$ (b w3 w1)"
114   "b w2 w3" "b w3 w2" "b w3 w3"
115   shows "((p  $\wedge$  (K_a p)  $\wedge$  (K_b p))  $\wedge$   $\neg$ (K_a (K_b p))) W w1"
116   unfolding Defs
117   nitpick[satisfy, atoms=w1 w2 w3] (* model *)
118   using assms(1) assms(5) assms(6) assms(7)
119   assms(9) assms(12) assms(21) assms(23) by blast (* proof *)
120 end

```

Figure 3: Testing the automation of PAL in HOL (contd.)

```

1 (* Sebastian Reiche and Christoph Benzmüller, 2021 *)
2 theory PAL_WiseMenPuzzle_4Agents imports PAL_definitions
3
4 begin
5 (* Parameter settings *)
6 declare [[smt_solver=cvc4,smt_oracle,smt_timeout=120]]
7
8 (** Encoding of the wise men puzzle in PAL **)
9 consts a::"α" b::"α" c::"α" d::"α" (* Agents modeled as accessibility relations *)
10 abbreviation Agent::"α⇒bool" ("A") where "A x ≡ x = a ∨ x = b ∨ x = c ∨ x = d"
11 axiomatization where group_S5: "S5Agents A"
12
13 (** Encoding of the wise men puzzle in PAL **)
14 (* Common knowledge: At least one of a, b, c and d has a white spot *)
15 consts ws::"α⇒σ"
16 axiomatization where WM1: "[C_A (A_ws a ∨ A_ws b ∨ A_ws c ∨ A_ws d)]"
17 axiomatization where
18   (* Common knowledge: If x does not have a white spot then y know this *)
19   WM2ab: "[C_A (¬(A_ws a) → (K_b (¬(A_ws a))))]" and
20   WM2ac: "[C_A (¬(A_ws a) → (K_c (¬(A_ws a))))]" and
21   WM2ad: "[C_A (¬(A_ws a) → (K_d (¬(A_ws a))))]" and
22   WM2ba: "[C_A (¬(A_ws b) → (K_a (¬(A_ws b))))]" and
23   WM2bc: "[C_A (¬(A_ws b) → (K_c (¬(A_ws b))))]" and
24   WM2bd: "[C_A (¬(A_ws b) → (K_d (¬(A_ws b))))]" and
25   WM2ca: "[C_A (¬(A_ws c) → (K_a (¬(A_ws c))))]" and
26   WM2cb: "[C_A (¬(A_ws c) → (K_b (¬(A_ws c))))]" and
27   WM2cd: "[C_A (¬(A_ws c) → (K_d (¬(A_ws c))))]" and
28   WM2da: "[C_A (¬(A_ws d) → (K_a (¬(A_ws d))))]" and
29   WM2db: "[C_A (¬(A_ws d) → (K_b (¬(A_ws d))))]" and
30   WM2dc: "[C_A (¬(A_ws d) → (K_c (¬(A_ws d))))]"
31
32 (* Positive introspection principles are implied *)
33 lemma WM2ab': "[C_A ((A_ws a) → K_b (A_ws a))]" using WM2ab group_S5 unfolding Defs by (smt (z3))
34 lemma WM2ac': "[C_A ((A_ws a) → K_c (A_ws a))]" using WM2ac group_S5 unfolding Defs by (smt (z3))
35 lemma WM2ad': "[C_A ((A_ws a) → K_d (A_ws a))]" using WM2ad group_S5 unfolding Defs by (smt (z3))
36 lemma WM2ba': "[C_A ((A_ws b) → K_a (A_ws b))]" using WM2ba group_S5 unfolding Defs by (smt (z3))
37 lemma WM2bc': "[C_A ((A_ws b) → K_c (A_ws b))]" using WM2bc group_S5 unfolding Defs by (smt (z3))
38 lemma WM2bd': "[C_A ((A_ws b) → K_d (A_ws b))]" using WM2bd group_S5 unfolding Defs by (smt (z3))
39 lemma WM2ca': "[C_A ((A_ws c) → K_a (A_ws c))]" using WM2ca group_S5 unfolding Defs by (smt (z3))
40 lemma WM2cb': "[C_A ((A_ws c) → K_b (A_ws c))]" using WM2cb group_S5 unfolding Defs by (smt (z3))
41 lemma WM2cd': "[C_A ((A_ws c) → K_d (A_ws c))]" using WM2cd group_S5 unfolding Defs by (smt (z3))
42 lemma WM2da': "[C_A ((A_ws d) → K_a (A_ws d))]" using WM2da group_S5 unfolding Defs by (smt (z3))
43 lemma WM2db': "[C_A ((A_ws d) → K_b (A_ws d))]" using WM2db group_S5 unfolding Defs by (smt (z3))
44 lemma WM2dc': "[C_A ((A_ws d) → K_c (A_ws d))]" using WM2dc group_S5 unfolding Defs by (smt (z3))
45
46 (* Automated solutions of the Wise Men Puzzle with 4 Agents*)
47 theorem whitespot_d: "[[!¬K_a(A_ws a)]]([!¬K_b(A_ws b)]]([!¬K_c(A_ws c)]](K_d (A_ws d))))]"
48   using WM1 WM2ba WM2ca WM2cb WM2da WM2db WM2dc
49   unfolding Defs by (smt (verit))
50
51 theorem whitespot_d':
52   "[[!¬((K_a (A_ws a)) ∨ (K_a (¬A_ws a)))]([!¬((K_b (A_ws b)) ∨ (K_b (¬A_ws b))))]([
53     [!¬((K_c (A_ws c)) ∨ (K_c (¬A_ws c)))](K_d (A_ws d))))]"
54   using whitespot_d
55   unfolding Defs sledgehammer[verbose]() (* finds proof *)
56   (* reconstruction timeout *)
57   oops
58
59 (* Consistency confirmed by nitpick *)
60 lemma True nitpick [satisfy] oops (* model found *)
61 end

```

Figure 4: Modeling and automating the wise men puzzle with four agents