Designing Normative Theories of Ethical Reasoning:
Formal Framework, Methodology, and Tool Support

Christoph Benzmüller\textsuperscript{a,b}, Xavier Parent\textsuperscript{a}, Leendert van der Torre\textsuperscript{a}

\textsuperscript{a}Computer Science and Communications, University of Luxembourg, Esch-sur-Alzette, Luxembourg
\textsuperscript{b}Department of Mathematics and Computer Science, Freie Universität Berlin, Berlin, Germany

Abstract

The area of formal ethics is experiencing a shift from a unique or standard approach to normative reasoning, as exemplified by so-called standard deontic logic, to a variety of application-specific theories. However, the adequate handling of normative concepts such as obligation, permission, prohibition, and moral commitment is challenging, as illustrated by the notorious paradoxes of deontic logic. In this article we introduce an approach to design and evaluate theories of normative reasoning. In particular, we present a formal framework based on higher-order logic, a design methodology, and we discuss tool support. Moreover, we illustrate the approach using an example of an implementation, we demonstrate different ways of using it, and we discuss how the design of normative theories is now made accessible to non-specialist users and developers.

Keywords: Knowledge Representation and Reasoning; Automated Reasoning and Theorem Proving; Normative Reasoning; Normative Systems; Philosophical and Ethical Issues; Semantical Embedding; Higher-Order Logic

1. Introduction

Many artificial moral agent architectures are based on explicit normative systems and normative reasoning developed in formal ethics. For example, Liao et al.\textsuperscript{[1]} describe a smart home visualized in Fig.\textsuperscript{[1]} which uses an argumentation engine\textsuperscript{[2]} to make ethical decisions based on normative systems of stakeholders. The argumentation-based mechanism is used to find an agreement in case of moral dilemmas, and to offer an explanation as to how a specific morally sensitive decision has been made. Liao et al. assume the existence of pre-defined

\textsuperscript{1}Benzmüller is funded by the VolkswagenStiftung under grant CRAP (Consistent Rational Argumentation in Politics). Parent and van der Torre are supported by the European Union’s Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement MIREL (MIning and REasoning with Legal texts) No 690974.
systems of norms, coming from the outside and constraining the agent’s behavior. Like humans, intelligent systems evolve in a highly regulated environment. For instance, a smart home processes personal data 24/7, and, as will be discussed in Sect. 6.1, there are legal rules one must comply with when processing personal data. For example, in the European Union the General Data Protection Regulation (GDPR, Regulation EU 2016/679) applies. Given a description of the current situation, the smart home should thus be able to, e.g., check compliance with these rules, and act accordingly in case of non-compliance.

In general, it becomes increasingly evident that adequate explicit representations of legal and ethical knowledge are beneficial, if not vital, to obtain ethical intelligent systems with increasing levels of autonomy in their decision making; see, e.g., [3][4]. There are grounds to believe that such competencies are needed not only for guaranteeing sufficient degrees of reliability and accountability, but also for achieving human intuitive interaction means regarding explainability and transparency of decisions. In particular, in very critical application areas, such as autonomous warfare and automated financial trading, suitable means for regulating, controlling, assessing and explaining the behaviour of intelligent systems are required. Bottom-up approaches alone, in which intelligent agents, e.g., learn to adequately behave in critical situation, are not sufficient to convincingly address the challenge. It seems appropriate to combine such methods with top-down means based on explicitly represented, governing upper principles from which appropriate legal and ethical behaviour can be inferred. This way an additional, orthogonal layer of control can be enforced in future intelligent autonomous systems which is based on explicit representation of legal and ethical theories. Such an additional competency layer also enables better explanations and justifications of the agents decisions. What counts, in particular, in highly critical contexts, is not how an agent computed the critical action it wants to perform, but whether this action passes the additional assessment by the upper-level explicit ethical reasoner before its execution. Figure[2]
Figure 2: Explicit ethical and legal control unit for intelligent autonomous systems.

graphically depicts our viewpoint. Based on information perceived from the environment and the intelligent systems internal knowledge, the AI Reasoner is computing suggestions on the next action(s) to be executed. It does not matter whether these computations are based on subsymbolic or symbolic techniques, or combinations of them, since the suggested actions are not executed immediately—at least not those considered most critical. Instead they are internally assessed, before execution, against some explicitly modelled ethical and legal theories, which govern and control the behaviour of the entire system, and which also support system verification and intuitive user explanations.

The area of formal ethics is experiencing a shift from a unique or standard approach to normative reasoning, as exemplified by so-called standard deontic logic, to a variety of application-specific theories. Deontic logic is the formal study of principles of normative reasoning, viz. reasoning about norms (obligation, permission, prohibition). It was originally developed as a tool for formalizing normative reasoning in ethical and legal contexts and has since been explored primarily by philosophical logicians and a few legal theorists. A number of systems have been proposed. In Sect. 3 we briefly describe the most prominent ones. This plurality of systems reflects the fact that the adequate handling of normative reasoning is a most challenging endeavour.

Moreover, the challenges of the adequate handling of normative concepts such as obligation, permission, prohibition, and moral commitment are also illustrated by the notorious paradoxes of normative reasoning. Deontic logics were
developed by logicians and philosophers, with little interest in automation and implementation. The need to strengthen the link between theory and practice motivated the creation of a bi-annual conference series in 1991, called Deontic Logic and Normative Systems (DEON), the publication of two handbooks [5, 6] and the publication of a textbook [7].

This article therefore addresses the practical development of computational tools for normative reasoning based on formal methods. Since ethical and legal theories for intelligent systems as well as suitable normative reasoning formalisms are both still under development [5], we introduce a flexible workbench to support empirical studies with such theories in which the preferred logic formalisms themselves can still be varied, complemented, assessed and compared. In this way we build a bridge between the deduction systems community and the formal ethics community.

The infrastructure we propose draws on both recent developments in universal logical reasoning in classical higher-order logic (HOL) [8] and the coalescence and improvements in interactive and automated theorem proving (ATP) in HOL, as witnessed by systems such as Isabelle/HOL [9], LEO-II [10] and Leo-III [11]. We are thus linking the historically rich research areas of HOL, automated theorem proving, normative reasoning and formal ethics; see also Fig. 3 which will be explained in more detail in Sect. 2.1.

