Faithful Logic Embeddings in HOL—Deep and Shallow

Christoph Benzmüller[1,2(✉)]

[1] Otto-Friedrich-Universität Bamberg, Kapuzinerstraße 16, 96047 Bamberg, Germany
[2] Freie Universität Berlin, Kaiserswerther Str. 16–18, 14195 Berlin, Germany
christoph.benzmueller@uni-bamberg.de, c.benzmueller@fu-berlin.de

Abstract. Deep and shallow embeddings of non-classical logics in classical higher-order logic have been explored, implemented, and used in various reasoning tools in recent years. This paper presents a method for the simultaneous deployment of deep and shallow embeddings of various degrees in classical higher-order logic. This enables flexible, interactive and automated theorem proving and counterexample finding at meta and object level, as well as automated faithfulness proofs between these logic embeddings. The method is beneficial for logic education, research and application and is illustrated here using a simple propositional modal logic. However, this approach is conceptual in nature and not limited to this simple logic context.

Keywords: Logic embeddings · Faithfulness · Automated Reasoning

1 Motivation and Introduction

Deep embeddings of logics, or more generally of domain-specific languages, in a suitable metalogic, such as classical higher-order logic (HOL) [6,28], are typically based on explicitly introduced abstract data types. They essentially axiomatically characterize the structure and the meaning of the new language. Shallow embeddings, on the other hand, identify the embedded languages directly as fragments of the metalogic and they typically require few or no axiom postulates.[1] Deep and shallow embeddings of logics in HOL effectively and elegantly support interactive and automated reasoning with and about such logics within existing HOL theorem proving systems. Consequently, they have received increased attention in recent years. Shallow embeddings, for example, are well suited to automate even quantified modal logics, enabling applications in areas such as

[1] Gibbons and Wu characterize the two notions as follows [35, p.2]: *"[...] a deep embedding consists of a representation of the abstract syntax as an algebraic datatype, together with some functions that assign semantics to that syntax by traversing the algebraic datatype. A shallow embedding eschews the algebraic datatype, and hence the explicit representation of the abstract syntax of the language; instead, the language is defined directly in terms of its semantics."*.

C. Barrett and U. Waldmann (Eds.): CADE 2025, LNAI 15943, pp. 280–301, 2025.
https://doi.org/10.1007/978-3-031-99984-0_16

computational metaphysics [17]. Deep embeddings, on the other hand, are well suited to enable reasoning about the syntactic structure of embedded logics.

The contributions of this paper are manifold: First, a technique for the simultaneous provision of deep and shallow embeddings in metalogic HOL is presented, with the effect that computer-supported faithfulness proofs between the embeddings are not only enabled, but may—at least for simple embedded logics—now even be fully automated in HOL. This is, to the best of the authors' knowledge, a novel result. Second, this paper discusses shallow embedding alternatives differing in the degree of semantic dependencies that are explicitly encoded and maintained. More precisely, heavyweight (maximal) and lightweight (minimal) shallow embeddings are contrasted as extremes, and doing so, the paper hints at the full spectrum that exists between these extremes, which is in fact relevant for practice. For example, while a minimal shallow embedding for a normal modal logic only needs to explicitly encode the crucial semantic dependency on possible worlds [11], the explicit maintenance of an additional dependency, namely on dynamically updated domains of possible worlds, becomes relevant in case the logic is extended by a public announcement operator [13]. Third, the paper hints at opportunities for logic education, and for research, about and with logic embeddings within modern proof assistants and theorem provers for HOL. These range from metalogical studies to object logic applications using interactive proof and disproof, as well as automated theorem proving and counterexample finding, while allowing precise control over the particular level (object- or metalevel) at which reasoning takes place. Fourth, the paper presents some preliminary studies on whether shallow or deep embeddings are better suited for certain reasoning tasks. Fifth, using the Löb axiom of provability logic, the paper briefly sketches how research on the detection of semantic correspondences could possibly be supported by the presented technique in the future.

Since this paper and associated material is intended to be used in logic education, simple propositional modal logic (PML) [19,34,56] has been chosen to illustrate the addressed ideas, and Isabelle/HOL [50] has been chosen as the host proof assistant system for the experiments conducted. Reasons for using Isabelle/HOL include (i) its very good proof automation support with e.g. the `sledgehammer` [20] tool, (ii) the availability of the (counter-)model finder `nitpick` [21], and (iii) its powerful user interface, which allows for user-friendly LaTeX-quality representations. In principle, this paper could have taken advantage of Isabelle/HOL's special support for computer-verifiable paper writing to generate a document directly from the Isabelle/HOL sources, as is the case, for example, with all the articles published in the Archive of Formal Proofs. However, the author has decided against this idea and in favor of screenshots of the source code in order to save space. Furthermore, the visual highlighting and indenting in these screenshots makes them more useful for teaching and illustration purposes by drawing attention to the most important information.

Organization of the paper: Sects. 2 and 3 briefly introduce PML and HOL, before Sects. 4–6 present deep, maximally shallow, and minimally shallow embeddings of PML in HOL. These are connected by automated faithfulness proofs given in Sect. 7. Various experiments with the presented embeddings are presented in Sect. 8, before Sect. 9 discusses related work and Sect. 10 concludes the paper.

2 Propositional Modal Logic (PML)

The *syntax* of PML is defined by the grammar $\varphi, \psi := a \mid \neg\varphi \mid \varphi \supset \psi \mid \Box\varphi$, where $a \in \mathcal{S}$ are propositional symbols declared in a (nonempty) signature $\mathcal{S}$. Further logical connectives can be defined as usual, e.g.: $\varphi \vee \psi := \neg\varphi \supset \psi$, $\varphi \wedge \psi := \neg(\varphi \supset \neg\psi)$, $\Diamond\varphi := \neg\Box\neg\varphi$, $\top := p \supset p$ (for some $p \in \mathcal{S}$), and $\bot := \neg\top$.

A *semantics* for PML, called relational semantics or possible-world semantics, determines the truth of a formula φ relative to a so-called possible world w and a relational model $\mathcal{M}$. A relational model $\mathcal{M}$ is a tuple $\langle \mathcal{W}, \mathcal{R}, \mathcal{V} \rangle$, where $\mathcal{W}$ is a set of possible worlds (the universe), $\mathcal{R}$ is a binary relation on W (called the accessibility relation), $\mathcal{V}$ is a valuation function that assigns a truth value to each symbol $a \in \mathcal{S}$ in each world $w \in \mathcal{W}$, where $\mathcal{S}$ is the signature of propositional symbols. $\langle \mathcal{W}, \mathcal{R} \rangle$ is called a *frame*. Formally, the truth of a formula φ with respect to a given model $\langle \mathcal{W}, \mathcal{R}, \mathcal{V} \rangle$ and a possible world $w \in \mathcal{W}$ is defined as follows:

$\langle \mathcal{W}, \mathcal{R}, \mathcal{V} \rangle, w \models a$ iff $\mathcal{V}(a)(w)$ is true
$\langle \mathcal{W}, \mathcal{R}, \mathcal{V} \rangle, w \models \neg\varphi$ iff not $\langle \mathcal{W}, \mathcal{R}, \mathcal{V} \rangle, w \models \varphi$
$\langle \mathcal{W}, \mathcal{R}, \mathcal{V} \rangle, w \models \varphi \supset \psi$ iff $\langle \mathcal{W}, \mathcal{R}, \mathcal{V} \rangle, w \models \varphi$ implies $\langle \mathcal{W}, \mathcal{R}, \mathcal{V} \rangle, w \models \psi$
$\langle \mathcal{W}, \mathcal{R}, \mathcal{V} \rangle, w \models \Box\varphi$ iff for all v with $\mathcal{R}wv$ we have $\langle \mathcal{W}, \mathcal{R}, \mathcal{V} \rangle, v \models \varphi$

Note that in this recursive definition of truth the only parameter altered is the possible world w in the $\Box$-clause, which states that a formula φ is 'necessary' in a world w if and only if (iff) φ is true at all worlds v that are reachable from w via the accessibility relation $\mathcal{R}$. Based on the above definitions a notion of validity can be defined: A formula φ is valid (in so called base modal logic **K**) iff $\langle \mathcal{W}, \mathcal{R}, \mathcal{V} \rangle, w \models \varphi$ for all $\mathcal{W}$, $\mathcal{R}$, $\mathcal{V}$, and $w \in \mathcal{W}$.

Different systems of modal logic beyond base logic **K** can be obtained by requiring (combinations of) so called *frame conditions*. For example, a frame is called (i) reflexive iff Rxx for all $x \in \mathcal{W}$, (ii) symmetric iff Rxy implies Ryx, for all $x, y \in \mathcal{W}$, and (iii) transitive iff Rxy and Ryz implies Rxz, for all $x, y, z \in \mathcal{W}$. Moreover, there are well-known correspondences, known as Sahlqvist correspondences [55], between such semantic conditions and the validity of certain axiom schemata. For example, (i) corresponds to the validity of axiom schema M: $\Box\varphi \supset \varphi$, (ii) to B: $\varphi \supset \Box\Diamond\varphi$ and (iii) to 4: $\Box\varphi \supset \Box\Box\varphi$.

A *proof system* for modal logic **K** is obtained by adding the schematic necessitation rule, stating that 'if φ is a theorem of **K**, then so is $\Box\varphi$' and the distribution axiom schema $\Box(\varphi \supset \psi) \supset (\Box\varphi \supset \Box\psi)$ to the principles of classical propositional logic. The latter can be captured by a simple Hilbert calculus [57,63] consisting of the modes ponens rule, 'if φ and $\varphi \supset \psi$ are theorems (of **K**), then so is ψ', and the three axiom schemata H1: $\varphi \supset (\psi \supset \varphi)$, H2: $(\varphi \supset (\psi \supset \gamma)) \supset ((\varphi \supset \psi) \supset (\varphi \supset \gamma))$ and H3: $(\neg\varphi \supset \neg\psi) \supset (\psi \supset \varphi)$).

Well known modal logics beyond **K** (which works without additional frame conditions) include the logic **S4** with e.g. the frame conditions (i) and (iii) and the logic **S5** with e.g. the frame conditions (i), (ii) and (iii). Thus, by adding the schematic axioms (M)+(4) or (M)+(B)+(4) to the proof system for modal logic **K** above, proof systems for **S4** and **S5** are obtained.

