Åqvist's Dyadic Deontic Logic **E** in HOL

Christoph Benzmüller
Freie Universität Berlin, Germany, and University of Luxembourg, Luxembourg
c.benzmueller@gmail.com

Ali Farjami
University of Luxembourg, Luxembourg
ali.farjami@uni.lu

Xavier Parent
University of Luxembourg, Luxembourg
xavier.parent@uni.lu

Abstract

We devise a shallow semantical embedding of Åqvist's dyadic deontic logic **E** in classical higher-order logic. This embedding is shown to be faithful, viz. sound and complete. This embedding is also encoded in Isabelle/HOL, which turns this system into a proof assistant for deontic logic reasoning. The experiments with this environment provide evidence that this logic *implementation* fruitfully enables interactive and automated reasoning at the meta-level and the object-level.

Keywords: Dyadic deontic logic; Preference semantics; Classical higher-order logic; Semantical embedding; Automated reasoning.

1 Introduction

Normative notions such as obligation and permission are the subject of deontic logic [24] and conditional obligations are addressed in so-called *dyadic deontic logic*.

This work has been supported by the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 690974 - MIREL - MIning and REasoning with Legal texts. Benzmüller has been funded by the Volkswagen Foundation under project CRAP — Consistent Rational Argumentation in Politics.

A landmark and historically important family of dyadic deontic logics has been proposed by B. Hansson [26]. These logics have been recast in the framework of possible world semantics by Åqvist [3]. They come with a preference semantics, in which a binary preference relation ranks the possible words in terms of betterness. The framework was motivated by the well-known paradoxes of *contrary-to-duty* (CTD) reasoning like Chisholm [20]'s paradox. In this paper, we focus on the class of all preference models, in which no specific properties (like reflexivity or transitivity) are required of the betterness relation. This class of models has a known axiomatic characterization, given by Åqvist's system **E** [31].

When applied as a meta-logical tool, *simple type theory* [21], aka classical Higher-Order Logic (HOL), can help to better understand semantical issues of embedded object logics. The syntax and semantics of HOL are well understood [10] and there exist automated proof tools for it; examples include Isabelle/HOL [29], LEO-II [16] and Leo-III [32].

In this paper we devise an *embedding* of **E** in HOL. This embedding utilizes the *shallow semantical embedding* approach that has been put forward by Benzmüller [8, 9] as a pragmatical solution towards universal logic reasoning. This approach uses HOL as (universal) meta-logic to specify, in a shallow way, the syntax and semantics of various object logics, in our case system **E**. The embedding has been encoded in Isabelle/HOL to enable syntactical and semantical experiments in deontic reasoning.

Benzmüller et al. [13, 12] developed an analogous shallow semantical embedding for the dyadic deontic logic proposed by Carmo and Jones [19]. A core difference concerns the notion of semantics employed in both papers, which leads to different semantical embeddings. Instead of the semantics based on preference models as employed by Hansson [26] and Åqvist [3], a neighborhood semantics is employed by Carmo and Jones [19]. Moreover, this methodology was applied for more recent deontic frameworks [14].

Deep semantical embeddings of non-classical logics have been studied in the related literature [23, 22]. The emphasis in these works typically is on interactive proofs of meta-logical properties. While meta-logical studies [11, 25] are also in reach for the methods presented here, our interest is in proof automation at object level, i.e., proof automation of Åqvist's system **E**. In other words, we are interested in practical normative reasoning applications of system **E** in which a high degree of automation at object level is required. Moreover, we are interested not only in the "propositional" system **E**, but also in quantified extensions of it. For this, we plan to accordingly adapt the achievements of previous work [15, 6]. Making deep semantical embeddings scale for quantified non-classical logics, on the contrary, seems more challenging and less promising regarding proof automation.

The article is structured as follows. Section 2 describes system **E** and Section 3

introduces HOL. The semantical embedding of **E** in HOL is then devised and studied in Section 4. This section also shows the faithfulness (viz. soundness and completeness) of the embedding. Section 5 discusses the implementation in Isabelle/HOL [29]. Section 6 concludes the paper.

2 Dyadic Deontic Logic E

The language of **E** is obtained by adding the following operators to the language of propositional logic: $\Box$ (for necessity); $\Diamond$ (for possibility); and $\bigcirc(-/-)$ (for conditional obligation) ; $P(-/-)$ (for conditional permission). $\bigcirc(\psi/\varphi)$ is read "If φ, then ψ is obligatory", and $P(\psi/\varphi)$ is read "If φ, then ψ is permitted'. The set of well-formed formulas (wffs) is defined in the straightforward way. Iteration of the modal and deontic operators is permitted, and so are "mixed" formulas, e.g., $\bigcirc(q/p) \wedge p$. We put $\top =_{df} \neg q \vee q$, for some atomic wff q, and $\bot =_{df} \neg\top$. $\Diamond$ is the dual of $\Box$, viz. $\Diamond\varphi =_{df} \neg\Box\neg\varphi$. P is also the dual of $\bigcirc$, viz. $P(\psi/\varphi) =_{df} \neg \bigcirc (\neg\psi/\varphi)$.

We recall the main difference between the Kripke relational semantics for so-called Standard Deontic Logic (SDL) [24] and the semantics for **E**. The first one uses a binary classification of worlds into good/bad (or green/red). The second one allows for gradations between these two extremes. The closer a world is to ideality, the better it is.

A preference model is a structure $M = \langle W, \succeq, V\rangle$ where:

- W is a non-empty set of possible worlds (W is called "universe");
- $\succeq \subseteq W \times W$ (intuitively, $\succeq$ is a betterness or comparative goodness relation; "$s \succeq t$" can be read as "world s is at least as good as world t");
- V is a function assigning to each atomic wff a set of worlds, i.e., $V(p) \subseteq W$ (intuitively, $V(p)$ is the set of worlds at which p is true).

No specific properties (like reflexivity or transitivity) are required of the betterness relation.

Given a preference model $M = \langle W, \succeq, V\rangle$ and a world $s \in W$, we define the satisfaction relation $M, s \vDash \varphi$ (read as "world s satisfies φ in model M") by induction on the structure of φ as described below. Intuitively, the evaluation rule for the dyadic obligation operator puts $\bigcirc(\psi/\varphi)$ true whenever all the best φ-worlds are ψ-worlds. Here best is defined in terms of optimality rather than maximality [31]. A φ-world is optimal if it is at least as good as any other φ-world. We define $V^M(\varphi) = \{s \in W \mid M, s \models \varphi\}$ and $\mathrm{opt}_{\succeq}(V^M(\varphi)) = \{s \in V^M(\varphi) \mid \forall t(t \vDash \varphi \rightarrow s \succeq t)\}$.

Whenever the model M is obvious from context, we write $V(\varphi)$ instead of $V^M(\varphi)$.

