Chapter 1
Dyadic Deontic Logic in HOL: Faithful Embedding and Meta-Theoretical Experiments

Christoph Benzmüller[1], Ali Farjami[2], and Xavier Parent[2]

Abstract A shallow semantical embedding of a dyadic deontic logic by Carmo and Jones in classical higher-order logic is presented. The embedding is proven sound and complete, that is, faithful. This result provides the theoretical foundation for the implementation and automation of dyadic deontic logic within off-the-shelf higher-order theorem provers and proof assistants. To demonstrate the practical relevance of our contribution, the embedding has been encoded in the Isabelle/HOL proof assistant. As a result a sound and complete (interactive and automated) theorem prover for the dyadic deontic logic of Carmo and Jones has been obtained. Experiments have been conducted which illustrate how the exploration and assessment of meta-theoretical properties of the embedded logic can be supported with automated reasoning tools integrated with Isabelle/HOL.

1.1 Introduction

Dyadic deontic logic is the logic for reasoning with dyadic obligations ("it ought to be the case that ... if it is the case that ..."). A particular dyadic deontic logic, tailored to so-called contrary-to-duty conditionals, has been proposed by Carmo and Jones [1]. We shall refer to it as DDL in the remainder. DDL comes with a neighborhood semantics and a weakly complete axiomatization over the class of finite models. The framework is immune to the well-known contrary-to-duty paradoxes, like Chisholm's paradox, and other related puzzles. However, the question of how to mechanise and automate reasoning tasks in DDL has not been studied yet.

This article adresses this challenge. We essentially devise a faithfull semantical embedding of DDL in classical higher-order logic (HOL). The latter logic thereby serves as an universal meta-logic [2]. Analogous to successful, recent work in the area of computational metaphysics (cf. Kirchner et al. [3] and the references therein),

Freie Universität Berlin, Berlin, Germany · University of Luxembourg, Esch-sur-Alzette, Luxembourg

the key motivation is to mechanise and automate DDL on the computer by reusing existing theorem proving technology for meta-logic HOL. The embedding of DDL in HOL as devised in this article enables just this.

The present work is part of the larger LogiKEy project [4]. This project aims at developing a reasoning infrastructure flexible enough to “host" a large spectrum of deontic formalisms, including the dyadic deontic logic of Carmo and Jones. Existing approaches are usually tied to a specific logical system. However, we do not think that there is a single, uniquely correct (deontic) logical system, but there may be many equally qualified choices, so that a particular choice of a logic, respectively, logic combination, is left to the user.

Due to the improved flexibility and expressivity as offered in the LogiKEy approach, highly non-trivial natural language arguments can now be more easily mechanized and assessed on the computer. A recent example is Alan Gewirth's argument for the *Principle of Generic Consistency* (PGC) [5, 6]. It was successfully encoded and verified on the computer [7, 8] via utilizing a suitable extension of the semantic embedding described in this paper.

Meta-logic HOL [9], as employed in this article, was originally devised by Church [10], and further developed by Henkin [11] and Andrews [12, 13, 14]. It bases both terms and formulas on simply typed λ-terms. The use of the λ-calculus has some major advantages. For example, λ-abstractions over formulas allow the explicit naming of sets and predicates, something that is achieved in set theory via the comprehension axioms. Another advantage is, that the complex rules for quantifier instantiation at first-order and higher-order types is completely explained via the rules of λ-conversion (the so-called rules of α-, β-, and η-conversion) which were proposed earlier by Church [15, 16]. These two advantages are exploited in our embedding of DDL in HOL.

Different notions of semantics for HOL have been thoroughly studied in the literature [17, 18]. In this article we assume HOL with Henkin semantics (cf. the detailed description by Benzmüller et al. [17]). For this notion of HOL, which does not suffer from Gödel's incompleteness results, several sound and complete theorem provers have been developed in the past decades [19]. We propose to reuse these systems for the automation of DDL. The semantical embedding as devised in this article provides both the theoretical foundation for the approach and the practical bridging technology that is enabling DDL applications within existing HOL theorem provers.

The article is structured as follows: Section 2 outlines the syntax and semantics of DDL, as far as needed for this article. Section 3 provides a comparably detailed introduction into HOL; this is needed to keep the article sufficiently self-contained. The semantical embedding of DDL in HOL is then devised and studied in Sec. 4. This section also presents the respective soundness and completeness proofs for the embedding; i.e. the embeddings faithfulness is shown. Section 5 then depicts and discusses the implementation of the devised embedding in the proof assistent system

Isabelle/HOL and presents examples of meta-theoretical experiments.[1] Section 6 concludes the paper.

1.2 The Dyadic Deontic Logic of Carmo and Jones

This section provides a concise introduction of DDL, the dyadic deontic logic proposed by Carmo and Jones. Definitions as required for the remainder are presented. For further details we refer to the literature [20, 1].

To define the formulas of DDL we start with a countable set P of propositional symbols, and we choose $\neg$ and $\vee$ as the only primitive connectives.

The set of *DDL formulas* is given as the smallest set of formulas obeying the following conditions:

- Each $p^j \in P$ is an (atomic) DDL formula.
- Given two arbitrary DDL formulas φ and ψ, then
  $\neg\varphi$ — *classical negation*,
  $\varphi \vee \psi$ — *classical disjunction*,
  $\bigcirc(\psi/\varphi)$ — *dyadic deontic obligation: "it ought to be ψ, given φ"*,
  $\Box\varphi$ — *in all worlds*,
  $\Box_a\varphi$ — *in all actual versions of the current world*,
  $\Box_p\varphi$ — *in all potential versions of the current world*,
  $\bigcirc_a\varphi$ — *monadic deontic operator for actual obligation*, and
  $\bigcirc_p\varphi$ — *monadic deontic operator for primary obligation*
  are also DDL formulas.

Further logical connectives can be defined as usual: $\varphi \wedge \psi := \neg(\neg\varphi \vee \neg\psi)$, $\varphi \rightarrow \psi := \neg\varphi \vee \psi$, $\varphi \longleftrightarrow \psi := (\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi)$, $\Diamond\varphi := \neg\Box\neg\varphi$, $\Diamond_a\varphi := \neg\Box_a\neg\varphi$, $\Diamond_p\varphi := \neg\Box_p\neg\varphi$, $\top := \neg q^j \vee q^j$, for some propositional symbol q^j, $\bot := \neg\top$, and $\bigcirc\varphi := \bigcirc(\varphi/\top)$.

A DDL *model* is a structure $M = \langle S, av, pv, ob, V\rangle$, where S is a non empty set of items called possible worlds, V is a function assigning a set of worlds to each atomic formula, that is, $V(p^j) \subseteq S$. av: $S \rightarrow \wp(S)$, where $\wp(S)$ is the power set of S, is a function mapping worlds to sets of worlds such that $av(s) \neq \emptyset$. $av(s)$ is the set of actual versions of the world s. pv: $S \rightarrow \wp(S)$ is another, similar mapping such that $av(s) \subseteq pv(s)$ and $s \in pv(s)$. $pv(s)$ is the set of potential versions of the world s. ob: $\wp(S) \rightarrow \wp(\wp(S))$ is a function mapping sets of worlds to sets of sets of worlds. $ob(\bar{X})$ is the set of propositions that are obligatory in context $\bar{X} \subseteq S$. The following conditions hold for ob (where $\bar{X}, \bar{Y}, \bar{Z}$ designate arbitrary subsets of S):

1. $\emptyset \notin ob(\bar{X})$.
2. If $\bar{Y} \cap \bar{X} = \bar{Z} \cap \bar{X}$, then $\bar{Y} \in ob(\bar{X})$ if and only if $\bar{Z} \in ob(\bar{X})$.

[1] The sources of our Isabelle/HOL encoding of the embedding and of the conducted experiments can be found at the website of the LogiKEy project: `logikey.org`.

3. Let $\bar{\beta} \subseteq ob(\bar{X})$ and $\bar{\beta} \neq \emptyset$. If $(\cap\bar{\beta}) \cap \bar{X} \neq \emptyset$ (where $\cap\bar{\beta} = \{s \in S \mid \text{for all } \bar{Z} \in \bar{\beta} \text{ we have } s \in \bar{Z}\}$), then $(\cap\bar{\beta}) \in ob(\bar{X})$.
4. If $\bar{Y} \subseteq \bar{X}$ and $\bar{Y} \in ob(\bar{X})$ and $\bar{X} \subseteq \bar{Z}$, then $(\bar{Z} \setminus \bar{X}) \cup \bar{Y} \in ob(\bar{Z})$.
5. If $\bar{Y} \subseteq \bar{X}$ and $\bar{Z} \in ob(\bar{X})$ and $\bar{Y} \cap \bar{Z} \neq \emptyset$, then $\bar{Z} \in ob(\bar{Y})$.

Satisfiability of a formula φ for a model $M = \langle S, av, pv, ob, V\rangle$ and a world $s \in S$ is expressed by writing that $M, s \models \varphi$ and we define $V^M(\varphi) = \{s \in S \mid M, s \models \varphi\}$. In order to simplify the presentation, whenever the model M is obvious from context, we write $V(\varphi)$ instead of $V^M(\varphi)$. Moreover, we often use "iff" as shorthand for "if and only if".

$$
\begin{array}{lll}
M, s \models & p^j & \text{iff } s \in V(p^j) \\
M, s \models & \neg\varphi & \text{iff } M, s \not\models \varphi \text{ (that is, not } M, s \models \varphi) \\
M, s \models & \varphi \vee \psi & \text{iff } M, s \models \varphi \text{ or } M, s \models \psi \\
M, s \models & \Box\varphi & \text{iff } V(\varphi) = S \\
M, s \models & \Box_a\varphi & \text{iff } av(s) \subseteq V(\varphi) \\
M, s \models & \Box_p\varphi & \text{iff } pv(s) \subseteq V(\varphi) \\
M, s \models & \bigcirc(\psi/\varphi) & \text{iff } V(\psi) \in ob(V(\varphi)) \\
M, s \models & \bigcirc_a\varphi & \text{iff } V(\varphi) \in ob(av(s)) \text{ and } av(s) \cap V(\neg\varphi) \neq \emptyset \\
M, s \models & \bigcirc_p\varphi & \text{iff } V(\varphi) \in ob(pv(s)) \text{ and } pv(s) \cap V(\neg\varphi) \neq \emptyset
\end{array}
$$

Our evaluation rule for $\bigcirc(\_/\_)$ is a simplified version of the one used by Carmo and Jones. Given the constraints placed on ob, the two rules are equivalent (cf. [21, result II-2-2]).

As usual, a DDL formula φ is *valid in a DDL model* $M = \langle S, av, pv, ob, V\rangle$, i.e. $M \models^{DDL} \varphi$, if and only if for all worlds $s \in S$ we have $M, s \models \varphi$. A formula φ is *valid*, denoted $\models^{DDL} \varphi$, if and only if it is valid in every DDL model.

1.3 Classical Higher-order Logic

In this section we introduce classical higher-order logic (HOL). The presentation, which has partly been adapted from [21], is rather detailed in order to keep the article sufficiently self-contained.

1.3.1 Syntax of HOL

For defining the syntax of HOL, we first introduce the set T of *simple types*. We assume that T is freely generated from a set of *basic types* $BT \supseteq \{o, i\}$ using the function type constructor $\rightarrow$. Type o denotes the (bivalent) set of Booleans, and i a non-empty set of individuals.

For the definition of HOL, we start out with a family of denumerable sets of typed constant symbols $(C_\alpha)_{\alpha\in T}$, called the HOL *signature*, and a family of denumerable sets of typed variable symbols $(V_\alpha)_{\alpha\in T}$.[2] We employ Church-style typing, where each term t_α explicitly encodes its type information in subscript α.

The *language of HOL* is given as the smallest set of terms obeying the following conditions.

- Every typed constant symbol $c_\alpha \in C_\alpha$ is a HOL term of type α.
- Every typed variable symbol $X_\alpha \in V_\alpha$ is a HOL term of type α.
- If $s_{\alpha\to\beta}$ and t_α are HOL terms of types $\alpha \to \beta$ and α, respectively, then $(s_{\alpha\to\beta}\, t_\alpha)_\beta$, 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 $(\lambda X_\alpha s_\beta)_{\alpha\to\beta}$, called *abstraction*, is an HOL term of type $\alpha \to \beta$.