3 Classical Higher-Order Logic (HOL)

The *syntax* of HOL is defined by $s,t := p_\alpha \mid X_\alpha \mid (\lambda X_\alpha s_\beta)_{\alpha\Rightarrow\beta} \mid (s_{\alpha\Rightarrow\beta}t_\alpha)_\beta$, where $\alpha,\beta,o \in \mathcal{T}$ and with $\mathcal{T}$ being a set of *simple types* defined by $\alpha,\beta := o \mid i \mid (\alpha \Rightarrow \beta)$. Type o, called `bool` in Isabelle/HOL, denotes truth values, i individuals, and $\Rightarrow$ is the function type constructor. The p_α in the syntax of HOL are typed constants symbols declared in the sets S_α of a given signature $\mathscr{S} = (S_\alpha)_{\alpha\in\mathcal{T}}$. The X_α are typed variables (distinct from the p_α). Complex HOL terms are constructed from given HOL terms via λ-abstraction $(\lambda X_\alpha s_\beta)_{\alpha\Rightarrow\beta}$ and function application $(s_{\alpha\Rightarrow\beta}t_\alpha)_\beta$. HOL is thus a logic of terms defined on top of the simply typed λ-calculus, and terms of type o are called *formulas*. The type of each term is given as a subscript and may be omitted if obvious in context. It is assumed that signature $\mathscr{S}$ contains the following *primitive logical connectives*: $\neg_{o\Rightarrow o}, \vee_{o\Rightarrow o\Rightarrow o}$, $=_{\alpha\Rightarrow\alpha\Rightarrow o}$ (short: $=^\alpha$) and $\Pi_{(\alpha\Rightarrow o)\Rightarrow o}$ (short: Π^α). Further logical connectives or shorthand notations can be introduced as usual; e.g. $\longrightarrow_{o\Rightarrow o\Rightarrow o} := \lambda X_o \lambda Y_o(\neg X \vee Y)$ and $\forall X_\alpha\, \varphi_o := \Pi^\alpha(\lambda X_\alpha\, \varphi)$.

The definition of a *semantics* for HOL starts with the notion of a *frame* $\mathcal{D}$, a collection $\{\mathcal{D}_\alpha\}_{\alpha\in T}$ of nonempty sets $\mathcal{D}_\alpha$, such that $\mathcal{D}_o = \{T, F\}$ (for the truth values true and false). $\mathcal{D}_i$ is chosen freely and $\mathcal{D}_{\alpha\Rightarrow\beta}$ are collections of functions mapping $\mathcal{D}_\alpha$ into $\mathcal{D}_\beta$. 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 $\mathcal{D}_\alpha$, called the *denotation* of p_α. The logical connectives $\neg, \vee, \Pi$ and $=$ are given their expected standard denotations, for example, $I(\neg)$ is such that $I(\neg)(T) = F$ and $I(\neg)(F) = T$, and $I(\Pi^\alpha)$ is such that for all $s \in \mathcal{D}_{\alpha\Rightarrow o}$ we have $I(\Pi^\alpha)(s) = T$ iff $s(a) = T$ for all $a \in \mathcal{D}_\alpha$.

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

The *denotation* $[\![s_\alpha]\!]^{M,g}$ 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 by $[\![p_\alpha]\!]^{\mathcal{M},g} = I(p_\alpha)$, $[\![X_\alpha]\!]^{\mathcal{M},g} = g(X_\alpha)$, $[\![(s_{\alpha\Rightarrow\beta}t_\alpha)_\beta]\!]^{\mathcal{M},g} = [\![s_{\alpha\Rightarrow\beta}]\!]^{\mathcal{M},g}([\![t_\alpha]\!]^{\mathcal{M},g})$, and $[\![(\lambda X_\alpha s_\beta)_{\alpha\Rightarrow\beta}]\!]^{\mathcal{M},g}$ = the $f : \mathcal{D}_\alpha \longrightarrow \mathcal{D}_\beta$ s.t. $\forall d \in \mathcal{D}_\alpha : f(d) = [\![s_\beta]\!]^{\mathcal{M},g[d/X_\alpha]}$. Note that from the given definitions it e.g. follows $[\![\forall X_\alpha\varphi_o]\!]^{\mathcal{M},g} = T$ iff$[\![\varphi_o]\!]^{\mathcal{M},g[d/X_\alpha]} = T$ for all $d \in \mathcal{D}_\alpha$.

In a *standard model* a domain $\mathcal{D}_{\alpha\Rightarrow\beta}$ is defined as the set of all total functions from $\mathcal{D}_\alpha$ to $\mathcal{D}_\beta$: $\mathcal{D}_{\alpha\Rightarrow\beta} = \{f \mid f : \mathcal{D}_\alpha \Rightarrow \mathcal{D}_\beta\}$. In a *Henkin model* (or general model) [2,39] 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, it is required that every term still denotes.

s_o is *valid in* $\mathcal{M}$ *under assignment* g, denoted as $\mathcal{M}, g \models^{\mathtt{HOL}} s_o$, iff $[\![s_o]\!]^{\mathcal{M},g} = T$. s_o is *valid in* $\mathcal{M}$, denoted as $\mathcal{M} \models^{\mathtt{HOL}} s_o$, iff $\mathcal{M}, g \models^{\mathtt{HOL}} s_o$ for all assignments g, and s_o is *valid*, denoted as $\models^{\mathtt{HOL}} s_o$, iff s_o is valid in all Henkin models $\mathcal{M}$.

Each standard model is obviously also a Henkin model. Consequently, if $\models^{\mathtt{HOL}} s_o$, then s_o is also valid in all standard models.

Due to Gödel [36], sound and complete mechanizations of HOL with standard semantics cannot be achieved. For HOL with Henkin semantics, however, sound and complete calculi exist; see e.g. [7,9]. Relevant for this paper is that Isabelle/HOL is also sound and complete for HOL, and that HOL as presented is sufficiently well suited as a metalogic for the work presented in this paper.

4 Deep Embedding of PML in HOL

This section presents a deep embedding of PML in HOL. Both the PML language and its semantics are formalized very closely following the definitions given in Sect. 2. This formalization, and all others presented in this paper, are based on some uniformly assumed conventions, resp. preliminaries. These preliminaries are provided in a file `PMLinHOL_preliminaries.thy`, see Fig. 1. Useful comments are provided; they are encapsulated in orange delimiters $-\!< >$.

Relevant at this point is that a type $\mathcal{S}$ is introduced to represent the signature $\mathcal{S}$ of PML, i.e. a (finite or infinite) set of propositional logic symbols; type `w` is the type of possible worlds, $\mathcal{W} := \texttt{w} \Rightarrow \texttt{bool}$ is the predicate type of possible worlds (we regard predicates of this type as characteristic functions for the associated sets of possible worlds), $\mathcal{R} := \texttt{w} \Rightarrow \texttt{w} \Rightarrow \texttt{bool}$ is the type of accessibility relations between possible worlds, and $\mathcal{V} := \mathcal{S} \Rightarrow \texttt{w} \Rightarrow \texttt{bool}$ is the type of evaluation functions (these definitions are shown in lines 4–9). Well known predicates and functions on relations are introduced (in lines 11–19), and $\forall w{:}W.\,\varphi$ (see lines 21–22) is a W-predicate guarded quantifier that unfolds into $\forall w.\,W\,w \longrightarrow \varphi$, where W is of type $\mathcal{W}$. An important remark concerns the sharing of, e.g., the signature type $\mathcal{S}$ between all embeddings introduced in this paper. This sharing is essential for enabling the faithfulness proofs between the embeddings as presented in Sect. 7.

The deep embedding of PML in HOL is defined in file `PMLinHOL_deep.thy` and presented at top of Fig. 2. After importing `PMLinHOL_preliminaries.thy`, this file starts out with defining an abstract datatype[2] introducing the inductively defined logic language PML, which introduces atomic formulas a^d (for $a \in \mathcal{S}$), negated formulas $\neg^d\varphi$, implication formulas $\varphi \supset^d \psi$, and boxed formulas $\Box^d\varphi$ (see line 4). The superscript ^{d} is used to mark all deeply embedded connectives so that they can be properly distinguished from their counterparts defined in the following sections. In these sections, the alternative superscripts ^{s} (maximal shallow) and ^{m} (minimal shallow) are used accordingly.

In accordance with Sect. 2 the further logical connectives $\vee^d$, $\wedge^d$, $\Diamond^d$, $\top^d$, and $\bot^d$ are introduced as defined terms (lines 6–10).[3] Then the primitive recursive definition of truth in a world w for a given model $\langle W, R, V\rangle$, termed `RelativeTruthD` and noted as $\langle W, R, V\rangle, w \models^d \varphi$, is given (lines 12–16). This definition follows one to one, even in presentation style, the definition introduced in Sect. 2. Based on this, the validity $\models^d \varphi$ of a deeply embedded formula φ is defined as relative truth w.r.t. all sets of worlds W, accessibility relations R, valuation functions V, and worlds $w \in W$ (line 18). Finally, in order to support a convenient, simultaneous unfolding of the introduced definitions, a bag `DefD` is defined (in lines 20–21), which bundles them and adds them also to the simplifier `simp` of Isabelle/HOL.

[2] Such a datatype declaration corresponds to, and unfolds into, a rather large set of axioms in HOL that could be postulated instead. Using the Isabelle/HOL command `print_theorems`, this set can be inspected by the user.

[3] Abbreviations could be used instead. They are always unfolded implicitly, while definition unfolding is an explicit operation, leading to more options for control. In

```
1  theory PMLinHOL_preliminaries  imports Main                    (* Christoph Benzmüller, 2025 *)
2  begin
3  —‹Type declarations common for both the deep and shallow embedding›
4   typedecl w —‹Type for possible worlds›
5   typedecl 𝒮 —‹Type for propositional constant symbols›
6   consts p::𝒮 q::𝒮 —‹Some propositional constant symbols›
7   type_synonym 𝒲 = "w⇒bool" —‹Type for sets of possible worlds›
8   type_synonym ℛ = "w⇒w⇒bool" —‹Type for accessibility relations›
9   type_synonym 𝒱 = "𝒮⇒w⇒bool" —‹Type for valuation functions›
10 —‹Some useful predicates for accessibility relations›
11  abbreviation(input) "reflexive     ≡ λR::ℛ. ∀x. R x x"
12  abbreviation(input) "symmetric     ≡ λR::ℛ. ∀x y. R x y ⟶ R y x"
13  abbreviation(input) "transitive    ≡ λR::ℛ. ∀x y z. (R x y ∧ R y z)  ⟶ R x z"
14  abbreviation(input) "equivrel      ≡ λR::ℛ. reflexive R ∧ symmetric R ∧ transitive R"
15  abbreviation(input) "irreflexive   ≡ λR::ℛ. ∀x. ¬R x x"
16  abbreviation(input) "euclidean     ≡ λR::ℛ. ∀x y z. R x y ∧ R x z ⟶ R y z"
17  abbreviation(input) "wellfounded   ≡ λR::ℛ. ∀P::𝒲. (∀x. (∀y. R y x ⟶ P y) ⟶ P x) ⟶  (∀x. P x)"
18  abbreviation(input) "converserel   ≡ λR::ℛ. λy. λx. R x y"
19  abbreviation(input) "conversewf    ≡ λR::ℛ. wellfounded (converserel R)"
20 —‹Bounded universal quantifier: ‹∀x:W. φ› stands for ‹∀x. W x ⟶ φ x››
21  abbreviation(input) "BAll W φ ≡ ∀x::w. W x ⟶ φ x" syntax "BAll"::"pttrn⇒𝒲⇒bool⇒bool" ("(3∀(_/:_)./_)" [0,0,10]10)
22  translations "∀x:W. φ" ⇌ "CONST BAll W (λx. φ)"
23 —‹Backward implication; useful to have› abbreviation(input) Bimp (infixr "⟵" 50) where "φ ⟵ ψ ≡ ψ ⟶ φ"
24 —‹Further settings› declare[[syntax_ambiguity_warning=false]] nitpick_params[user_axioms,expect=genuine]
25 end
```

Fig. 1. Preliminaries.