$M, s \models p$ if and only if $s \in V(p)$
$M, s \models \neg\varphi$ if and only if $M, s \not\models \varphi$ (that is, not $M, s \models \varphi$)
$M, s \models \varphi \vee \psi$ if and only if $M, s \models \varphi$ or $M, s \models \psi$
$M, s \models \Box\varphi$ if and only if $V(\varphi) = W$
$M, s \models \bigcirc(\psi/\varphi)$ if and only if $\mathrm{opt}_{\succeq}(V(\varphi)) \subseteq V(\psi)$

As usual, a formula φ is valid in a preference model $M = \langle W, \succeq, V\rangle$ (notation: $M \models \varphi$) if and only if, for all worlds $s \in W$, $M, s \models \varphi$. A formula φ is valid (notation: $\models \varphi$) if and only if it is valid in every preference model. The notions of semantic consequence and satisfiability in a model are defined as usual.

System **E** is defined by the following axioms and rules:

Axiom schemata for propositional logic	(PL)
S5-schemata for $\Box$ and $\Diamond$	(S5)
$\bigcirc(\psi_1 \to \psi_2/\varphi) \to (\bigcirc(\psi_1/\varphi) \to \bigcirc(\psi_2/\varphi))$	(COK)
$\bigcirc(\psi/\varphi) \to \Box\bigcirc(\psi/\varphi)$	(Abs)
$\Box\psi \to \bigcirc(\psi/\varphi)$	(Nec)
$\Box(\varphi_1 \leftrightarrow \varphi_2) \to (\bigcirc(\psi/\varphi_1) \leftrightarrow \bigcirc(\psi/\varphi_2))$	(Ext)
$\bigcirc(\varphi/\varphi)$	(Id)
$\bigcirc(\psi/\varphi_1 \wedge \varphi_2) \to \bigcirc(\varphi_2 \to \psi/\varphi_1)$	(Sh)
If $\vdash \varphi$ and $\vdash \varphi \to \psi$ then $\vdash \psi$	(MP)
If $\vdash \varphi$ then $\vdash \Box\varphi$	(N)

The notions of theoremhood, deducibility and consistency are defined as usual.

The following theorem tells us that system **E** is the weakest system that characterizes preference models. It also tells us that the assumptions of reflexivity and totalness of $\succeq$ do not modify the logic, in the sense that they do not add new validities (or theorems).

Theorem 1. *System* **E** *is sound and complete with respect to the class of all preference models. System* **E** *is also sound and complete with respect to the class of those in which $\succeq$ is reflexive, and with respect to the class of those in which $\succeq$ is total (for all $s, t \in W$, $s \succeq t$ or $t \succeq s$).*

Proof. See Parent [31]. □

E is first in a family of three systems. Consider the condition of limitedness. Its role is to rule out infinite chains of strictly better worlds. Formally: if $V(\varphi) \neq \emptyset$, then $\mathrm{opt}_{\succeq}(V(\varphi)) \neq \emptyset$. Such a condition boosts the logic to system **F**, obtained by supplementing **E** with D*:

$$\Diamond\varphi \rightarrow (\bigcirc(\psi/\varphi) \rightarrow P(\psi/\varphi)) \tag{D*}$$

Similarly, the additional assumption of transitivity of $\succeq$ boosts the logic to system **G**, obtained by supplementing **F** with Sp:

$$(P(\psi/\varphi) \wedge \bigcirc((\psi \rightarrow \chi)/\varphi) \rightarrow \bigcirc(\chi/(\varphi \wedge \psi)) \tag{Sp}$$

None of **F** and **G** will concern us in this paper.

3 Classical Higher-Order Logic

In this section we introduce classical higher-order logic (HOL). The presentation, which has been adapted from [7], is rather detailed in order to keep the article sufficiently self-contained.

3.1 Syntax of HOL

To define 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 $(V_\alpha)_{\alpha\in T}$.[1] 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 $s_{\alpha\rightarrow\beta}$ and t_α are HOL terms of types $\alpha \rightarrow \beta$ and α, respectively, then $(s_{\alpha\rightarrow\beta}\, t_\alpha)_\beta$, called *application*, is an HOL term of type β.

[1] For example in Section 4 we assume constant symbol r, with type $i \rightarrow i \rightarrow o$ as part of the signature.

- 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 $=_{\alpha\to\alpha\to o}\in C_{\alpha\to\alpha\to o}$, abbreviated as $=^\alpha$. The symbols $\Pi_{(\alpha\to o)\to o}$ and $=_{\alpha\to\alpha\to o}$ 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.[2] For example, we may define $s \wedge t := \neg(\neg s \vee \neg t)$, $s \to t := \neg s \vee t$, $s \longleftrightarrow t := (s \to t) \wedge (t \to s)$, $\top := (\lambda X_i X) = (\lambda X_i X)$, $\bot := \neg\top$ and $\exists X_\alpha s := \neg\forall X_\alpha \neg s$.

Each occurrence of a variable in a term is either bound by a λ or free. We use $free(s)$ to denote the set of variables with a free occurrence in s. We consider two terms to be *equal* if the terms are the same up to the names of bound variables, that is, we consider α-conversion implicitly.

Substitution of a term s_α for a variable X_α in a term t_β is denoted by $[s/X]t$. Since we consider α-conversion implicitly, we assume the bound variables of t to avoid variable capture.

Well-known operations and relations on HOL terms include *$\beta\eta$-normalization* and *$\beta\eta$-equality*, denoted by $s =_{\beta\eta} t$, *β-reduction* and *η-reduction*. A *β-redex* $(\lambda X s)t$ β-reduces to $[t/X]s$. An *η-redex* $\lambda X(sX)$, where $X \notin free(s)$, η-reduces to s. We

[2] As demonstrated by Andrews [2], 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.

write $s =_\beta t$ to mean s can be converted to t by a series of β-reductions and expansions. Similarly, $s =_{\beta\eta} t$ means s can be converted to t using both β and η.

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 literature [10].

The semantics of choice for the remainder is Henkin semantics, i.e., we work with Henkin's general models [27]. 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:[3]

- $I(\neg_{o\to o}) = not \in D_{o\to o}$ 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:

[3]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 [1]: 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.

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

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. The proof is straightforward, for instance we prove the first one. $\|(\neg_{o\to o}\, s_o)_o\|^{M,g} = T$ iff $\|\neg_{o\to o}\|^{M,g}(\|s_o\|^{M,g}) = T$ iff $not(\|s_o\|^{M,g}) = T$ iff $\|s_o\|^{M,g} = F$. □

An HOL formula s_o is *true* in a Henkin model M under assignment g if and only if $\|s_o\|^{M,g} = T$; this is also expressed by writing that $M, g \models^{HOL} s_o$. An HOL formula s_o is called *valid* in M, which is expressed by writing that $M \models^{HOL} s_o$, if

and only if $M, g \models^{HOL} s_o$ for all assignments g. Moreover, a formula s_o is called *valid*, expressed by writing that $\models^{HOL} s_o$, if and only if s_o is valid in all Henkin models M.

