I/O Logic in HOL

Ali Farjami
University of Luxembourg, Luxembourg
farjami110@gmail.com

Paul Meder
University of Luxembourg, Luxembourg
paul.meder@uni.lu

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

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

Abstract

A shallow semantical embedding of Input/output logic in classical higher-order logic is presented, and shown to be faithful (sound and complete). This embedding has been implemented in Isabelle/HOL, a proof assistant tool. We assess General Data Protection Regulation as a practical application for our reasoning framework.

Keywords: Input/output logic; Classical higher-order logic; Isabelle/HOL; General Data Protection Regulation.

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.

1 Introduction

Deontic logic is a reasoning framework about normative concepts such as obligation, permission, and prohibition. On one hand, we have the family of traditional deontic logics which includes Standard Deontic Logic (SDL), a modal logic of type **KD**, and Dyadic Deontic Logic (DDL) [18, 19]. On the other hand, we have the so-called "norm-based" deontic logics. Here the frameworks do not evaluate the deontic operators with regard to a set of possible worlds but with reference to a set of norms. Such a framework investigates which norms apply for a given input set, referred to as facts, and a set of explicitly given conditional norms, referred to as normative system. A particular framework that falls within this category, is called Input/Output (I/O) logic. It gained high recognition in the AI community and is also addressed as a chapter in the handbook of deontic logic [18]. The framework is expressive enough for dealing with legal concepts such as constitutive, prescriptive and defensible rules [13].

We propose a deontic reasoner based on I/O logic. *Basic Output* and *Basic Reusable Output* are two important I/O semantics that can be formulated with possible worlds semantics. We encode these two I/O semantics in classical Higher-Order Logic (HOL), also known as simple type theory. The syntax and semantics of HOL are well understood [6] and there exist automated proof tools for it; examples include Isabelle/HOL [23] and Leo-II [11]. For the embeddings of *Basic Output* and *Basic Reusable Output* in HOL, we use the shallow semantical embedding of Kripke semantics (**K** and **KT**) in classical higher-order logic. Both of the semantical embeddings are faithful.

The embeddings have been encoded in Isabelle/HOL to enable experiments for deontic reasoning frameworks. We have examined an application of General Data Protection Regulation (GDPR) as a practical example in our implementation. The experiments with this environment provide evidence that the logic's implementation fruitfully enables interactive and automated reasoning at the meta-level and the object level.

The article is structured as follows: Sec. 2 gives a quick review of modal logic and higher-order logic whereas Sec. 3 introduces I/O logic. The semantical embeddings of *Basic Output* and *Basic Reusable Output* in HOL are then devised and studied in Sec. 4. This section also shows the faithfulness (viz. soundness and completeness) of the embeddings. In Sec. 5, we use GDPR as a use-case example for our deontic reasoning framework.

2 Preliminaries

In this section, we give a quick overview of modal logic and HOL, and briefly remind the most important notions.

2.1 Modal logic K

The syntax of modal logic **K** is based on propositional logic with an additional modal operator $\Box$, and is generated as follows:

$$\varphi, \psi ::= p | \neg\varphi | \varphi \vee \psi | \Box\varphi$$

where p denotes an atomic formula. Other logical connectives such as $\wedge$, $\rightarrow$ and $\Diamond$, can be defined in the usual way. In terms of axioms, we have that the axiomatization consists of all the classical tautologies. Moreover, it is characterized by formulas of the form $\Box(\varphi \rightarrow \psi) \rightarrow (\Box\varphi \rightarrow \Box\psi)$, called axiom K, and by the rules such as *Modus ponens* (from φ and $\varphi \rightarrow \psi$ infer ψ) and *Necessitation* (from φ infer $\Box\varphi$).

A Kripke frame for **K** is a pair $\langle W, R\rangle$, where W is a non-empty set of possible worlds and R is a binary relation on W, called accessibility relation. A Kripke model for **K** is a triple $M = \langle W, R, V\rangle$, where $\langle W, R\rangle$ is a Kripke frame, and V is a function assigning a set of worlds to each atomic formula, that is, $V(p) \subseteq W$.

Satisfiability of a formula φ for a model $M = \langle W, R, V\rangle$ and a world $s \in W$ is expressed by the notation $M, s \models \varphi$ and we define $V(\varphi) = \{s \in W | M, s \models \varphi\}$. The satisfaction relation $\models$ is then defined as follows:

$M, s \models p$	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	for all $t \in W$, sRt $M, t \models \varphi$

As usual, a modal formula φ is *valid in a Kripke model* $M = \langle W, R, V\rangle$, i.e., $M \models \varphi$, if and only if for all worlds $s \in W$, we have $M, s \models \varphi$. A formula φ is called *valid*, denoted as $\models^{\mathbf{K}} \varphi$, if and only if it is valid in every Kripke model.

System **K** is sound and complete with respect to the class of all Kripke models. By adding additional axioms to system **K** such as $\mathbf{T} : \Box\varphi \rightarrow \varphi$, $\mathbf{4} : \Box\varphi \rightarrow \Box\Box\varphi$ or $\mathbf{5} : \Diamond\varphi \rightarrow \Box\Diamond\varphi$, the corresponding class of Kripke models has to satisfy the property of *reflexivity*, *transitivity* or *euclideaness*, respectively.

2.2 Classical higher-order logic

Classical higher-order logic is based on simple typed λ-calculus. We assume that the set T of simple types is freely generated from a set of basic types $\{o, i\}$ using

the function type constructor $\rightarrow$. Type o denotes the set of Booleans whereas type i refers to a non-empty set of individuals.