5 Maximal Shallow Embedding of PML in HOL

The shallow embedding of PML in HOL presented in the middle part of Fig. 2, which is formalized in file `PMLinHOL_shallow.thy`, is a *maximal* (heavyweight) shallow embedding in the following sense: PML formulas φ are associated here with HOL predicate terms φ_σ of type $\sigma := \mathcal{W} \Rightarrow \mathcal{R} \Rightarrow \mathcal{V} \Rightarrow \mathtt{w} \Rightarrow \mathtt{bool}$ (see line 4), thus explicitly encoding the entire semantic dependencies of a PML formula φ_σ on a set of worlds W, an accessibility relation R, a valuation function V and a world w. Remember, that $\mathcal{W}$, $\mathcal{R}$, $\mathcal{V}$, and $\mathtt{w}$ have been uniformly declared for all embeddings presented in this paper in the imported preliminaries file.

Exploiting λ-abstraction and the logical connectives $\neg$ and $\longrightarrow$ from metalogic HOL, the definitions of deeply embedded negation $\neg^s$ and implication $\supset^s$ (in lines 6–8), recursively pass on all the dependencies as mentioned:[4] $\neg^s\varphi := \lambda W R V w.\, \neg(\varphi\, W R V\, w)$ and $\varphi \supset^s \psi := \lambda W R V w.\, (\varphi\, W R V\, w) \longrightarrow (\psi\, W R V\, w)$. Atoms a^s, for $a \in \mathcal{S}$, are evaluated (in line 5) using the explicitly maintained valuation function V, that is, $a^s := \lambda W R V w.\, V\, a\, w$. The only case where parts of the recursively passed on dependencies are actually modified is (in line 8) for $\Box^s\varphi$, since here the world in which φ is recursively evaluated may change with the transition, via the accessibility relation R, to world v: $\Box^s\varphi := \lambda W R V w. \forall v.\ R\, w\, v \longrightarrow (\varphi\, W R V\, v)$. The definition of the other logical connectives $\vee^s$, $\wedge^s$, $\Diamond^s$, $\top^s$ and $\bot^s$ (in lines 10–14) is analogous to the deep embedding and as given in Sect. 3.

any case, readers new to Isabelle/HOL can simply focus on the indented definition equations here (and in other figures), at least at first reading.

[4] Dot-notation is used, whereby the scope opened by . is reaching as far to the right as consistent with the formula structure. The term $\lambda W R V w.\, \neg(\varphi\, W R V\, w)$ thus corresponds to $\lambda W \lambda R \lambda V \lambda w (\neg(\varphi\, W R V\, w))$.

```
theory PMLinHOL_deep imports PMLinHOL_preliminaries                    (* Christoph Benzmüller, 2025 *)
begin
—‹Deep embedding (of propositional modal logic in HOL)›
 datatype PML = AtmD 𝒮 ("_ᵈ") | NotD PML ("¬ᵈ") | ImpD PML PML (infixr "⊃ᵈ" 93) | BoxD PML ("□ᵈ")
—‹Further logical connectives as definitions›
 definition OrD    (infixr "∨ᵈ" 92) where   "φ∨ᵈψ ≡ ¬ᵈφ ⊃ᵈ ψ"
 definition AndD   (infixr "∧ᵈ" 95) where   "φ∧ᵈψ ≡ ¬ᵈ(φ ⊃ᵈ ¬ᵈψ)"
 definition DiaD   ("◇ᵈ_")          where     "◇ᵈφ ≡ ¬ᵈ(□ᵈ(¬ᵈφ)) "
 definition TopD   ("⊤ᵈ")           where      "⊤ᵈ  ≡  pᵈ ⊃ᵈ pᵈ"
 definition BotD   ("⊥ᵈ")           where      "⊥ᵈ  ≡ ¬ᵈ ⊤ᵈ"
—‹Definition of truth of a formula relative to a model ‹⟨W,R,V⟩› and a possible world w›
 primrec RelativeTruthD :: "𝒲⇒ℛ⇒𝒱 ⇒w⇒PML⇒bool" ("⟨_,_,_⟩,_ ⊨ᵈ _") where
    "⟨W,R,V⟩, w ⊨ᵈ aᵈ       = (V a w)"
  | "⟨W,R,V⟩, w ⊨ᵈ ¬ᵈφ      = (¬ ⟨W,R,V⟩, w ⊨ᵈ φ)"
  | "⟨W,R,V⟩, w ⊨ᵈ φ ⊃ᵈ ψ   = (⟨W,R,V⟩, w ⊨ᵈ φ  ⟶  ⟨W,R,V⟩, w ⊨ᵈ ψ)"
  | "⟨W,R,V⟩, w ⊨ᵈ □ᵈφ      = (∀v:W. R w v ⟶ ⟨W,R,V⟩, v ⊨ᵈ φ)"
—‹Definition of validity›
 definition ValD ("⊨ᵈ _") where "(⊨ᵈ φ) ≡ (∀W R V. ∀w:W. ⟨W,R,V⟩,w ⊨ᵈ φ)"
—‹Collection of definitions in a bag called DefD›
 named_theorems DefD declare OrD_def[DefD,simp] AndD_def[DefD,simp] DiaD_def[DefD,simp]
   TopD_def[DefD,simp] BotD_def[DefD,simp] RelativeTruthD_def[DefD,simp] ValD_def[DefD,simp]
end
```

```
theory PMLinHOL_shallow imports PMLinHOL_preliminaries                 (* Christoph Benzmüller, 2025 *)
begin
—‹Shallow embedding (of propositional modal logic in HOL)›
 type_synonym σ = "𝒲⇒ℛ⇒𝒱⇒w⇒bool"
 definition AtmS::"𝒮⇒σ"     ("_ˢ")          where      "aˢ ≡ λW R V w. V a w"
 definition NegS::"σ⇒σ"     ("¬ˢ")          where    "¬ˢ φ ≡ λW R V w.  ¬(φ W R V w)"
 definition ImpS::"σ⇒σ⇒σ" (infixr "⊃ˢ" 93) where  "φ ⊃ˢ ψ ≡ λW R V w. (φ W R V w) ⟶ (ψ W R V w)"
 definition BoxS::"σ⇒σ"     ("□ˢ")          where    "□ˢ φ ≡ λW R V w. ∀v:W. R w v ⟶ (φ W R V v)"
—‹Further logical connectives as definitions›
 definition OrS    (infixr "∨ˢ" 92) where   "φ ∨ˢ ψ ≡ ¬ˢφ ⊃ˢ ψ"
 definition AndS   (infixr "∧ˢ" 95) where   "φ ∧ˢ ψ ≡ ¬ˢ(φ ⊃ˢ ¬ˢψ)"
 definition DiaS   ("◇ˢ")           where     "◇ˢφ ≡ ¬ˢ (□ˢ (¬ˢφ)) "
 definition TopS   ("⊤ˢ")           where      "⊤ˢ ≡  pˢ ⊃ˢ pˢ"
 definition BotS   ("⊥ˢ")           where      "⊥ˢ ≡ ¬ˢ ⊤ˢ"
—‹Definition of truth of a formula relative to a model ‹⟨W,R,V⟩› and a possible world w›
 definition RelativeTruthS :: "𝒲⇒ℛ⇒𝒱 ⇒w⇒σ⇒bool" ("⟨_,_,_⟩,_ ⊨ˢ _") where "⟨W,R,V⟩,w ⊨ˢ φ ≡ φ W R V w"
—‹Definition of validity›
 definition  ValS ("⊨ˢ _") where "⊨ˢ φ  ≡  ∀W R V. ∀w:W. ⟨W,R,V⟩,w ⊨ˢ φ"
—‹Collection of definitions in a bag called DefS›
 named_theorems DefS declare AtmS_def[DefS,simp] NegS_def[DefS,simp] ImpS_def[DefS,simp]
   BoxS_def[DefS,simp] OrS_def[DefS,simp] AndS_def[DefS,simp] DiaS_def[DefS,simp] TopS_def[DefS,simp]
   BotS_def[DefS,simp] RelativeTruthS_def[DefS,simp] ValS_def[DefS,simp]
end
```

```
theory PMLinHOL_shallow_minimal  imports PMLinHOL_preliminaries        (* Christoph Benzmüller, 2025 *)
begin
—‹The acessibility relation R and the valuation function V are introduced as constants at the meta-level HOL›
 consts R::ℛ V::𝒱
—‹Shallow embedding (of propositional modal logic in HOL)›
 type_synonym σ = "w⇒bool"
 definition AtmM::"𝒮⇒σ"     ("_ᵐ")          where      "aᵐ ≡ λw. V a w"
 definition NegM::"σ⇒σ"     ("¬ᵐ")          where    "¬ᵐ φ ≡ λw.  ¬φ w"
 definition ImpM::"σ⇒σ⇒σ" (infixr "⊃ᵐ" 93) where  "φ ⊃ᵐ ψ ≡ λw. φ w ⟶ ψ w"
 definition BoxM::"σ⇒σ"     ("□ᵐ")          where    "□ᵐ φ ≡ λw. ∀v. R w v ⟶ φ v"
—‹Further logical connectives as definitions›
 definition OrM    (infixr "∨ᵐ" 92) where   "φ ∨ᵐ ψ ≡ ¬ᵐφ ⊃ᵐ ψ"
 definition AndM   (infixr "∧ᵐ" 95) where   "φ ∧ᵐ ψ ≡ ¬ᵐ(φ ⊃ᵐ ¬ᵐψ)"
 definition DiaM   ("◇ᵐ_")          where     "◇ᵐφ ≡ ¬ᵐ (□ᵐ (¬ᵐφ)) "
 definition TopM   ("⊤ᵐ")           where      "⊤ᵐ ≡ pᵐ ⊃ᵐ pᵐ"
 definition BotM   ("⊥ᵐ")           where      "⊥ᵐ ≡ ¬ᵐ ⊤ᵐ"
—‹Definition of truth of a formula relative to a model ‹⟨W,R,V⟩› and a possible world w›
 definition RelativeTruthM :: "w⇒σ⇒bool" ("_ ⊨ᵐ _") where "w ⊨ᵐ φ ≡ φ w"
—‹Definition of validity›
 definition  ValM ("⊨ᵐ _") where "⊨ᵐ φ  ≡  ∀w::w. w ⊨ᵐ φ"
—‹Collection of definitions in a bag called DefM›
 named_theorems DefM declare AtmM_def[DefM,simp] NegM_def[DefM,simp] ImpM_def[DefM,simp]
   BoxM_def[DefM,simp] OrM_def[DefM,simp] AndM_def[DefM,simp] DiaM_def[DefM,simp] TopM_def[DefM,simp]
   BotM_def[DefM,simp] RelativeTruthM_def[DefM,simp] ValM_def[DefM,simp]
end
```

Fig. 2. PML in HOL: deep (top), maximal shallow (mid), and minimal shallow (bot).