4 Embedding E into HOL

4.1 Semantical embedding

The formulas of **E** 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 symbol $r_{i\to\tau}$. Moreover, for each atomic propositional symbol p^j of **E**, the HOL signature must contain the corresponding constant symbol p^j_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 a formula φ of **E** into a term $\lfloor\varphi\rfloor$ of HOL of type τ. The mapping is defined recursively:

$$\begin{array}{lcl} \lfloor p^j \rfloor & = & p^j_\tau \\ \lfloor \neg\varphi \rfloor & = & \neg_{\tau\to\tau} \lfloor\varphi\rfloor \\ \lfloor \varphi \vee \psi \rfloor & = & \vee_{\tau\to\tau\to\tau} \lfloor\varphi\rfloor\lfloor\psi\rfloor \\ \lfloor \Box\varphi \rfloor & = & \Box_{\tau\to\tau} \lfloor\varphi\rfloor \\ \lfloor \bigcirc(\psi/\varphi) \rfloor & = & \bigcirc_{\tau\to\tau\to\tau} \lfloor\varphi\rfloor\lfloor\psi\rfloor \end{array}$$

$\neg_{\tau\to\tau}$, $\vee_{\tau\to\tau\to\tau}$, $\Box_{\tau\to\tau}$ and $\bigcirc_{\tau\to\tau\to\tau}$ abbreviate the following terms of HOL:

$$\begin{array}{ll} \neg_{\tau\to\tau} & = \lambda A_\tau \lambda X_i \neg(A\,X) \\ \vee_{\tau\to\tau\to\tau} & = \lambda A_\tau \lambda B_\tau \lambda X_i (A\,X \vee B\,X) \\ \Box_{\tau\to\tau} & = \lambda A_\tau \lambda X_i \forall Y_i (A\,Y) \\ \bigcirc_{\tau\to\tau\to\tau} & = \lambda A_\tau \lambda B_\tau \lambda X_i \forall W_i (\,(\lambda V_i (A\,V \wedge (\forall Y_i (A\,Y \to r_{i\to\tau} V\,Y))))\,W \to B\,W)^4 \end{array}$$

Analyzing the truth of formula φ, represented by the HOL term $\lfloor\varphi\rfloor$, in a particular world w, represented by the term w_i, corresponds to evaluating the application $(\lfloor\varphi\rfloor\,w_i)$. In line with previous work [15], we define $vld_{\tau\to o} = \lambda A_\tau \forall S_i (A\,S)$. With this definition, validity of a formula s in **E** corresponds to the validity of the formula $(vld\,\lfloor\varphi\rfloor)$ in HOL, and vice versa.

[4] If $\mathrm{opt}_{\succeq}(A)$ is taken as a abbreviation for $\lambda V_i (AV \wedge (\forall Y_i (AY \to r_{i\to\tau} V\,Y)))$, then this can be simplified to $\bigcirc_{\tau\to\tau\to\tau} = \lambda A_\tau \lambda B_\tau \lambda X_i (\mathrm{opt}_{\succeq}(A) \subseteq B)$.

4.2 Soundness and completeness

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

Definition 1 (Preference model ⇒ Henkin model)**.** *Let $M = \langle W, \succeq, V\rangle$ be a preference model. Let $p^1, ..., p^m$ for $m \geq 1$ be atomic propositional symbols and $\lfloor p^j \rfloor = p^j_\tau$ for $j = 1, ..., m$. A Henkin model $H^M = \langle \{D_\alpha\}_{\alpha\in T}, I\rangle$ for M is defined as follows: D_i is chosen as the set of possible worlds W and 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, in particular, it is required that D_τ and $D_{i\to\tau}$ contain the elements Ip^j_τ and $Ir_{i\to\tau}$. Interpretation I is constructed as follows:*

1. *For $1 \leq i \leq m$, $Ip^j_\tau \in D_\tau$ is chosen such that $Ip^j_\tau(s) = T$ iff $s \in V(p^j)$ in M.*
2. *$Ir_{i\to\tau} \in D_{i\to\tau}$ is chosen such that $Ir_{i\to\tau}(s, u) = T$ iff $s \succeq u$ in M.*

Since we assume that there are no other symbols (besides the r, the p^j and the primitive logical connectives) 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 a total evaluation function.

Lemma 1. *Let H^M be a Henkin model for a preference model M. For all formulas δ of* **E***, all 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. □

Lemma 2 (Henkin model ⇒ Preference model)**.** *For every Henkin model $H = \langle \{D_\alpha\}_{\alpha\in T}, I\rangle$ there exists a corresponding preference model M. Corresponding here means that for all formulas δ of* **E** *and for all assignments g and worlds s,*

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

Proof. Suppose that $H = \langle \{D_\alpha\}_{\alpha\in T}, I\rangle$ is a Henkin model. Without loss of generality, we can assume that the domains of H are denumerable [27]. We construct the corresponding preference model M as follows:

- $W = D_i$.
- $s \succeq u$ for $s, u \in W$ iff $Ir_{i\to\tau}(s, u) = T$.

- $s \in V(p^j_\tau)$ iff $Ip^j_\tau(s) = T$ for all p^j.

Moreover, the above construction ensures that H is a Henkin model for M. Hence, Lemma 1 applies. This ensures that for all formulas δ of **E**, for all assignments g and all worlds s we have $\|\lfloor\delta\rfloor S_i\|^{H,g[s/S_i]} = T$ if and only if $M, s \vDash \delta$. □

Theorem 2 (Soundness and completeness of the embedding)**.**

$$\models \varphi \textit{ if and only if } \models^{HOL} \mathit{vld}\lfloor\varphi\rfloor$$

Proof. (Soundness, $\leftarrow$) The proof is by contraposition. Assume $\not\models \varphi$, i.e, there is a preference model $M = \langle W, \succeq, V\rangle$, and a world $s \in W$, such that $M, s \not\models \varphi$. By Lemma 1 for an arbitrary assignment g it holds that $\|\lfloor\varphi\rfloor\, S_i\|^{H^M,g[s/S_i]} = F$ in Henkin model $H^M = \langle\{D_\alpha\}_{\alpha\in T}, I\rangle$. Thus, by definition of $\|.\|$, it holds that $\|\forall S_i(\lfloor\varphi\rfloor\, S_i)\|^{H^M,g} = \|\mathit{vld}\lfloor\varphi\rfloor\|^{H^M,g} = F$. Hence, $H^M \not\models^{HOL} \mathit{vld}\lfloor\varphi\rfloor$. By definition $\not\models^{HOL} \mathit{vld}\lfloor\varphi\rfloor$.