The above definition encompasses the simply typed λ-calculus. In order to extend this base framework into logic HOL we simply ensure that the signature $(C_\alpha)_{\alpha\in T}$ provides a sufficient selection of primitive logical connectives. Without loss of generality, we here assume the following *primitive logical connectives* to be part of the signature: $\neg_{o\to o} \in C_{o\to o}$, $\vee_{o\to o\to o} \in C_{o\to o\to o}$, $\Pi_{(\alpha\to o)\to o} \in C_{(\alpha\to o)\to o}$ and $=_{\alpha\to\alpha\to\alpha} \in C_{\alpha\to\alpha\to\alpha}$, abbreviated as $=^\alpha$. The symbols $\Pi_{(\alpha\to o)\to o}$ and $=_{\alpha\to\alpha\to\alpha}$ are generally assumed for each type $\alpha \in T$. The denotation of the primitive logical connectives is fixed below according to their intended meaning. *Binder notation* $\forall X_\alpha\, s_o$ is used as an abbreviation for $\Pi_{(\alpha\to o)\to o}\lambda X_\alpha s_o$. Universal quantification in HOL is thus modeled with the help of the logical constants $\Pi_{(\alpha\to o)\to o}$ to be used in combination with lambda-abstraction. That is, the only binding mechanism provided in HOL is lambda-abstraction.

HOL is a logic of terms in the sense that the *formulas of HOL* are given as the terms of type o. In addition to the primitive logical connectives selected above, we could assume *choice operators* $\epsilon_{(\alpha\to o)\to\alpha} \in C_{(\alpha\to o)\to\alpha}$ (for each type α) in the signature. We are not pursuing this here.

Type information as well as brackets may be omitted if obvious from the context, and we may also use infix notation to improve readability. For example, we may write $(s \vee t)$ instead of $((\vee_{o\to o\to o} s_o) t_o)$.

From the selected set of primitive connectives, other logical connectives can be introduced as abbreviations.[3] For example, we may define $s \wedge t := \neg(\neg s \vee \neg t)$, $s \longrightarrow t := \neg s \vee t$, $s \longleftrightarrow t := (s \longrightarrow t) \wedge (t \longrightarrow s)$, $\top := (\lambda X_i X) = (\lambda X_i X)$, $\bot := \neg\top$ and $\exists X_\alpha s := \neg\forall X_\alpha \neg s$.

---

[2] For example in Section 4 we will assume constant symbols av, pv and ob with types $i \to i \to o$, $i \to i \to o$ and $(i \to o) \to (i \to o) \to o$ as part of the signature.

[3] As demonstrated by Andrews [9], we could in fact start out with only primitive equality in the signature (for all types α) and introduce all other logical connectives as abbreviations based on it. Alternatively, we could remove primitive equality from the above signature, since equality can be defined in HOL from these other logical connectives by exploiting Leibniz' principle, expressing that two objects are equal if they share the same properties. *Leibniz equality* $\doteq^\alpha$ at type α is thus defined as $s_\alpha \doteq^\alpha t_\alpha := \forall P_{\alpha\to o}(Ps \longleftrightarrow Pt)$. The motivation for the redundant signature as selected here is to stay close to the the choices taken in implemented theorem provers such as LEO-II and Leo-III and also to theory paper [17], which is recommended for further details.

The notions of *free variables*, α-*conversion*, $\beta\eta$-*equality* (denoted as $=_{\beta\eta}$) and *substitution* of a term s_α for a variable X_α in a term t_β (denoted as $[s/X]t$) are defined as usual.

1.3.2 Semantics of HOL

The semantics of HOL is well understood and thoroughly documented. The introduction provided next focuses on the aspects as needed for this article. For more details we refer to the previously mentioned literature [17].

The semantics of choice for the remainder is Henkin semantics, i.e., we work with Henkin's general models [11]. Henkin models (and standard models) are introduced next. We start out with introducing frame structures.

A *frame* D is a collection $\{D_\alpha\}_{\alpha\in\mathrm{T}}$ of nonempty sets D_α, such that $D_o = \{T, F\}$ (for truth and falsehood). The $D_{\alpha\to\beta}$ are collections of functions mapping D_α into D_β.

A *model* for HOL is a tuple $M = \langle D, I\rangle$, where 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 D_α, called the *denotation of* p_α. The logical connectives $\neg$, $\vee$, Π and $=$ are always given their expected, standard denotations:[4]

- $I(\neg_{o\to o}) = not \in D_{o\to o}$ such that $not(T) = F$ and $not(F) = T$.
- $I(\vee_{o\to o\to o}) = or \in D_{o\to o\to o}$ such that $or(a, b) = T$ iff ($a = T$ or $b = T$).
- $I(=_{\alpha\to\alpha\to o}) = id \in D_{\alpha\to\alpha\to o}$ such that for all $a, b \in D_\alpha$, $id(a, b) = T$ iff a is identical to b.
- $I(\Pi_{(\alpha\to o)\to o}) = all \in D_{(\alpha\to o)\to o}$ such that for all $s \in D_{\alpha\to o}$, $all(s) = T$ iff $s(a) = T$ for all $a \in D_\alpha$; i.e., s is the set of all objects of type α.

Variable assignments are a technical aid for the subsequent definition of an interpretation function $\|.\|^{M,g}$ for HOL terms. This interpretation function is parametric over a model M and a variable assignment g.

A *variable assignment* g maps variables X_α to elements in 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* $\|s_\alpha\|^{M,g}$ of an HOL term s_α on a model $M = \langle D, I\rangle$ under assignment g is an element $d \in D_\alpha$ defined in the following way:

$$\|p_\alpha\|^{M,g} = I(p_\alpha)$$
$$\|X_\alpha\|^{M,g} = g(X_\alpha)$$

---

[4] Since $=_{\alpha\to\alpha\to o}$ (for all types α) is in the signature, it is ensured that the domains $D_{\alpha\to\alpha\to o}$ contain the respective identity relations. This addresses an issue discovered by Andrews [13]: if such identity relations did not exist in the $D_{\alpha\to\alpha\to o}$, then Leibniz equality in Henkin semantics might not denote as intended.

$$\|(s_{\alpha\to\beta}\, t_\alpha)_\beta\|^{M,g} = \|s_{\alpha\to\beta}\|^{M,g}(\|t_\alpha\|^{M,g})$$
$$\|(\lambda X_\alpha s_\beta)_{\alpha\to\beta}\|^{M,g} = \text{the function } f \text{ from } D_\alpha \text{ to } D_\beta \text{ such that } f(d) = \|s_\beta\|^{M,g[d/X_\alpha]} \text{ for all } d \in D_\alpha$$

A model $M = \langle D, I\rangle$ is called a *standard model* if and only if for all $\alpha, \beta \in T$ we have $D_{\alpha\to\beta} = \{f \mid f : D_\alpha \longrightarrow D_\beta\}$. In a *Henkin model (general model)* function spaces are not necessarily full. Instead it is only required that for all $\alpha, \beta \in T$, $D_{\alpha\to\beta} \subseteq \{f \mid f : D_\alpha \longrightarrow D_\beta\}$. However, it is required that the valuation function $\|\cdot\|^{M,g}$ from above is total, so that every term denotes. Note that this requirement, which is called *Denotatpflicht*, ensures that the function domains $D_{\alpha\to\beta}$ never become too sparse, that is, the denotations of the lambda-abstractions as devised above are always contained in them.

Corollary 1 *For any Henkin model $M = \langle D, I\rangle$ and variable assignment g:*

1. $\|(\neg_{o\to o}\, s_o)_o\|^{M,g} = T$ *iff* $\|s_o\|^{M,g} = F$.
2. $\|((\vee_{o\to o\to o}\, s_o)\, t_o)_o\|^{M,g} = T$ *iff* $\|s_o\|^{M,g} = T$ *or* $\|t_o\|^{M,g} = T$.
3. $\|((\wedge_{o\to o\to o}\, s_o)\, t_o)_o\|^{M,g} = T$ *iff* $\|s_o\|^{M,g} = T$ *and* $\|t_o\|^{M,g} = T$.
4. $\|((\to_{o\to o\to o}\, s_o)\, t_o)_o\|^{M,g} = T$ *iff* (*if* $\|s_o\|^{M,g} = T$ *then* $\|t_o\|^{M,g} = T$).
5. $\|((\longleftrightarrow_{o\to o\to o}\, s_o)\, t_o)_o\|^{M,g} = T$ *iff* ($\|s_o\|^{M,g} = T$ *iff* $\|t_o\|^{M,g} = T$).
6. $\|\top\|^{M,g} = T$.
7. $\|\bot\|^{M,g} = F$.
8. $\|(\forall X_\alpha s_o)_o\|^{M,g} = T$ *iff for all* $d \in D_\alpha$ *we have* $\|s_o\|^{M,g[d/X_\alpha]} = T$.
9. $\|(\exists X_\alpha s_o)_o\|^{M,g} = T$ *iff there exists* $d \in D_\alpha$ *such that* $\|s_o\|^{M,g[d/X_\alpha]} = T$.

Proof We leave the proof as an exercise to the reader. □

An HOL formula s_o is *true* in an Henkin model M under assignment g if and only if $\|s_o\|^{M,g} = T$; this is also expressed by writing that $M, g \models^{\text{HOL}} s_o$. An HOL formula s_o is called *valid* in M, which is expressed by writing that $M \models^{\text{HOL}} s_o$, if and only if $M, g \models^{\text{HOL}} s_o$ for all assignments g. Moreover, a formula s_o is called *valid*, expressed by writing that $\models^{\text{HOL}} s_o$, if and only if s_o is valid in all Henkin models M. Finally, we define $\Sigma \models^{\text{HOL}} s_o$ for a set of HOL formulas Σ if and only if $M \models^{\text{HOL}} s_o$ for all Henkin models M with $M \models^{\text{HOL}} t_o$ for all $t_o \in \Sigma$.

Note that any standard model is obviously also a Henkin model. Hence, validity of a HOL formula s_o for all Henkin models implies validity of s_o for all standard models.

1.4 Modeling DDL as a Fragment of HOL

This section, the core contribution of this article, presents a shallow semantical embedding of DDL in HOL and proves its soundness and completeness. In contrast to a deep logical embedding, where the syntax and semantics of logic L would be formalized in full detail (using structural induction and recursion), only the core differences in the semantics of both DDL and meta-logic HOL are explicitly encoded here.

1.4.1 Semantical Embedding

DDL formulas are identified in our semantical embedding with certain HOL terms (predicates) of type $i \to o$. They can be applied to terms of type i, which are assumed to denote possible worlds. That is, the HOL type i is now identified with a (non-empty) set of worlds. Type $i \to o$ is abbreviated as τ in the remainder. The HOL signature is assumed to contain the constant symbols $av_{i\to\tau}$, $pv_{i\to\tau}$ and $ob_{\tau\to\tau\to o}$. Moreover, for each propositional symbol p^i of DDL, the HOL signature must contain the corresponding constant symbol p^i_τ. Without loss of generality, we assume that besides those symbols and the primitive logical connectives of HOL, no other constant symbols are given in the signature of HOL.