Truth in a world w for a given model $\langle W, R, V\rangle$, called `RelativeTruthS` and noted as $\langle W, R, V\rangle, w \models^s \varphi$, is then defined (see line 16) as the application of φ to W, R, V and w. Analogous to the deep embedding from above, validity $\models^s \varphi$ of an embedded formula φ is defined as relative truth w.r.t. all sets of worlds W, accessibility relations R, valuation functions V, and worlds $w \in W$ (line 18). Also analogous to above, a bag `DefS` is defined (lines 20–22), which conveniently bundles all the definitions given in this file and adds them to the simplifier. Note that not a single axiom is being introduced in this maximal shallow embedding, with the effect, that PML is being characterized here as a proper, albeit quite heavyweight, fragment of the language of HOL.

6 Minimal Shallow Embedding of PML in HOL

The maximal shallow embedding presented in Sect. 5 explicitly maintains more dependencies than seemingly necessary, since the set of worlds W, the accessibility relation R, and the valuation function V are never modified in any of the given definitions, but simply passed on. Therefore, it makes sense to treat these static dependencies as parameters in the metalogic HOL. This is the core idea of the *minimal* (lightweight) shallow embedding of PML in HOL as shown at the bottom of Fig. 2 and formalized in file `PMLinHOL_shallow_minimal.thy`.

To implement this idea, W is now implicitly identified with type `w` of HOL at the metalevel, and the constant symbols `R` and `V` are introduced in metalogic HOL to represent the semantic notions R and V of PML (see line 4). Most importantly, the modified HOL type $\sigma := \mathtt{w} \Rightarrow \mathtt{bool}$ is now taken as the type of the embedded formulas φ_σ, indicating that PML formulas in this minimal embedding now depend only on possible worlds.[5] Hence, instead of maintaining dependencies on W, R, V, and w, the definitions of the logical connectives a^m, $\neg^m$, $\supset^m$, and $\Box^m$ now only explicitly maintain a dependency on worlds w (lines 7–10). Their definitions are thus, where `V` and `R` are the above-mentioned constant symbols in metalogic HOL: $a^s := \lambda w.\, \mathtt{V}\, a\, w$, $\neg^m \varphi := \lambda w. \neg(\varphi\, w)$ and $\varphi \supset^m \psi := \lambda w.\, \varphi\, w \longrightarrow \psi\, w$, and $\Box^m \varphi := \lambda w. \forall v.\, \mathtt{R}\, w\, v \longrightarrow \varphi\, v$. The definition of the other logical connectives $\vee^m$, $\wedge^m$, $\Diamond^m$, $\top^m$, and $\bot^m$ (in lines 12–16) is analogous to before. Truth in a world w, for a given model $\langle W, R, V\rangle$ maintained at the meta-level HOL, called `RelativeTruthM` and noted as $w \models^m \varphi$, is then defined (see line 18) as the application of the embedded formula φ to w. Consequently, validity $\models^m \varphi$ of an embedded formula φ is defined as relative truth w.r.t. to all worlds w (line 20). The bag `DefM` defined (in lines 22–24) is analogous to `DefD` and `DefS` from before. Again, not a single axiom is introduced. So PML is once again characterized as a proper fragment of HOL, but now more lightweight.

[5] Note that this is also the core idea of the well-known standard translation of PML into first-order logic [19], which, in fact, is implemented here by exploiting λ-conversion and compositionality directly in the metalogical HOL without the need for explicit recursive definitions.

7 Faithfulness Automated

The deep embedding presented in Sect. 4 and the maximal and minimal shallow embeddings presented in Sect. 5 and Sect. 6 have been carefully designed to allow for automated proofs of mutual faithfulness within the metalogic HOL. A relevant aspect is the sharing of the signature $\mathcal{S}$ of propositional symbols between all embeddings. The automated faithfulness proofs are shown in Fig. 3.

```
1  theory PMLinHOL_faithfulness imports PMLinHOL_deep PMLinHOL_shallow PMLinHOL_shallow_minimal (* C.B., 2025 *)
2  begin
3  —‹Mappings: deep to maximal shallow and deep to minimal shallow›
4    primrec DpToShMax ("(_)") where "(a^d) = a^s" | "(¬^d φ) = ¬^s (φ)" | "(φ ⊃^d ψ) = (φ) ⊃^s (ψ)" | "(□^d φ) = □^s (φ)"
5    primrec DpToShMin ("[_]") where "[a^d] = a^m" | "[¬^d φ] = ¬^m [φ]" | "[φ ⊃^d ψ] = [φ] ⊃^m [ψ]" | "[□^d φ] = □^m [φ]"
6  —‹Proving faithfulness between deep and maximal shallow›
7    theorem Faithful1a:  "∀W R V. ∀w:W. ⟨W,R,V⟩,w ⊨^d φ ⟷ ⟨W,R,V⟩,w ⊨^s (φ)"   apply induct by auto
8    theorem Faithful1b:  "⊨^d φ ⟷ ⊨^s (φ)"                                  using Faithful1a by auto
9  —‹Proving faithfulness between deep and minimal shallow›
10   theorem Faithful2:   "∀w. (λx::w. True),R,V⟩,w ⊨^d φ ⟷ w ⊨^m [φ]"          apply induct by auto
11 —‹Proving faithfulness maximal shallow and minimal shallow›
12   theorem Faithful3:   "∀w. (λx::w. True),R,V⟩,w ⊨^s (φ) ⟷ w ⊨^m [φ]"       apply induct by auto
13 —‹Additional check for soundness for the minimal shallow embedding›
14   lemma Sound1: "⊨^m ψ ⟷ (∃φ. ψ=[φ] ∧ ⊨^d φ)" by (smt Faithful2 DefM DefD RelativeTruthD.simps ext[of ψ "[x⊃^dx]"])
15   lemma Sound2: "⊨^m ψ ⟷ (∃φ. ψ=[φ] ∧ ⊨^m [φ])" using Sound1 by blast
16 end
```

Fig. 3. Automated faithfulness proofs between different embeddings of PML in HOL.

The starting points are the mapping functions $(\!|.|\!)$ and $[\![.]\!]$ (in lines 4–5), which map deeply embedded formulas to maximally and minimally embedded counterparts, respectively. Using these mappings the following faithfulness statements, which are formulated relative to the deeply embedded proof language PML, are proved automatically by induction (see lines 7–12):

$$\forall W R V.\forall w{:}W.\ \langle W,R,V\rangle, w \models^d \varphi \longleftrightarrow \langle W,R,V\rangle, w \models^s (\!|\varphi|\!) \quad (\texttt{Faithful1a})$$

$$\models^d \varphi \longleftrightarrow \models^s (\!|\varphi|\!) \quad (\texttt{Faithful1b})$$

$$\forall w.\ \langle(\lambda x_w.\top),\mathtt{R},\mathtt{V}\rangle, w \models^d \varphi \longleftrightarrow w \models^m [\![\varphi]\!] \quad (\texttt{Faithful2})$$

$$\forall w.\ \langle(\lambda x_w.\top),\mathtt{R},\mathtt{V}\rangle, w \models^s (\!|\varphi|\!) \longleftrightarrow w \models^m [\![\varphi]\!] \quad (\texttt{Faithful3})$$

Theorems `Faithful1a` and `Faithful1b` are rather obvious; they establish the desired faithfulness correspondence between deeply embedded formulas and their maximally shallow counterparts. Theorems `Faithful2` and `Faithful3`, on the other hand, require some explanation. The former states that if W is taken as $\lambda x_{\mathtt{w}}.\top$, i.e. the universe of possible worlds denoted by type `w` in metalogic HOL, and if R and V are replaced by the metalogical constant symbols `R` and `V` in HOL, denoting accessibility between worlds and evaluation of propositional symbols from $\mathcal{S}$ in worlds, then each deeply embedded formula φ faithfully corresponds to its minimally shallow counterpart $[\![\varphi]\!]$. The latter formulates an analogous correspondence between maximal shallow formulas $(\!|\varphi|\!)$ and minimal shallow formulas $[\![\varphi]\!]$. Together, these theorems show the intended correspondences between

all the introduced embeddings, and they provide a valuable foundation for working with them simultaneously and interchangeably in applications. Moreover, the surjection theorems `Sound1` and `Sound2` can be automated by `sledgehammer` for the minimal shallow embedding, but not yet for the maximal one.

8 Experiments

The possibilities that the linked embeddings offer, e.g., for logic education, are illustrated with some experiments. The experiments performed also test the soundness of our embeddings, and assess proof automation and counterexample finding for them with `sledgehammer` and `nitpick`. The embedding that has performed best in this regard is the minimal shallow embedding. This is why file `PMLinHOL_shallow_minimal_tests.thy`, see Fig. 4, is presented here.[6]

```
1  theory PMLinHOL_shallow_minimal_tests imports PMLinHOL_shallow_minimal      (* Christoph Benzmüller, 2025 *)
2  begin
3  —‹Hilbert calculus: proving that the schematic axioms and rules are implied by the embedding›
4    lemma H1: "⊨ᵐ φ ⊃ᵐ (ψ ⊃ᵐ φ)" by auto
5    lemma H2: "⊨ᵐ (φ ⊃ᵐ (ψ ⊃ᵐ γ)) ⊃ᵐ ((φ ⊃ᵐ ψ) ⊃ᵐ (φ ⊃ᵐ γ)) " by auto
6    lemma H3: "⊨ᵐ (¬ᵐφ ⊃ᵐ ¬ᵐψ) ⊃ᵐ (ψ ⊃ᵐ φ)" by auto
7    lemma MP: "⊨ᵐ φ ⟹ ⊨ᵐ (φ ⊃ᵐ ψ) ⟹ ⊨ᵐ ψ" by auto
8  —‹Reasoning with the Hilbert calculus: interactive and fully automated›
9    lemma HCderived1: "⊨ᵐ (φ ⊃ᵐ φ)" —‹sledgehammer(H1 H2 H3 MP) returns: by (metis H1 H2 MP)›
10     proof -  have 1: "⊨ᵐ φ ⊃ᵐ ((ψ ⊃ᵐ φ) ⊃ᵐ φ)" using H1 by auto
11              have 2: "⊨ᵐ (φ ⊃ᵐ ((ψ ⊃ᵐ φ) ⊃ᵐ φ)) ⊃ᵐ ((φ ⊃ᵐ (ψ ⊃ᵐ φ)) ⊃ᵐ (φ ⊃ᵐ φ))" using H2 by auto
12              have 3: "⊨ᵐ (φ ⊃ᵐ (ψ ⊃ᵐ φ)) ⊃ᵐ (φ ⊃ᵐ φ)" using 1 2 MP by meson
13              have 4: "⊨ᵐ φ ⊃ᵐ (ψ ⊃ᵐ φ)" using H1 by auto
14              thus ?thesis using 3 4 MP by meson qed
15   lemma HCderived2: "⊨ᵐ φ ⊃ᵐ (¬ᵐφ ⊃ᵐ ψ) " by (metis H1 H2 H3 MP)
16   lemma HCderived3: "⊨ᵐ (¬ᵐφ ⊃ᵐ φ) ⊃ᵐ φ" by (metis H1 H2 H3 MP)
17   lemma HCderived4: "⊨ᵐ (φ ⊃ᵐ ψ) ⊃ᵐ (¬ᵐψ ⊃ᵐ ¬ᵐφ) " by auto
18 —‹Modal logic: the schematic necessitation rule and distribution axiom are implied›
19   lemma Nec: "⊨ᵐ φ ⟹ ⊨ᵐ □ᵐφ" by (smt DefM)
20   lemma Dist:  "⊨ᵐ □ᵐ(φ ⊃ᵐ ψ) ⊃ᵐ (□ᵐφ ⊃ᵐ □ᵐψ) " by auto
21 —‹Correspondence theory: correct statements›
22   lemma cM:  "reflexive R ⟷ (∀φ. ⊨ᵐ □ᵐφ ⊃ᵐ φ)" by auto
23   lemma cBa: "symmetric R ⟶ (∀φ. ⊨ᵐ φ ⊃ᵐ □ᵐ◇ᵐφ)" by auto
24   lemma cBb: "symmetric R ⟵ (∀φ. ⊨ᵐ φ ⊃ᵐ □ᵐ◇ᵐφ)" by (metis DefM)
25   lemma c4a: "transitive R ⟶ (∀φ. ⊨ᵐ □ᵐφ ⊃ᵐ □ᵐ(□ᵐφ))" by (smt DefM)
26   lemma c4b: "transitive R ⟵ (∀φ. ⊨ᵐ □ᵐφ ⊃ᵐ □ᵐ(□ᵐφ))" by auto
27 —‹Correspondence theory: incorrect statements›
28   lemma "reflexive R ⟶ (∀φ. ⊨ᵐ □ᵐφ ⊃ᵐ □ᵐ(□ᵐφ))" nitpick[card w=3,show_all] oops —‹nitpick: Counterexample›
29 —‹Simple, incorrect validity statements›
30   lemma "⊨ᵐ φ ⊃ᵐ □ᵐφ" nitpick[card w=2, card 𝒮= 1] oops —‹nitpick: Counterexample: modal collapse not implied›
31   lemma "⊨ᵐ □ᵐ( □ᵐφ ⊃ᵐ □ᵐψ) ∨ᵐ □ᵐ( □ᵐψ ⊃ᵐ □ᵐφ)" nitpick[card w=3] oops —‹nitpick: Counterexample›
32   lemma "⊨ᵐ (◇ᵐ(□ᵐ φ)) ⊃ᵐ □ᵐ(◇ᵐ φ)" nitpick[card w=2] oops —‹nitpick: Counterexample›
33 —‹Implied axiom schemata in S5›
34   lemma KB: "symmetric R ⟶ (∀φ ψ.  ⊨ᵐ (◇ᵐ(□ᵐφ)) ⊃ᵐ  □ᵐ(◇ᵐφ))" by auto
35   lemma K4B: "symmetric R ∧ transitive R ⟶ (∀φ ψ. ⊨ᵐ □ᵐ( □ᵐφ ⊃ᵐ  □ᵐψ) ∨ᵐ □ᵐ( □ᵐψ ⊃ᵐ  □ᵐφ))" by (smt DefM)
36 end
```