(Completeness, $\rightarrow$) The proof is again by contraposition. Assume $\not\models^{HOL} \mathit{vld}\lfloor\varphi\rfloor$, i.e., there is a Henkin model $H = \langle\{D_\alpha\}_{\alpha\in T}, I\rangle$ and an assignment g such that $\|\mathit{vld}\lfloor\varphi\rfloor\|^{H,g} = F$. By Lemma 2, there is a preference model M such that $M \not\vDash \varphi$. Hence, $\not\models \varphi$. □

Remark: In contrast to a deep logical embedding, in which the syntactical structure and the semantics of logic L would be formalized in full detail (using e.g., structural induction and recursion), only the core differences in the semantics of both system **E** and meta-logic HOL have been explicitly encoded in our shallow semantical embedding. In a certain sense we have thus shown, that system **E** can, in fact, be identified and handled as a natural fragment of HOL.

5 Implementation in Isabelle/HOL

5.1 Implementation

The semantical embedding as devised in Section 4 has been implemented in the higher-order proof assistant Isabelle/HOL [29]. Figure 1 displays the respective encoding. Some explanations are in order:

- On line 3, the type i for possible words is introduced
- On line 4, the type τ for formulas is introduced
- On line 5, a designated constant for the actual world (aw) is introduced
- On line 6, the constant r is introduced. r encodes the preference relation $\succeq$
- Lines 8–14 define the Boolean connectives in the usual way

```
1  theory DDLE imports Main
2  begin
3  typedecl i (* type for possible worlds *)
4  type_synonym τ = "(i⇒bool)" (* type for propositions *)
5  consts aw::i (* actual world *)
6  consts r :: "i⇒τ"  (infixr "r" 70) (* comparative goodness relation *)
7
8  definition ddetop          :: "τ" ("⊤")              where "⊤  ≡ λw. True"
9  definition ddebot          :: "τ" ("⊥")              where "⊥  ≡ λw. False"
10 definition ddeneg          :: "τ⇒τ" ("¬_"[52]53)     where "¬φ ≡ λw. ¬φ(w)"
11 definition ddeand          :: "τ⇒τ⇒τ" (infixr"∧"51) where "φ∧ψ ≡ λw. φ(w)∧ψ(w)"
12 definition ddeor           :: "τ⇒τ⇒τ" (infixr"∨"50) where "φ∨ψ ≡ λw. φ(w)∨ψ(w)"
13 definition ddeimp          :: "τ⇒τ⇒τ" (infixr"→"49) where "φ→ψ ≡ λw. φ(w)⟶ψ(w)"
14 definition ddeequivt       :: "τ⇒τ⇒τ" (infixr"↔"48) where "φ↔ψ ≡ λw. φ(w)⟷ψ(w)"
15
16 definition ddebox  :: "τ⇒τ" ("□") where "□ ≡ λφ w.  ∀v. φ(v)"
17 definition ddediomond  :: "τ⇒τ" ("◇") where "◇ ≡ λφ w.  ∃v. φ(v)"
18
19 definition ddeopt :: "τ⇒τ" ("opt<_>") (* obligation/permission operators *)
20   where "opt<φ> ≡ (λv. ( (φ)(v) ∧ (∀x. ((φ)(x)  ⟶  v r x)) ) )"
21 abbreviation(input) msubset :: "τ⇒τ⇒bool" (infix "⊆" 53)
22   where "φ ⊆ ψ ≡ ∀x. φ x ⟶ ψ x"
23 definition ddecond :: "τ⇒τ⇒τ" ("○<_|_>")
24   where "○<ψ|φ> ≡ λw. opt<φ> ⊆ ψ"
25 definition ddeperm :: "τ⇒τ⇒τ" ("P<_|_>")
26   where "P<ψ|φ> ≡ ¬○<¬ψ|φ>"
27
28 definition ddevalid :: "τ⇒bool" ("⌊_⌋"[8]109) (* global validity *)
29   where "⌊p⌋ ≡ ∀w. p w"
30 definition ddeactual :: "τ⇒bool" ("⌊_⌋ₗ"[7]105) (* local validity *)
31   where "⌊p⌋ₗ ≡ p(aw)"
32
33 lemma True nitpick [satisfy,user_axioms,show_all,expect=genuine] oops (* consistency check *)
34 end
```

Figure 1: Shallow semantical embedding of system **E** in Isabelle/HOL

- Lines 16 and 17 introduce the alethic operators $\Box$ and $\Diamond$
- The dyadic deontic operators are handled in lines 19–26. Lines 19–20 define the notion of optimal φ-world, and lines 23–26 define the dyadic operators using this notion.
- Lines 28–31 introduce the notion of global validity (i.e, truth in all worlds) and local validity (truth at the actual world).

A sample query is run on line 33. The model finder Nitpick [17] confirms the consistency of the definitions.

In the remainder of this section, we illustrate how the implementation in Isabelle/HOL can be used.

5.2 CTD scenarios

In this section we apply the framework to one of the benchmark problems of deontic logic, the problem of CTD reasoning. We give two examples of CTD scenarios discussed in the deontic logic literature: Chisholm's scenario [20]; Reykjavic's scenario [5].

Chisholm's scenario. The scenario involves the following four sentences:

1. It ought to be that a certain man goes (to the assistance of his neighbours);
2. It ought to be that if he goes he tells them he is coming;
3. If he does not go, he ought not to tell them he is coming;
4. He does not go.

We briefly recall the problem raised by CTDs in SDL. (For more on CTDs the reader may wish to consult [28].) When representing a conditional obligation sentence $\bigcirc(\psi/\varphi)$ in SDL, one separates the contribution of *if ... then* and that of *ought*. The *ought* operator can then take either wide scope ("It ought to be the case that, if φ, then ψ") or narrow scope ("If φ, then it ought to be the case that ψ"). There are thus different possible formalisations of the scenario depending on the choice being made. It turns out that none rendering is satisfactory. The formalisation of these sentences is either inconsistent or the sentences are logically dependent. The Chisholm set is therefore called a paradox.

System **E** is known to provide a solution to Chisholm's paradox: the formalisation of 1–4 is consistent, and each sentence remains logically independent one from the others. These two facts are confirmed by our implementation. This is documented further by Figure 2. On line 1, the theory embedding **E** in Isabelle/HOL (as described in Figure 1) is loaded. On lines 11–14, the Chisholm scenario is encoded. On line 17, a consistency check query is run. Nitpick confirms consistency of 1–4, and outputs the Henkin model described in Figure 3. One can easily read off the preference model this Henkin model encodes. We have

- $W = \{i_1, i_2, i_3, i_4\}$
- $\succeq = \{(i_3, i_1), (i_3, i_2)\}$
- $V(go) = \emptyset$ and $V(tell) = \{i_4\}$.