This article is structured as follows. Section 2 further motivates the need for a flexible normative reasoning infrastructure and presents our methodology. Section 3 briefly surveys and discusses the challenging area of normative reasoning. Moreover, this section explains which particular deontic logics have been implemented in our approach. A concrete example implementation of a state-of-the-art deontic logic, Åqvist system \( E \), is presented in further detail in Sect. 4. Tool support is discussed in Sect. 5 and subsequently two case studies are presented in Sect. 6. The first case study illustrates contrary-to-duty compliant reasoning in the context of the general data protection regulation. A second, larger case study shows how our framework scales for the formalization and automation of challenging ethical theories (Gewirth’s “Principle of Generic Consistency” [12]) on the computer. To enable the latter work an extended contrary-to-duty compliant higher-order deontic logic has been provided and utilized in our framework. Sections 7 and 8 discuss related work and further research, and Sect. 9 concludes the article.
2. The SSE approach: HOL as a universal meta-logic

Our core motivation has been to develop an easy-to-use and flexible suite of reasoning tools supporting the experimentation with different normative theories, in different application scenarios. However, the implementation of specialist theorem provers and model finders for deontic logics is very tedious and requires expertise. Our approach therefore focuses on reuse and adaptation of existing technology rather than new implementations from first principles. In particular, we want to enable novices to become acquainted with the area of normative reasoning in short time in a computer-assisted hands-on fashion; this should enable them to gradually acquire much needed background expertise. A main objective of our work therefore has been to build up a starting basis of mechanized deontic logics and ethical and legal theories, and to make them accessible for experimentation to students and researchers. Our approach therefore pays much attention to intuitive interaction within sophisticated user-interfaces to access, explore, assess and modify both the used deontic logics and the encoded

---

With formal ethics community we refer to Normative Multi-Agent Systems [http://icr.uni.lu/normas], Deontic Logic and Normative Systems [http://deonticlogic.org] and Formal Ethics [http://www.fe2019.ugent.be]. The deduction systems community is further split into the area of Automated Reasoning [http://aarinc.org] and Interactive Theorem Proving [https://itp2018.inria.fr]. With our work we bridge between all of these communities; see also Fig. 3 and the explanations in Sect. 2.
ethical theories. The tools we reuse are state-of-the-art reasoning systems for classical logic that are actively developed by the deduction systems community. That relieves us from resource intensive investments in the implementation of new technology. However, to enable this reuse of technology a bridge between deontic logics and classical logics was needed. This bridge is provided by an adaptation of the shallow semantical embeddings \[8\] approach, which is discussed in more detail in Sect. 2.1 below.

The methodology and technology we contribute requires some modest background knowledge to be acquired. The developers’ perspective is a bit more ambitious and requires some background knowledge in meta-logic HOL (see Sect. 2.2) and a good mastery of the semantical embedding technique (see Sect. 4). However, also at that level we support the adoption of initial skills by providing a library of example encodings of deontic logics to start with. This, e.g., enables copy and pasting from these encodings. Several successful student projects at BSc, MSc and PhD level meanwhile provide good evidence for the practical relevance of our approach at the developers level \[13, 14, 15, 16\].

2.1. Methodology: Faithful semantical embeddings in HOL

Our methodology, depicted graphically in Fig. 3, is based on the reuse of existing reasoning tools enabled by the shallow semantical embedding approach \[17, 8\]. HOL is utilized in this approach as a universal meta-logic in which different deontic logics are semantically embedded. We have extended this approach in our project for ambitious deontic logics. This methodology enables the use of interactive proof assistants, such as Isabelle/HOL, which comes with a sophisticated user-interface and, in addition, integrates various state-of-the-art reasoning tools. The ATPs integrated with Isabelle/HOL via the Sledgehammer \[18\] tool comprise higher-order ATPs, first-order ATPs and satisfiability modulo theories (SMT) solvers, and many of these systems internally again employ efficient SAT solving technology. Isabelle/HOL also provides two model finders, Nitpick \[19\] and Nunchaku.

The SSE approach utilizes the syntactic capabilities of the higher-order theorem prover (a) to represent the semantics of a target logic, and (b) to define the original syntax of the target theory within the prover. The overall infrastructure, in combination with the SSE approach, meets our demanding requirements regarding flexibility along different axes; cf. Sect. 5.

An initial focus in the SSE approach has been on quantified modal logics \[17\]. One core aspects is that the standard translation \[20\] from propositional modal logic to first-order logic can be modelled (i.e., embedded) within HOL, without requiring an external translation mechanism. The modal operator \(\Box\), for example, can be explicitly defined by the \(\lambda\)-term \(\lambda \varphi. \lambda w. \exists v. (Rw v \land \varphi v)\), where

\[^3\text{Shallow semantical embeddings are different from deep semantic embeddings. In the latter the syntax of the target logic is represented using an inductive data structure (e.g., following the definition of the language). The semantics of a formula is then evaluated by recursively traversing the data structure. Deep semantical embeddings often require technical inductive proofs that can be avoided when shallow semantical embeddings are used instead.}\]
\( R \) denotes the accessibility relation associated with \( \Diamond \). This definition, however, can be hidden from the user, who can construct new modal logic formulas involving \( \Diamond \varphi \) and use them to represent and proof theorems.

Most importantly, however, such an embedding of modal logic operators in HOL can be extended to also include quantifiers. We briefly illustrate this idea using an example (omitting types, as above, for better readability). First, it is relevant to note that \( \forall x. \phi_x \) is shorthand in HOL for \( \Pi(\lambda x. \phi_x) \), where the logical constant symbol \( \Pi \) is given an obvious semantics, namely to check whether the set of objects denoted by \( (\lambda x. \phi_x) \) is the set of all objects (of the respective type). \( \exists x. \phi_x \) is hence shorthand for \( \neg \Pi(\lambda x. \neg \phi_x) \). The important and interesting aspect thus is that additional binding mechanisms for universal and existential quantifiers can be avoided in HOL by reusing \( \lambda \)-notation. This principle can now be applied also to obtain SSEs for quantified modal logics (and for many other quantified non-classical logics). For example, \( \Diamond \forall x. \phi_x \) is represented as \( \Diamond \Pi'(\lambda x. \lambda w. P_xw) \), where \( \Pi' \) stands for the \( \lambda \)-term \( \lambda \Phi. \lambda w. \Pi(\lambda x. \Phi xw) \) and where the \( \Diamond \) gets resolved as described above. The following series of conversions explains this encoding in more detail:

\[
\begin{align*}
\Diamond \forall x. \phi_x & \equiv \Diamond \Pi'(\lambda x. \lambda w. P_xw) \\
& \equiv \Diamond((\lambda \Phi. \lambda w. \Pi(\lambda x. \Phi xw))(\lambda x. \lambda w. P_xw)) \\
& \equiv \Diamond((\lambda w. \Pi((\lambda x. \lambda w. P_xw)xw))) \\
& \equiv \Diamond(\lambda w. \Pi((\lambda x. P_xw))) \\
& \equiv (\lambda \varphi. \lambda w. \exists v. (Rwv \land \varphi v))(\lambda w. \Pi((\lambda x. P_xw))) \\
& \equiv (\lambda \varphi. \lambda w. \neg \Pi(\lambda v. \neg (Rwv \land \varphi v)))(\lambda w. \Pi((\lambda x. P_xw))) \\
& \equiv (\lambda w. \neg \Pi(\lambda v. \neg (Rwv \land \Pi((\lambda x. P_xw)v)))) \\
& \equiv (\lambda w. \neg \Pi(\lambda v. \neg (Rwv \land \forall x. P_xv))) \\
& \equiv (\lambda w. \exists v. Rwv \land \forall x. P_xv)
\end{align*}
\]

This illustrates the embedding of \( \Diamond \forall x. P_x \) in HOL.\(^4\) Moreover, this embedding can be accompanied with different notions of validity. For example, we say \( \Diamond \forall x. P_x \) is globally valid (valid for all worlds \( w \)) if and only if \( \forall w. ((\Diamond \forall x. P_x) w) \) holds. Local validity for a particular actual world, denoted by a constant symbol \( aw \), then amounts to checking whether \( ((\Diamond \forall x. P_x) aw) \) is true in HOL.