Fig. 4. Experiments: Hilbert system and Sahlqvist correspondences.

Derived Hilbert Calculus. The first experiments (see lines 4–7) confirm that the schematic axioms `H1-H3` and the modus ponens rule `MP` of the propositional

[6] Corresponding files for the deep and the maximal shallow embeddings are shown in the appendix of [5]. The Isabelle/HOL sources are available at http://logikey.org/tree/master/CoursesAndTutorials/2025-CADE-DeepShallow and also in [15].

Hilbert calculus from Sect. 2 are implied by the minimal shallow embedding (local versions of `H1-H3` and `MP` rule are also implied; this is not shown here). The `auto` tactic, and with it, the simplifier `simp` are used, which has access to the definitions in bundle `DefM`. The φ, ψ and γ are schematic variables of type σ in the metalogic HOL. The proofs were originally found by calling `sledgehammer`.

Hilbert Style Proofs Automated and Interactive. The next experiment (in lines 9–14) demonstrates automated and interactive reasoning using the schematic axioms and rules of the Hilbert calculus introduced earlier. It is shown that the formula schema $\varphi \supset^m \varphi$ is derivable in the given Hilbert calculus. By preventing access to the definitions in the bag `DefM`, e.g. with the command `sledgehammer(H1 H2 H3 MP)`, the automated theorem provers orchestrated by `sledgehammer` can be forced to search for proofs in the given Hilbert calculus only, and they succeed here. They find out that only `H1` and `H2` and `MP` are needed to obtain the result. In the given situation, the reconstruction of a proof from `H1` and `H2` and `MP` is then possible with Isabelle's trusted `metis` tool. For educational purposes it is relevant that proofs from the literature can be easily mapped one to one to proofs in our setting. This is illustrated here with an interactive derivation (in lines 10–14) that is following a proof from a Wikipedia entry [63] on Hilbert systems. Further derived axiom schemata are proved automatically at the level of the Hilbert calculus (in lines 15–16). If such attempts fail (as was initially the case in line 17), then either interactive proof is needed or, as is demonstrated here with the call of `auto`, the unfolding of the definitions of the shallow embedding in metalogic HOL (within `auto`) may in fact help to still find a proof automatically. Depending on prior knowledge and teaching objectives, students can be asked to solely provide interactive proofs in such exercises rather than using automation tools. It is also important to note that Isabelle/HOL's proof tactics support, and in particular the Eisbach package, can be used to further enhance the automation of proofs at the level of embedded logic.[7]

Correspondence Theory Automated. Modal logic textbooks typically include exercises on the well known Sahlqvist correspondences [55] (see lines 22–26). Using the definitions in bag `DefM`, such correspondences are here automated in Isabelle/HOL. More exhaustive experiments on such (and similar) correspondences have been reported in prior work [3,8,43,51].

Disproving Correspondences. Even more interesting than automatically proving the Sahlqvist correspondences is disproving falsely assumed correspondences. For example, when `nitpick` is called on the false conjecture, that in PML models with reflexive accessibility relations R the axiom schema $\Box^m\varphi \supset^m \Box^m\Box^m\varphi$ is implied (see line 28), the counterexample shown in Fig. 5 is reported back.

As prior work demonstrates, see e.g. [8,14,51], `nitpick` often provides intuitive and sometimes even surprising counterexamples when used in conjunction with shallow embeddings.[8] This is not only useful for teaching logic, but can

[7] The Eisbach package was e.g. used in [41,42] to define and automate proof methods for a non-trivial shallow embedding of abstract object theory AOT [64] in HOL.

[8] Prior work [8,43] also illustrates how counterexamples reported by `nitpick` can be systematically converted into statements that verifiably disprove the false conjectures; the method presented there could indeed be fully automated.

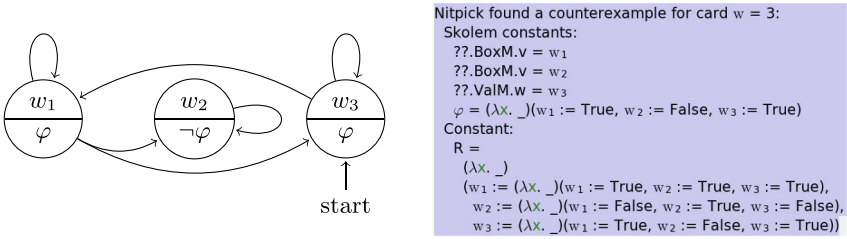

Fig. 5. Counterexample reported by `nitpick` in line 28 of Fig. 4

also be useful for research studies, see e.g. [14]. However, some non-classical logics have only infinite models, and then the finite model finder `nitpick` is no longer applicable; this is e.g. the case for Lorini's T-STIT logic [46], which was embedded in HOL by Meder [47]. Developing infinite model finding tools and integrating them into proof assistants such as Isabelle/HOL is thus an important research challenge. So too is the automated, domain-specific provision of intuitive counterexample diagrams like the one on the left in Fig. 5.

For further false conjectures (in lines 30–32 of Fig. 4), including the modal collapse formula $\varphi \supset^m \Box^m\varphi$, intuitive counterexamples are generated by `nitpick`. Finally, examples of simple modal theorems (in lines 34–35), such as the KB theorem $\Diamond^m\Box^m\varphi \supset^m \Box^m\Diamond^m\varphi$, are automated efficiently using `smt` solvers in combination with definition unfolding in metalogic HOL.

Reasoning at Object Level. Figure 6 presents further experiments. After proving the schematic modal principle $K\Diamond$ (in line 4) an object level proof example [56, Ex. 6.10] is studied (in lines 6–14): $\Box^m p^m \supset^m (\Diamond^m q^m \supset^m \Diamond^m(p^m \wedge^m q^m))$, where p and q are propositional symbols from the PML signature $\mathcal{S}$. This statement is again effectively automated via definition unfolding at the metalevel HOL (line 6), while the presented interactive proof (in lines 8–14) uses the propositional Hilbert calculus from before in combination with the modal rule of necessitation, the distribution axiom of base logic **K** and the newly introduced $K\Diamond$-principle.

```
1  theory PMLinHOL_shallow_minimal_further_tests imports PMLinHOL_shallow_minimal_tests   (* C.B., 2025 *)
2  begin
3  —‹Implied modal principle›
4    lemma K_Dia: "⊨ᵐ (□ᵐ(φ ⊃ᵐ ψ)) ⊃ᵐ ((◇ᵐφ) ⊃ᵐ ◇ᵐψ)"   by auto
5  —‹Example 6.10 of Sider (2009) Logic for Philosophy›
6    lemma T1a: "⊨ᵐ □ᵐpᵐ ⊃ᵐ ((◇ᵐqᵐ) ⊃ᵐ ◇ᵐ (pᵐ ∧ᵐ qᵐ))" by auto  —‹fast automation in meta-logic HOL›
7    lemma T1b: "⊨ᵐ □ᵐpᵐ ⊃ᵐ ((◇ᵐqᵐ) ⊃ᵐ ◇ᵐ (pᵐ ∧ᵐ qᵐ))"
8      proof -                                  —‹alternative interactive proof in modal object logic K›
9        have 1: "⊨ᵐ pᵐ ⊃ᵐ (qᵐ ⊃ᵐ (pᵐ ∧ᵐ qᵐ))"        unfolding AndM_def using H1 H2 H3 MP by metis
10       have 2: "⊨ᵐ □ᵐ(pᵐ ⊃ᵐ (qᵐ ⊃ᵐ (pᵐ ∧ᵐ qᵐ)))"                  using 1 Nec      by metis
11       have 3: "⊨ᵐ □ᵐpᵐ ⊃ᵐ □ᵐ(qᵐ ⊃ᵐ (pᵐ ∧ᵐ qᵐ))"                 using 2 Dist MP  by metis
12       have 4: "⊨ᵐ (□ᵐ(qᵐ ⊃ᵐ (pᵐ ∧ᵐ qᵐ))) ⊃ᵐ ((◇ᵐqᵐ) ⊃ᵐ ◇ᵐ(pᵐ ∧ᵐ qᵐ))" using K_Dia      by metis
13       have 5: "⊨ᵐ □ᵐpᵐ ⊃ᵐ ((◇ᵐqᵐ) ⊃ᵐ ◇ᵐ (pᵐ ∧ᵐ qᵐ))"                 using 3 4 H1 H2 MP by metis
14       thus ?thesis . qed
15 end
```

Fig. 6. Experiments: Simple proof in modal logic **K**.