The mapping $\lfloor\cdot\rfloor$ translates DDL formulas φ into HOL terms $\lfloor\varphi\rfloor$ of type τ. The mapping is recursively[5] defined:

$$
\begin{aligned}
\lfloor p^j \rfloor &= p^j_\tau \\
\lfloor \neg\varphi \rfloor &= \neg_{\tau\to\tau} \lfloor\varphi\rfloor \\
\lfloor \varphi \vee \psi \rfloor &= \vee_{\tau\to\tau\to\tau} \lfloor\varphi\rfloor\lfloor\psi\rfloor \\
\lfloor \Box\varphi \rfloor &= \Box_{\tau\to\tau} \lfloor\varphi\rfloor \\
\lfloor \bigcirc(\psi/\varphi) \rfloor &= \bigcirc_{\tau\to\tau\to\tau} \lfloor\varphi\rfloor\lfloor\psi\rfloor \\
\lfloor \Box_a\varphi \rfloor &= \Box^a_{\tau\to\tau} \lfloor\varphi\rfloor \\
\lfloor \Box_p\varphi \rfloor &= \Box^p_{\tau\to\tau} \lfloor\varphi\rfloor \\
\lfloor \bigcirc_a\varphi \rfloor &= \bigcirc^a_{\tau\to\tau} \lfloor\varphi\rfloor \\
\lfloor \bigcirc_p\varphi \rfloor &= \bigcirc^p_{\tau\to\tau} \lfloor\varphi\rfloor
\end{aligned}
$$

$\neg_{\tau\to\tau}$, $\vee_{\tau\to\tau\to\tau}$, $\Box_{\tau\to\tau}$, $\bigcirc_{\tau\to\tau\to\tau}$, $\Box^a_{\tau\to\tau}$, $\Box^p_{\tau\to\tau}$, $\bigcirc^a_{\tau\to\tau}$ and $\bigcirc^p_{\tau\to\tau}$ thereby abbreviate the following HOL terms:

$$
\begin{aligned}
\neg_{\tau\to\tau} &= \lambda A_\tau \lambda X_i \neg(A\,X) \\
\vee_{\tau\to\tau\to\tau} &= \lambda A_\tau \lambda B_\tau \lambda X_i (A\,X \vee B\,X) \\
\Box_{\tau\to\tau} &= \lambda A_\tau \lambda X_i \forall Y_i (A\,Y) \\
\bigcirc_{\tau\to\tau\to\tau} &= \lambda A_\tau \lambda B_\tau \lambda X_i (ob\,A\,B) \\
\Box^a_{\tau\to\tau} &= \lambda A_\tau \lambda X_i \forall Y_i (\neg(av\,X\,Y) \vee A\,Y) \\
\Box^p_{\tau\to\tau} &= \lambda A_\tau \lambda X_i \forall Y_i (\neg(pv\,X\,Y) \vee A\,Y) \\
\bigcirc^a_{\tau\to\tau} &= \lambda A_\tau \lambda X_i ((ob\,(av\,X)\,A) \wedge \exists Y_i (av\,X\,Y \wedge \neg(A\,Y))) \\
\bigcirc^p_{\tau\to\tau} &= \lambda A_\tau \lambda X_i ((ob\,(pv\,X)\,A) \wedge \exists Y_i (pv\,X\,Y \wedge \neg(A\,Y)))
\end{aligned}
$$

[5] A recursive definition is actually not needed in practice. By inspecting the equations below it should become clear that only the abbreviations for the logical connectives of DDL are required in combination with a type-lifting for the propositional constant symbols; cf. also Fig. 1.1.

Analyzing the truth of a translated formula $\lfloor\varphi\rfloor$ in a world represented by term w_i corresponds to evaluating the application $(\lfloor\varphi\rfloor\, w_i)$. In line with previous work [22], we define $\text{vld}_{\tau\to o} = \lambda A_\tau \forall S_i (A\, S)$. With this definition, validity of a DDL formula φ in DDL corresponds to the validity of formula $(\text{vld}\,\lfloor\varphi\rfloor)$ in HOL, and vice versa.

1.4.2 Soundness and Completeness

To prove the soundness and completeness, that is, faithfulness, of the above embedding, a mapping from DDL models into Henkin models is employed.

Definition 1 (Henkin model H^M for DDL model M)

For any DDL model $M = \langle S, av, pv, ob, V\rangle$, we define a corresponding Henkin model H^M. Thus, let a DDL model $M = \langle S, av, pv, ob, V\rangle$ be given. Moreover, assume that $p^j \in P$, for $j \geq 1$, are the only propositional symbols of DDL. Remember that our embedding requires the corresponding signature of HOL to provide constant symbols p^j_τ such that $\lfloor p^j\rfloor = p^j_\tau$ for $j = 1, \ldots, m$.

A Henkin model $H^M = \langle\{D_\alpha\}_{\alpha\in T}, I\rangle$ for M is now defined as follows: D_i is chosen as the set of possible worlds S; all other sets $D_{\alpha\to\beta}$ are chosen as (not necessarily full) sets of functions from D_α to D_β. For all $D_{\alpha\to\beta}$ the rule that every term $t_{\alpha\to\beta}$ must have a denotation in $D_{\alpha\to\beta}$ must be obeyed (Denotatpflicht). In particular, it is required that D_τ, $D_{i\to\tau}$ and $D_{\tau\to\tau\to o}$ contain the elements Ip^j_τ, $Iav_{i\to\tau}$, $Ipv_{i\to\tau}$ and $Iob_{\tau\to\tau\to o}$. The interpretation function I of H^M is defined as follows:

1. For $j = 1, \ldots, m$, $Ip^j_\tau \in D_\tau$ is chosen such that $Ip^j_\tau(s) = T$ iff $s \in V(p^j)$ in M.
2. $Iav_{i\to\tau} \in D_{i\to\tau}$ is chosen such that $Iav_{i\to\tau}(s, u) = T$ iff $u \in av(s)$ in M.
3. $Ipv_{i\to\tau} \in D_{i\to\tau}$ is chosen such that $Ipv_{i\to\tau}(s, u) = T$ iff $u \in pv(s)$ in M.
4. $Iob_{\tau\to\tau\to o} \in D_{\tau\to\tau\to o}$ is such that $Iob_{\tau\to\tau\to o}(\bar{X}, \bar{Y}) = T$ iff $\bar{Y} \in ob(\bar{X})$ in M.
5. For the logical connectives $\neg$, $\vee$, Π and $=$ of HOL the interpretation function I is defined as usual (see the previous section).

Since we assume that there are no other symbols (besides the p^i, av, pv, ob and $\neg$, $\vee$, Π, and $=$) in the signature of HOL, I is a total function. Moreover, the above construction guarantees that H^M is a Henkin model: $\langle D, I\rangle$ is a frame, and the choice of I in combination with the Denotatpflicht ensures that for arbitrary assignments g, $\|.\|^{H^M,g}$ is an total evaluation function.

Lemma 1 *Let H^M be a Henkin model for a DDL model M. In H^M we have for all $s \in D_i$ and all $\bar{X}, \bar{Y}, \bar{Z} \in D_\tau$ (cf. the conditions on DDL models as stated on page 3):*[6]

[6] In the proof of the lemma we implicitly employ curring and uncurring, and we associate sets with their characteristic functions. This analogously applies to the remainder of this article.

(av) $Iav_{i\rightarrow\tau}(s) \neq \emptyset$.
(pv1) $Iav_{i\rightarrow\tau}(s) \subseteq Ipv_{i\rightarrow\tau}(s)$.
(pv2) $s \in Ipv_{i\rightarrow\tau}(s)$.
(ob1) $\emptyset \notin Iob_{\tau\rightarrow\tau\rightarrow o}(\bar{X})$.
(ob2) *If* $\bar{Y} \cap \bar{X} = \bar{Z} \cap \bar{X}$, *then* $(\bar{Y} \in Iob_{\tau\rightarrow\tau\rightarrow o}(\bar{X})$ *iff* $\bar{Z} \in Iob_{\tau\rightarrow\tau\rightarrow o}(\bar{X}))$.
(ob3) *Let* $\bar{\beta} \subseteq Iob_{\tau\rightarrow\tau\rightarrow o}(\bar{X})$ *and* $\bar{\beta} \neq \emptyset$.
If $(\cap\bar{\beta}) \cap \bar{X} \neq \emptyset$, *where* $\cap\bar{\beta} = \{s \in S \mid$ *for all* $\bar{Z} \in \bar{\beta}$ *we have* $s \in \bar{Z}\}$,
then $(\cap\bar{\beta}) \in Iob_{\tau\rightarrow\tau\rightarrow o}(\bar{X})$.
(ob4) *If* $\bar{Y} \subseteq \bar{X}$ *and* $\bar{Y} \in Iob_{\tau\rightarrow\tau\rightarrow o}(\bar{X})$ *and* $\bar{X} \subseteq \bar{Z}$,
then $(\bar{Z} \setminus \bar{X}) \cup \bar{Y} \in Iob_{\tau\rightarrow\tau\rightarrow o}(\bar{Z})$.
(ob5) *If* $\bar{Y} \subseteq \bar{X}$ *and* $\bar{Z} \in Iob_{\tau\rightarrow\tau\rightarrow o}(\bar{X})$ *and* $\bar{Y} \cap \bar{Z} \neq \emptyset$,
then $\bar{Z} \in Iob_{\tau\rightarrow\tau\rightarrow o}(\bar{Y})$.

Proof See Appendix 1.6

Lemma 2 *Let* $H^M = \langle\{D_\alpha\}_{\alpha\in T}, I\rangle$ *be a Henkin model for a DDL model* M. *We have* $H^M \models^{HOL} \Sigma$ *for all* $\Sigma \in \{AV, PV1, PV2, OB1, ..., OB5\}$, *where*
AV is $\forall W_i \exists V_i (av_{i\rightarrow\tau} W_i V_i)$
PV1 is $\forall W_i \forall V_i (av_{i\rightarrow\tau} W_i V_i \rightarrow pv_{i\rightarrow\tau} W_i V_i)$
PV2 is $\forall W_i (pv_{i\rightarrow\tau} W_i W_i)$
OB1 is $\forall X_\tau \neg ob_{\tau\rightarrow\tau\rightarrow o} X_\tau (\lambda X_\tau \bot)$
OB2 is $\forall X_\tau Y_\tau Z_\tau (\ (\forall W_i ((Y_\tau W_i \wedge X_\tau W_i) \longleftrightarrow (Z_\tau W_i \wedge X_\tau W_i)))$
$\rightarrow (ob_{\tau\rightarrow\tau\rightarrow o} X_\tau Y_\tau \longleftrightarrow ob_{\tau\rightarrow\tau\rightarrow o} X_\tau Z_\tau))$
OB3 is $\forall \beta_{\tau\rightarrow\tau\rightarrow o} \forall X_\tau$
$(\ ((\forall Z_\tau (\beta_{\tau\rightarrow\tau\rightarrow o} Z_\tau \rightarrow ob_{\tau\rightarrow\tau\rightarrow o} X_\tau Z_\tau)) \wedge \exists Z_\tau (\beta_{\tau\rightarrow\tau\rightarrow o} Z_\tau))$
$\rightarrow (\ (\exists Y_i (((\lambda W_i \forall Z_\tau (\beta_{\tau\rightarrow\tau\rightarrow o} Z_\tau \rightarrow Z_\tau W_i)) Y_i) \wedge X_\tau Y_i))$
$\rightarrow ob_{\tau\rightarrow\tau\rightarrow o} X_\tau (\lambda W_i \forall Z_\tau (\beta_{\tau\rightarrow\tau\rightarrow o} Z_\tau \rightarrow Z_\tau W_i))))$
OB4 is $\forall X_\tau Y_\tau Z_\tau$
$(\ (\forall W_i (Y_\tau W_i \rightarrow X_\tau W_i) \wedge ob_{\tau\rightarrow\tau\rightarrow o} X_\tau Y_\tau \wedge \forall X_\tau (X_\tau W_i \rightarrow Z_\tau W_i))$
$\rightarrow ob_{\tau\rightarrow\tau\rightarrow o} Z_\tau (\lambda W_i ((Z_\tau W_i \wedge \neg X_\tau W_i) \vee Y_\tau W_i)))$
OB5 is $\forall X_\tau Y_\tau Z_\tau$
$(\ (\forall W_i (Y_\tau W_i \rightarrow X_\tau W_i) \wedge ob_{\tau\rightarrow\tau\rightarrow o} X_\tau Z_\tau \wedge \exists W_i (Y_\tau W_i \wedge Z_\tau W_i))$
$\rightarrow ob_{\tau\rightarrow\tau\rightarrow o} Y_\tau Z_\tau)$

Proof See Appendix 1.6

Lemma 3 *Let* H^M *be a Henkin model for a DDL model* M. *For all DDL formulas* δ, *arbitrary variable assignments* g *and worlds* s *it holds:*

$$M, s \models \delta \text{ if and only if } \|\lfloor\delta\rfloor\, S_i\|^{H^M, g[s/S_i]} = T$$

Proof See Appendix 1.6

Lemma 4 *For every Henkin model* $H = \langle\{D_\alpha\}_{\alpha\in T}, I\rangle$ *such that* $H \models^{HOL} \Sigma$ *for all* $\Sigma \in$ *{AV, PV1, PV2, OB1,..., OB5}, there exists a corresponding DDL model* M. *Corresponding means that for all DDL formulas* δ *and for all assignments* g *and worlds* s, $\|\lfloor\delta\rfloor S_i\|^{H, g[s/S_i]} = T$ *if and only if* $M, s \models \delta$.