On the one hand, $i_4 \models \neg go$. On the other hand, each obligation in Chisholm's set (ax1, ax2 and ax3) is vacuously true, because the set of best antecedent-worlds is empty:

$$\mathrm{opt}_{\succeq}(V(\top)) = \mathrm{opt}_{\succeq}(V(\neg go)) = \mathrm{opt}_{\succeq}(V(go)) = \emptyset$$

The above model does the job, but it is not a very interesting one. One can enforce some aspects of the outputted model, by putting a number of suitable constraints on this one, like those shown on lines 19 and 20. First, all the possible

```
1  theory Chisholm_Scenario imports DDLE
2  begin
3  (* Defining some parameters for Nitpick *)
4  nitpick_params [user_axioms,show_all,expect=genuine,format=2]
5
6  (* Constants for the Chisholm scenario *)
7  consts go :: "τ"  tell :: "τ"
8
9  context (*Chisholm scenario is formalized *)
10 assumes
11   ax1: "⌊○<go|⊤>⌋" and (* It ought to be that a certain man goes. *)
12   ax2: "⌊○<tell|go>⌋" and (* It ought to be that if he goes he tells them he is coming. *)
13   ax3: "⌊○<¬tell|¬go>⌋" and (* If he does not go, he ought not to tell them he is coming. *)
14   ax4: "⌊¬go⌋ι" (* He does not go. *)
15
16 begin (* Consistency is confirmed by Nitpick *)
17 lemma True nitpick [satisfy,card=4] oops
18 lemma assumes
19       "⌊◇(go ∧ tell)⌋" and "⌊◇(go ∧ ¬tell)⌋" and "⌊◇(¬go ∧ tell)⌋" and "⌊◇(¬go ∧ ¬tell)⌋"
20       and limitedness: "(∀φ. (∃x. (φ)x) ⟶ (∃x. opt<φ>x))"
21       shows True nitpick [satisfy,card=4]  oops
22 end
23
24 (* Independence confirmed by Nitpick; countermodels are produced *)
25 lemma assumes "⌊○<go|⊤> ∧ ○<tell|go> ∧ ○<¬tell|¬go>⌋ι" shows "⌊¬go⌋ι" nitpick oops
26 lemma assumes "⌊○<tell|go> ∧ ○<¬tell|¬go> ∧ ¬go⌋ι" shows "⌊○<go|⊤>⌋ι" nitpick oops
27 lemma assumes "⌊○<go|⊤> ∧ ○<¬tell|¬go> ∧ ¬go⌋ι" shows "⌊○<tell|go>⌋ι" nitpick oops
28 lemma assumes "⌊○<go|⊤> ∧ ○<tell|go>∧ ¬go⌋ι" shows "⌊○<¬tell|¬go>⌋ι" nitpick oops
29 end
```

Figure 2: Chisholm's paradox in Isabelle/HOL

```
Nitpicking formula...
Nitpick found a model for card i = 4:

  Constants:
    go = (λx. _)(i₁ := False, i₂ := False, i₃ := False, i₄ := False)
    tell = (λx. _)(i₁ := False, i₂ := False, i₃ := False, i₄ := True)
    aw = i₄
    (r) =
      (λx. _)
      ((i₁, i₁) := False, (i₁, i₂) := False, (i₁, i₃) := False, (i₁, i₄) := False, (i₂, i₁) := False,
       (i₂, i₂) := False, (i₂, i₃) := False, (i₂, i₄) := False, (i₃, i₁) := True, (i₃, i₂) := True,
       (i₃, i₃) := False, (i₃, i₄) := False, (i₄, i₁) := False, (i₄, i₂) := False, (i₄, i₃) := False,
       (i₄, i₄) := False)
```

Figure 3: Henkin model for the Chisholm scenario

truth assignments for the relevant propositional letters must be considered (line 19). Second, the condition of limitedness (which boots us to system **F**) must be verified (line 20). The combination of these two constraints has the effect of preventing the obligations ax1, ax2 and ax3 from being vacuously true.

Lines 25 to 28 in Figure 2 confirm that the representation of the scenario in **E** meets the requirement of independence. This is confirmed by showing that no sentence follows logically for the other three.

Reykjavic's scenario. It consists of the following five sentences:

1. You should not tell the secret to Reagan;
2. You should not tell the secret to Gorbachev;
3. You should tell Reagan if you tell Gorbachev;
4. You should tell Gorbachev if you tell Reagan;
5. You told the secret to Gorbachev.

```
1  theory Reykjavic_Scenario imports DDLE
2  begin
3  consts (* We introduce special constants *)
4  tell_Reagan:: "τ"  tell_Gorbachev:: "τ"
5  context (*Reykjavic scenario*)
6  assumes
7    ax1:"⌊○<¬ tell_Reagan|⊤>⌋" and (* You should not tell the secret to Reagan. *)
8    ax2:"⌊○<¬tell_Gorbachev|⊤>⌋" and (* You should not tell the secret to Gorbachev. *)
9    ax3:"⌊○<tell_Reagan|tell_Gorbachev>⌋" and (* You should tell Reagan if you tell Gorbachev. *)
10   ax4:"⌊○<tell_Gorbachev|tell_Reagan>⌋" and (* You should tell Gorbachev if you tell Reagan. *)
11   ax5:"⌊tell_Gorbachev⌋₁" (* You told the secret to Gorbachev. *)
12
13 begin
14 lemma True  nitpick [satisfy,user_axioms,show_all,expect=genuine,card=2,format=2] oops
15 end  (* Consistency is confirmed by Nitpick *)
16 end
```

Figure 4: The Reykjavic scenario in system **E**

On line 14 in Figure 4 Nitpick confirms that the set of formulas ax1–ax5 (=the representation of 1–5 in **E**) is consistent.

5.3 Automatic verification of validities

Automatic verification of valid formulas is also possible. In Figure 5 Isabelle/HOL confirms the validity of each and every axiom and primitive rule of **E** by using the Sledgehammer tool [18] that gives access to automatic theorem provers (ATPs). Figure 6 gives the example of four "reduction" laws identified by Belanyek et al. [4]. They use these reduction laws to establish a more general result concerning iterated modalities in **G**, to the effect that any formula containing nested modal operators is equivalent to some formula with no nesting. The reduction laws are:

$$\bigcirc(\varphi|(\pi \vee (\chi \wedge \bigcirc(\gamma|\eta)))) \leftrightarrow ((\bigcirc(\gamma|\eta) \wedge \bigcirc(\varphi|(\pi \vee \chi))) \vee (\neg \bigcirc(\gamma|\eta) \wedge \bigcirc(\varphi|\pi)))$$
$$\bigcirc(\varphi|(\pi \vee (\chi \wedge \neg \bigcirc(\gamma|\eta)))) \leftrightarrow ((\neg \bigcirc(\gamma|\eta) \wedge \bigcirc(\varphi|(\pi \vee \chi))) \vee (\bigcirc(\gamma|\eta) \wedge \bigcirc(\varphi|\pi)))$$
$$\bigcirc(\pi \vee (\chi \wedge \bigcirc(\gamma|\eta))|\psi) \leftrightarrow ((\bigcirc(\gamma|\eta) \wedge \bigcirc(\pi \vee \chi|\psi)) \vee (\neg \bigcirc(\gamma|\eta) \wedge \bigcirc(\pi|\psi)))$$
$$\bigcirc(\pi \vee (\chi \wedge \neg \bigcirc(\gamma|\eta))|\psi) \leftrightarrow ((\neg \bigcirc(\gamma|\eta) \wedge \bigcirc(\pi \vee \chi|\psi)) \vee (\bigcirc(\gamma|\eta) \wedge \bigcirc(\pi|\psi)))$$

```
theory Axioms imports DDLE
begin
lemma COK:"⌊○<(ψ1→ψ2)|φ> → (○<ψ1|φ> → ○<ψ2|φ>)⌋" sledgehammer
  by (simp add: ddecond_def ddeimp_def ddevalid_def)

lemma Abs:"⌊○<ψ|φ> → □○<ψ|φ>⌋" sledgehammer
  by (simp add: ddebox_def ddecond_def ddeimp_def ddevalid_def)

lemma Nec:"⌊□ψ → ○<ψ|φ>⌋" sledgehammer
  by (simp add: ddebox_def ddecond_def ddeimp_def ddevalid_def)

lemma Ext:"⌊□(φ1↔φ2) → (○<ψ|φ1> ↔ ○<ψ|φ2>)⌋" unfolding Defs sledgehammer
  by (simp add: ddecond_def ddeopt_def)

lemma Id:"⌊○<φ|φ>⌋" sledgehammer
  by (simp add: ddecond_def ddeopt_def ddevalid_def)

lemma Sh:"⌊○<ψ|φ1∧φ2> → ○<(φ2→ψ)|φ1>⌋" sledgehammer
  by (simp add: ddeand_def ddecond_def ddeimp_def ddeopt_def ddevalid_def)

lemma MP:"(⌊φ⌋∧⌊φ→ψ⌋)⟹⌊ψ⌋" unfolding Defs sledgehammer by simp

lemma N:"⌊φ⌋⟹⌊□φ⌋" unfolding Defs sledgehammer by simp
end
```

Figure 5: Verifying the validity of the axioms and rules of system **E**

On lines 3-13 in Figure 6 , Isabelle/HOL confirms that the proofs of these equivalences carry over from **G** to **E**. However, the more general result concerning iterated

```
theory Reduction_laws  imports DDLE
begin
lemma "⌊○<φ|(π∨(χ∧○<γ|η>))> ↔ ((○<γ|η>∧○<φ|(π∨χ)>)∨(¬○<γ|η>∧○<φ|π>))⌋"
  unfolding Defs sledgehammer by (smt ddecond_def ddeopt_def)

lemma "⌊○<φ|(π∨(χ∧¬○<γ|η>))> ↔ ((¬○<γ|η>∧○<φ|(π∨χ)>)∨(○<γ|η>∧○<φ|π>))⌋"
  unfolding Defs sledgehammer by (smt ddecond_def ddeopt_def)

lemma "⌊○<(π∨(χ∧○<γ|η>))|ψ> ↔ ((○<γ|η>∧○<(π∨χ)|ψ>)∨(¬○<γ|η>∧○<π|ψ>))⌋"
  unfolding Defs sledgehammer using ddecond_def by auto

lemma "⌊○<(π∨(χ∧¬○<γ|η>))|ψ> ↔ ((¬○<γ|η>∧○<(π∨χ)|ψ>)∨(○<γ|η>∧○<π|ψ>))⌋"
  unfolding Defs sledgehammer using ddecond_def by auto

lemma "⌊□φ ↔ ○<⊥|¬φ>⌋" nitpick [satisfy,user_axioms,show_all,expect=genuine] oops
end
```

Figure 6: Reduction laws in system **E**

modalities does not. To establish that one, the authors appeal to the fact that in **G**, $\Box$ is definable in terms of $\bigcirc(-/-)$: $\Box\varphi \leftrightarrow \bigcirc(\bot/\neg\varphi)$. Nitpick confirms that this equivalence is falsifiable in the class of all preference models (line 15).

5.4 Correspondence theory

The aim of correspondence theory is to establish connections between properties of Kripke frames and the formulas in modal logic that are true in all Kripke frames with these properties. Figure 7 shows some first experimentations in correspondence theory. Lines 8–9 tell us that limitedness is equivalent with (and thus corresponds to) D*. Lines 11–13 tell us that limitedness and transitivity are conjointly enough to get both D* and Sp. However, on lines 15–16, Isabelle/HOL fails to show that they are necessary conditions too. The problem is with the proof of the property of transitivity (lines 23–24). The good news is: we do not get a counter-model to the implication (calls for countermodel search with nitpick are not displayed here).

```
1  theory  Correspondence_theory  imports DDLE
2  begin
3  abbreviation limitedness  where "limitedness ≡ (∀φ. (∃x. (φ)x) ⟶ (∃x. opt<φ>x))"
4  abbreviation Dstar_valid  where "Dstar_valid ≡ (∀φ ψ. ⌊◇φ → (○<ψ|φ>  → ¬○<¬ψ|φ>)⌋)"
5  abbreviation transitivity where "transitivity ≡ (∀x y z. (x r y ∧ y r z) ⟶ x r z)"
6  abbreviation Sp_valid     where "Sp_valid ≡ (∀φ ψ χ. ⌊(¬○<¬ψ|φ> ∧ ○<ψ→χ|φ>) → ○<χ|φ∧ψ>⌋)"
7
8  lemma "limitedness ⟷ Dstar_valid"
9   unfolding ddecond_def ddediomond_def ddeimp_def ddeneg_def ddevalid_def by auto
10
11 lemma "(limitedness ∧ transitivity) ⟶ (Sp_valid ∧ Dstar_valid)"
12  unfolding ddecond_def ddediomond_def ddeimp_def ddeneg_def ddeand_def ddevalid_def ddeopt_def
13  sledgehammer by smt (*This direction is provable*)
14
15 lemma "(Sp_valid ∧ Dstar_valid) ⟶ (limitedness ∧ transitivity)"
16  unfolding ddecond_def ddediomond_def ddeimp_def ddeneg_def ddeand_def ddevalid_def ddeopt_def oops
17  (*This direction unfortunately not yet, but we also do not get a countermodel*)
18
19 lemma "(Sp_valid ∧ Dstar_valid) ⟶ limitedness"
20  unfolding ddecond_def ddediomond_def ddeimp_def ddeneg_def ddeand_def ddevalid_def ddeopt_def
21  sledgehammer by auto (*Splitting the conjunction, limitednedd is easy for the ATPs*)
22
23 lemma "(Sp_valid ∧ Dstar_valid) ⟶ transitivity"
24  unfolding ddecond_def ddediomond_def ddeimp_def ddeneg_def ddeand_def ddevalid_def oops
25  (*Splitting the conjunction, transitivity is too hard for the ATPs*)
26  (*This direction unfortunately not yet, but we also do not get a countermodel*)
27 end
```

Figure 7: Experiments in correspondence theory

6 Conclusion

A shallow semantical embedding of Åqvist's dyadic deontic logic **E** in classical higher-order logic has been presented and shown to be faithful (sound and complete). The work presented here and in Benzmüller et al. [13] provides the theoretical foundation for the implementation and automation of dyadic deontic logic within existing theorem provers and proof assistants for HOL. We do not define new logics. Instead, we provide an empirical infrastructure for assessing practical aspects of ambitious, state-of-the-art deontic logics; this has not been done before.