Larger object-level proof examples could now be presented, studied and automated using the minimal shallow embedding. Prior studies have shown that this approach is generally competitive, in particular, when it comes to automating quantified modal logics; see e.g. the recent experiments reported in [60]. Recipes for embedding modal quantifiers and extensive application studies for quantified modal logics have been presented in prior works; see e.g. [12,17].

```
1  theory PMLinHOL_shallow_minimal_Loeb_tests imports PMLinHOL_shallow_minimal  (* Christoph Benzmüller, 2025 *)
2  begin
3  —‹Löb axiom: with the minimal shallow embedding automated reasoning tools are still partly responsive›
4    lemma Loeb1:  "∀φ. ⊨ᵐ □ᵐ( □ᵐφ ⊃ᵐ φ) ⊃ᵐ □ᵐφ" nitpick[card w=1,card 𝒮=1] oops —‹nitpick: Counterexample.›
5    lemma Loeb2:  "(conversewf R ∧ transitive R) ⟶ (∀φ. ⊨ᵐ □ᵐ(□ᵐφ ⊃ᵐφ) ⊃ᵐ □ᵐφ)"  —‹sh: Proof found› oops
6    lemma Loeb3:  "(conversewf R ∧ transitive R) ⟵ (∀φ. ⊨ᵐ □ᵐ(□ᵐφ ⊃ᵐφ) ⊃ᵐ □ᵐφ)"  —‹sh: No Proof› oops
7    lemma Loeb3a: "conversewf R ⟵ (∀φ. ⊨ᵐ □ᵐ(□ᵐφ ⊃ᵐφ) ⊃ᵐ □ᵐφ)" unfolding DefM by blast
8    lemma Loeb3b: "transitive R ⟵ (∀φ. ⊨ᵐ □ᵐ(□ᵐφ ⊃ᵐφ) ⊃ᵐ □ᵐφ)" —‹sledgehammer: No Proof› oops
9    lemma Loeb3c: "irreflexive R ⟵ (∀φ. ⊨ᵐ □ᵐ(□ᵐφ ⊃ᵐφ) ⊃ᵐ □ᵐφ)"  —‹sledgehammer: Proof found› oops
10 end
```

Fig. 7. Experiments: Studying the Löb-axiom of provability logic.

Experiments with the Löb-Axiom. Figure 7 presents automation experiments with the Löb-axiom of provability logic [24]. A first observation, interesting for education and research, is that via definition unfolding the schematic axioms are automatically converted into corresponding formulas in metalogic HOL. For the Löb-axiom (in line 4) we thus obtain after definition unfolding the HOL formula $\forall \varphi\, w.\, (\forall v.\, \mathtt{R} wv \longrightarrow (\forall z.\, \mathtt{R} vz \longrightarrow \varphi z) \longrightarrow \varphi v) \longrightarrow (\forall v.\, \mathtt{R} wv \longrightarrow \varphi v)$. When using the maximal shallow embedding instead, the Löb-axiom unfolds into the expected much heavier HOL formula $\forall \varphi\, W R V\, x.\, W\, x \longrightarrow (\forall z.\, W\, z \longrightarrow Rxz \longrightarrow (\forall x.\, Wx \longrightarrow Rzx \longrightarrow \varphi\, W R V x) \longrightarrow \varphi\, W R V z) \longrightarrow (\forall z.\, Wz \longrightarrow Rxz \longrightarrow \varphi\, W R V z)$, which also explains why automated reasoning is generally much harder for maximal shallow embeddings than for minimal shallow embeddings. The utilisation of the features of the presented framework enables students and researchers to explore frame properties for non-trivial properties independently and to subsequently simplify them. Proof automation or interactive proof can then be employed to investigate their equivalence to already known properties, as illustrated in Fig. 7 (in lines 5–9). As shown, such proof automation attempts succeed for the following conjectures: (line 5) the converse well-foundedness of the accessibility relation `R` in combination with the transitivity of `R` implies the Löb axiom, (line 7) the Löb axiom implies converse well-foundedness of `R`, and (line 9) the Löb axiom implies the irreflexivity of `R`. Whenever transitivity is an implicant to be proved, proof automation still fails (see lines 6 and 8). Here interactive proof is still required or, alternatively, improved HOL theorem proving systems, in particular provers that can suitably guess non-trivial instantiations for higher-order variables.[9]

Performance: Examples so Far. The performance of `sledgehammer` and `nitpick` for the different embeddings for the example proof problems as discussed so far

[9] For more on this challenge topic see e.g. the experiments reported in [16].

is briefly discussed:[10] *Minimal shallow embedding:* Proofs for 36 out of 38 valid statements (theorems, lemmas and substatements in interactive proofs) were found by calls to `sledgehammer`. `Nitpick` reported counterexamples for 5 out of 5 invalid statements. Reconstruction of `sledgehammer` proofs with trusted tactics failed in 2 cases. *Maximal shallow embedding:* Proofs for 34 out of 38 valid statements were found by `sledgehammer`. Countermodels for 4 out of 5 invalid statements were found by `nitpick`. Reconstruction of `sledgehammer` proofs with trusted tactics failed in 4 cases. *Deep embedding:* Proofs for 32 out of 38 valid statements were found by `sledgehammer`; in particular, there were no proofs delivered for the Löb examples. Countermodels for 5 out of 5 invalid statements were found by nitpick. Reconstruction of proofs failed in 2 cases.

Performance: Formula Classification. Further experiments were carried out in order to assess the performance of the different embeddings more rigorously. These tests focused on the classification of PML formula examples with respect to the logics in the modal logic cube [34] defined by the schematic axioms $T := \Box\varphi \supset \varphi$, $B := \varphi \supset \Box\Diamond\varphi$ and $4 := \Box\varphi \supset \Box\Box\varphi$, respectively by their semantic counterparts $Ts :=$ `reflexiv` R, $Bs :=$ `symmetric` R and $4s :=$ `transitiv` R (with R being the accessibility relation). The example formulas studied were:

F_1: $\Diamond\Diamond\varphi \supset \Diamond\varphi$
F_2: $\Diamond\Box\varphi \supset \Box\Diamond\varphi$
F_3: $\Diamond\Box\varphi \supset \Box\varphi$
F_4: $\Box\Diamond\Box\Diamond\varphi \supset \Box\Diamond\varphi$
F_5: $\Diamond(\varphi \wedge \Diamond\psi)) \supset (\Diamond\varphi \wedge \Diamond\psi)$
F_6: $((\Box(\varphi \supset \psi)) \wedge \Diamond\Box\neg\psi) \supset \neg\Diamond\psi$
F_7: $\Diamond\varphi \supset \Box(\varphi \vee \Diamond\varphi)$
F_8: $\Diamond(\Box\varphi) \supset (\varphi \vee \Diamond\varphi)$
F_9: $(\Box\Diamond\varphi \wedge \Box\Diamond\neg\varphi) \supset \Diamond\Diamond\varphi$
F_{10}: $(\Box(\varphi \supset \Box\psi) \wedge \Box\Diamond\neg\psi) \supset \neg\Box\psi$

The automatically identified weakest logics in the modal cube in which these formulas are valid are as follows: F_1: **K4**, F_2: **KB**, F_3: **KB4**, F_4: **KB**/**K4**, F_5: **K4**, F_6: **KB4**, F_7: **KB4**, F_8: **KT**/**KB**, F_9: **KT**, and F_{10}: **KT**.

Explained in abstract terms, for all $M \subseteq \{T, B, 4\}$, $N \subseteq \{Ts, Bs, 4s\}$ and $F \in \{F_1, \ldots, F_{10}\}$, `sledgehammer` and `nitpick` were called to assess the validity of the following sets of proof problems (each one consisting of $3^2 \times 10 = 80$ entries): $M \models^d F$, $N \models^d F$, $M \models^s F$, $N \models^s F$, $M \models^m F$, $N \models^m F$. `sledgehammer` and `nitpick` were thereby invoked in two modes, with and without prior application of Isabelle/HOL's simplifier `simp` to the proof problem under consideration. Consequently, for each set of problems, a total of 160 calls to each tool were conducted (80 with prior simplification and 80 without).

The results of these experiments, conducted with the same hardware as mentioned in Footnote 10, are presented in Fig. 8. The best results are highlighted. These are the maxima for the informative cases 'counterexample found' and 'no (finite) counterexample found' for `nitpick`, and 'proof found' for `sledgehammer`. For the 'unknown' and 'no proof' cases, which don't provide any useful information, the minima are shown as the best results. A overall score is reported, which was computed by adding the results obtained for the informative cases.

[10] A MacBook Pro 16,1 (6-Core Intel Core i7 processor, 2,6 GHz, 6 Cores, 256 KB L2 Cache per Core, 12 MB L3 Cache, 16 GB Memory) and Isabelle 2025 in standard configuration was used, which sets a 30 s prover timeout for `sledgehammer`.

	nitpick			sledgehammer		
	A	B	C	D	E	score
Problems	counterex.	no counterex.	unknown	proof found	no proof	
$M \models^m F$	**84(42/42)**	72(36/36)	4(2/2)	73(**38**/35)	87(**42**/45)	229
$N \models^m F$	**84(42/42)**	**76(38/38)**	**0(0/0)**	**76(38/38)**	**84(42/42)**	**236**
$M \models^s F$	32(16/16)	0(0/0)	128(64/64)	32(24/8)	128(56/72)	64
$N \models^s F$	**84(42/42)**	0(0/0)	76(38/38)	66(**38**/28)	94(**42**/52)	150
$M \models^d F$	16(10/6)	74(0/74)	70(70/**0**)	33(16/17)	127(64/63)	123
$N \models^d F$	32(8/24)	56(0/56)	72(72/**0**)	70(**38**/32)	90(**42**/48)	158

Fig. 8. Performance results of the automated PML formula classification task conducted with `nitpick` and `sledgehammer` (explanation: the first entry 73(**38**/35) in column D, e.g., states that 73 proofs were reported in total by `sledgehammer` for the problem set $M \models^m F$, with 38 proofs found with prior application of the simplifier `simp` and 35 without; analogous for the other entries).

The results obtained are briefly discussed. Perfect experimentation results were obtained for the problem set $N \models^m F$, that is, the minimal shallow embedding when used in combination with the semantical frame conditions Ts, Bs and $4s$. In this setting `nitpick` either provided a counterexample for each formula classification task or reported that there was no (finite) counterexample. The latter judgments were then supplemented by proofs from `sledgehammer` that confirmed the validity of these formulas. The second best performance was achieved with the minimal shallow embedding when used in combination with the axiom schemata T, B and 4. The maximal shallow embedding returned some good results for the problem set $N \models^s F$ regarding counterexample finding with `nitpick` and regarding theorem proving with `sledgehammer` when prior simplification was applied. However, in this setting, `nitpick` was unable to identify any formulas with no (finite) countermodels. For the deep embedding the performance results obtained with `nitpick` regarding counterexample finding significantly dropped in comparison to the minimal shallow embedding. The numbers reported for the deep embedding regarding the identification of formulas with no (finite) countermodels are still good though.[11] Interestingly, the maximal number of 38 proofs was still obtained by `sledgehammer` in the deep embedding when prior simplification was applied. Future work should extend the experiments presented to more challenging test formulas F and also include the further axiom schemata (and their corresponding semantic conditions) from the modal logic cube [34]. Additionally, it seems useful to replay, now using all the embeddings as provided in this paper, the metalogical cube verification experiments reported in prior work [3,8].