What has been sketched above is an SSE for a first-order quantified modal logic \( K \) with a possibilist notion of quantification. However, depending on the type we assign to variable \( x \) in \( \Diamond \forall x. P_x \), the sketched solution scales for arbitrary higher-order types. Since provers such as Isabelle/HOL and Leo-III support (restricted forms of) polymorphism, respective universal and existential quantifiers for the entire type hierarchy can be introduced with a single definition (see, e.g., lines 14–17 in Fig. \( \text{[4]} \) where respective polymorphic quantifier definitions, including binder notation, are provided for Aqvist’s system \( \mathbf{E} \)).

\(^4\) In the implementation of our approach in Isabelle/HOL such conversions are hidden by default, so that the user may interact with the system at the level of the target logic and enter formulas such as \( \Diamond \forall x. P_x \). Definition unfolding is handled in the background, but can made visible upon request by the user.
Further details on the semantical embedding of quantified modal logics in HOL, including a proof of faithfulness, are provided in [17]. Standard deontic logic (modal logic KD) can easily be obtained from this work. To do so we simply postulate in metalogic HOL that the accessibility relation $R$ underlying the $\diamond$ operator is serial. The corresponding $\textbf{D}$ axiom $\Box \phi \supset \neg \Box \neg \phi$ (or equivalently, $\neg (\Box \phi \land \Box \neg \phi)$, see also Sect. 3.1.1 below) then becomes derivable as a corollary from this postulate and the SSE (and so does the K-schema and the necessitation rule, already in base logic K). Further emendations of the presented framework to obtain multi-modal logics and an actualist notion of quantification have been proposed by Benzmüller et al. (see [8] and the references therein for further details).

2.2. Source or meta-logic: HOL

HOL has its roots in the logic of Frege’s Begriffsschrift [21]. However, the version of HOL as addressed here refers to a (simply) typed logic of functions, which has been put foward by Church [22]. It provides $\lambda$-notation, as an elegant and useful means to denote unnamed functions, predicates and sets. Types in HOL eliminate paradoxes and inconsistencies.

To keep this article sufficiently self-contained we briefly introduce HOL (the reader eventually wants to skip this subsection and get back again later). More detailed information on HOL and its automation can be found in the literature [22, 23, 24].

**Definition 1** (Types). The set $T$ of simple types in HOL is freely generated from a set of basic types $BT \supseteq \{o, i\}$ using the function type constructor $\to$. Usually, type $o$ denotes the (bivalent) set of Booleans, and $i$ denotes a non-empty set of individuals. Further base types may be added.

**Definition 2** (Terms). The terms of HOL are defined as follows (where $C_\alpha$ denotes typed constants and $x_\alpha$ typed variables distinct from $C_\alpha$; $\alpha, \beta, o \in T$):

$$s, t ::= C_\alpha | x_\alpha | (\lambda x_\alpha. s_\beta)_{\alpha \to \beta} | (s_\alpha \to t_\beta)_{\beta}$$

Complex typed HOL terms are thus constructed via $\lambda$-abstraction and function application, and HOL terms of type $o$ are called formulas.

As primitive logical connectives we choose $\neg_{o \to o}, \lor_{o \to o}$ and $\Pi_{(a \to o) \to o}$ (for each type $\alpha$), that is, we assume that these symbols are always contained in the signature. Binder notation $\forall x_\alpha. s_\alpha$ is used as an abbreviation for $\Pi_{(a \to o) \to o} \lambda x_\alpha. s_\alpha$.

Additionally, description or choice operators $\epsilon_{(\alpha \to o) \to o}$ (for each type $\alpha$) or primitive equality $=_{\alpha \to \alpha \to o}$ (for each type $\alpha$), abbreviated as $=^\alpha$, may be added. From the selected set of primitive logical connectives, other logical connectives can be introduced as abbreviations. Equality can also be defined by exploiting Leibniz’ principle, expressing that two objects are equal if they share the same properties. Type information as well as brackets may be omitted if obvious from the context.

We consider two terms to be equal if the terms are the same up to the names of bound variables (i.e., $\alpha$-conversion is handled implicitly).
Definition 3 (Substitution). Substitution of a term $s_\alpha$ for a variable $x_\alpha$ in a term $t_\beta$ is denoted by $[s/x]t$. Since we consider $\alpha$-conversion implicitly, we assume the bound variables of $t$ avoid variable capture.

Definition 4 ($\lambda$-conversion). Prominent operations and relations on HOL terms include $\beta\eta$-normalization and $\beta\eta$-equality, $\beta$-reduction and $\eta$-reduction. It is well known, that for each simply typed $\lambda$-term there is a unique $\beta$-normal form and a unique $\beta\eta$-normal form. Two terms $l$ and $r$ are $\beta\eta$-equal, denoted as $l =_{\beta\eta} r$, if their $\beta\eta$-normal forms are identical (up to $\alpha$-conversion). Examples of $\lambda$-conversions have been presented on p. 7 (types were omitted there).

The semantics of HOL is well understood and thoroughly documented in the literature [23, 25]. The semantics of choice for our work is Henkin [26]’s general semantics. The following sketch of standard and Henkin semantics for HOL closely follows Andrews [22].

Definition 5 (Frame). A frame is a collection $\{D_\alpha\}_{\alpha \in T}$ of nonempty sets called domains such that $D_\alpha = \{T, F\}$ where $T$ represents truth and $F$ falsehood, $D_i \neq \emptyset$ and $D_u \neq \emptyset$ are chosen arbitrary, and $D_\alpha \rightarrow \beta$ are collections of total functions mapping $D_\alpha$ into $D_\beta$.

Definition 6 (Interpretation). An interpretation is a tuple $\langle \{D_\alpha\}_{\alpha \in T}, I \rangle$ where $\{D_\alpha\}_{\alpha \in T}$ is a frame and where function $I$ maps each typed constant symbol $c_\alpha$ to an appropriate element of $D_\alpha$, which is called the denotation of $c_\alpha$. The denotations of $\neg, \vee$ and $\Pi_{(\alpha \to \beta)}$ are always chosen as usual. A variable assignment $\phi$ maps variables $X_\alpha$ to elements in $D_\alpha$.

Definition 7 (Henkin model). An interpretation is a Henkin model (general model) if and only if there is a binary valuation function $V$ such that $V(\phi, s_\alpha) \in D_\alpha$ for each variable assignment $\phi$ and term $s_\alpha$, and the following conditions are satisfied for all $\phi$, variables $x_\alpha$, constants $C_\alpha$, and terms $l_\alpha \rightarrow \beta, r_\alpha, s_\beta$ ($\alpha, \beta \in T$): $V(\phi, x_\alpha) = \phi(x_\alpha)$, $V(\phi, C_\alpha) = I(C_\alpha)$, $V(\phi, l_\alpha \rightarrow \beta, r_\alpha) = V(\phi, l_\alpha \rightarrow \beta)V(\phi, r_\alpha)$, and $V(\phi, \lambda x_\alpha s_\beta)$ represents the function from $D_\alpha$ into $D_\beta$ whose value for each argument $z \in D_\alpha$ is $V(\phi[z/x_\alpha], s_\beta)$, where $\phi[z/x_\alpha]$ is that assignment such that $\phi[z/x_\alpha](z) = z$ and $\phi[z/x_\alpha](y_\beta) = \phi(y_\beta)$ when $y_\beta \neq x_\alpha$.