Proof Suppose that $H = \langle\{D_\alpha\}_{\alpha\in T}, I\rangle$ is a Henkin model such that $H \models^{\mathrm{HOL}} \Sigma$ for all $\Sigma \in$ {AV, PV1, PV2, OB1,..,OB5}. Without loss of generality, we can assume that the domains of H are denumerable [11]. We construct the corresponding DDL model M as follows:

1. $S = D_i$,
2. $u \in av(s)$ for $s, u \in S$ iff $Iav_{i\to\tau}(s, u) = T$,
3. $u \in pv(s)$ for $s, u \in S$ iff $Ipv_{i\to\tau}(s, u) = T$,
4. $\bar{Y} \in ob(\bar{X})$ for $\bar{X}, \bar{Y} \in D_i \longrightarrow D_o$ iff $Iob_{\tau\to\tau\to o}(\bar{X}, \bar{Y}) = T$, and
5. $s \in V(p^j)$ iff $Ip^j_\tau(s) = T$.

Since $H \models^{\mathrm{HOL}} \Sigma$ for all $\Sigma \in$ {AV, PV1, PV2, OB1, .., OB5}, it is straightforward (but tedious) to verify that av, pv and ob satisfy the conditions as required for a DDL model.

Moreover, the above construction ensures that H is a Henkin model H^M for DDL model M. Hence, Lemma 3 applies. This ensures that for all DDL formulas δ, for all assignment g and all worlds s we have $\|\lfloor\delta\rfloor S_i\|^{H,g[s/S_i]} = T$ if and only if $M, s \models \delta$.□

Theorem 1 (Soundness and Completeness of the Embedding)

$$\models^{DDL} \varphi \textit{ if and only if } \{\textit{AV, PV1, PV2, OB1,..,OB5}\} \models^{HOL} \textit{vld} \lfloor\varphi\rfloor$$

Proof (Soundness, $\longleftarrow$) The proof is by contraposition. Assume $\not\models^{DDL} \varphi$, that is, there is a DDL model $M = \langle S, av, pv, ob, V\rangle$, and world $s \in S$, such that $M, s \not\models \varphi$. Now let H^M be a Henkin model for DDL model M. By Lemma 3, for an arbitrary assignment g, it holds that $\|\lfloor\varphi\rfloor\, S_i\|^{H^M,g[s/S_i]} = F$. Thus, by definition of $\|.\|$, it holds that $\|\forall S_i(\lfloor\varphi\rfloor\, S)\|^{H^M,g} = \|\text{vld}\,\lfloor\varphi\rfloor\|^{H^M,g} = F$. Hence, $H^M \not\models^{\mathrm{HOL}} \text{vld}\,\lfloor\varphi\rfloor$. Furthermore, $H^M \models^{\mathrm{HOL}} \Sigma$ for all $\Sigma \in$ {AV, PV1, PV2, OB1,. . . ,OB5} by Lemma 2. Thus, {AV, PV1, PV2, OB1,..,OB5} $\not\models^{\mathrm{HOL}} \text{vld}\,\lfloor\varphi\rfloor$.

(Completeness, $\longrightarrow$) The proof is again by contraposition. Assume {AV, PV1, PV2, OB1,..,OB5} $\not\models^{\mathrm{HOL}} \text{vld}\,\lfloor\varphi\rfloor$, that is, there is a Henkin model $H = \langle\{D_\alpha\}_{\alpha\in T}, I\rangle$ such that $H \models^{\mathrm{HOL}} \Sigma$ for all $\Sigma \in$ {AV, PV1, PV2, OB1,..,OB5}, but $\|\text{vld}\,\lfloor\varphi\rfloor\|^{H,g} = F$ for some assignment g . By Lemma 4, there is a DDL model M such that $M \not\models \varphi$. Hence, $\not\models^{DDL} \varphi$. □

Each DDL reasoning problem thus represents a particular HOL problem. The embedding presented in this section, which is based on simple abbreviations, tells us how the two logics are connected.

1.5 Implementation and Experiments in Isabelle/HOL

The semantical embedding from Section 1.4.1 has been implemented in the higher-order proof assistant Isabelle/HOL [23]. Figure 1.1 displays the entire encoding. We provide some explanations:

- Line 4: the primitive type i for possible words is introduced.

```
1 theory CJ_DDL imports Main          (* Christoph Benzmüller & Xavier Parent & Ali Farjami, 2020 *)
2
3 begin (* DDL: Dyadic Deontic Logic by Carmo and Jones *)
4  typedecl i (*type for possible worlds*)
5  type_synonym τ = "(i⇒bool)"
6  type_synonym γ = "τ⇒τ"
7  type_synonym ϱ = "τ⇒τ⇒τ"
8
9  consts av::"i⇒τ" pv::"i⇒τ" ob::"τ⇒(τ⇒bool)" (*accessibility, resp. neighborhood, relations*)
10         cw::i (*current world*)
11
12  axiomatization where
13   ax_3a: "∀w.∃x. av(w)(x)" and
14   ax_4a: "∀w x. av(w)(x) ⟶ pv(w)(x)" and
15   ax_4b: "∀w. pv(w)(w)" and
16   ax_5a: "∀X.¬ob(X)(λx. False)" and
17   ax_5b: "∀X Y Z. (∀w. ((Y(w) ∧ X(w)) ⟷ (Z(w) ∧ X(w)))) ⟶ (ob(X)(Y) ⟷ ob(X)(Z))" and
18   ax_5c: "∀X Y Z. (((∃w. (X(w) ∧ Y(w) ∧ Z(w))) ∧ ob(X)(Y) ∧ ob(X)(Z))
19                   ⟶ ob(X)(λw. Y(w) ∧ Z(w)))" and
20   ax_5d: "∀X Y Z. ((∀w. Y(w) ⟶ X(w)) ∧ ob(X)(Y) ∧ (∀w. X(w) ⟶ Z(w)))
21                   ⟶ ob(Z)(λw. (Z(w) ∧ ¬X(w)) ∨ Y(w))" and
22   ax_5e: "∀X Y Z. ((∀w. Y(w) ⟶ X(w)) ∧ ob(X)(Z) ∧ (∃w. Y(w) ∧ Z(w))) ⟶ ob(Y)(Z)"
23
24  abbreviation ddltop::τ ("⊤") where "⊤ ≡ λw. True"
25  abbreviation ddlbot::τ ("⊥") where "⊥ ≡ λw. False"
26  abbreviation ddlneg::γ ("¬_"[52]53) where "¬A ≡ λw. ¬A(w)"
27  abbreviation ddland::ϱ (infixr"∧"51) where "A∧B ≡ λw. A(w)∧B(w)"
28  abbreviation ddlor::ϱ (infixr"∨"50) where "A∨B ≡ λw. A(w)∨B(w)"
29  abbreviation ddlimp::ϱ (infixr"→"49) where "A→B ≡ λw. A(w)⟶B(w)"
30  abbreviation ddlequiv::ϱ (infixr"↔"48) where "A↔B ≡ λw. A(w)⟷B(w)"
31  abbreviation ddlbox::γ ("□") where "□A ≡ λw.∀v. A(v)"
32  abbreviation ddlboxa::γ ("□a") where "□aA ≡ λw. (∀x. av(w)(x) ⟶ A(x))"
33  abbreviation ddlboxp::γ ("□p") where "□pA ≡ λw. (∀x. pv(w)(x) ⟶ A(x))"
34  abbreviation ddldia::γ ("◇") where "◇A ≡ ¬□(¬A)"
35  abbreviation ddldiaa::γ ("◇a") where "◇aA ≡ ¬□a(¬A)"
36  abbreviation ddldiap::γ ("◇p") where "◇pA ≡ ¬□p(¬A)"
37  abbreviation ddlo::ϱ ("O⟨_|_⟩"[52]53) where "O⟨B|A⟩ ≡ λw. ob(A)(B)"
38  abbreviation ddloa::γ ("Oa") where "OaA ≡ λw. ob(av(w))(A) ∧ (∃x. av(w)(x) ∧ ¬A(x))"
39  abbreviation ddlop::γ ("Op") where "OpA ≡ λw. ob(pv(w))(A) ∧ (∃x. pv(w)(x) ∧ ¬A(x))"
40
41  abbreviation ddlvalid::"τ ⇒ bool" ("⌊_⌋"[7]105) where "⌊A⌋ ≡ ∀w. A w"  (*global validity*)
42  abbreviation ddlvalidcw::"τ ⇒ bool" ("⌊_⌋ι"[7]105) where "⌊A⌋ι ≡ A cw"  (*local validity (in cw)*)
43
44 (* A is obligatory (monadic operator). *)
45  abbreviation ddlobl::γ ("○<_>") where "○<A> ≡  O⟨A|⊤⟩"
46
47 (* Consistency *)
48  lemma True nitpick [satisfy,user_axioms,show_all] oops
49 end
```

Fig. 1.1 Shallow semantical embedding of DDL in Isabelle/HOL.

- Line 5: a type abbreviation τ for type $i \to o$ is declared; τ is the type of DDL formulas, which are encoded as predicates on worlds in HOL.
- Lines 6–7: further type abbreviations γ and ϱ for (τ-lifted) unary and binary DDL connectives in HOL are introduced.
- Line 9: the constants av, pv and ob are declared; they denote accessibility relations, resp. neighborhood relations, and they are used below to define the operators $\Box_a$, $\Box_p$ and $\bigcirc(\_/\_)$.
- Line 10: a designated constant for the actual/current world (cw) is introduced.
- Lines 12–22: the axioms for av, pv and ob are postulated.
- Lines 24–30: the (τ-lifted) Boolean connectives are defined in the usual way [22].

- Lines 31–33: the three necessity operators $\Box$, $\Box_a$ ("in all actual worlds") and $\Box_p$ ("in all possible worlds") are introduced; the former is declared as a universal (S5) modal operator and the latter two use av and pv as guards in their definitions.
- Lines 34–36: the dual possibility operators $\Diamond$, $\Diamond_a$ and $\Diamond_p$ are introduced.
- Line 37: using the neighborhood relation ob, the dyadic obligation operator $\bigcirc$ ("it ought to be . . . , given . . . ") is defined.
- Lines 38–39: using av, pv and ob, the actual and primary obligation operators $\bigcirc_a$ (actual obligation) and $\bigcirc_p$ (primary obligation) are defined.
- Lines 41–42: the notions of global validity (i.e, truth in all worlds) and local validity (truth at the actual world) are introduced.
- Line 45: a monadic obligation operator is defined based on dyadic obligation.
- Line 48: the model finder Nitpick [24] confirms the consistency of the introduced theory; the reported model (not displayed here) consists of a single world i_1, which is self-connected via the accessibility relations av and pv, whereas the neighborhood relation ob is the empty relation.