[11] However, some seemingly self-contradictory answers by `nitpick` in this setting with and without prior simplification for some problems require further consideration.

9 Related Work and Implications for LogiKEy

The notions of deep and shallow embeddings appear to originate from the work of Boulton et al. [25], and the connections between the two have been discussed in numerous related papers, see e.g. [40,53,61,62] and the references therein. However, to the best of the author's knowledge, no previous work has succeeded in integrating deep and (various degrees of) shallow embeddings in the form presented and, in particular, in enabling mutual faithfulness proofs between them.

Deep Embeddings. Exemplary developments and applications of non-trivial deep logic embeddings include self-verifications of the HOL light logic [1,38], a formalization of the Core Why3 logic [29], and a provably sound implementation of a differential dynamic logic [23], to name a few. For examples of deep embeddings of logics and their calculi in the Coq proof assistant [18], see e.g. [58] and the references therein, and Isabelle/HOL examples are presented in [22,31–33]. Further related work on the self-verification or bootstrapping of theorem provers can be found in [26], and more recently in [27] in the context of Lean [48].

Shallow Embeddings. Exemplary work in this category, which Nipkow called the extensional approach [49], includes a mechanization of HOL reasoning in first-order set theory [37], the techniques presented in [40] to overcome some problems of shallow embedding for reasoning over the syntax of embedded languages, a shallow embedding for pure type systems in first-order logic [30], and various other works such as [44,54]; see also the references given in [62]. Note also that the automation support for various quantified non-classical logics as realized in the HOL theorem prover Leo3 [59,60] is based on minimal shallow embeddings.

Implications for LogiKEy. The logic-pluralistic knowledge and reasoning methodology LogiKEy [10], see Fig. 9, leverages higher-order logic (HOL) as a unifying metalogic to enable the modeling of diverse object logics, so far, primarily

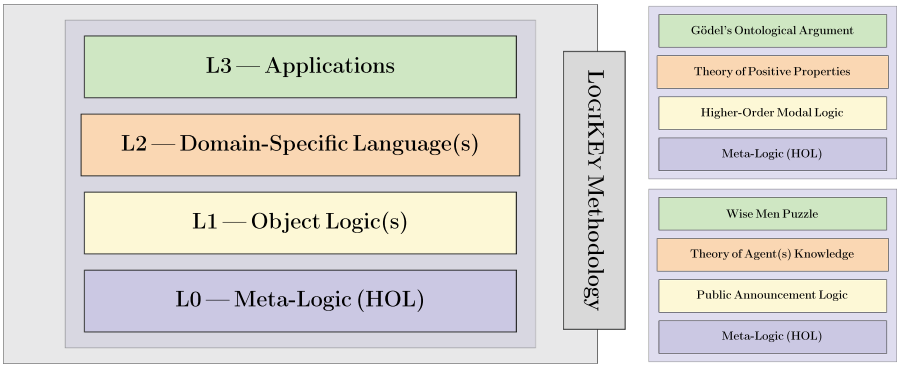

Fig. 9. The logic-pluralistic knowledge representation and reasoning methodology LogiKEy (left) and two concrete applications of it (right): Studies on Gödel's ontological argument [17] (top) and an automation of the wise men puzzle [13] (bottom).

via using shallow semantic embeddings. LogiKEy has been used in a range of application domains and contexts. Very different kinds of object logics have been embedded LogiKEy over the past decade, including, for example, free logic [14], modal higher-order logic [17], public announcement logic [13], conditional logic [4] and dyadic deontic logic with a preference-based semantics [51].

This paper shows how the LogiKEy approach could benefit in the future from providing faithful, deep and shallow embeddings of all object logics under consideration. This would allow the embeddings to be used interchangeably in applications. The technique presented has also the potential to replace or complement pen-and-paper faithfulness proofs in prior works, such as the ones reported in [4,12,13], by computer-verified proofs in Isabelle/HOL. This in turn will be relevant for enabling applications in areas that require highest levels of trust. For this note that a particular application direction of LogiKEy is normative and legal reasoning in the context of AI regulation, see also [10,45].

This paper also illustrates the conceptual modifications applied in prior LogiKEy papers when transitioning from lightweight shallow embeddings of non-classical logics, such as those used to encode higher-order modal logic [12] or conditional logic [4], which consider only dependencies on possible worlds, to more heavyweight shallow embeddings, as, for example, required to embed public announcement logic [13]; in the latter work additional dependencies on evaluation domains had to be encoded. Note on the other hand, that previous work has often avoided the explicit modeling of atoms, signatures, and evaluations, which has been essential for the work presented in this paper. Moreover, the presented technique should eventually help to overcome challenges in application contexts that require the combined use of deep and shallow embeddings. The Fatio protocol for multi-agent argumentation [52] is such an example. Most importantly, the presented technique for linking deep and shallow embeddings will be very fruitful for furthering the use of LogiKEy in university-level logic courses, including the author's courses on universal logical reasoning that have been taught for many years at FU Berlin and now at the University of Bamberg.

10 Conclusion

This paper has demonstrated how the simultaneous deployment of deep logic embeddings and shallow logic embeddings of various degrees in classical higher-order logic can be achieved. This technique allows for the mechanization of faithfulness proofs between the interlinked embeddings, and, as has been shown in this paper, this technique can even support the automation of such proofs.

The work presented is relevant to future research in various areas. Due to its conciseness and elegance, as well as its support for interactive and automated theorem proving and counterexample finding, it can also be used to support logic education effectively. The latter application direction was the main motivation for illustrating this approach using simple propositional modal logic in this paper. This logic is routinely used as introductory material in logic courses worldwide. The Isabelle/HOL sources provided and explained in this paper should be easily

reusable to support such courses, and they should also be easily transferable to other higher-order proof assistant systems.

Acknowledgements. I am grateful to the reviewers of this paper for their valuable feedback. I would also like to thank D. Fuenmayor, D. Kirchner, L. Pasetto and E. Zalta for their comments and all those who have contributed to LogiKEy in the past. My thanks also go to Deepl for the fruitful exchange that (hopefully) helped to improve the quality of the wording of this paper.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Abrahamsson, O., Myreen, M.O., Kumar, R., Sewell, T.: Candle: a verified implementation of HOL light. In: Andronick, J., de Moura, L. (eds.) 13th International Conference on Interactive Theorem Proving, ITP 2022, 7–10 August 2022, Haifa, Israel. LIPIcs, vol. 237, pp. 3:1–3:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2022). https://doi.org/10.4230/LIPICS.ITP.2022.3
2. Andrews, P.B.: General models and extensionality. J. Symb. Log. **37**(2), 395–397 (1972). https://doi.org/10.2307/2272982
3. Benzmüller, C.: Verifying the modal logic cube is an easy task (for higher-order automated reasoners). In: Siegler, S., Wasser, N. (eds.) Verification, Induction, Termination Analysis. LNCS (LNAI), vol. 6463, pp. 117–128. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-17172-7_7
4. Benzmüller, C.: Cut-elimination for quantified conditional logic. J. Philos. Log. **46**(3), 333–353 (2016). https://doi.org/10.1007/s10992-016-9403-0
5. Benzmüller, C.: Faithful logic embeddings in HOL — deep and shallow (2025). https://doi.org/10.48550/arXiv.2502.19311
6. Benzmüller, C., Andrews, P.: Church's type theory. In: Zalta, E.N., Nodelman, U. (eds.) The Stanford Encyclopedia of Philosophy, 2024 edn. Metaphysics Research Lab, Stanford University. Spring (2024). https://plato.stanford.edu/archives/spr2024/entries/type-theory-church/
7. Benzmüller, C., Brown, C., Kohlhase, M.: Higher-order semantics and extensionality. J. Symb. Log. **69**(4), 1027–1088 (2004). https://doi.org/10.2178/jsl/1102022211
8. Benzmüller, C., Claus, M., Sultana, N.: Systematic verification of the modal logic cube in Isabelle/HOL. In: Kaliszyk, C., Paskevich, A. (eds.) PxTP 2015, vol. 186, pp. 27–41. EPTCS (2015). https://doi.org/10.4204/EPTCS.186.5
9. Benzmüller, C., Miller, D.: Automation of higher-order logic. In: Gabbay, D.M., Siekmann, J.H., Woods, J. (eds.) Handbook of the History of Logic, Volume 9—Computational Logic, pp. 215–254. Elsevier, North Holland (2014). https://doi.org/10.1016/B978-0-444-51624-4.50005-8
10. Benzmüller, C., Parent, X., van der Torre, L.: Designing normative theories for ethical and legal reasoning: LogiKEy framework, methodology, and tool support. Artif. Intell. **287**, 103348 (2020). https://doi.org/10.1016/j.artint.2020.103348
11. Benzmüller, C., Paulson, L.C.: Multimodal and intuitionistic logics in simple type theory. Log. J. IGPL **18**(6), 881–892 (2010). https://doi.org/10.1093/jigpal/jzp080