For $\alpha, \beta, o \in T$, the *language of HOL* is generated as follows:

$$s, t ::= p_\alpha | X_\alpha | (\lambda X_\alpha s_\beta)_{\alpha\rightarrow\beta} | (s_{\alpha\rightarrow\beta}\, t_\alpha)_\beta | (\neg_{o\rightarrow o} s_o)_o | (s_o \vee_{o\rightarrow o\rightarrow o} t_o)_o | (\Pi_{(\alpha\rightarrow o)\rightarrow o} s_{\alpha\rightarrow o})_o$$

where p_α denotes a typed constant symbol and X_α represents a typed variable symbol. $(\lambda X_\alpha s_\beta)_{\alpha\rightarrow\beta}$ and $(s_{\alpha\rightarrow\beta}\, t_\alpha)_\beta$ are called *abstraction* and *application*, respectively. HOL is a logic of terms in the sense that the *formulas of HOL* are given as terms of type o. Our selected primitive logical connectives are $\neg_{o\rightarrow o}$, $\vee_{o\rightarrow o\rightarrow o}$ and $\Pi_{(\alpha\rightarrow o)\rightarrow o}$. We often write $\forall X_\alpha s_o$ as syntactic sugar for $\Pi_{(\alpha\rightarrow o)\rightarrow o}(\lambda X_\alpha s_o)$.

The notions of *free variables*, *α-conversion*, *$\beta\eta$-equality* 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.

The semantics of HOL are well understood and thoroughly documented [6]. In the remainder, the semantics of choice are Henkin's general models [21].

A *frame* D is a collection $\{D_\alpha\}_{\alpha\in T}$ of nonempty sets D_α, such that $D_o = \{T, F\}$, denoting truth and falsehood, respectively. $D_{\alpha\rightarrow\beta}$ represents a collection 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_α to appropriate elements of D_α, called the *denotation of* p_α. The logical connectives $\neg$, $\vee$, Π and $=$ are always given in their expected standard denotations. 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 the variable W, which is now mapped to d. The *denotation* $\|s_\alpha\|^{M,g}$ of a 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:

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

Since $I(\neg_{o\rightarrow o})$, $I(\vee_{o\rightarrow o\rightarrow o})$ and $I(\Pi_{(\alpha\rightarrow o)\rightarrow o})$ always denote the standard truth functions, we have:

1. $\|(\neg_{o\rightarrow o}\, s_o)_o\|^{M,g} = T$ iff $\|s_o\|^{M,g} = F$.
2. $\|((\vee_{o\rightarrow o\rightarrow o}\, s_o)\, t_o)_o\|^{M,g} = T$ iff $\|s_o\|^{M,g} = T$ or $\|t_o\|^{M,g} = T$.

3. $\|(\forall X_\alpha s_o)_o\|^{M,g} = \|(\Pi_{(\alpha\to o)\to o}(\lambda X_\alpha s_o))_o\|^{M,g} = T$ iff for all $d \in D_\alpha$ we have $\|s_o\|^{M,g[d/X_\alpha]} = T$.

A HOL formula s_o is *true* in a Henkin model M under the assignment g if and only if $\|s_o\|^{M,g} = T$. This is also expressed by the notation $M, g \models^{HOL} s_o$. A HOL formula s_o is called *valid* in M, denoted as $M \models^{HOL} s_o$, if and only if $M, g \models^{HOL} s_o$ for all assignments g. Moreover, a formula s_o is called *valid*, denoted as $\models^{HOL} s_o$, if and only if s_o is valid in all Henkin models M. Finally, we define $\Sigma \models^{HOL} s_o$ for a set of HOL formulas Σ if and only if $M \models^{HOL} s_o$ for all Henkin models M with $M \models^{HOL} t_o$ for all $t_o \in \Sigma$.

3 Input/Output logic

Input/output logic was initially developed by Makinson and van der Torre, and introduced in [22]. I/O logic focuses on the reasoning and studying of conditional norms. The literature has presented several different I/O operators. In this paper, we will focus on *Basic Output* and *Basic Reusable Output*.

3.1 Syntax

$G \subseteq \mathcal{L} \times \mathcal{L}$ is called a normative system, with $\mathcal{L}$ representing the set of all the formulas of propositional logic. A pair $(a, x) \in G$ is referred to as a conditional norm or obligation, where a and x are formulas of propositional logic. a is called the body and represents some situation or condition, whereas x is called the head and represents what is obligatory or desirable in that situation. Thus the pair (a, x) is read as "given a, it is obligatory that x".

3.2 Semantics

For a set of formulas A, we have that $G(A) = \{x \mid (a, x) \in G \text{ for some } a \in A\}$ and $Cn(A) = \{x \mid A \vdash x\}$ with $\vdash$ denoting the classical propositional consequence relation. A set of formulas is considered as *complete* if it is either *maximal consistent* or equal to $\mathcal{L}$.

Definition 1 (Basic Output)**.** *Given a set of conditional norms G and an input set A of propositional formulas,*

$$out_2(G, A) = \bigcap\{Cn(G(V)) \mid A \subseteq V, V \text{ complete}\}$$

Definition 2 (Basic Reusable Output)**.** *Given a set of conditional norms G and an input set A of propositional formulas,*

$$out_4(G, A) = \bigcap\{Cn(G(V)) \mid A \subseteq V \supseteq G(V), V \text{ complete}\}$$

Besides those traditional formulations of the operators, the paper [22] documents modal formulations for out_2 and out_4.

Theorem 1. *$x \in out_2(G, A)$ if and only if $x \in Cn(G(\mathcal{L}))$ and $G^{\Box} \cup A \vdash_{\mathbf{S}} \Box x$ for any modal logic $\boldsymbol{S}$ with $\boldsymbol{K_0} \subseteq \boldsymbol{S} \subseteq \boldsymbol{K45}$.*

Theorem 2. *$x \in out_4(G, A)$ if and only if $x \in Cn(G(\mathcal{L}))$ and $G^{\Box} \cup A \vdash_{\mathbf{S}} \Box x$ for any modal logic $\boldsymbol{S}$ with $\boldsymbol{K_0 T} \subseteq \boldsymbol{S} \subseteq \boldsymbol{KT45}$.*

$\mathbf{K_0}$ is a subsystem of system $\mathbf{K}$ with axiom K, *Modus ponens* and the passage from ψ to $\Box\psi$ for every classical tautology ψ. The notation $G^{\Box}$ denotes the set of all modal formulas of the form $b \to \Box y$, such that $(b, y) \in G$. We have that $G^{\Box} \cup A \vdash_{\mathbf{S}} \Box x$ if for a finite subset Y of $G^{\Box} \cup A$, it holds that $(\bigwedge Y \to \Box x) \in \mathbf{S}$. The notation $\bigwedge Y$ stands for the conjunction of all the elements $y_1, y_2, \ldots, y_n$ in Y, i.e., $y_1 \wedge y_2 \wedge \cdots \wedge y_n$.