We end this paper by listing a number of topics for future research. First, it would be worthwhile to study the shallow semantical embedding of the stronger systems **F** and **G** in HOL. Second, it would be interesting to look at the three systems from the point of view of a semantics defining best in terms of maximality rather than optimality [30, 31]. Third, we could employ our implementation to systematically inspect and verify some meta-logical properties of these systems within Isabelle/HOL. Fourth, it would be interesting to study the quantified extensions of these systems. Previous work has focused on monadic modal logic and conditional logic [6, 7, 15]. Last, but not least, experiments could investigate whether the provided implementation already supports non-trivial applications in practical normative reasoning, or whether further improvements are required.

Acknowledgements

We thank an anonymous reviewer for valuable comments.

References

[1] Andrews, P.B.: General models and extensionality. Journal of Symbolic Logic **37**(2), 395–397 (1972)

[2] Andrews, P.B.: Church's type theory. In: Zalta, E.N. editor, The Stanford Encyclopedia of Philosophy. Metaphysics Research Lab, Stanford University, spring 2014 edition (2014)

[3] Åqvist, L.: Deontic logic. In: Handbook of philosophical logic, pp. 147–264, Springer, Dordrecht (2002)

[4] Belanyek, A., Grossi, D., van der Hoek, W.: A note on nesting in dyadic deontic logic. arXiv preprint, arXiv:1710.03481 (2017)

[5] Belzer., M.: A logic of deliberation. In Proceedings of the Fifth National Conference on Artificial Intelligence, pp. 38–43 (1986)

[6] Benzmüller, C.: Automating quantified conditional logics in HOL. In: Rossi, F. (eds.) 23rd International Joint Conference on Artificial Intelligence, IJCAI-13, Beijing, China, pp. 746–753, AAAI Press (2013)

[7] Benzmüller, C.: Cut-elimination for quantified conditional logic. Journal of Philosophical Logic, **46**(3), 333–353 (2017)

[8] Benzmüller, C.: Universal (meta-)logical reasoning: Recent successes. Science of Computer Programming, **172**, 48–62 (2019)

[9] Benzmüller, C.: Universal (meta-)logical reasoning: The Wise Men Puzzle (Isabelle/HOL dataset). Data in Brief, **24**, no. 103774 (2019)

[10] Benzmüller, C., Brown, C., Kohlhase, M.: Higher-order semantics and extensionality. Journal of Symbolic Logic, **69**(4), 1027–1088 (2004)

[11] Benzmüller, C., Claus, M., Sultana, N.: Systematic verification of the modal logic cube in Isabelle/HOL. In: Kaliszyk, C., Paskevich, A. (eds.) PxTP 2015, Berlin, Germany, EPTCS, vol. 186, pp. 24–41 (2015)

[12] Benzmüller, C., Farjami, A., Parent., X.: Faithful semantical embedding of a dyadic deontic logic in HOL. arXiv preprint, arXiv:1802.08454 (2018)

[13] Benzmüller, C., Farjami, A., Parent., X.: A dyadic deontic logic in HOL. In: Broersen, J., Condoravdi, C., Nair, S., Pigozzi, G. (eds.) Deontic Logic and Normative Systems — 14th International Conference, DEON 2018, Utrecht, The Netherlands, 3-6 July, 2018, pp. 33–50, College Publications, UK (2018). (John-Jules Meyer Best Paper Award).

[14] Benzmüller, C., Farjami, A., Meder, P., Parent., X.: I/O logic in HOL. Journal of Applied Logics – IfCoLoG Journal of Logics and their Applications (2019). (To appear)

[15] Benzmüller, C., Paulson, L.C.: Quantified multimodal logics in simple type theory. Logica Universalis (Special Issue on Multimodal Logics), **7**(1), 7–20 (2013)

[16] Benzmüller, C., Sultana, N., Paulson, L. C., Theiß, F.: The higher-order prover LEO-II. Journal of Automated Reasoning, **55**(4), 389–404 (2015)

[17] 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.) International Conference on Interactive Theorem Proving 2010, LNCS, vol. 6172, pp. 131–146, Springer (2010)

[18] Blanchette, J. C., Paulson, L. C.: Hammering away - A user's guide to Sledgehammer for Isabelle/HOL (2017)

[19] Carmo, J. M. C. L. M., Jones, A. J. I.: Completeness and decidability results for a logic of contrary-to-duty conditionals. Journal of Logic and Computation **23**(3), 585–626 (2013)

[20] Chisholm, R. M.: Contrary-to-duty imperatives and deontic logic. Analysis, **24**(2), 33–36 (1963)

[21] Church, A.: A formulation of the simple theory of types. Journal of Symbolic Logic, **5**(2), 56–68 (1940)

[22] Doczkal, C., Bard, J.: Completeness and decidability of converse PDL in the constructive type theory of Coq. In: Andronick, J., Felty, A. P. (eds.) International Conference on Certified Programs and Proofs, CPP 2018, Los Angeles, USA, Proceedings of the 7th ACM SIGPLAN, pp. 42–52, ACM, New York, USA (2018)

[23] Doczkal, C., Smolka., G.: Completeness and decidability results for CTL in constructive

type theory. Journal of Automated Reasoning, **56**(32), 343–365 (2016)

[24] Gabbay, D., Horty, J., Parent, X., van der Meyden, R., van der Torre, L.: Handbook of Deontic Logic and Normative Systems. Volume 1. College Publications, UK (2013)

[25] Kirchner, D., Benzmüller, C., Zalta, E.: Mechanizing principia logico-metaphysica in functional type theory. CoRR `https://arxiv.org/abs/1711.06542` (2017)

[26] Hansson, B.: An analysis of some deontic logics. Nous, 373–398 (1969)

[27] Henkin, L.: Completeness in the theory of types. Journal of Symbolic Logic, **5**(2), 81–91 (1950)

[28] Hilpinen, R., McNamara, P.: Deontic logic. In Gabbay, D., Horty, J., Parent, X., van der Meyden, R., van der Torre, L. (eds.) *Handbook of Deontic logic and Normative Systems*, chapter 1, pp. 3–136. College Publications, UK (2013)

[29] Nipkow, T., Paulson, L.C., Wenzel., M.: Isabelle/HOL — A proof assistant for higher-order logic. volume 2283 of Lecture Notes in Computer Science, Springer (2002)