Figure 1.2 reports on some meta-theoretical experiments. We briefly explain them:

- Lines 4–7: it is shown that the rules of modus ponens and necessitation for DDL are implied by the semantic embedding as provided in Fig. 1.1; their validity is automatically proved here by Isabelle/HOL's simplifier "simp".[7]
- Lines 10–12: it is proved that $\Box$ is a S5 modal operator.
- Line 15: it is proved that $\Box_p$ validates the T axiom; $\Box_p$ is hence a modal operator of type KT (in Chellas [26]'s nomenclature).
- Lines 16–17: it is confirm that $\Box_p$ is not a S5 modality; Nitpick finds countermodels for the axioms 4 and B.
- Line 20: it is shown that $\Box_a$ validates the D axiom; $\Box_a$ is hence a modal operator of type KD. Lines 21–23: it is confirmed that $\Box_a$ is not a S5 modality; Nitpick finds countermodels for the axioms T, S4 and B.
- Lines 26–27: inclusion relations for $\Box$, $\Box_a$ and $\Box_p$ are confirmed.
- Lines 30–31: the observation II-2-1 of Carmo and Jones [20] is proved.
- Lines 34–44: the validity of a number of laws involving the dyadic obligation operator are verified.

Figure 1.3 continues the meta-theoretical experiments:

- Lines 47–50: the validity of a number of laws involving $\bigcirc_a$, $\bigcirc_p$, $\Box_a$ and $\Box_p$ is verified.
- Lines 53–54: it is proved that the so-called law of factual detachment holds in two versions.

---

[7] The proofs in our experiments have actually been provided by first calling the "sledgehammer" tool [25] in Isabelle/HOL, which then, after automatically proving the goals with state-of-the-art automated theorem proving systems, suggested the use of more trusted tactics, such as Isabelle/HOL's simplifier "simp", to close the proof goals. Only occasionally sledgehammer failed to directly prove the given statements. In such cases, some intermediate proof steps may be interactively provided by the user to assist the automated theorem provers. An example is given in lines 41–44, where one intermediate proof step (line 42) is stipulated in order to help the automated reasoning tools to prove the lemma stated in line 40.

```
theory CJ_DDL_Tests imports CJ_DDL  (* Christoph Benzmüller & Ali Farjami & Xavier Parent, 2020 *)

begin (* Modus Ponens and Necessitation of the embedded DDL are implied. *)
lemma MP: "⟦⌊A⌋; ⌊A → B⌋⟧ ⟹ ⌊B⌋" by simp
lemma Nec: "⌊A⌋ ⟹ ⌊□A⌋" by simp
lemma Neca: "⌊A⌋ ⟹ ⌊□aA⌋"  by simp
lemma Necp: "⌊A⌋ ⟹ ⌊□p A⌋"  by simp

(* "□" is an S5 modality *)
lemma C_1_refl: "⌊□A → A⌋" by simp
lemma C_1_trans: "⌊□A → (□(□A))⌋"  by simp
lemma C_1_sym: "⌊A → (□(◇A))⌋"  by simp

(* "□p" is an KT modality *)
lemma C_9_p_refl: "⌊□pA → A⌋"  by (simp add: ax_4b)
lemma "⌊□pA → (□p(□pA))⌋" nitpick [user_axioms] oops (* countermodel *)
lemma "⌊A → (□p(◇pA))⌋"   nitpick [user_axioms] oops (* countermodel *)

(* "□a" is an KD modality *)
lemma C_10_a_serial: "⌊□aA → ◇aA⌋"  by (simp add: ax_3a)
lemma "⌊□aA → A⌋" nitpick [user_axioms] oops (* countermodel *)
lemma "⌊□aA → (□a(□aA))⌋" nitpick [user_axioms] oops (* countermodel *)
lemma "⌊A → (□a(◇aA))⌋" nitpick [user_axioms] oops (* countermodel *)

(* Relationship between "□,□a,□p" *)
lemma C_11: "⌊□A → □pA⌋"  by simp
lemma C_12: "⌊□pA → □aA⌋" using ax_4a by auto

(* Observation II-2-1 *)
lemma ax_5b': "ob X Y ⟷ ob X (λz. X z ∧ Y z)" by (metis (no_types, lifting) ax_5b)
lemma ax_5b'': "ob X Y ⟷ ob X (λz. Y z ∧ X z)" by (metis (no_types, lifting) ax_5b)

(* Characterisation of "O" *)
lemma C_2: "⌊O⟨A|B⟩ → ◇(B ∧ A)⌋"  by (metis ax_5a ax_5b)
lemma C_3: "⌊(◇(A ∧ B ∧ C) ∧ O⟨B|A⟩ ∧ O⟨C|A⟩ ) → O⟨(B ∧ C)|A⟩⌋" using ax_5c by auto
lemma C_4: "⌊(□(A → B) ∧ (◇(A ∧ C)) ∧ O⟨C|B⟩) → O⟨C|A⟩⌋"   using ax_5e by blast
lemma C_5: "⌊□(A ↔ B) → (O⟨C|A⟩ ↔ O⟨C|B⟩)⌋"  by presburger
lemma C_6: "⌊□(C → (A ↔ B)) → (O⟨A|C⟩ ↔ O⟨B|C⟩)⌋"  by (smt ax_5b)
lemma C_7: "⌊O⟨B|A⟩ → □(O⟨B|A⟩)⌋"  by blast
lemma C_8: "⌊O⟨B|A⟩ → O⟨(A → B)|⊤⟩⌋"
 proof -
   have "∀X Y Z. (ob X Y ∧ (∀w. X w  ⟶ Z w)) ⟶ ob Z (λw. (Z w ∧ ¬X w) ∨ Y w)"
     by (smt ax_5d ax_5b ax_5b'')
   thus ?thesis  using ax_5b by fastforce qed
```

Fig. 1.2 Experiments (meta-theory) with the embedding of DDL in Isabelle/HOL.

- Line 57–63: the observation II-3-1 of Carmo and Jones [20], which is required for the proof of their soundness theorem, is proved.
- Lines 66–93: a number of observations and results as reported by Carmo and Jones [20] are proved automatically.

1.6 Conclusion

A shallow semantical embedding of Carmo and Jones's logic of contrary-to-duty conditionals in classical higher-order logic has been presented and shown to be faithful (sound an complete). This embedding has been implemented in the proof assistant Isabelle/HOL, resulting in the first interactive and automated theorem prover for this logic that we are aware of. Moreover, the work reported in this paper has pro-

```
(* Relationship between "O_a,O_p,□_a,□_p" *)
lemma C_13_a: "⌊□_aA → (¬O_aA ∧ ¬O_a(¬A))⌋"  by (metis (full_types) ax_5a ax_5b)
lemma C_13_b: "⌊□_pA → (¬O_pA ∧ ¬O_p(¬A))⌋"   by (metis (full_types) ax_5a ax_5b)
lemma C_14_a: "⌊□_a(A ↔ B) → (O_aA ↔ O_aB)⌋"  by (metis ax_5b)
lemma C_14_b: "⌊□_p(A ↔ B) → (O_pA ↔ O_pB)⌋"  by (metis ax_5b)

(* Relationship between "O,O_a,O_p,□_a,□_p" *)
lemma C_15_a: "⌊(O⟨B|A⟩ ∧ □_aA ∧ ◇_aB ∧ ◇_a(¬B)) → O_aB⌋"  using ax_5e by blast
lemma C_15_b: "⌊(O⟨B|A⟩ ∧ □_pA ∧ ◇_pB ∧ ◇_p(¬B)) → O_pB⌋"  using ax_5e by blast

(* Soundness and consistency *)
lemma II_3_1: "((⌊O⟨B|A⟩⌋) ∧ (∃x. Z(x) ∧ A(x) ∧ B(x))) ⟶ ob(Z)(A → B)"
  proof
   assume "(⌊O⟨B|A⟩⌋) ∧ (∃x. Z(x) ∧ A(x) ∧ B(x))"
   hence "ob (λz. A z ∧ Z z) (λz. A z ∧ Z z ∧ B z)" using ax_5e ax_5b  ax_5b' ax_5d by smt
   hence "ob (λz. Z z ∧ A z) (λz. Z z ∧ A z ∧ B z)" using ax_5e ax_5b  ax_5b' ax_5d by smt
   hence "ob Z (λw. (Z w ∧ ¬(Z w ∧ A w)) ∨ (Z w ∧ A w ∧ B w))" by (metis (mono_tags) ax_5d)
   from this show  L19: "ob(Z)(A → B)" by (smt ax_5b)  qed

(* Some theorems and derived (proof) rules *)
lemma II_4_1: "⌊□(A ↔ B) → (C(A) ↔ C(B))⌋"  using ext by blast
lemma obs_II_4_1_a: "⌊A ↔ B⌋ ⟹ ⌊C(A) ↔ C(B)⌋"  using ext by blast
lemma obs_II_4_1_b: "⌊A ↔ B⌋ ⟹ ⌊(◇(A ∧ C) ∧ O⟨C|B⟩) → O⟨C|A⟩⌋"  using ax_5e by blast
lemma obs_II_4_1_c_1: "⌊◇(O⟨B|A⟩) → ◇(□(O⟨B|A⟩))⌋"  by blast
lemma obs_II_4_1_c_2: "⌊◇(□(O⟨B|A⟩)) → ◇(O⟨B|A⟩)⌋"  by auto
lemma obs_II_4_1_c_3: "⌊◇(O⟨B|A⟩) → □(O⟨B|A⟩)⌋"  by blast
lemma obs_II_4_1_c_4: "⌊◇(¬(O⟨B|A⟩)) → □(¬(O⟨B|A⟩))⌋"  by blast
lemma res_II_4_1_a_1: "⌊¬(O⟨⊥|A⟩)⌋"  by (simp add: ax_5a)
lemma res_II_4_1_a_2: "⌊(◇_p(A ∧ B ∧ C) ∧ O⟨B|A⟩ ∧ O⟨C|A⟩) → O⟨(B ∧ C)|A⟩⌋" using C_3 by auto
lemma res_II_4_1_a_3: "⌊O⟨B|A⟩ → O⟨B|(A ∧ B)⟩⌋"  by (smt ax_5a ax_5b ax_5e)
lemma res_II_4_1_a_4: "⌊◇_p(O⟨B|A⟩) → □_p(O⟨B|(A ∧ B)⟩)⌋"  by (smt ax_5a ax_5b ax_5e)
lemma res_II_4_1_a_5: "⌊(◇_p(A ∧ B ∧ C) ∧ O⟨C|A⟩) → O⟨C|(A ∧ B)⟩⌋"  by (smt ax_5a ax_5b ax_5e)
lemma res_II_4_1_b_1: "⌊A ↔ B⌋ ⟹ ⌊O⟨C|A⟩ ↔ O⟨C|B⟩⌋"  by (smt ax_5a ax_5b ax_5e)
lemma res_II_4_1_b_2: "⌊C → (A ↔ B)⌋ ⟹ ⌊O⟨A|C⟩ ↔ O⟨B|C⟩⌋"  by (smt ax_5b)
lemma obs_II_4_2_1: "⌊(O⟨B|A⟩ ∧ ◇_a(A ∧ B) ∧ ◇_a(A ∧ ¬B))
                        → (O⟨B|A⟩ ∧ ◇_a(A → B) ∧ ◇_a(¬(A → B)))⌋" by blast
lemma obs_II_4_2_2: "⌊O⟨B|A⟩ → O⟨(A → B)|⊤⟩⌋" by (simp add: C_8)
lemma obs_II_4_2_3: "⌊(O⟨(A → B)|⊤⟩ ∧ □_a⊤ ∧ ◇_a(A → B) ∧ ◇_a(¬(A → B)))
                        → O_a(A → B)⌋" by (smt ax_5e)
lemma obs_II_4_2_4: "⌊□_a⊤⌋" by simp
lemma obs_II_4_2_5: "⌊(O⟨(A → B)|⊤⟩ ∧ ◇_a(A → B) ∧ ◇_a(¬(A → B))) → O_a(A → B)⌋" by (smt ax_5e)
lemma obs_II_4_2_6: "⌊(O⟨B|A⟩ ∧ ◇_a(A ∧ B) ∧ ◇_a(A ∧ ¬B)) → O_a(A → B)⌋" by (simp add: II_3_1)
lemma obs_II_4_2_6_p: "⌊(O⟨B|A⟩ ∧ ◇_p(A ∧ B) ∧ ◇_p(A ∧ ¬B)) → O_p(A → B)⌋" by (simp add: II_3_1)

lemma Oa_C: "⌊◇_a(A ∧ B) ∧ O_aA ∧ O_aB →  O_a(A ∧ B)⌋"  using ax_5c by auto
lemma Op_C: "⌊◇_p(A ∧ B) ∧ O_pA ∧ O_pB →  O_p(A ∧ B)⌋"  using ax_5c by auto
lemma Oa_DD: "⌊(O_aA ∧ O⟨B|A⟩ ∧ ◇_a(A ∧ B)) → O_a(A ∧ B)⌋" using ax_5b ax_5c obs_II_4_2_6 by smt
lemma Op_DD: "⌊(O_pA ∧ O⟨B|A⟩ ∧ ◇_p(A ∧ B)) → O_p(A ∧ B)⌋" using ax_5b ax_5c obs_II_4_2_6_p by smt
end
```

Fig. 1.3 Experiments (meta-theory) with the embedding of DDL in Isabelle/HOL (cont'd from Fig. 1.2).

vided important inspiration and impetus for the development of the larger LogiKEy [4] framework and methodology for pluralistic, expressive normative reasoning. In the context of this larger project further case studies with extensions of the logic by Carmo and Jones have successfully been conducted [7, 8], which in turn motivates much further work towards the practical employment of the presented approach.