3.3 Proof Theory

In terms of proof theory, I/O logics are characterized by derivation rules about norms. Given a set of norms G, a derivation system is the smallest set of norms which extends G and is closed under certain derivation rules.

- (SI) Strengthening the input: from (a, x) to (b, x) whenever we have $\vdash b \to a$
- (WO) Weakening the output: from (a, x) to (a, y) whenever we have $\vdash x \to y$
- (AND) Conjunction of the output: from (a, x) and (a, y) to $(a, x \wedge y)$
- (OR) Disjunction of the input: from (a, x) and (b, x) to $(a \vee b, x)$
- (CT) Cumulative transitivity: from (a, x) and $(a \wedge x, y)$ to (a, y)

$deriv_2$ denotes the derivation system for the I/O operator out_2 and is formed by the rules SI, WO, AND and OR. The derivation system for out_4 is called $deriv_4$ and it is closed under all of the five rules.

4 Shallow semantical embedding

The so-called *shallow semantical embedding* is an approach proposed by Benzmüller [5], that uses classical higher-order logic as a meta-logic in order to represent and model the syntactic and semantical elements of a specific target logic. This methodology is documented and studied for Kripke semantics in [9] and for neighborhood semantics in [7]. This section presents shallow semantical embeddings of the I/O operators out_2 and out_4 in HOL and provides proofs for the soundness and completeness of both operators. To realize these embeddings, we use the provided modal formulations of the operators.

4.1 Semantical embedding of K in HOL

By introducing a new type i to denote possible worlds, the propositions of **K** are identified with certain HOL terms (predicates) of type $i \to o$. The type $i \to o$ is abbreviated as τ in the remainder. This allows us to represent the propositional formulas of **K** as functions from possible worlds to truth values in HOL and therefore the truth of a formula can explicitly be evaluated in a particular world. The HOL signature is assumed to further contain the constant symbol $r_{i\to i\to o}$. Moreover, for each propositional symbol p^j of **K**, 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 **K** into a formula $\lfloor\varphi\rfloor$ of HOL of type τ. The mapping is defined recursively:

$$\begin{array}{lcl} \lfloor p^j \rfloor & = & p^j_\tau \\ \lfloor \neg s \rfloor & = & \neg_{\tau\to\tau} \lfloor s \rfloor \\ \lfloor s \vee t \rfloor & = & \vee_{\tau\to\tau\to\tau} \lfloor s \rfloor \lfloor t \rfloor \\ \lfloor \Box s \rfloor & = & \Box_{\tau\to\tau} \lfloor s \rfloor \end{array}$$

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

$$\begin{array}{ll} \neg_{\tau\to\tau} & = \lambda A_\tau \lambda X_i \neg (A\, X) \\ \vee_{\tau\to\tau\to\tau} & = \lambda A_\tau \lambda B_\tau \lambda X_i (A\, X \vee B\, X) \\ \Box_{\tau\to\tau} & = \lambda A_\tau \lambda X_i \forall Y_i (\neg (r_{i\to i\to o} X\, Y) \vee A\, Y) \end{array}$$

Analyzing the truth of a translated formula $\lfloor s \rfloor$ in a particular world w, represented by the term w_i, corresponds to evaluating the application $(\lfloor s \rfloor\, w_i)$. In line with the previous work [10], we define $vld_{\tau\to o} = \lambda A_\tau \forall S_i (A\, S)$. With this definition,

validity of a formula s in **K** corresponds to the validity of the formula $(vld \lfloor s \rfloor)$ in HOL, and vice versa.

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

Lemma 1 (Aligning Henkin models H^M with Kripke models M)**.** *For every Kripke model $M = \langle W, R, V \rangle$ there exists a corresponding Henkin model H^M, such that for all formulas δ of* **K***, 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 [9]. □

Lemma 2 (Aligning Kripke models M_H with Henkin models H)**.** *For every Henkin model $H = \langle \{D_\alpha\}_{\alpha \in T}, I \rangle$ there exists a corresponding Kripke model M_H, such that for all formulas δ of* **K** *and for all assignments g and worlds s it holds*

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

Proof. See [9]. □

The following table summarizes the alignment of Kripke models and Henkin models. For the class of Kripke models $\langle W, R, V \rangle$ that satisfies some properties, such as *reflexivity*, the corresponding class of Henkin models also needs to satisfy the higher-order counter part of this property. For instance, in system **KT** the class of Kripke models satisfies the property of *reflexivity*, which corresponds to axiom **T**. The higher-order counter part of this property is represented as REF : $\forall X_i (r_{i \to i \to o} X_i X_i)$ and has to be satisfied by the constant $r_{i \to i \to o}$.

Kripke model $\langle W, R, V \rangle$	Henkin model $\langle D, I \rangle$
Possible worlds $s \in W$	Set of individuals $s_i \in D_i$
Accessibility relation R sRu	Binary predicates $r_{i \to i \to o}$ $Ir_{i \to i \to o}(s_i, u_i) = \top$
Propositional letters p^j Valuation function $s \in V(p^j)$	Unary predicates $p^j_{i \to o}$ Interpretation function $Ip^j_{i \to o}(s_i) = \top$

These correspondences between Kripke and Henkin models include the assumptions that have been formulated at the beginning of this section.