If an interpretation $H = \langle \{D_\alpha\}_{\alpha \in T}, I \rangle$ is an Henkin model the function $V$ is uniquely determined and $V(\phi, s_\alpha) \in D_\alpha$ is called the denotation of $s_\alpha$.

Definition 8 (Standard model). $H = \langle \{D_\alpha\}_{\alpha \in T}, I \rangle$ is called a standard model if and only if for all $\alpha$ and $\beta$, $D_\alpha \rightarrow \beta$ is the set of all functions from $D_\alpha$ into $D_\beta$. Obviously each standard model is also a Henkin model.

Definition 9 (Validity). A formula $c$ of HOL is valid in a Henkin model $H$ if and only if $V(\phi, c) = T$ for all variable assignments $\phi$. In this case we write $H \models_{HOL} c$ is (Henkin) valid, denoted as $\models_{HOL} c$, if and only if $H \models_{HOL} c$ for all Henkin models $H$.

The following theorem verifies that the logical connectives behave as intended. The proof is straightforward.
Theorem 1. Let $V$ be the valuation function of a Henkin model $H$. The following properties hold for all variable assignments $\phi$, terms $s_o, t_o, l_\alpha, r_\alpha$, and variables $x_\alpha, w_\alpha$: $V(\phi, \neg s_o) = T$ if and only if $V(\phi, s_o) = F$, $V(\phi, s_o \lor t_o) = T$ if and only if $V(\phi, s_o) = T$ or $V(\phi, t_o) = T$, $V(\phi, \forall x_\alpha, s_o) = V(\phi, \Pi (\alpha \rightarrow o) \rightarrow \lambda x_\alpha, s_o) = T$ if and only if for all $v \in D_\alpha$ holds $V(\phi[v/w_\alpha], (\lambda x_\alpha s_\alpha) w_\alpha) = T$, and if $l_\alpha = \beta \eta r_\alpha$ then $V(\phi, l_\alpha) = V(\phi, r_\alpha)$.

A HOL formula $c$ that is Henkin-valid is obviously also valid in all standard models. Thus, when a Henkin-sound theorem prover for HOL finds a proof for $c$, then we know that $c$ is also theorem in standard semantics. More care has to be taken when model finders for HOL return models or countermodels, since theoretically these models could be non-standard. In practice this has not been an issue, since the available model finders for HOL return finite models only, and finite models in HOL are known to be standard [22]. Most importantly, however, the returned models in Isabelle/HOL can always be inspected by the user.

3. Theories of normative reasoning covered by the SSE approach

To explain what theories of normative reasoning are covered by the SSE approach, we briefly survey the area of deontic logic.

3.1. Deontic logic

Deontic logic [27, 5, 7] is the field of logic that is concerned with normative concepts such as obligation, permission, and prohibition. Alternatively, a deontic logic is a formal system capturing the essential logical features of these concepts. Typically, a deontic logic uses $O p$ to mean that it is obligatory that $p$, (or it ought to be the case that $p$), and $P p$ to mean that it is permitted, or permissible, that $p$. Deontic logic can be used for reasoning about normative multiagent systems, i.e., about multiagent organizations with normative systems in which agents can decide whether to follow the explicitly represented norms, and the normative systems specify how, and to which extent, agents can modify the norms. Normative multiagent systems need to combine normative reasoning with agent interaction, and thus raise the challenge to relate the logic of normative systems to aspects of agency.

There are two main paradigms in deontic logic, which we briefly describe in the next two subsections.

3.1.1. Modal logic paradigm

Traditional (or “standard”) deontic logic (SDL) is a normal propositional modal logic of type KD, which means that it extends the propositional tautologies with the axioms $K: O (p \rightarrow q) \rightarrow (O p \rightarrow O q)$ and $D: \neg (O p \rightarrow O \neg p)$, and it is closed under the inference rules modus ponens $p, p \rightarrow q \rightarrow q$ and generalization or necessitation $p / O p$. Prohibition and permission are defined by $F p = O \neg p$ and $P p = \neg O \neg p$. SDL is an unusually simple and elegant theory. An advantage of
its modal-logical setting is that it can easily be extended with other modalities such as epistemic or temporal operators and modal accounts of action.

Dyadic deontic logic (DDL) introduces a conditional operator \( O(p/q) \), to be read as “it ought to be the case that \( p \), given \( q \)”. Many DDLs have been proposed to deal with so-called contrary-to-duty (CTD) reasoning, cf. [28] for an overview on this area. In brief, the CTD problem is mostly about how to represent conditional obligation sentences dealing with norm violation, and an example is provided Sect. 6.1. Two full-blooded DDLs are: the DDL proposed by Hansson [29], Åqvist [30, 31] and Kratzer [32], and the one proposed by Carno and Jones [28, 33]. A notable feature of the DDL by Åqvist is that it also provides support for reasoning about so-called prima facie obligations; cf. [34]. A prima facie obligation is one that leaves room for exceptions.

To enable ethical agency a model of decision needs to be integrated in the deontic frames. Horty’s deontic STIT logic [35], which combines deontic logic with a modal logic of action, has been proposed as a starting point. The semantic condition for the STIT-ought is a utilitarian generalization of the SDL view that “it ought be that \( A \)” means that \( A \) holds in all deontically optimal worlds. To the best of our knowledge, the axiomatization problem for this logic is open.

3.1.2. Norm-based paradigm

The term “norm-based” deontic logic has been coined by Hansen [36] to refer to a family of frameworks analysing the deontic modalities not with reference to a set of possible worlds (some of them being more ideal than others), but with reference to a set of explicitly given norms. In such a framework, the central question is: given some input (e.g. a fact) and a set of explicitly given conditional norms (a normative system), what norms apply? Thus, the perspective is slightly different from the traditional setting, focusing on inference patterns [37].

A specific norm-based deontic logic is called input/output (I/O) logic. Initially devised by Makinson [38] and further developed over the past years by van der Torre and colleagues, I/O logic has gained increased recognition in the AI community. This is evidenced by the fact that the framework has its own chapter in the Handbook of Deontic Logic and Normative Systems [5]. Other examples of norm-based deontic logic are: Horty’s theory of reasons based on Reiter’s default logic; Hansen’s logic of prioritized conditional obligations. This proposed classification of paradigms (norm-based vs modal logic) is not meant to be exhaustive or exclusive. Some frameworks (like adaptive deontic logic [41, 42]) combine the two.

I/O logic can be viewed as a rule-based system. The knowledge base takes the form of a set of rules of the form \((a,b)\) to be read as “if \( a \) then \( b \)”. The key feature of I/O logic is that it uses an operational semantics, based on the notion of detachment, rather than a truth-functional one in terms of truth-values and possible worlds. On the semantical side, the meaning of the deontic concepts is given in terms of a set of procedures, called I/O operations, yielding outputs (e.g., obligations) for inputs (facts). On the syntactical side, the proof-theory is formulated as a set of inference rules manipulating pairs of formulas rather than individual formulas. The framework covers functionalities that are
unanimously regarded as characteristic of the legal domain, and thus required to enable effective legal reasoning:

1. Support for the modelling of constitutive rules, which define concepts or constitute activities that cannot exist without such rules (e.g. legal definitions such as “property”), and prescriptive rules, which regulate actions by making them obligatory, permitted, or prohibited.