Acknowledgements We want to thank . . .

References

1. Carmo J, Jones AJI. Completeness and decidability results for a logic of contrary-to-duty conditionals. J Log Comput. 2013;23(3):585–626. Available from: http://dx.doi.org/10.1093/logcom/exs009.
2. Benzmüller C. Universal (Meta-)Logical Reasoning: Recent Successes. Science of Computer Programming. 2019;172:48–62.
3. Kirchner D, Benzmüller C, Zalta EN. Computer Science and Metaphysics: A Cross-Fertilization. Open Philosophy. 2019;2:230–251.
4. Benzmüller C, Parent X, van der Torre L. Designing Normative Theories for Ethical and Legal Reasoning: LogiKEy Framework, Methodology, and Tool Support. Artificial Intelligence. 2020;Accepted; preprint arXiv:1903.10187.
5. Gewirth A. Reason and Morality. University of Chicago Press; 1981.
6. Beyleveld D. The Dialectical Necessity of Morality: An Analysis and Defense of Alan Gewirth's Argument to the Principle of Generic Consistency. University of Chicago Press; 1991.
7. Fuenmayor D, Benzmüller C. Harnessing Higher-Order (Meta-)Logic to Represent and Reason with Complex Ethical Theories. In: PRICAI 2019: Trends in Artificial Intelligence. vol. 11670 of LNCS. Springer; 2019. p. 418–432.
8. Fuenmayor D, Benzmüller C. Mechanised Assessment of Complex Natural-Language Arguments using Expressive Logic Combinations. In: Frontiers of Combining Systems, 12th International Symposium, FroCoS 2019. vol. 11715 of LNAI. Springer; 2019. p. 112–128.
9. Benzmüller C, Andrews P. Church's Type Theory. In: Zalta EN, editor. The Stanford Encyclopedia of Philosophy. summer 2019 ed. Metaphysics Research Lab, Stanford University; 2019. p. 1–62 (in pdf version). Available from: https://plato.stanford.edu/entries/type-theory-church/.
10. Church A. A formulation of the simple theory of types. Journal of Symbolic Logic. 1940;5(2):56–68.
11. Henkin L. Completeness in the theory of types. Journal of Symbolic Logic. 1950;15(2):81–91.
12. Andrews PB. Resolution in type theory. Journal of Symbolic Logic. 1971;36(3):414–432.
13. Andrews PB. General models and extensionality. Journal of Symbolic Logic. 1972;37(2):395–397.
14. Andrews PB. General models, descriptions, and choice in type theory. Journal of Symbolic Logic. 1972;37(2):385–394.
15. Church A. A set of postulates for the foundation of logic. Annals of Mathematics. 1932;33(3):346–366.
16. Church A. An unsolvable problem of elementary number theory. American Journal of Mathematics. 1936;58(2):354–363.
17. Benzmüller C, Brown C, Kohlhase M. Higher-Order Semantics and Extensionality. Journal of Symbolic Logic. 2004;69(4):1027–1088. Available from: http://christoph-benzmueller.de/papers/J6.pdf.
18. Muskens R. Intensional models for the theory of types. Journal of Symbolic Logic. 2007;75(1):98–118.
19. Benzmüller C, Miller D. Automation of Higher-Order Logic. In: Gabbay DM, Siekmann JH, Woods J, editors. Handbook of the History of Logic, Volume 9 — Computational Logic. North Holland, Elsevier; 2014. p. 215–254. Available from: http://christoph-benzmueller.de/papers/B5.pdf.
20. Carmo J, Jones AJI. Deontic logic and contrary-to-duties. In: Gabbay DM, Guenthner F, editors. Handbook of Philosophical Logic: Volume 8. Dordrecht: Springer Netherlands; 2002. p. 265–343.
21. Benzmüller C. Cut-Elimination for Quantified Conditional Logic. Journal of Philosophical Logic. 2017;46(3):333–353. Available from: http://christoph-benzmueller.de/papers/J31.pdf.
22. Benzmüller C, Paulson LC. Quantified Multimodal Logics in Simple Type Theory. Logica Universalis (Special Issue on Multimodal Logics). 2013;7(1):7–20. Available from: http://christoph-benzmueller.de/papers/J23.pdf.

23. Nipkow T, Paulson LC, Wenzel M. Isabelle/HOL — A Proof Assistant for Higher-Order Logic. vol. 2283 of LNCS. Springer; 2002.
24. Blanchette JC, Nipkow T. Nitpick: A Counterexample Generator for Higher-Order Logic Based on a Relational Model Finder. In: ITP 2010. No. 6172 in LNCS. Springer; 2010. p. 131–146.
25. Blanchette JC, Böhme S, Paulson LC. Extending Sledgehammer with SMT Solvers. J of Automated Reasoning. 2013;51(1):109–128.
26. Chellas B. Modal Logic. Cambridge: Cambridge University Press; 1980.

Appendix

Proof of Lemma 1

Proof Each statement follows by construction of H^M for M.

(av) By definition of av for $s \in S$ in M, $av(s) \neq \emptyset$; hence, there is $u \in S$ such that $u \in av(s)$. By definition of H^M, $Iav_{i \to \tau}(s, u) = T$, so $u \in Iav_{i \to \tau}(s)$ and hence $Iav_{i \to \tau}(s) \neq \emptyset$ in H^M.
(pv1) By definition of av and pv for $s \in S$ in M, $av(s) \subseteq pv(s)$; hence, for every $u \in av(s)$ we have $u \in pv(s)$. In H^M this means, if $Iav_{i \to \tau}(s, u) = T$, then $Ipv_{i \to \tau}(s, u) = T$. So, $Iav_{i \to \tau}(s) \subseteq Ipv_{i \to \tau}(s)$ in H^M.
(pv2) This case is similar to (av).
(ob1) By definition of ob, we have $\emptyset \notin ob(\bar{X})$; hence, in H^M, $Iob_{\tau \to \tau \to o}(\bar{X}, \emptyset) = F$, that is $\emptyset \notin Iob_{\tau \to \tau \to o}(\bar{X})$.
(ob2) Suppose $\bar{Y} \cap \bar{X} = \bar{Z} \cap \bar{X}$. In M we have $\bar{Y} \in ob(\bar{X})$ iff $\bar{Z} \in ob(\bar{X})$. By definition of H^M we have $Iob_{\tau \to \tau \to o}(\bar{X}, \bar{Y}) = T$ iff $Iob_{\tau \to \tau \to o}(\bar{X}, \bar{Z}) = T$. Hence, $\bar{Y} \in Iob_{\tau \to \tau \to o}(\bar{X})$ iff $\bar{Z} \in Iob_{\tau \to \tau \to o}(\bar{X})$ in H^M.
(ob3) Suppose $\bar{\beta} \subseteq Iob_{\tau \to \tau \to o}(\bar{X})$ and $\bar{\beta} \neq \emptyset$. If $(\cap\bar{\beta}) \cap \bar{X} \neq \emptyset$, by definition of ob in M we have $(\cap\bar{\beta}) \in ob(\bar{X})$. Hence, in H^M, $Iob_{\tau \to \tau \to o}(\bar{X}, (\cap\bar{\beta})) = T$ and then $(\cap\bar{\beta}) \in Iob_{\tau \to \tau \to o}(\bar{X})$.
(ob4) and (ob5) are similar to (ob2).

Proof of Lemma 2

Proof We present detailed arguments for most cases.

AV: For all $s \in D_i$: $Iav_{i \to \tau}(s) \neq \emptyset$ (by Lemma 1 (av))
$\Leftrightarrow$ For all $s \in D_i$, there exists $u \in D_i$ such that $Iav_{i \to \tau}(s, u) = T$
$\Leftrightarrow$ For all assignments g, for all $s \in D_i$, there exists $u \in D_i$ such that $\|av\, W\, V\|^{H^M, g[s/W_i][u/V_i]} = T$
$\Leftrightarrow$ For all g, all $s \in D_i$ we have $\|\exists V(av\, W\, V)\|^{H^M, g[s/W_i]} = T$
$\Leftrightarrow$ For all g we have $\|\forall W \exists V(av\, W\, V)\|^{H^M, g} = T$

$\Leftrightarrow$ $H^M \models^{\text{HOL}} AV$

PV1: Given an arbitary assignment g, and arbitary $s, u \in D_i$ such that $\|av\, W\, V\|^{H^M, g[s/W_i][u/V_i]} = T$

$\Leftrightarrow$ $Iav_{i\to\tau}(s,u) = T$

$\Rightarrow$ $Ipv_{i\to\tau}(s,u) = T$ $\quad (Iav_{i\to\tau}(s) \subseteq Ipv_{i\to\tau}(s)$, by Lemma 1 (pv1))

$\Leftrightarrow$ $\|pv\, W\, V\|^{H^M, g[s/W_i][u/V_i]} = T$

Hence by definition of $\|.\|$, for all g, for all $s, u \in D_i$ we have:

$\|av\, W\, V\|^{H^M, g[s/W_i][u/V_i]} = T$ implies $\|pv\, W\, V\|^{H^M, g[s/W_i][u/V_i]} = T$

$\Leftrightarrow$ For all g, all $s, u \in D_i$ we have $\|av\, W\, V \rightarrow pv\, W\, V\|^{H^M, g[s/W_i][u/V_i]} = T$

$\Leftrightarrow$ For all g, all $s \in D_i$ we have $\|\forall V\ (av\, W\, V \rightarrow pv\, W\, V)\|^{H^M, g[s/W_i]} = T$

$\Leftrightarrow$ For all g we have $\|\forall W\, \forall V\ (av\, W\, V \rightarrow pv\, W\, V)\|^{H^M, g} = T$

$\Leftrightarrow$ $H^M \models^{\text{HOL}} PV1$

PV2: This case is analogous to AV.