Theorem 3 (Faithfulness of the embedding of System **K** in HOL)**.**

$$\models^{\mathbf{K}} \varphi \text{ if and only if } \models^{HOL} vld \lfloor \varphi \rfloor$$

Proof. See [9, 10]. □

Theorem 4 (Faithfulness of the embedding of System **KT** in HOL)**.**

$$\models^{\mathbf{KT}} \varphi \textit{ if and only if } \{REF\} \models^{HOL} vld\lfloor\varphi\rfloor$$

Proof. See [9, 10] for a proof of the faithfulness of the embedding of modal logic **K** in HOL. The result for logic **KT** follows as a simple corollary; see also section 3.3 in [12]. □

4.2 Semantical embedding of I/O logic in HOL

Given a finite set of conditional norms G and an input set A of propositional formulas. For the embeddings of out_2 and out_4, we first use the corresponding translation into modal logic and afterwards we apply theorem 3 and theorem 4, respectively, in order to prove faithfulness.

Theorem 5 (Faithfulness of the embedding of out_2 in HOL)**.**

$$\varphi \in out_2(G, A)$$

if and only if

$$\models^{HOL} vld\lfloor\bigwedge(G^{\Box} \cup A) \rightarrow \Box\varphi\rfloor \textit{ and } \models^{HOL} vld\lfloor\bigwedge G(\mathcal{L}) \rightarrow \varphi\rfloor$$

Proof. The proof is directly by choosing $\mathbf{S} = \mathbf{K}$ in theorem 1 and then apply theorem 3.

$$\varphi \in out_2(G, A)$$

if and only if

$$G^{\Box} \cup A \vdash_{\mathbf{K}} \Box\varphi \text{ and } \varphi \in Cn(G(\mathcal{L}))$$

if and only if

$$\models^{\mathbf{K}} \bigwedge(G^{\Box} \cup A) \rightarrow \Box\varphi \text{ and } \bigwedge G(\mathcal{L}) \vdash \varphi$$

if and only if[1]

$$\models^{\mathbf{K}} \bigwedge(G^{\Box} \cup A) \rightarrow \Box\varphi \text{ and } \models^{\mathbf{K}} \bigwedge G(\mathcal{L}) \rightarrow \varphi$$

if and only if

$$\models^{HOL} vld\lfloor\bigwedge(G^{\Box} \cup A) \rightarrow \Box\varphi\rfloor \text{ and } \models^{HOL} vld\lfloor\bigwedge G(\mathcal{L}) \rightarrow \varphi\rfloor$$

□

[1] It holds that $\vdash p$ if and only if $\models^{\mathbf{K}} p$ for propositional formulas.

Theorem 6 (Faithfulness of the embedding of out_4 in HOL)**.**

$$\varphi \in out_4(G, A)$$

if and only if

$$\{REF\} \models^{HOL} vld\lfloor\bigwedge(G^{\Box} \cup A) \rightarrow \Box\varphi\rfloor \text{ and } \{REF\} \models^{HOL} vld\lfloor\bigwedge G(\mathcal{L}) \rightarrow \varphi\rfloor$$

Proof. The proof is directly by choosing $\mathbf{S} = \mathbf{KT}$ in theorem 2 and then apply theorem 4.

$$\varphi \in out_4(G, A)$$

if and only if

$$G^{\Box} \cup A \vdash_{\mathbf{KT}} \Box\varphi \text{ and } \varphi \in Cn(G(\mathcal{L}))$$

if and only if

$$\models^{\mathbf{KT}} \bigwedge(G^{\Box} \cup A) \rightarrow \Box\varphi \text{ and } \bigwedge G(\mathcal{L}) \vdash \varphi$$

if and only if [2]

$$\models^{\mathbf{KT}} \bigwedge(G^{\Box} \cup A) \rightarrow \Box\varphi \text{ and } \models^{\mathbf{KT}} \bigwedge G(\mathcal{L}) \rightarrow \varphi$$

if and only if

$$\{REF\} \models^{HOL} vld\lfloor\bigwedge(G^{\Box} \cup A) \rightarrow \Box\varphi\rfloor \text{ and } \{REF\} \models^{HOL} vld\lfloor\bigwedge G(\mathcal{L}) \rightarrow \varphi\rfloor$$

□

4.3 Implementation of I/O logic in Isabelle/HOL

The semantical embeddings of out_2 and out_4 as devised in the previous section have been implemented into the higher-order proof assistant tool Isabelle/HOL [23]. The embedding of the operator out_2 is based on a system **K**. We declare the type i to denote possible worlds, introduce an Isabelle/HOL constant r to represent the relation R for the corresponding class of Kripke models and define the connectives of a modal logic as HOL formulas.

Let the set of conditional norms G be composed of the elements (a, e) and (b, e), where a, b and e are propositional symbols, and let the input set A correspond to the singleton set containing $a \vee b$. According to the provided translation, $e \in out_2(G, A)$ if and only if $G^{\Box} \cup A \vdash_{\mathbf{K}} \Box e$ and $e \in Cn(G(\mathcal{L}))$. The first part makes use of a modal logic of type **K** whereas the second part uses classical propositional logic. Theorem

[2] It holds that $\vdash p$ if and only if $\models^{\mathbf{KT}} p$ for propositional formulas.