2. Management of the reification of rules that are objects with properties, such as jurisdiction, authority, temporal attributes [43].

3. Implementation of defeasibility—see [43, 44]; when the antecedent of a rule is satisfied by the facts of a case (or via other rules), the conclusion of the rule presumably holds, but is not necessarily true.

3.2. Theories of normative reasoning implemented

The following theories of normative reasoning have been “implemented” by utilising the SSE approach:

- SDL: All logics from the modal logic cube, including logic KD, i.e. SDL, have meanwhile been faithfully implemented in the SSE approach [17]. These implementations scale for FO and even HO extensions.

- DDL: the DDL by Åqvist [30, 31] and the DDL by Carmo and Jones [33]: Faithful semantical embeddings of these logics in Isabelle/HOL are already available [45, 46], and most recently the ATP Leo-III has been adapted to accept DDL as input.

- I/O logic [38]: The main challenge comes from the fact that the framework does not have a truth-functional semantics, but an operational one. First experiments with the semantical embedding of the I/O-operator “out1” (called simple-minded) and “out2” (called basic) in Isabelle/HOL have been presented in [13, 47]. Some relevant I/O logic variants have very recently been studied [48, 49], and we conjecture that some of these variants are related to certain non-normal modal logics, e.g., conditional logics with a selection function semantics or similar logics with a neighbourhood semantics. However, the embedding of such logics has already been studied in the first authors previous work [50, 51]. It should thus be possible to benefit from these existing results in the given context.

4. Sample embedding: Åqvist’s system E

In this section, to illustrate our approach, we describe our embedding of Åqvist [30]’s dyadic deontic logic in HOL. The system is called E. The axiomatization problem for this logic was resolved only recently [31]. The reason for
giving the example of $E$ is threefold. First, and foremost, this logic is a recognized standard for normative reasoning. This is because it provides support for both contrary-to-duty (CTD) reasoning and reasoning about so-called *prima facie* obligations §. Second, a proof of the faithfulness of the embedding is available, and the establishment of this result is our success criterium. Third, its implementation scales for first- and higher-order extension.

4.1. Target logic: System $E$

**Definition 10.** The language of $E$ is generated by the following BNF:

$$
\phi ::= p \mid \neg \phi \mid \phi \land \phi \mid \Box \phi \mid \Box (\phi / \phi)
$$

$\Box \phi$ is read as “$\phi$ is settled as true”, and $\Box (\psi / \phi)$ as “$\psi$ is obligatory, given $\phi$”. $\Box \phi$ (“$\phi$ is unconditionally obligatory”) is short for $\Box (\phi / \top)$.

Traditionally so-called preference models are used as models for the language.

**Definition 11.** A preference model $M = (W, \succeq, V)$ is a tuple where:

- $W$ is a (non-empty) set of possible worlds;
- $\succeq$ is a binary relation over $W$ ordering the worlds according to their betterness or comparative goodness. $s \succeq t$ is read as “$s$ is at least as good as $t$”;
- $V$ is a valuation assigning to each propositional letter $p$ a set of worlds (the set of those at which $p$ holds).

For $E$, no specific properties are assumed of the betterness relation $\succeq$. It is known that the assumptions of reflexivity and totalness (every worlds are pairwise comparable) do not “affect” the logic; cf. Theorem 2.

Intuitively the evaluation rule for $\Box (\psi / \phi)$ puts $\Box (\psi / \phi)$ true if and only if the best (according to $\succeq$) $\phi$-worlds are all $\psi$-worlds. Formally:

**Definition 12 (Satisfaction).** Given a preference model $M = (W, \succeq, V)$ and a world $s \in W$, we define the satisfaction relation $M, s \models \phi$ as usual except for the following two new clauses

- $M, s \models \Box \phi$ iff for all $t \in M$, $M, t \models \phi$
- $M, s \models \Box (\psi / \phi)$ iff $\text{opt}_\succeq(||\phi||) \subseteq ||\psi||$

where $||\phi||$ is the set of worlds at which $\phi$ holds and $\text{opt}_\succeq(||\phi||)$ is the subset of those that are optimal according to $\succeq$:

$$
\text{opt}_\succeq(||\phi||) = \{ s \in ||\phi|| \mid \forall t \ (t \models \phi \rightarrow s \succeq t) \}
$$

**Definition 13 (Validity).** A formula $\phi$ is valid in the class $P$ of all preference models (notation: $\models^P \phi$) if and only if, for all preference models $M$ and all worlds $s$ in $M$, $M, s \models \phi$. 

13
Definition 14. E is the proof system consisting of the following axiom schemata and rule schemata (the labels are from [52]):

\[
\begin{align*}
\phi, & \text{ where } \phi \text{ is a tautology from PL} & \text{(PL)} \\
\Box(\phi \rightarrow \psi) & \rightarrow (\Box \phi \rightarrow \Box \psi) & \text{(K)} \\
\Box \phi & \rightarrow \Box \Box \phi & \text{(4)} \\
\neg \Box \phi & \rightarrow \Box \neg \Box \phi & \text{(5)} \\
\bigcirc (\psi \rightarrow \chi/\phi) & \rightarrow \bigcirc(\bigcirc(\psi/\phi) \rightarrow \bigcirc(\chi/\phi)) & \text{(COK)} \\
\bigcirc (\phi/\phi) & \text{Id} \\
\bigcirc (\chi/(\phi \land \psi)) & \rightarrow \bigcirc((\psi \rightarrow \chi)/\phi) & \text{(Sh)} \\
\bigcirc (\psi/\phi) & \rightarrow \Box \bigcirc (\psi/\phi) & \text{(Abs)} \\
\Box \psi & \rightarrow \bigcirc(\psi/\phi) & \text{(Nec)} \\
\Box(\phi \leftrightarrow \psi) & \rightarrow (\bigcirc(\chi/\phi) \leftrightarrow \bigcirc(\chi/\psi)) & \text{(Ext)} \\
\text{If } \vdash \phi \text{ and } \vdash \phi \rightarrow \psi \text{ then } \vdash \psi & \text{(MP)} \\
\text{If } \vdash \phi \text{ then } \vdash \Box \phi & \text{(N)} 
\end{align*}
\]

The notions of theorem and consistency are defined as usual.

The following theorems resolve two long-standing open problems in deontic logic. The second and third clauses in the statement of Theorem 2 mean that the assumptions of reflexivity and totalness do not have any import on the logic.

**Theorem 2.** E is sound and complete with respect to the following three classes of preference models:
1. the class of all preference models;
2. the class of preference models in which the betterness relation is required to be reflexive;
3. the class of preference models in which the betterness relation is required to be total (for all \(s\) and \(t\), either \(s \succeq t\) or \(t \succeq s\)).

**Proof.** The proof can be found in Parent [31].

**Theorem 3.** The theoremhood problem in E (“Is \(\phi\) a theorem in E?”) is decidable.

**Proof.** The proof can be found in Parent [53].

Stronger systems may be obtained by adding further constraints on \(\geq\), like transitivity and the so-called limit assumption (which rules out infinite sequences of strictly better worlds).