OB1: For all $\bar{X} \in D_\tau : \emptyset \notin Iob_{\tau\to\tau\to o}(\bar{X})$ $\quad$ (by Lemma 1 (ob1))

$\Leftrightarrow$ For all g, all $\bar{X} \in D_\tau$ we have $\|\neg ob\, X\, (\lambda X.\bot)\|^{H^M, g[\bar{X}/X_\tau]} = T$

$\Leftrightarrow$ For all g we have $\|\forall X\, \neg(ob\, X\, (\lambda X_\tau \bot))\|^{H^M, g[\bar{X}/X_\tau]} = T$

$\Leftrightarrow$ $H^M \models^{\text{HOL}} OB1$

OB2: Given an arbitary assignment g, and arbitary $\bar{X}, \bar{Y}, \bar{Z} \in D_\tau$ such that $\|\forall W((Y\, W \wedge X\, W) \longleftrightarrow (Z\, W \wedge X\, W))\|^{H^M, g[\bar{X}/X_\tau][\bar{Y}/Y_\tau][\bar{Z}/Z_\tau]} = T$

$\Leftrightarrow$ For all $s \in D_i$ we have
$\|(Y\, W \wedge X\, W) \longleftrightarrow (Z\, W \wedge X\, W)\|^{H^M, g[\bar{X}/X_\tau][\bar{Y}/Y_\tau][\bar{Z}/Z_\tau][s/W_i]} = T$

$\Leftrightarrow$ For all $s \in D_i$ we have $\|Y\, W \wedge X\, W\|^{H^M, g[\bar{X}/X_\tau][\bar{Y}/Y_\tau][\bar{Z}/Z_\tau][s/W_i]} = T$ iff $\|Z\, W \wedge X\, W\|^{H^M, g[\bar{X}/X_\tau][\bar{Y}/Y_\tau][\bar{Z}/Z_\tau][s/W_i]} = T$

$\Leftrightarrow$ For all $s \in D_i$ we have $s \in \bar{Y} \cap \bar{X}$ iff $s \in \bar{Z} \cap \bar{X}$

$\Leftrightarrow$ $\bar{Y} \cap \bar{X} = \bar{Z} \cap \bar{X}$

$\Rightarrow$ $Iob_{\tau\to\tau\to o}(\bar{X}, \bar{Y}) = T$ iff $Iob_{\tau\to\tau\to o}(\bar{X}, \bar{Z}) = T$ $\quad$ (by Lemma 1 (ob2))

$\Leftrightarrow$ $\|ob\, X\, Y)\|^{H^M, g[\bar{X}/X_\tau][\bar{Y}/Y_\tau][\bar{Z}/Z_\tau]} = T$ iff $\|ob\, X\, Z\|^{H^M, g[\bar{X}/X_\tau][\bar{Y}/Y_\tau][\bar{Z}/Z_\tau]} = T$

$\Leftrightarrow$ $\|ob\, X\, Y \longleftrightarrow ob\, X\, Z\|^{H^M, g[\bar{X}/X_\tau][\bar{Y}/Y_\tau][\bar{Z}/Z_\tau]} = T$

Hence, by definition of $\|.\|$, for all g, for all $\bar{X}, \bar{Y}, \bar{Z} \in D_\tau$ we have:

$\|(\forall W(\ ((Y\, W \wedge X\, W) \longleftrightarrow (Z\, W \wedge X\, W)) \rightarrow$
$(ob\, X\, Y \longleftrightarrow ob\, X\, Z))\|^{H^M, g[\bar{X}/X_\tau][\bar{Y}/Y_\tau][\bar{Z}/Z_\tau]} = T$

$\Leftrightarrow$ For all g we have $\|\forall XYZ(\forall W(\ ((Y\, W \wedge X\, W) \longleftrightarrow (Z\, W \wedge X\, W)) \rightarrow (ob\, X\, Y \longleftrightarrow ob\, X\, Z))\|^{H^M, g} = T$

$\Leftrightarrow$ $H^M \models^{\text{HOL}} OB$

OB3: Given assignment g, and $\bar{\beta} \in D_{\tau\to o}, \bar{X} \in D_\tau$ such that $\|\forall Z(\beta\, Z \rightarrow ob\, X\, Z)\|^{H^M, g[\bar{\beta}/\beta_{\tau\to o}][\bar{X}/X_\tau]} = T$ and $\|\exists Z(\beta\, Z)\|^{H^M, g[\bar{\beta}/\beta_{\tau\to o}]} = T$ and $\|\exists Y(((\lambda W \forall Z(\beta\, Z \rightarrow Z\, W))\, Y) \wedge X\, Y)\|^{H^M, g[\bar{\beta}/\beta_{\tau\to o}][\bar{X}/X_\tau]} = T$

$\Leftrightarrow$ For all $\bar{Z} \in D_\tau$ we have $\|\beta Z\|^{H^M, g[\bar{\beta}/\beta_{\tau\to o}][\bar{X}/X_\tau][\bar{Z}/Z_\tau]} = T$ implies $\|ob\, X\, Z\|^{H^M, g[\bar{\beta}/\beta_{\tau\to o}][\bar{X}/X_\tau][\bar{Z}/Z_\tau]} = T$ and there exists $\bar{Z} \in D_\tau$ such that $\|\beta Z\|^{H^M, g[\bar{\beta}/\beta_{\tau\to o}][\bar{Z}/Z_\tau]} = T$ and there exists $s \in D_i$ such that $\|(\lambda W \forall Z(\beta Z \to Z\, W))\, Y \wedge X\, Y\|^{H^M, g[\bar{\beta}/\beta_{\tau\to o}][\bar{X}/X_\tau][s/Y_i]} = T$

$\Leftrightarrow$ For all $\bar{Z} \in D_\tau$ we have $\bar{Z} \in \beta$ implies $\bar{Z} \in Iob_{\tau\to\tau\to o}(\bar{X})$ and there exists $\bar{Z} \in D_\tau$ such that $\bar{Z} \in \bar{\beta}$ and there exists $s \in D_i$ such that $s \in \cap\bar{\beta}$ and $s \in \bar{X}$ (see **Justification** *)[8]

$\Leftrightarrow$ $\bar{\beta} \subseteq Iob_{\tau\to\tau\to o}(\bar{X})$ and $\bar{\beta} \neq \emptyset$ and $(\cap\bar{\beta}) \cap \bar{X} \neq \emptyset$

$\Rightarrow$ $Iob_{\tau\to\tau\to o}(\bar{X}, (\cap\bar{\beta})) = T$ (by Lemma 1 (ob3))

$\Leftrightarrow$ $\|ob\, X\, (\lambda W \forall Z(\beta Z \to Z\, W))\|^{H^M, g[\bar{\beta}/\beta_{\tau\to o}][\bar{X}/X_\tau]} = T$

Hence by definition of $\|.\|$, for all g, all $\bar{\beta} \in D_{\tau\to o}$, all $\bar{X} \in D_\tau$ we have:

$\|((\forall Z(\beta Z \to ob\, X\, Z)) \wedge (\exists Z(\beta Z))) \to ((\exists Y(((\lambda W \forall Z(\beta Z \to Z\, W))Y) \wedge X\, Y)) \to ob\, X\, (\lambda W \forall Z(\beta Z \to Z\, W)))\|^{H^M, g[\bar{\beta}/\beta_{\tau\to o}][\bar{X}/X_\tau]} = T$

$\Leftrightarrow$ For all g, we have $\|\forall\beta\forall X(((\forall Z(\beta Z \to ob\, X\, Z)) \wedge (\exists Z(\beta Z))) \to ((\exists Y(((\lambda W \forall Z(\beta Z \to Z\, W))Y) \wedge X\, Y)) \to ob\, X\, (\lambda W \forall Z(\beta Z \to Z\, W))))\|^{H^M, g} = T$

$\Leftrightarrow$ $H^M \models^{\text{HOL}} OB3$

OB4: Given assignment g, and $\bar{X}, \bar{Y}, \bar{Z} \in D_\tau$ such that $\|\forall W(Y\, W \to X\, W) \wedge ob\, X\, Y \wedge \forall W(X\, W \to Z\, W)\|^{H^M, g[\bar{X}/X_\tau][\bar{Y}/Y_\tau][\bar{Z}/Z_\tau]} = T$

$\Leftrightarrow$ $\|\forall W(Y\, W \to X\, W)\|^{H^M, g[\bar{X}/X_\tau][\bar{Y}/Y_\tau][\bar{Z}/Z_\tau]} = T$ and
$\|ob\, X\, Y\|^{H^M, g[\bar{X}/X_\tau][\bar{Y}/Y_\tau][\bar{Z}/Z_\tau]} = T$ and
$\|\forall W(X\, W \to Z\, W)\|^{H^M, g[\bar{X}/X_\tau][\bar{Y}/Y_\tau][\bar{Z}/Z_\tau]} = T$

$\Leftrightarrow$ For all $s \in D_i$ we have ($s \in \bar{Y}$ implies $s \in \bar{X}$) and $\bar{Y} \in Iob_{\tau\to\tau\to o}(\bar{X})$ and ($s \in \bar{X}$ implies $s \in \bar{Z}$)

$\Leftrightarrow$ $\bar{Y} \subseteq \bar{X}$ and $\bar{Y} \in Iob_{\tau\to\tau\to o}(\bar{X})$ and $\bar{X} \subseteq \bar{Z}$

$\Rightarrow$ $(\bar{Z} \setminus \bar{X}) \cup \bar{Y} \in Iob_{\tau\to\tau\to o}(\bar{Z})$ (by Lemma 1 (ob4))

$\Leftrightarrow$ $\|ob\, Z\, (\lambda W((Z\, W \wedge \neg X\, W) \vee Y\, W))\|^{H^M, g[\bar{X}/X_\tau][\bar{Y}/Y_\tau][\bar{Z}/Z_\tau]} = T$ (see **Justification ****)[9]

Hence by definition of $\|.\|$ for all g, all $\bar{X}, \bar{Y}, \bar{Z} \in D_\tau$ we have:

$\|(\forall W(Y\, W \to X\, W) \wedge ob\, X\, Y \wedge \forall W(X\, W \to Z\, W)) \to ob\, Z\, (\lambda W((Z\, W \wedge \neg X\, W) \vee Y\, W))\|^{H^M, g[\bar{X}/X_\tau][\bar{Y}/Y_\tau][\bar{Z}/Z_\tau]} = T$

$\Leftrightarrow$ For all g we have $\|\forall XYZ((\forall W(Y\, W \to X\, W) \wedge ob\, X\, Y \wedge \forall W(X\, W \to Z\, W)) \to ob\, Z\, (\lambda W((Z\, W \wedge \neg X\, W) \vee Y\, W)))\|^{H^M, g} = T$

$\Leftrightarrow$ $H^M \models^{\text{HOL}} OB4$

OB5: This case is analogous to OB4.

Proof of Lemma 3

Proof The proof of the lemma is by induction on the structure of δ.
In the base case we have $\delta = p^j$ for some $p^j \in P$:

$$
\begin{array}{ll}
 & \|\lfloor p^j \rfloor S\|^{H^M, g[s/S_i]} = T \\
\Leftrightarrow & \|p^j_\tau S\|^{H^M, g[s/S_i]} = T \\
\Leftrightarrow & I p^j_\tau(s) = T \\
\Leftrightarrow & s \in V(p^j) \quad \text{(by definition of } H^M\text{)} \\
\Leftrightarrow & M, s \vDash p^j
\end{array}
$$

For proving the inductive cases we apply the induction hypothesis, which is formulated as follows: For all δ' that are structurally smaller than δ, for all assignments g and all s we have $\|\lfloor \delta' \rfloor S\|^{H^M, g[s/S_i]} = T$ if and only if $M, s \vDash \delta'$.