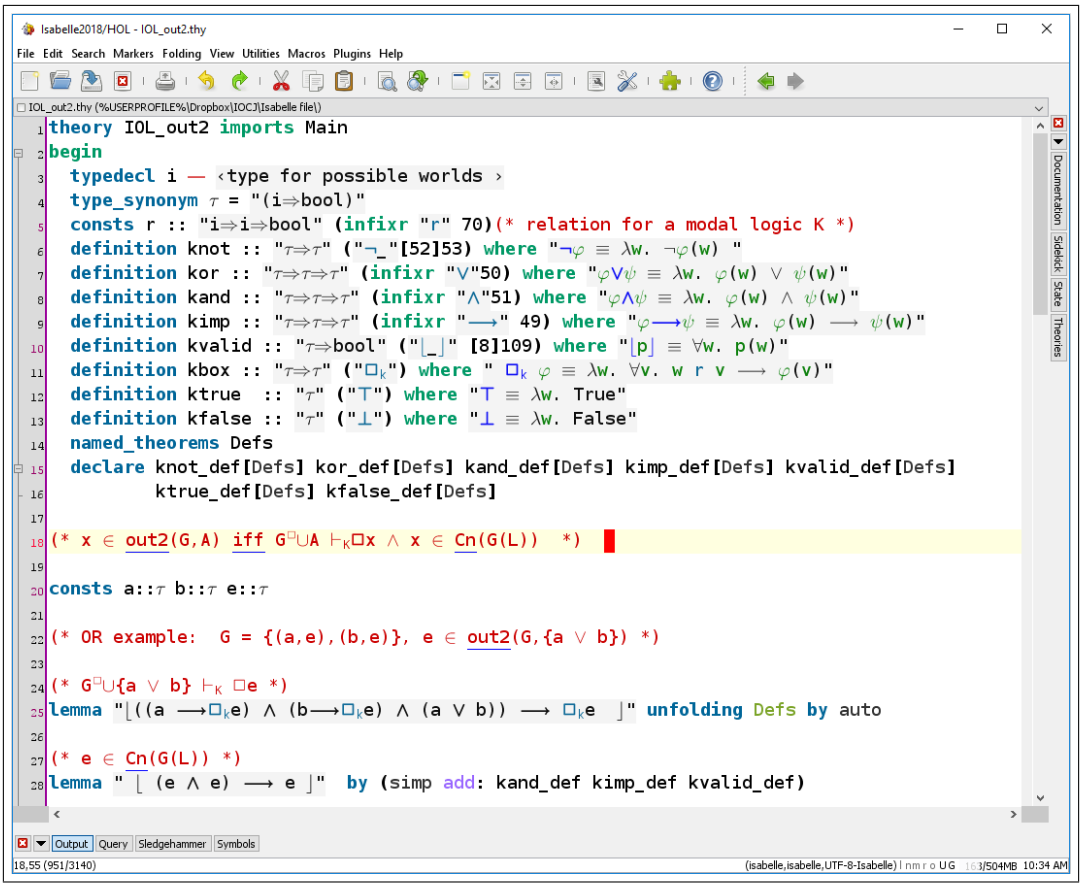

Figure 1: Semantical embedding of out_2 in Isabelle/HOL

5 provides us now with higher-order formulations for both of these statements, i.e., $\models^{HOL} vld\lfloor\bigwedge(G^{\Box} \cup A) \rightarrow \Box\varphi\rfloor$ and $\models^{HOL} vld\lfloor\bigwedge G(\mathcal{L}) \rightarrow \varphi\rfloor$, respectively. Regarding the implementation, the propositional symbols a, b and e have to be declared as constants of type τ. The framework's integrated automatic theorem provers (ATPs), called via the Sledgehammer tool [15], are able to prove both of the statements. This is depicted in figure 1. This small example shows that our encoded out_2 operator satisfies the rule of disjunction (OR).

Consider a set of conditional norms $N = \{(a, b), (a \wedge b, e)\}$ and an input set $A = \{a\}$. The rule of cumulative transitivity (CT) is not satisfied by the out_2 operator. This can also be verified with our implementation. The model finder Nitpick [14] is able to generate a countermodel for the statement $N^{\Box} \cup A \vdash_{\mathbf{K}} \Box e$ and therefore we were able to show that $e \notin out_2(N, A)$. In particular, Nitpick came