**4.2. Embedding of E in HOL**

The formulas of E are identified in our semantical embedding with certain HOL terms (predicates) of type \(i \rightarrow o\), where terms of type \(i\) are assumed to denote possible worlds and \(o\) denotes the (bivalent) set of Booleans.
That is, the HOL type $i$ is now identified with a (non-empty) set of worlds. Type $i \rightarrow o$ is abbreviated as $\tau$ in the remainder. The HOL signature is assumed to contain the constant symbol $R_{i,\tau}$. Moreover, for each propositional symbol $P^j$ of $E$, the HOL signature must contain the corresponding constant symbol $P^j_{\tau}$. 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.

**Definition 15.** The mapping $[\cdot]$ translates a formula $\varphi$ of $E$ into a formula $[\varphi]$ of HOL of type $\tau$. The mapping is defined recursively in the usual way except for the following two new clauses:

\[
[\Box \varphi] = \Box_{\tau \rightarrow \tau} [\varphi] \\
[\Diamond (\psi/\phi)] = \Diamond_{\tau \rightarrow \tau} [\psi] [\phi]
\]

where $\Box_{\tau \rightarrow \tau}$ and $\Diamond_{\tau \rightarrow \tau}$ abbreviate the following formulas of HOL:

\[
\Box_{\tau \rightarrow \tau} = \lambda \phi_{\tau}. \lambda x_1. \forall y_1.(\phi y) \\
\Diamond_{\tau \rightarrow \tau} = \lambda \phi_{\tau}. \lambda \psi_\tau. \lambda x_1. \forall w_1.((\lambda v_1.(\phi v \land (\forall y_1.(\phi y \rightarrow R_{i,\tau} v y)))) w \rightarrow \psi w)
\]

The basic idea is to make the modal logic’s possible worlds structure explicit by introducing a distinguished predicate symbol $R$ to represent the betterness relation, and to translate a formula directly according to its semantics. For instance $\Diamond (\psi/\phi)$ translates into

\[
\lambda x.((\forall w.(\phi w \land (\forall y.(\phi y \rightarrow R_{i,\tau} y))) w \rightarrow \psi w))
\]

**Definition 16** (Validity of an embedded formula). Global validity ($vld$) of an embedded formula $\phi$ of $E$ in HOL is defined by the equation

\[
vld [\phi] = \forall z. [\phi] z
\]

For example, checking the global validity of $\Diamond (\psi/\phi)$ in $E$ is hence reduced to checking the validity of the formula

\[
\forall w.(\phi w \land (\forall y.(\phi y \rightarrow R_{i,\tau} w y))) \rightarrow \psi w)
\]

in HOL.

This definition will be hidden from the user, who can construct now deontic logic formulas involving $\Diamond (\psi/\phi)$ and use them to represent and proof theorems.

### 4.3. Faithfulness of the embedding

It can be shown that the embedding is faithful, in the sense given by theorem 4. Remember that the establishment of such a result is our main success criterium. Intuitively, theorem 4 says that a formula $\phi$ (in the language of $E$) is valid in the class of all preference models if and only if its translation $[\phi]$ (in the language of HOL) is valid in the class of Henkin models in the sense of definition 16.
Theorem 4 (Faithfulness of the embedding).

\[ \models^P \phi \text{ if and only if } \models^{HOL} \text{vld} [\phi] \]

Proof. The proof can be found in Benzmüller et al. [45]. The crux of the argument consists in relating preference models with Henkin models in a truth-preserving way.

4.4. Encoding in Isabelle/HOL

The practical employment of the above described semantical embedding for E in Isabelle/HOL is straightforward and can be done in a separate theory file. This way, for a concrete application scenario, we can simply import the embedding without dealing with any technical details. The complete embedding is quite short (approx. 30 lines of code with line breaks) and is displayed in Fig. 4.

The embedding has been extended to include quantifiers as well. The possibilist quantifiers (cf. Sect. 2.1) are introduced in lines 13–17—this amounts to having a fixed domain of individuals rather than a world-relative domain (the actualist semantics). If needs be, actualist quantifiers can also be introduced in a few lines of code; see the study [8], where analogous definitions of both possibilist and actualist quantifiers are presented for higher-order conditional logic. The betterness relation \( R \) is introduced as an uninterpreted constant symbol in line 20, and the conditional obligation operator is defined in line 26. Its definition mirrors the one given in Definition [15]. The unconditional obligation operator is introduced in line 27. It is defined in terms of its dyadic counterpart in the usual way. Last, global and local validity (cf. Sect. 2.1) are introduced in lines 30–33.

Lines 35–40 show sample queries. On line 35, consistency of the embedding is confirmed. Lines 37–40 illustrate how Isabelle/HOL can be used as a heuristic tool in correspondence theory. The focus is on the assumption of transitivity of the betterness relation. One would like to know what its syntactical counterpart is. Isabelle/HOL confirms that such an assumption has the effect of validating the axiom Lewis [54] called CV (line 37). Isabelle/HOL also confirms that transitivity is not equivalent to CV: The model finder Nitpick [19] integrated with Isabelle/HOL finds a model validating CV in which the betterness relation is not transitive (lines 39 and 40).

5. Tool support

5.1. Support for different reasoning tasks

In a nutshell, a reasoner is a tool that can perform reasoning tasks in a given application domain. Reasoning thereby refers to the process of deriving or concluding information that is not explicitly encoded in the knowledge base. Which information is derivable and which is not is thereby dependent on the particular choice of logic. The reasoning tasks that are particularly relevant in our context, for example, include:
Figure 4: Embedding of the semantics of system E in Isabelle/HOL.
Compliance checking: Is the current situation, resp. an intended action in a given situation, compliant with a given regulation (a set of formally represented norms)?

Non-compliance analysis: If non-compliance is the result of a compliance check, can the reasons be revealed (and explained)?

Entailment checking: Does such-and-such obligation or legal interpretation follow from a given regulation?

Non-entailment analysis: If entailment checking fails, can the reasons be revealed (and explained)?

Consistency checking: Is a given regulation consistent? Is such-and-such norm, as part of a given regulation, consistent with this other set of norms, stemming from another regulation? Is such-and-such legal interpretation consistent with another one?

Inconsistency analysis: If consistency checking fails, can a minimal set of conflicting norms be revealed and the inconsistency be explained?

Consistency checking, non-compliance analysis and non-entailment analysis are well supported by model finders respectively counter-model finders, while the other tasks generally require theorem proving technology. A powerful deontic reasoner should thus ideally provide both (counter-)model finding and theorem proving. Moreover, intuitive proof objects and adequate presentations of (counter-)models are desirable to enable user explanations.

5.2. Flexibility along different axes

While the above reasoning tasks are comparably easy to provide for many decidable propositional fragments of deontic logics, it becomes much less so for their quantified extensions. In fact, we are not aware of any system, besides the work presented in this article, that still meets all these requirements for some contrary-to-duty compliant first-order or higher-order deontic logic; cf. the related work in Sect. 7.

Our approach addresses challenges along different axes, which are motivated by the following observations:

Different logics: The quest for “a most suitable deontic logic” is still open, and it eventually will remain so for quite some time. I/O logics are favoured in our group, but we also study and implement alternative proposals, such as the DDL by Hansson or the one by Carmo and Jones. Eventually there is no single best solution, and different deontic logics qualify best in different application contexts.