We consider each inductive case in turn:

$\delta = \neg\varphi$:

$$
\begin{array}{ll}
 & \|\lfloor \neg\varphi \rfloor S\|^{H^M, g[s/S_i]} = T \\
\Leftrightarrow & \|(\neg_{\tau\to\tau} \lfloor \varphi \rfloor) S\|^{H^M, g[s/S_i]} = T \\
\Leftrightarrow & \|\neg(\lfloor \varphi \rfloor S)\|^{H^M, g[s/S_i]} = T \qquad \text{(since } (\neg_{\tau\to\tau} \lfloor \varphi \rfloor) S =_{\beta\eta} \neg(\lfloor \varphi \rfloor S)\text{)} \\
\Leftrightarrow & \|\lfloor \varphi \rfloor S\|^{H^M, g[s/S_i]} = F \\
\Leftrightarrow & M, s \nvDash \varphi \quad \text{(by induction hypothesis)} \\
\Leftrightarrow & M, s \vDash \neg\varphi
\end{array}
$$

$\delta = \varphi \vee \psi$:

$$
\begin{array}{ll}
 & \|\lfloor \varphi \vee \psi \rfloor S\|^{H^M, g[s/S_i]} = T \\
\Leftrightarrow & \|(\lfloor \varphi \rfloor \vee_{\tau\to\tau\to\tau} \lfloor \psi \rfloor) S\|^{H^M, g[s/S_i]} = T \\
\Leftrightarrow & \|(\lfloor \varphi \rfloor S) \vee (\lfloor \psi \rfloor S)\|^{H^M, g[s/S_i]} = T \\
 & \qquad\qquad \text{(since } (\lfloor \varphi \rfloor \vee_{\tau\to\tau\to\tau} \lfloor \psi \rfloor) S =_{\beta\eta} ((\lfloor \varphi \rfloor S) \vee (\lfloor \psi \rfloor S))\text{)} \\
\Leftrightarrow & \|\lfloor \varphi \rfloor S\|^{H^M, g[s/S_i]} = T \text{ or } \|\lfloor \psi \rfloor S)\|^{H^M, g[s/S_i]} = T \\
\Leftrightarrow & M, s \vDash \varphi \text{ or } M, s \vDash \psi \quad \text{(by induction hypothesis)} \\
\Leftrightarrow & M, s \vDash \varphi \vee \psi
\end{array}
$$

$\delta = \Box\varphi$:

$$
\begin{array}{ll}
 & \|\lfloor \Box\varphi \rfloor S\|^{H^M, g[s/S_i]} = T \\
\Leftrightarrow & \|(\lambda X \forall Y (\lfloor \varphi \rfloor Y)) S\|^{H^M, g[s/S_i]} = T \\
\Leftrightarrow & \text{For all } a \in D_i \text{ we have } \|\lfloor \varphi \rfloor Y\|^{H^M, g[s/S_i][a/Y_i]} = T
\end{array}
$$

---

[8] **Justification *:** By definition of $\|.\|$, $\|\lambda W_i \forall Z_\tau (\beta_{\tau\to o} Z_\tau \to Z_\tau W_i)\|^{H^M, g[\bar{\beta}/\beta_{\tau\to o}][\bar{X}/X_\tau][s/Y_i]}$ is denoting the function f from D_i to D_o such that for all $d \in D_i$, $f(d) = \|\forall Z_\tau (\beta_{\tau\to o} Z_\tau \to Z_\tau W_i)\|^{H^M, g[\bar{\beta}/\beta_{\tau\to o}][\bar{X}/X_\tau][s/Y_i][d/W_i]}$. By definition of $\|.\|$, $\|\forall Z_\tau (\beta_{\tau\to o} Z_\tau \to Z_\tau W_i)\|^{H^M, g[\bar{\beta}/\beta_{\tau\to o}][\bar{X}/X_\tau][s/Y_i][d/W_i]} = T$ iff for all $\bar{Z} \in \bar{\beta}$ we have $d \in \bar{Z}$. Thus, f is the characteristic function of the set $\cap\bar{\beta}$. By the Denotatpflicht, which is obeyed in H^M, we know that $f\ (= \cap\bar{\beta}) \in D_\tau$.

[9] **Justification **:** Similar to justification *, we can convince ourselves that $\|\lambda W((Z\,W \wedge \neg X\,W) \vee Y\,W)\|^{H^M, g[\bar{X}/X_\tau][\bar{Y}/Y_\tau][\bar{Z}/Z_\tau][\bar{Z}/Z_\tau]}$ is denoting the characteristic function f of the set $(\bar{Z} \setminus \bar{X}) \cup \bar{Y}$. By the Denotatpflicht, which is obeyed in H^M, we know that $f\ (= (\bar{Z} \setminus \bar{X}) \cup \bar{Y}) \in D_\tau$.

$\Leftrightarrow$ For all $a \in D_i$ we have $\|\lfloor\varphi\rfloor Y\|^{H^M, g[a/Y_i]} = T \quad (S \notin free(\lfloor\varphi\rfloor))$
$\Leftrightarrow$ For all $a \in S$ we have $M, a \models \varphi$ (by induction hypothesis)
$\Leftrightarrow$ $M, s \models \Box\varphi$

$\delta = \Box_a\varphi$:

$\|\lfloor\Box_a\varphi\rfloor S\|^{H^M, g[s/S_i]} = T$
$\Leftrightarrow$ $\|(\lambda X \forall Y(\neg av\, X\, Y \vee \lfloor\varphi\rfloor Y))\, S\|^{H^M, g[s/S_i]} = T$
$\Leftrightarrow$ For all $a \in D_i$ we have $\|\neg av\, S\, Y \vee \lfloor\varphi\rfloor Y\|^{H^M, g[s/S_i][a/Y_i]} = T$
$\Leftrightarrow$ For all $a \in D_i$ we have $\|av\, S\, Y\|^{H^M, g[s/S][a/Y]} = F$ or $\|\lfloor\varphi\rfloor Y\|^{H^M, g[s/S_i][a/Y_i]} = T$
$\Leftrightarrow$ For all $a \in D_i$ we have $Iav_{i\to\tau}(s, a) = F$ or $\|\lfloor\varphi\rfloor Y\|^{H^M, g[a/Y_i]} = T \quad (S \notin free(\lfloor\varphi\rfloor))$
$\Leftrightarrow$ For all $a \in S$ we have $a \notin av(s)$ or $M, a \models \varphi$ (by induction hypothesis)
$\Leftrightarrow$ $M, s \models \Box_a\varphi$

$\delta = \Box_p\varphi$.

The argument is analogous to $\delta = \Box_a\varphi$.

$\delta = \bigcirc(\psi/\varphi)$:[10]

$\|\lfloor\bigcirc(\psi/\varphi)\rfloor S\|^{H^M, g[s/S_i]} = T$
$\Leftrightarrow$ $\|(\lambda X(ob\lfloor\psi\rfloor\lfloor\varphi\rfloor))S\|^{H^M, g[s/S_i]} = T$
$\Leftrightarrow$ $\|ob\lfloor\psi\rfloor\lfloor\varphi\rfloor\|^{H^M, g[s/S_i]} = T$
$\Leftrightarrow$ $Iob_{\tau\to\tau\to o}(\|\lfloor\psi\rfloor\|^{H^M, g[s/S_i]})(\|\lfloor\varphi\rfloor\|^{H^M, g[s/S_i]}) = T$
$\Leftrightarrow$ $\|\lfloor\varphi\rfloor\|^{H^M, g[s/S_i]} \in Iob_{\tau\to\tau\to o}(\|\lfloor\psi\rfloor\|^{H^M, g[s/S_i]})$
$\Leftrightarrow$ $V(\varphi) \in Iob_{\tau\to\tau\to o}(V(\psi))$ (see **Justification** ***)
$\Leftrightarrow$ $V(\varphi) \in ob(V(\psi))$
$\Leftrightarrow$ $M, s \models \bigcirc(\psi/\varphi)$

$\delta = \bigcirc_a(\varphi)$:

$\|\lfloor\bigcirc_a(\varphi)\rfloor S\|^{H^M, g[s/S_i]} = T$
$\Leftrightarrow$ $\|(\lambda X(ob\,(av\, X)\lfloor\varphi\rfloor \wedge \exists Y(av\, X\, Y \wedge \neg(\lfloor\varphi\rfloor Y)))S\|^{H^M, g[s/S_i]} = T$

---

[10] **Justification ***:** We need to show that $\|\lfloor\varphi\rfloor\|^{H^M, g[s/S_i]}$ is identified with $V(\varphi) = \{s \in S \mid M, s \models \varphi\}$ (analogous for ψ). By induction hypothesis, for all assignments g and world s, we have $\|\lfloor\varphi\rfloor S\|^{H^M, g[s/S_i]} = T$ if and only if $M, s \models \varphi$. We expand the details of this equivalence. For all assignments g and all worlds $s \in D_i$ we have

$s \in \|\lfloor\varphi\rfloor\|^{H^M, g[s/S_i]}$ (charact. functions are associated with sets)
$\Leftrightarrow$ $\|\lfloor\varphi\rfloor\|^{H^M, g[s/S_i]}(s) = T$
$\Leftrightarrow$ $\|\lfloor\varphi\rfloor\|^{H^M, g[s/S_i]}(\|S\|^{H, g[s/S_i]}) = T$
$\Leftrightarrow$ $\|\lfloor\varphi\rfloor S\|^{H^M, g[s/S_i]} = T$
$\Leftrightarrow$ $M, s \models \varphi$ (induction hypothesis)
$\Leftrightarrow$ $s \in V(\varphi)$

Hence, $s \in \|\lfloor\varphi\rfloor\|^{H^M, g[s/S_i]}$ if and only if $s \in V(\varphi)$. By extensionality we thus know that $\|\lfloor\varphi\rfloor\|^{H^M, g[s/S_i]} = V(\varphi)$. Moreover, since H^M obeys the Denotatpflicht we know that $V(\varphi) \in D_\tau$.

$\Leftrightarrow$ $\|ob\ (av\ S)\lfloor\varphi\rfloor \wedge \exists Y(av\ S\,Y \wedge \neg(\lfloor\varphi\rfloor Y))\|^{H^M,g[s/S_i]} = T$

$\Leftrightarrow$ $\|ob\ (av\ S)\lfloor\varphi\rfloor\|^{H^M,g[s/S_i]} = T$ and
$\|\exists Y(av\ S\,Y \wedge \neg(\lfloor\varphi\rfloor Y))\|^{H^M,g[s/S_i]} = T$

$\Leftrightarrow$ $\|ob\ (av\ S)\lfloor\varphi\rfloor\|^{H^M,g[s/S_i]} = T$ and
there exists $a \in D_i$ such that $\|av\ S\,Y \wedge \neg(\lfloor\varphi\rfloor Y)\|^{H^M,g[s/S_i][a/Y_i]} = T$

$\Leftrightarrow$ $Iob_{\tau\to\tau\to o}(\|av\ S\|^{H^M,g[s/S_i]})(\|\lfloor\varphi\rfloor\|^{H^M,g[s/S_i]}) = T$ and
there exists $a \in D_i$ such that
$\|av\ X\,Y\|^{H^M,g[s/S_i][a/Y_i]} = T$ and $\|\lfloor\varphi\rfloor Y\|^{H^M,g[s/S_i][a/Y_i]} = F$

$\Leftrightarrow$ $\|\lfloor\varphi\rfloor\|^{H^M,g[s/S_i]} \in Iob_{\tau\to\tau\to o}(\|av\ S\|^{H^M,g[s/S_i]})$ and
there exists $a \in D_i$ such that
$\|av\ X\,Y\|^{H^M,g[s/S_i][a/Y_i]} = T$ and $\|\lfloor\varphi\rfloor Y\|^{H^M,g[s/S_i][a/Y_i]} = F$

$\Leftrightarrow$ $V(\varphi) \in Iob_{\tau\to\tau\to o}(\|av\ S\|^{H^M,g[s/S_i]})$ and **(similar to ***)**
there exists $a \in D_i$ such that
$\|av\ X\,Y\|^{H^M,g[a/Y_i]} = T$ and $\|\lfloor\varphi\rfloor Y\|^{H^M,g[a/Y_i]} = F$

$\Leftrightarrow$ $V(\varphi) \in Iob_{\tau\to\tau\to o}(av(s))$ and **(similar to ***)**
there exists $a \in D_i$ such that
$\|av\ X\,Y\|^{H^M,g[a/Y_i]} = T$ and $\|\lfloor\varphi\rfloor Y\|^{H^M,g[a/Y_i]} = F$ $(S \notin free(\lfloor\varphi\rfloor))$

$\Leftrightarrow$ $V(\varphi) \in ob(av(s))$ and
there exists $a \in S$ such that
$a \in av(s)$ and $M, a \not\models \varphi$ (by induction hypothesis)

$\Leftrightarrow$ $V(\varphi) \in ob(av(s))$ and
there exists $a \in S$ such that $a \in av(s)$ and $a \notin V(\varphi)$

$\Leftrightarrow$ $V(\varphi) \in ob(av(s))$ and
there exists $a \in S$ such that $a \in av(s) \cap V(\neg\varphi)$

$\Leftrightarrow$ $V(\varphi) \in ob(av(s))$ and $av(s) \cap V(\neg\varphi) \neq \emptyset$

$\Leftrightarrow$ $M, s \models \bigcirc_a(\varphi)$

$\delta = \bigcirc_p(\varphi)$:

The argument is analogous to $\delta = \bigcirc_a(\varphi)$. □