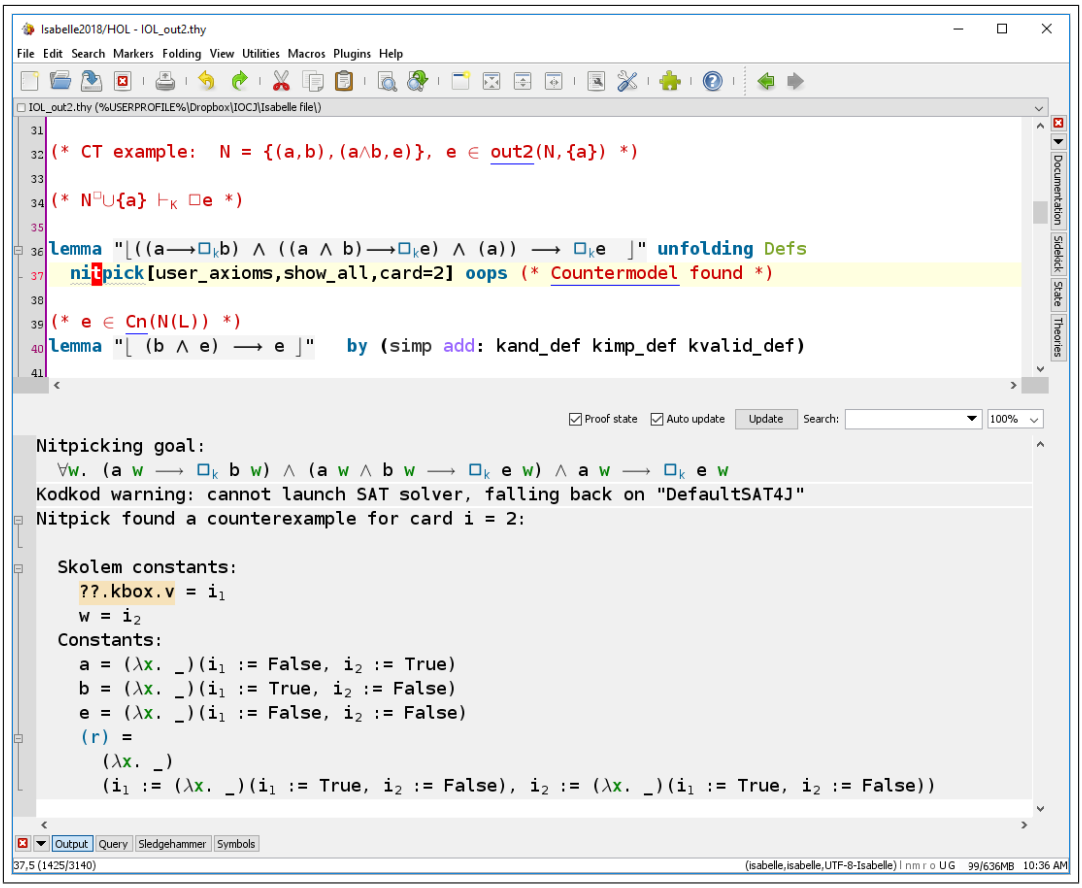

Figure 2: Further experiments with the embedding of out_2 in Isabelle/HOL

up with a model M consisting of two possible worlds i_1 and i_2. We have that a is evaluated to true at i_2 by the valuation function V, and $V(b) = \{i_1\}$ and $V(e) = \emptyset$. For the relation R, we have that the world i_1 is related to itself and i_2 is related to i_1. This is denoted by the set $R = \{(i_1, i_1), (i_2, i_1)\}$. Since the relation R is not reflexive, the formula $((a \rightarrow \Box b) \wedge ((a \wedge b) \rightarrow \Box e) \wedge a) \rightarrow \Box e$ is not valid. The formulation of the example and the generation of the countermodel is illustrated in Figure 2.

The embedding of the out_4 operator is based on a system **KT** which means that the corresponding class of Kripke models satisfy the property of reflexivity. In our implementation, the accessibility relation for this system is denoted by the constant r_t which we declare as reflexive. Due to this property, the Sledgehammer tool is able to prove the statement $N^{\Box} \cup A \vdash_{\mathbf{KT}} \Box e$ and thus we can verify that

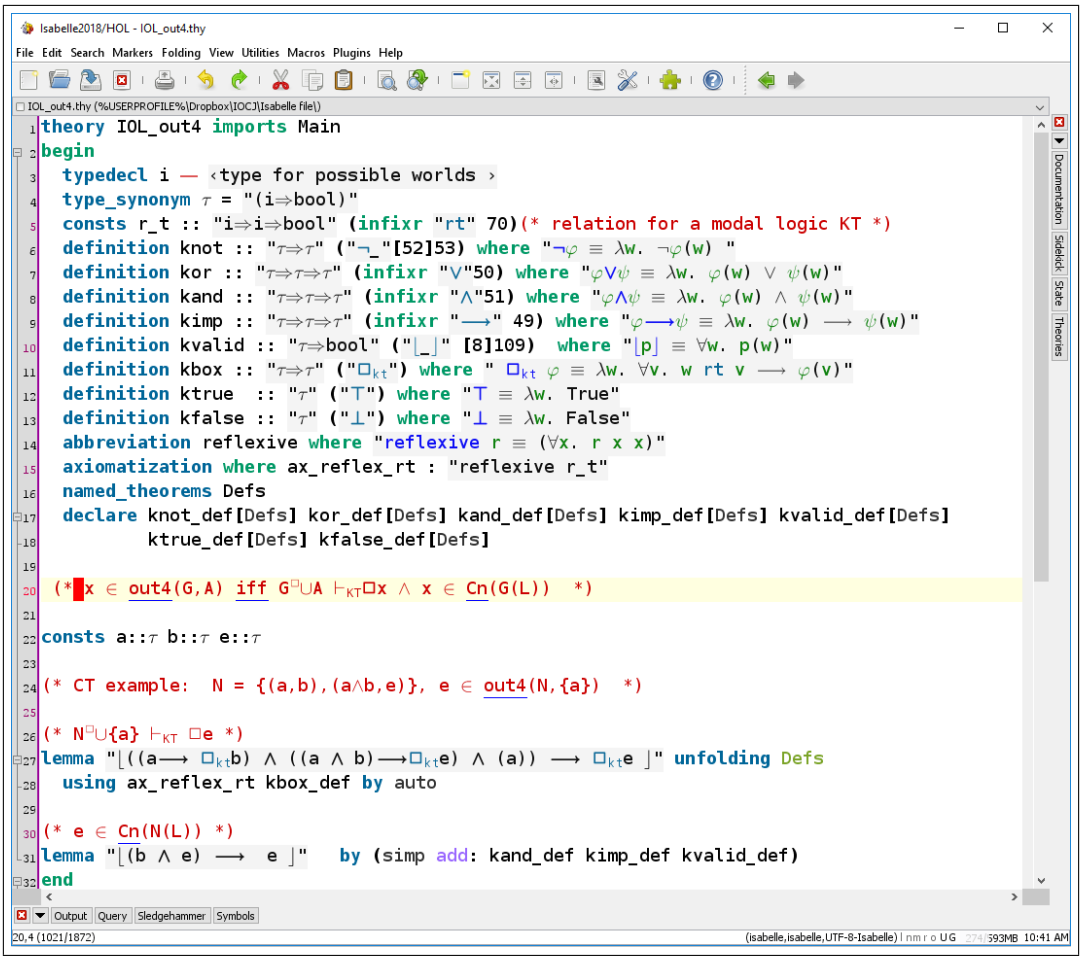

Figure 3: Semantical embedding of out_4 in Isabelle/HOL

$e \in out_4(N, A)$. Figure 3 shows the encoding of the out_4 operator in Isabelle/HOL and the verification of the CT example.

5 Application in legal reasoning

The paper [8] already documents some practical experiments of automated I/O logic in the domain of legal reasoning. In particular, General Data Protection Regulation (GDPR, Regulation EU 2016/679) is used as an application scenario. By this regulation, the European Parliament, the Council of the European Union and the

European Commission aim to strengthen and unify data protection for all the individuals within the European Union. The following two norms are part of GDPR:

1. Personal data shall be processed lawfully (Art. 5). For example, the data subject must have given consent to the processing of his or her personal data for one or more specific purposes (Art. 6/1.a).

2. If the personal data have been processed unlawfully (none of the requirements for a lawful processing applies), the controller has the obligation to erase the personal data in question without delay (Art. 17.d, right to be forgotten).