Expressivity levels: It is highly questionable whether normative knowledge and regulatory texts can always be abstracted and simplified to the point that pure propositional logic encodings are feasible and justified. In
Sect. 6.2, for example, we outline the encoding of an ambitious ethical theory, Gewirth’s “Principle of Generic Consistency (PGC)” \[12\,\underline{55}\], in an extended higher-order DDL. It can actually be doubted that this complex ethical theory can be mechanized also in a propositional setting without trivialising it to a point of vacuity. The need for quantified deontic logics is also evidenced by related work such as Govindarajulu and Bringsjord’s encoding of the “Doctrine of Double Effect” \[56\].

- **Logic combinations**: In concrete applications normative aspects often meet with other challenges that can be addressed by suitable logic extensions, respectively, by combining a deontic logic with other suitable logics as required. An example is provided again by the encoding of the PGC as outlined in Sect. 6.2.

These issues should be addressed in empirical studies in which the different choices of logics, different expressivity levels and different logic combinations are systematically compared and assessed within well selected application studies. However, for such empirical work to be feasible, implementations of the different deontic candidate logics (and its combinations with other logics) have to be provided first, both on the propositional level and ideally also on the first-order and higher-order level. Moreover, it is reasonable to ensure that these implementations remain maximally comparable regarding the technological foundations they are based on, since this may improve the fairness and the significance of (conceptual) empirical evaluations.

Our framework is graphically depicted in Fig. 5. The framework supports experimentation with different normative theories, in different application scenarios, and it is not tied to a specific deontic logic. The enabling technology are higher-order theorem proving systems such as Isabelle/HOL or Leo-III via the shallow semantical embedding technique. The figure well illustrates the different components and aspects that we consider relevant in our work. Different target logics (grey circles) and their combinations are provided in the higher-order meta-logic of the host system. In our case this one is either the proof assistant Isabelle/HOL or the higher-order automated theorem prover Leo-III. In provided target logics different ethical or legal theories can then be encoded. Concrete examples are given in Sect. 6. This set-up enables a two-way fertilization: On the one hand, the studied theories are themselves assessed, e.g. for consistency, for entailed knowledge, etc. On the other hand the properties of the different target logics are investigated. For example, while one of the target logics might suffer from paradoxes in a concrete application context, another target logic might well be sufficiently stable against paradoxes in the same application context. An illustration of this aspect is given in Sect. 6.1. After arriving at consistent formalizations of an ethical or legal theory in a suitable logic or suitable combination of logics, empirical studies can be conducted. To this end the agent-based LeoPARD framework \[57\], which is underlying the Leo-III \[11\] prover, will be adapted and utilized in future work to e.g. embody upper ethical theories in virtual agents and to conduct empirical studies about

19
the agents behaviour when reasoning with such upper principles in a simulated environment.

6. Case studies

6.1. Data protection

A well-known problem in the study of deontic logic is the proper representation of contrary-to-duty structures, situations in which there is a primary obligation and what we might call a secondary or CTD (contrary-to-duty) obligation, which comes into effect when the primary obligation is violated. The paradox arises when we try to symbolise certain intuitively consistent sets of ordinary language sentences, sets that include at least one contrary-to-duty obligation sentence, by means of ordinary counterparts available in various monadic deontic logics, such as SDL and similar systems. The formal representations often turn out to be inconsistent, in the sense that it is possible to deduce contradictions from them, or else they might violate some other intuitively plausible condition, for example that the sentences in the formalization should be independent of each other. It is not the purpose of this article to discuss in any greater depth the paradox. The interested reader should consult, e.g., paper [28].

5The problem was first pointed out by Chisholm in relation with SDL.
We present below a CTD structure from the General Data Protection Regulation (GDPR, Regulation EU 2016/679). It is a regulation by which the European Parliament, the Council of the European Union and the European Commission intend to strengthen and unify data protection for all individuals within the European Union. The regulation became enforceable from 25 May 2018. Here is the CTD structure in question:

|   |   |
|---|---|
| 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). |
| 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. |

When combined with the following a typical CTD-structure is exhibited.

|   |   |
|---|---|
|   |   |
|   |   |

The latter information pieces are not explicit part of the GDPR. Instead they are to be seen as implicit. [3] comes from another regulation, with which the GDPR has to co-exists. [7] is a factual information—it is exactly the kind of world situations the GDPR wants to regulate.

Fig. 6.1 illustrates the problem raised by CTD scenarios, when the inference engine is based on SDL. The knowledge base is encoded in lines 7–25. The relevant obligations and the assumed situation as described by (1)–(4) above are formalized in lines 19–25. Subsequently, three different kinds of queries are answered by the reasoning tools integrated with Isabelle/HOL. The first query asks whether the encoded knowledge base is consistent, and Nitpick answers negatively to this question in line 28. The failed attempt to compute a model is highlighted in pink. The second query asks whether falsum is derivable, and Isabelle’s prover metis returns a proof within a few milliseconds in line 29. Notice that the proof depends on the seriality axiom $D$ of SDL, which is imported from SDL’s encoding in file SDL.thy—that is not shown here. The query in line 29 asks whether an arbitrarily weird and unethical conclusion such as the obligation to kill Mary follows, and the prover answers positively to this query.

These results are clearly not desirable, and confirm the need to use a logic other than SDL for application scenarios in which norm violation play a key role.

Fig. 7 shows that our SSE based implementation of system $E$ is in contrast
theory GDPR_SDL
   (*GDPR CTD Example. C. Benzmüller & Xavier Parent, 2019*)
   imports SDL
   begin

   datatype data = d1 | d2 (*We exemplarily introduce concrete data objects d1 and d2.*)
   datatype indiv = Mary | Peter (*We exemplarily introduce individuals Mary and Peter.*)

   consts process_lawfully :: "data => 'a" erase :: 'a => 'a is_protected_by_GDPR :: 'a => bool belongs_to :: "data => indiv => 'a => bool"

   axiomatization where

   (*Data belonging to Europeans is protected by the GDPR.*)
   A0: "\forall x. \forall d. (is_european x \land belongs_to d x) \rightarrow is_protected_by_GDPR d" and
   (*Data d1 is belonging to the European Peter.*)
   F1: "\exists d1. belongs_to d1 Peter \land is_european Peter" and

   (*It is an obligation to process data lawfully.*)
   A1: "\forall d. is_protected_by_GDPR d \rightarrow process_lawfully d" and
   (*If data was not processed lawfully, then it is an obligation to erase the data.*)
   A2: "\forall d. (is_protected_by_GDPR d \land \neg process_lawfully d) \rightarrow \neg erase d" and
   (*An implicit: It is an obligation to keep the data if it was processed lawfully.*)
   A3: "\forall d. (\neg is_protected_by_GDPR d \land process_lawfully d) \rightarrow \neg erase d" and
   (*Given a situation where data is processed unlawfully.*)
   A4: "\neg process_lawfully d1" and

   (**Some Experiments**)
   lemma True nitpick [satisfy] oops (*Consistency-check fails; Nitpick finds no model.*)
   lemma "False" by (metis A0 A1 A2 F1 Implicit Situation) (*Prove of Falsum.*)

   (*Should the data be erased? — Yes, proof found by ATPs*)
   lemma "\neg erase d1" by (metis A0 A2 F1 Situation)
   (*Should the data be kept? — Yes, proof found by ATPs*)
   lemma "\neg \neg erase d1" by (metis A0 A1 F1 Implicit)
   (*Should Mary be killed? — Yes, proof found by ATPs*)
   lemma "\neg kill Mary" by (metis A0 A1 A2 F1 Implicit Situation)