12. Benzmüller, C., Paulson, L.C.: Quantified multimodal logics in simple type theory. Log. Univers. **7**(1), 7–20 (2013). https://doi.org/10.1007/s11787-012-0052-y
13. Benzmüller, C., Reiche, S.: Automating public announcement logic with relativized common knowledge as a fragment of HOL in LogiKEy. J. Log. Comput. **33**(6), 1243–1269 (2023). https://doi.org/10.1093/logcom/exac029
14. Benzmüller, C., Scott, D.S.: Automating free logic in HOL, with an experimental application in category theory. J. Autom. Reason. **64**(1), 53–72 (2019). https://doi.org/10.1007/s10817-018-09507-7
15. Benzmüller, C.: Faithful logic embeddings in HOL—deep and shallow (Isabelle/HOL dataset). Arch. Formal Proofs (2025). https://isa-afp.org/entries/FaithfulPMLinHOL.html
16. Benzmüller, C., Fuenmayor, D., Steen, A., Sutcliffe, G.: Who finds the short proof? Log. J. IGPL **32**, 442–464 (2024). https://doi.org/10.1093/jigpal/jzac082
17. Benzmüller, C., Scott, D.S.: Notes on Gödel's and Scott's variants of the ontological argument. Monatshefte für Mathematik (2025). https://doi.org/10.1007/s00605-025-02078-x
18. Bertot, Y., Casteran, P.: Interactive Theorem Proving and Program Development. Springer (2004)
19. Blackburn, P., van Benthem, J.: Modal logic: a semantic perspective. In: Blackburn, P., van Benthem, J.F.A.K., Wolter, F. (eds.) Handbook of Modal Logic, Studies in Logic and Practical Reasoning, North-Holland, vol. 3, pp. 1–84 (2007). https://doi.org/10.1016/S1570-2464(07)80004-8
20. Blanchette, J.C., Böhme, S., Paulson, L.C.: Extending sledgehammer with SMT solvers. In: Bjørner, N., Sofronie-Stokkermans, V. (eds.) CADE 2011. LNCS (LNAI), vol. 6803, pp. 116–130. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-22438-6_11
21. Blanchette, J.C., Nipkow, T.: Nitpick: a counterexample generator for higher-order logic based on a relational model finder. In: Kaufmann, M., Paulson, L.C. (eds.) ITP 2010. LNCS, vol. 6172, pp. 131–146. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-14052-5_11
22. Blanchette, J.C., Popescu, A., Traytel, D.: Soundness and completeness proofs by coinductive methods. J. Autom. Reason. **58**(1), 149–179 (2017). https://doi.org/10.1007/S10817-016-9391-3
23. Bohrer, R., Rahli, V., Vukotic, I., Völp, M., Platzer, A.: Formally verified differential dynamic logic. In: Proceedings of the 6th ACM SIGPLAN Conference on Certified Programs and Proofs, CPP 2017, pp. 208–221. Association for Computing Machinery, New York, NY, USA (2017). https://doi.org/10.1145/3018610.3018616
24. Boolos, G.S.: The Logic of Provability. Cambridge University Press (1994)
25. Boulton, R.J., Gordon, A.D., Gordon, M.J.C., Harrison, J., Herbert, J., Tassel, J.V.: Experience with embedding hardware description languages in HOL. In: Stavridou, V., Melham, T.F., Boute, R.T. (eds.) Theorem Provers in Circuit Design, Proceedings of the IFIP TC10/WG 10.2 International Conference on Theorem Provers in Circuit Design: Theory, Practice and Experience, Nijmegen, The Netherlands, 22–24 June 1992, Proceedings. IFIP Transactions, North-Holland, vol. A-10, pp. 129–156 (1992)
26. Carneiro, M.: Metamath zero: designing a theorem prover prover. In: Benzmüller, C., Miller, B. (eds.) CICM 2020. LNCS (LNAI), vol. 12236, pp. 71–88. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-53518-6_5
27. Carneiro, M.: Lean4Lean: towards a verified typechecker for Lean, in Lean (2024). https://doi.org/10.48550/arXiv.2403.14064

28. Church, A.: A formulation of the simple theory of types. J. Symb. Log. **5**(2), 56–68 (1940). https://doi.org/10.2307/2266170
29. Cohen, J.M., Johnson-Freyd, P.: A formalization of core Why3 in Coq. Proc. ACM Program. Lang. **8**(POPL), 1789–1818 (2024). https://doi.org/10.1145/3632902
30. Czajka, L.: A shallow embedding of pure type systems into first-order logic. In: Ghilezan, S., Geuvers, H., Ivetic, J. (eds.) 22nd International Conference on Types for Proofs and Programs (TYPES 2016). Leibniz International Proceedings in Informatics (LIPIcs), vol. 97, pp. 9:1–9:39. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany (2018). https://doi.org/10.4230/LIPIcs.TYPES.2016.9
31. From, A.H., Jacobsen, F.K.: Verifying a sequent calculus prover for first-order logic with functions in Isabelle/HOL. In: Andronick, J., de Moura, L. (eds.) 13th International Conference on Interactive Theorem Proving (ITP 2022). Leibniz International Proceedings in Informatics (LIPIcs), vol. 237, pp. 13:1–13:22. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany (2022). https://doi.org/10.4230/LIPIcs.ITP.2022.13
32. From, A.H.: A Naive prover for first-order logic. Arch. Formal Proofs (2022). https://isa-afp.org/entries/FOL_Seq_Calc3.html
33. From, A.H., Villadsen, J.: Soundness and completeness of implicational logic. Arch. Formal Proofs (2022). https://isa-afp.org/entries/Implicational_Logic.html
34. Garson, J.: Modal logic. In: Zalta, E.N., Nodelman, U. (eds.) The Stanford Encyclopedia of Philosophy, 2024 edn. Metaphysics Research Lab, Stanford University. Spring (2024). https://plato.stanford.edu/archives/spr2024/entries/logic-modal/
35. Gibbons, J., Wu, N.: Folding domain-specific languages: deep and shallow embeddings (functional pearl). In: Jeuring, J., Chakravarty, M.M.T. (eds.) Proceedings of the 19th ACM SIGPLAN International Conference on Functional Programming, Gothenburg, Sweden, 1–3 September 2014, pp. 339–347. ACM (2014). https://doi.org/10.1145/2628136.2628138
36. Gödel, K.: Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I. Monatshefte für Mathematik und Physik **38**(1), 173–198 (1931). https://doi.org/10.1007/BF01700692
37. Guilloud, S., Gambhir, S., Gilot, A., Kuncak, V.: Mechanized HOL reasoning in set theory. In: Bertot, Y., Kutsia, T., Norrish, M. (eds.) 15th International Conference on Interactive Theorem Proving, ITP 2024, 9–14 September 2024, Tbilisi, Georgia. LIPIcs, vol. 309, pp. 18:1–18:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2024). https://doi.org/10.4230/LIPICS.ITP.2024.18
38. Harrison, J.: Towards self-verification of HOL light. In: Furbach, U., Shankar, N. (eds.) IJCAR 2006. LNCS (LNAI), vol. 4130, pp. 177–191. Springer, Heidelberg (2006). https://doi.org/10.1007/11814771_17
39. Henkin, L.: Completeness in the theory of types. J. Symb. Log. **15**(2), 81–91 (1950). https://doi.org/10.2307/2266967
40. Kaposi, A., Kovács, A., Kraus, N.: Shallow embedding of type theory is morally correct. In: Hutton, G. (ed.) MPC 2019. LNCS, vol. 11825, pp. 329–365. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-33636-3_12
41. Kirchner, D.: Computer-verified foundations of metaphysics and an ontology of natural numbers in Isabelle/HOL. Ph.D. thesis, Freie Universität, Berlin (2022). https://doi.org/10.17169/refubium-35141
42. Kirchner, D., Benzmüller, C., Zalta, E.N.: Mechanizing principia Logico-Metaphysica in functional type theory. Rev. Symb. Log. **13**(1), 206–218 (2020). https://doi.org/10.1017/S1755020319000297

43. Lachnitt, H.: Systematic verification of the intuitionistic modal logic cube in Isabelle/HOL. Bachelor thesis, Freie Universität, Berlin (2017)
44. Lammich, P., Meis, R.: A separation logic framework for imperative HOL. Arch. Formal Proofs **2012** (2012). https://www.isa-afp.org/entries/Separation_Logic_Imperative_HOL.shtml
45. Lawniczak, L., Benzmüller, C.: Logical modalities within the European AI act: an analysis. In: Maranhão, J. (ed.) Proceedings of the 20th International Conference on Artificial Intelligence and Law. Association for Computing Machinery, New York, NY, USA (2025, in print)
46. Lorini, E.: Temporal logic and its application to normative reasoning. J. Appl. Non-Class. Log. **23**(4), 372–399 (2013). https://doi.org/10.1080/11663081.2013.841359
47. Meder, P.: Deontic agency and moral luck. Master's thesis, University of Luxembourg (2018). https://hdl.handle.net/10993/39770
48. de Moura, L., Kong, S., Avigad, J., van Doorn, F., von Raumer, J.: The Lean theorem prover (system description). In: Felty, A.P., Middeldorp, A. (eds.) CADE 2015. LNCS (LNAI), vol. 9195, pp. 378–388. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-21401-6_26
49. Nipkow, T.: Hoare logics in Isabelle/HOL. In: Schwichtenberg, H., Steinbrüggen, R. (eds.) Proof and System-Reliability, pp. 341–367. Springer, Netherlands, Dordrecht (2002). https://doi.org/10.1007/978-94-010-0413-8_11
50. Nipkow, T., Wenzel, M., Paulson, L.C. (eds.): Isabelle/HOL. LNCS, vol. 2283. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-45949-9
51. Parent, X., Benzmüller, C.: Conditional normative reasoning as a fragment of HOL. J. Appl. Non-Class. Log. **34**, 561–592 (2024). https://doi.org/10.1080/11663081.2024.2386917
52. Pasetto, L., Benzmüller, C.: Implementing the Fatio protocol for multi-agent argumentation in LogiKEy. In: Benzmüller, C., Otten, J., Ramanayake, R. (eds.) Proceedings of the 5th International Workshop on Automated Reasoning in Quantified Non-Classical Logics (ARQNL 2024), Nancy, France, 1 July 2024, vol. CEUR-3875, pp. 38–47. CEUR Workshop Proceedings. CEUR-WS.org, Nancy, France (2024). https://ceur-ws.org/Vol-3875/ARQNL2024_paper4.pdf
53. Prinz, J., Kavvos, G.A., Lampropoulos, L.: Deeper shallow embeddings. In: Andronick, J., de Moura, L. (eds.) 13th International Conference on Interactive Theorem Proving, ITP 2022, 7–10 August 2022, Haifa, Israel. LIPIcs, vol. 237, pp. 28:1–28:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2022). https://doi.org/10.4230/LIPICS.ITP.2022.28
54. Rabe, M.N., Lammich, P., Popescu, A.: A shallow embedding of hyperCTL. Arch. Formal Proofs **2014** (2014). https://www.isa-afp.org/entries/HyperCTL.shtml
55. Sahlqvist, H.: Completeness and correspondence in the first and second order semantics for modal logic. Stud. Log. Found. Math. **82**, 110–143 (1975). https://doi.org/10.1016/S0049-237X(08)70728-6
56. Sider, T.: Logic for Philosophy. Oxford University Press, New York (2009)
57. Smullyan, R.: A Beginner's Guide to Mathematical Logic. Dover Books on Mathematics, Dover Publications (2014). https://books.google.ae/books?id=n6S-AwAAQBAJ
58. Sozeau, M., et al.: The MetaCoq project. J. Autom. Reason. **64**(5), 947–999 (2020). https://doi.org/10.1007/S10817-019-09540-0
59. Steen, A., Benzmüller, C.: Extensional higher-order paramodulation in Leo-III. J. Autom. Reason. **65**(6), 775–807 (2021). https://doi.org/10.1007/s10817-021-09588-x

60. Steen, A., Sutcliffe, G., Benzmüller, C.: Solving quantified modal logic problems by translation to classical logics. J. Log. Comput., 1–23 (2025). https://doi.org/10.1093/logcom/exaf006
61. Svenningsson, J., Axelsson, E.: Combining deep and shallow embedding for EDSL. In: Loidl, H.-W., Peña, R. (eds.) TFP 2012. LNCS, vol. 7829, pp. 21–36. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40447-4_2
62. Wang, Z., Cao, Q., Tao, Y.: Verifying programs with logic and extended proof rules: deep embedding vs. shallow embedding. J. Autom. Reason. **68**(3), 18 (2024). https://doi.org/10.1007/S10817-024-09706-5
63. Wikipedia contributors: Hilbert system — Wikipedia, the free encyclopedia (2024). https://en.wikipedia.org/w/index.php?title=Hilbert_system&oldid=1257978821. Accessed 18 Feb 2025
64. Zalta, E.N.: Principia Logico-Metaphysica (Draft/Excerpt) (2025). https://mally.stanford.edu/principia.pdf. Accessed 27 May 2025