The authors [8] added the following two knowledge units. This establishes a typical Contrary-To-Duty (CTD) scenario which was then analyzed in the context of I/O logic.

3. It is obligatory e.g. as part of a respective agreement between a customer and a company) to keep the personal data (as relevant to the agreement) provided that it is processed lawfully.

4. Some data in the context of such an agreement has been processed unlawfully.

We formulated this GDPR scenario in Isabelle/HOL as an application scenario for the embedded out_2 operator; cf. Figure 4. The lines 48-50 show the set of *Norms* which is composed of:

- $(\top, process\_data\_lawfully)$
  This norm states that it is obligatory to process data lawfully.

- $(\neg process\_data\_lawfully, erase\_data)$
  This norms states that if the data was not processed lawfully then it is obligatory to erase the data.

- $(process\_data\_lawfully, \neg erase\_data)$
  This norms states that if the data has been processed lawfully then it is obligatory to keep the data.

Line 51 shows the *Input* set. We assume a situation where the data has not been processed lawfully, formally $Input = \{\neg process\_data\_lawfully\}$. At line 54, we introduce the constants symbols which are representing the propositions. Just like for the previous examples, each proposition is declared as a constant of type τ. Next, we want to check that $erase\_data$ is outputted in the context of $\neg process\_data\_lawfully$, meaning that the data should be erased in the

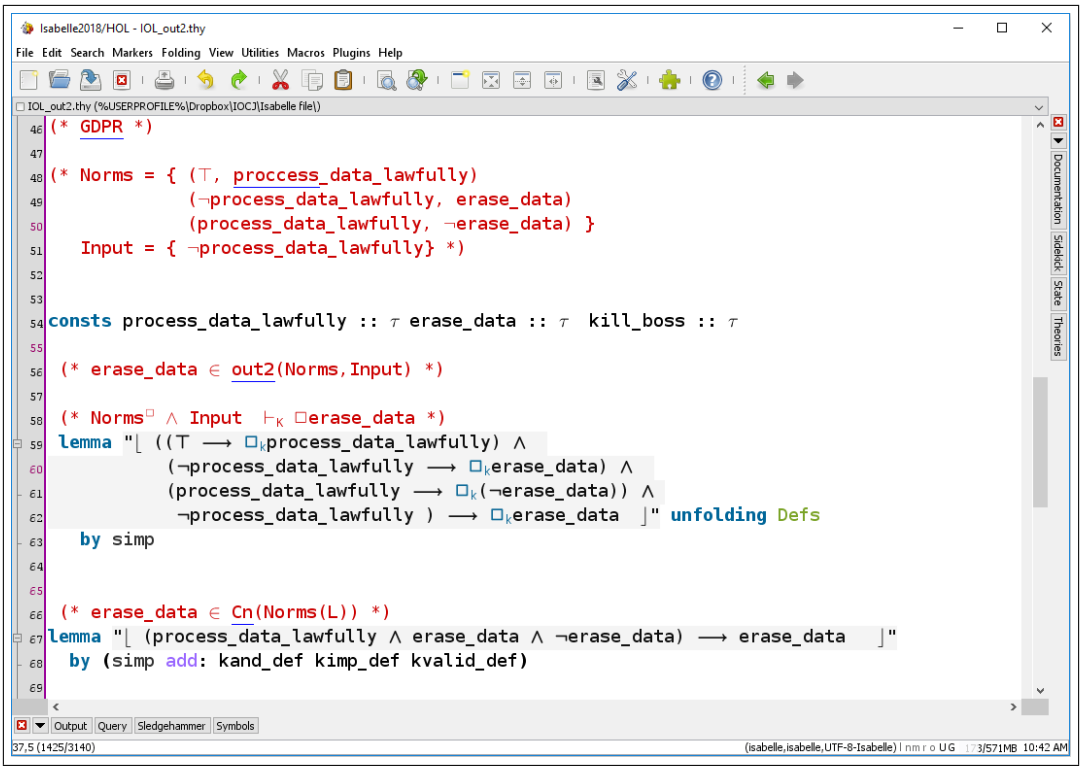

Figure 4: GDPR scenario in Isabelle/HOL

situation when the data has not been processed lawfully. So we have to verify that $erase\_data \in out_2(Norms, Input)$. By the modal translation of this operator, we have to check that $Norms^{\Box} \cup Input \vdash_{\mathbf{K}} \Box erase\_data$ and $earse\_data \in Cn(Norms(\mathcal{L}))$. The lines 59-62 and 67 show the formulations for those statements, respectively. Both of them could be by proven by the integrated ATPs of Isabelle/HOL.

Like in a related paper [8], we also showed that we are not able to derive any weird or unethical conclusions such as killing the boss, cf. Figure 5. The model finder Nitpick is able to generate a countermodel for the following statement:

$$Norms^{\Box} \cup Input \vdash_{\mathbf{K}} \Box kill\_boss$$

Therefore it showed that $kill\_boss \notin out_2(Norms, Input)$. Nitpick found a model M consisting of two possible worlds i_1 and i_2. For the valuation function V, we have that $V(erase\_data) = \{i_1\}$, $V(proccess\_data\_lawfully) = \{i_1\}$ and $V(kill\_boss) = \{i_2\}$ and for the relation R, we have that $R = \{(i_1, i_1), (i_2, i_1)\}$. The world i_2