[30] Parent, X.: Maximality vs optimality in dyadic deontic logic - Completeness results for systems in Hansson's tradition. Journal of Philosophical Logic, **43**(6), 1101–1128 (2014)

[31] Parent, X.: Completeness of Åqvist's systems E and F. The Review of Symbolic Logic, **8**(1), 164–177 (2015)

[32] Steen, A., Benzmüller, C.: The higher-order prover Leo-III. In: Galmiche, D., Schulz, S., Sebastiani, R. (eds.) Automated Reasoning. IJCAR 2018, LNCS, vol. 10900, pp. 108-116, Springer (2018)

Appendix

Proof for Lemma 1

In the proof we implicitly employ curring and uncurring, and we associate sets with their characteristic functions. Throughout the proof whenever possible we omit types in order to avoid making the notation too cumbersome. The proof of Lemma 1 is by induction on the structure of δ. We start with the case where δ is p^j. We have

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

In the inductive cases we make use of the following **induction hypothesis**: *For sentences δ' structurally smaller than δ we have: For all assignments g and states s, $\|\lfloor \delta' \rfloor S\|^{H^M, g[s/S_i]} = T$ if and only if $M, s \vDash \delta'$.*

We consider each inductive case in turn:

(a) $\delta = \varphi \vee \psi$. In this case:

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

(b) $\delta = \neg\varphi$. In this case:

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

(c) $\delta = \Box\varphi$. We have the following chain of equivalences:

$$
\begin{array}{ll}
 & \|\lfloor \Box\varphi \rfloor S\|^{H^M, g[s/S_i]} = T \\
\Leftrightarrow & \|(\lambda X \forall Y (\lfloor \varphi \rfloor Y)) S\|^{H^M, g[s/S_i]} = T \\
\Leftrightarrow & \|\forall Y (\lfloor \varphi \rfloor Y)\|^{H^M, g[s/S_i]} = T
\end{array}
$$

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

(d) $\delta = \bigcirc(\psi/\varphi)$. We have the following chain of equivalences:
$\|\lfloor\bigcirc(\psi/\varphi)\rfloor S\|^{H^M,g[s/S_i]} = T$
$\Leftrightarrow$ $\|(\lambda X \forall W(\,(\lambda V(\lfloor\varphi\rfloor\, V \wedge (\forall Y(\lfloor\varphi\rfloor\, Y \to r\, V\, Y))))\, W \to \lfloor\psi\rfloor\, W))S\|^{H^M,g[s/S_i]} = T$
$\Leftrightarrow$ $\|\forall W(\,(\lambda V(\lfloor\varphi\rfloor\, V \wedge (\forall Y(\lfloor\varphi\rfloor\, Y \to r\, V\, Y))))\, W \to \lfloor\psi\rfloor\, W)\|^{H^M,g[s/S_i]} = T$
$\Leftrightarrow$ For all $u \in D_i$ we have:
$\|(\lambda V(\lfloor\varphi\rfloor\, V \wedge (\forall Y(\lfloor\varphi\rfloor\, Y \to r\, V\, Y))))\, W \to \lfloor\psi\rfloor\, W\|^{H^M,g[s/S_i][u/W_i]} = T$
$\Leftrightarrow$ For all $u \in D_i$ we have:
If $\|(\lambda V(\lfloor\varphi\rfloor\, V \wedge (\forall Y(\lfloor\varphi\rfloor\, Y \to r\, V\, Y))))\, W\|^{H^M,g[s/S_i][u/W_i]} = T$,
then $\|\lfloor\psi\rfloor W\|^{H^M,g[s/S_i][u/W_i]} = T$
$\Leftrightarrow$ For all $u \in D_i$ we have:
If $\|\lfloor\varphi\rfloor W\|^{H^M,g[s/S_i][u/W_i]} = T$ and
$\|\forall Y(\lfloor\varphi\rfloor\, Y \to r\, W\, Y)\|^{H^M,g[s/S_i][u/W_i]} = T$,
then $\|\lfloor\psi\rfloor V\|^{H^M,g[s/S_i][u/W_i]} = T$
$\Leftrightarrow$ For all $u \in D_i$ we have:
If $\|\lfloor\varphi\rfloor W\|^{H^M,g[s/S_i][u/W_i]} = T$ and
for all $t \in D_i$ we have $\|\forall Y(\lfloor\varphi\rfloor\, Y \to r\, W\, Y)\|^{H^M,g[s/S_i][u/W_i][t/Y_i]} = T$,
then $\|\lfloor\psi\rfloor W\|^{H^M,g[s/S_i][u/W_i]} = T$
$\Leftrightarrow$ For all $u \in D_i$ we have:
If $\|\lfloor\varphi\rfloor W\|^{H^M,g[s/S_i][u/W_i]} = T$ and
for all $t \in D_i$ we have $\|\lfloor\varphi\rfloor Y\|^{H^M,g[s/S_i][u/W_i][t/Y_i]} = T$ implies $Ir_{i\to\tau}(u,t) = T$,
then $\|\lfloor\psi\rfloor W\|^{H^M,g[s/S_i][u/W_i]} = T$
$\Leftrightarrow$ For all $u \in D_i$ we have:
If $u \in V(\varphi)$ and
for all $t \in D_i$ we have $t \in V(\varphi)$ implies $u \succeq t$,
then $u \in V(\psi)$ (**see the justification ***)
$\Leftrightarrow$ $\mathrm{opt}_{\succeq}(V(\varphi)) \subseteq V(\psi)$
$\Leftrightarrow$ $M, s \models \bigcirc(\psi/\varphi)$

Justification *: What we need to show is: $\|\lfloor\varphi\rfloor\|^{H^M,g[s/S_i]}$ is identified with $V(\varphi)$ (analogously ψ). By induction hypothesis, for all assignments g and states s, we have $\|\lfloor\varphi\rfloor S\|^{H^M,g[s/S_i]} = T$ if and only if $M, s \models \varphi$. Expanding the details of this equivalence we have: for all assignments g and states s

$$s \in \|\lfloor\varphi\rfloor\|^{H^M,g[s/S_i]} \qquad \text{(functions to type } o \text{ are associated with sets)}$$

$$
\begin{aligned}
&\Leftrightarrow \|\lfloor\varphi\rfloor\|^{H^M,g[s/S_i]}(s) = T \\
&\Leftrightarrow \|\lfloor\varphi\rfloor\|^{H^M,g[s/S_i]}\|S\|^{H^M,g[s/S_i]} = T \\
&\Leftrightarrow \|\lfloor\varphi\rfloor S\|^{H^M,g[s/S_i]} = T \\
&\Leftrightarrow M, s \vDash \varphi \\
&\Leftrightarrow s \in V(\varphi)
\end{aligned}
$$

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 $\|\lfloor\varphi\rfloor\|^{H^M,g[s/S_i]}$ is identified with $V(\varphi)$. Moreover, since H^M obeys the Denotatpflicht we know that $V(\varphi) \in D_\tau$.

Received \jreceived