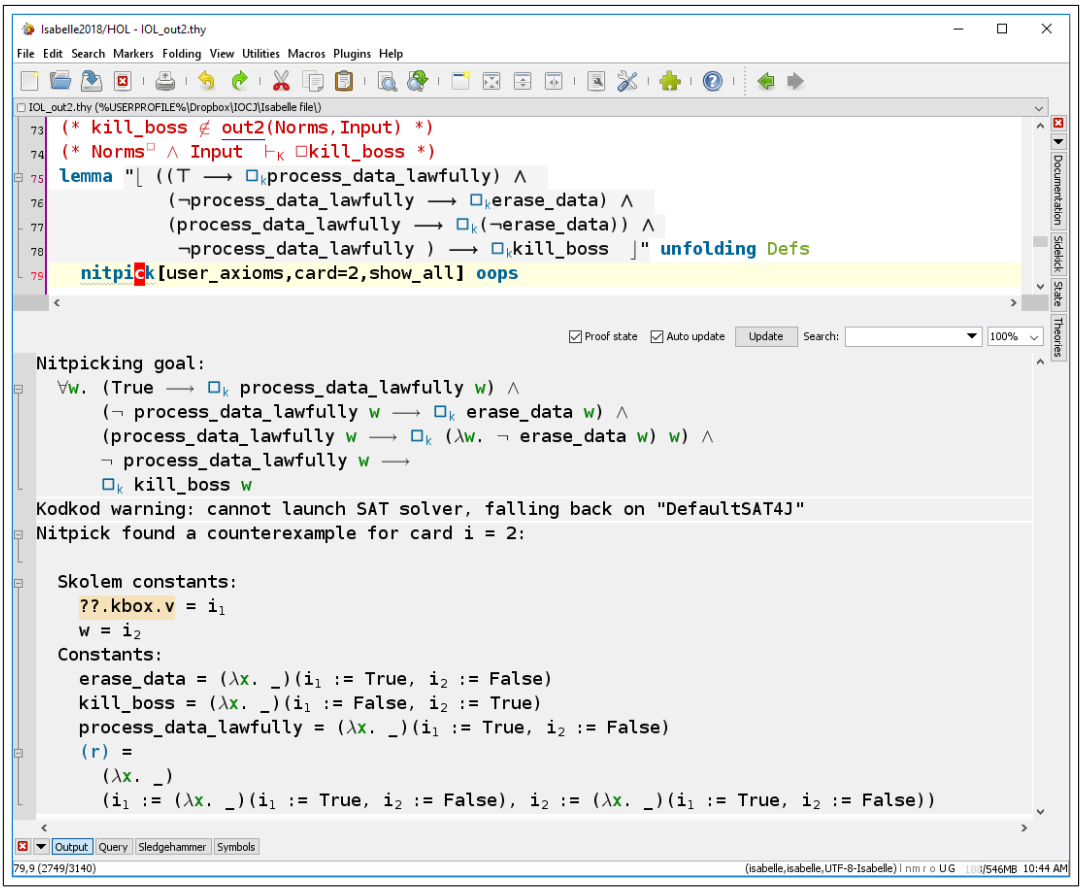

Figure 5: Further experiments with the GDPR scenario in Isabelle/HOL

satisfies the following formulas:

- $\top \rightarrow \Box process\_data\_lawfully$
- $\neg process\_data\_lawfully \rightarrow \Box erase\_data$
- $process\_data\_lawfully \rightarrow \Box \neg erase\_data$
- $\neg process\_data\_lawfully$

However, the formula $\Box kill\_boss$ is not satisfied in the possible world i_2.

6 Conclusion

We have presented straightforward embeddings of two I/O operators in HOL and we have shown that those embeddings are sound and complete, i.e. faithful. The work presented here along with Benzmüller et al. [7] provide the theoretical foundation for the implementation and automation of deontic logic within existing theorem provers and proof assistants for HOL. Future research should investigate whether the provided implementation already supports non-trivial applications in practical normative reasoning such as legal reasoning or multi-agent systems, or whether further improvements are required. We could also employ our implementation to systematically study some meta-logical properties of I/O logic within Isabelle/HOL. Moreover, we could similarly implement the intuitionistic I/O logic [25].

Acknowledgements

We thank the anonymous reviewers for their valuable feedback and 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] Benzmüller, C.: Cut-elimination for quantified conditional logic. Journal of Philosophical Logic, **46**(3), 333–353, (2017)

[5] Benzmüller, C.: Universal (Meta-)Logical Reasoning: Recent Successes. Science of Computer Programming, in print, (2018)

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

[7] 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)

[8] Benzmüller, C., Parent, X., van der Torre, L.: A Deontic Logic Reasoning Infrastructure. In: Manea, F., Miller, R., Nowotka, D. (eds.) 14th Conference on Computability in Europe, CiE 2018, Kiel, Germany, July 30-August, 2018, LNCS, vol. 10936, Springer, (2018)

[9] Benzmüller, C., Paulson, L.: Multimodal and intuitionistic logics in simple type theory. The Logic Journal of the IGPL **18**(6), 881–892 (2010)

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

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

[12] Benzmüller, C., Woltzenlogel Paleo, B.: Higher-Order Modal Logics: Automation and Applications. In Paschke, A., Wolfgang, F. (eds.) Reasoning Web 2015 (Invited paper), LNCS, vol. 9203, pp. 32–74, Springer, (2015)

[13] Boella, G., van der Torre, L.: Regulative and Constitutive Norms in Normative Multiagent Systems. In: Dubois, D., and Welty, C., Williams, M. (eds.) Principles of Knowledge Representation and Reasoning: Proceedings of the Ninth International Conference (KR2004), Whistler, Canada, June 2-5, 2004, pp. 255–266, AAAI Press, USA, (2004)

[14] 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)

[15] Blanchette, J. C., Paulson, L. C.: Hammering Away - A User's Guide to Sledgehammer for Isabelle/HOL, (2017)

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

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

[18] 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)

[19] Gabbay, D., Horty, J., Parent, X., van der Meyden, R., van der Torre, L.: Handbook of deontic logic and normative systems. Volume 2. College Publications, UK, (2018)

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

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

[22] Makinson, D., van der Torre, L.: Input/output logics. Journal of Philosophical Logic, **29**(4), 383–408, (2000)

[23] 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)

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

[25] Parent, X.: A modal translation of an intuitionistic I/O operation. Presented at the 7th Workshop on Intuitionistic Modal Logic and Applications (IMLA 2017), organized by V. de Paiva and S. Artemov at the University of Toulouse (France), 17-28 July, 2017 (satellite workshop of ESSLI17). Post-conference version of the paper under review.

Received \jreceived