   Figure 6: Failed analysis of the GDPR example in SDL.
not suffering from this effect. The prescriptive rules of the GDPR scenario are modelled in lines 18–22:

\[
\begin{align*}
A_1: & \forall d \; \bigcirc (\text{process lawfully } d \land \text{is protected by GDPR } d) \\
A_2: & \forall d \; \bigcirc (\text{erase } d \land \text{is protected by GDPR } d \land \neg \text{process lawfully } d) \\
\text{Implicit}: & \forall d \; \bigcirc (\neg \text{erase } d \land \text{is protected by GDPR } d \land \text{process lawfully } d)
\end{align*}
\]

The current situation, in which we have that Peter’s personal data \( d_1 \) are not processed lawfully, is defined in line 25:

\[
\text{Situation}: \neg \text{process lawfully } d_1
\]

The three same queries as before are run, but this time with success. In line 28 we are told that the knowledge base has a model. The computed model can be inspected in full detail a separate window. In line 29 we are told that falsum is no longer derivable. In line 30 we are told that the obligation to kill Mary no longer follows.

### 6.2. Gewirth’s Principle of Generic Consistency

In the papers [16, 59], our flexible reasoning infrastructure has been adapted and utilized for the mechanization and assessment of a challenging explicit ethical theory on the computer. This work pushes existing boundaries in knowledge representation and reasoning and it demonstrates that intuitive encodings of ambitious ethical theories and their mechanization, resp. automation, on the computer are no longer antipodes.

The exemplarily studied explicit theory in normative ethics is Alan Gewirth’s “Principle of Generic Consistency (PGC)” [12, 55], which he proposed as an emendation of the *Golden Rule*, i.e., the principle of treating others as one’s self would wish to be treated. Gewirth’s PGC aims at justifying a closely related upper moral principle, according to which any intelligent agent, by virtue of its self-understanding as an agent, is rationally committed to asserting that (i) it has rights to freedom and well-being, and (ii) all other agents have those same rights. The argument used by Gewirth to derive the PGC is by no means trivial and has stirred much controversy in legal and moral philosophy during the last decades. It has also been discussed in political philosophy as an argument for the a priori necessity of human rights. Perhaps more relevant for us, the PGC has lately been proposed as a means to bound the impact of artificial general intelligence [60].

Gewirth’s PGC has been taken an illustrative showcase in [16] to exemplarily assess its logical validity with our flexible deontic logic reasoning machinery. The success of this work provides evidence for the claims made in the present article. It is the first time that the PGC has been formalized and assessed on the computer at such a level of detail (i.e. without trivializing it by abstraction means) and while upholding intuitive formula representations and user-interaction means. And as a side-effect, the automated theorem provers integrated with Isabelle/HOL have helped in this work to reveal and fix some (minor) issues in Gewirth’s argument.
Figure 7: Successful analysis of the GDPR example scenario in system E.
To encode the PGC an extended embedding of the DDL by Carmo and Jones [28] in HOL was employed. Conditional obligation has been combined in [16], among others, with further modalities, with Kaplan’s notion of context-dependent logical validity, and with both first-order and higher-order quantifiers. It was then verified that the combined logic is still immune to known paradoxes in deontic logic.

Figure 8 provides a good impression of this work, which we cannot discuss here in full detail due to space restrictions. What can be seen though without further background knowledge is that readable formula presentations are supported, which in turn enable quite intuitive user-interactions in combination with proof automation means. It is demonstrated here that the interactive assessment of Gewirth’s argument is supported at an adequate level of granularity: the proof steps as presented in Fig. 8 correspond to actual steps in the Gewirth’s argument and further details in the verification process can be delegated to automated reasoning tools provided in Isabelle/HOL. Proof automation can of course be further pushed, but this has not been the focus so far in this work.

7. Related work

7.1. Machine ethics and deontic logic

The questions how transparency, explainability, and verifiability can best be achieved in future intelligent systems and whether bottom-up or top-down architectures should be preferred are discussed in a range of related articles; see e.g. [61, 62, 63, 64, 65, 66, 67] and the references therein. For example, Dennis et al. [67] make a compelling case for the use of so-called formal verification—a well-established technique for proving correctness of computer systems—in the area of machine ethics. The idea is to use formal methods to check whether specific ethical rules of behavior (e.g., the Rules of the Air) have been observed by the autonomous system (for, e.g., an unmanned civilian aircraft) when making decisions. An ethical rule is represented as a formula of the form “do(a) ⇒ e ¬Eφ”, denoting that doing action a counts as a violation of ethical principle φ. However, they do not specify in full the syntax and semantics of their operator E. It may be valuable to further explore the relationship between this work and the approach outlined in the present paper. As the authors observe, on p. 6 of their paper, the E modal operator resembles the obligation operator ◦ used in deontic logic.

Further related work includes a range of implemented theorem proving systems. A lean but powerful connection-based theorem prover for first-order modal logics, covering also SDL, has been developed by Otten [68]. A tableaux-based propositional reasoner is employed in the work of Furbach and Schon [68, 69] and first-order resolution methods for modal logics have been contributed.
Figure 8: Mechanization of Gewirth's argument in Isabelle/HOL.
Further related work includes a reasoner for propositional defeasible modal logic by Governatori and his team \[71\]. Their reasoner supports defeasible reasoning, but it is less flexible than ours, because it does not allow the user to easily switch between different systems of normative reasoning and explore their properties. A reasoner for expressive contextual deontic reasoning was proposed by Bringsjord et al. \[72\]. Pereira and Saptawijaya \[73, 74, 75\] present a solution that implements deontic and counterfactual reasoning in Prolog.

We are not aware of any attempts to automate, within a single framework, such a wide portfolio of contrary-to-duty resistant propositional, first-order and higher-order deontic logics as we report it in this article. Note that in addition to the features of the above related systems our solution also supports intuitive user-interaction and most flexible logic combinations, e.g., with the quantified conditional logics from \[51, 50\].

The SSE approach has also been implemented in the Leo-III theorem prover, so that the prover now provides native language support for a wide range of modal logics and for DDL \[14\]. A recent, independent study \[76\] shows that Leo-III, due to its wide range of directly supported logics, has become the most powerful and most widely applicable automated theorem prover existent to date.\[7\]

The flexibility of the SSE approach has been exploited and demonstrated in particular in the case study on Gewirth’s PGC that we have presented in Sect. 6.2. Impressive related work on the mechanization of ambitious ethical theories includes the already mentioned automation of the “Doctrine of Double Effect” by Govindarajulu and Bringsjord \[50\].

### 7.2. Universal logical reasoning

Related experiments with the universal (meta-)logical reasoning approach have been conducted in metaphysics \[77, 78\]. An initial focus thereby has been on computer-supported assessments of rational arguments, in particular, of modern, modal logic variants of the ontological argument for the existence of God. In the course of these experiments, in which the SSE approach was applied for automating different variants of higher-order quantified modal logics, the theorem prover LEO-II even detected an previously unnoticed inconsistency in Gödel’s \[29\] modal variant of the ontological argument, while the soundness of the emended variant by Scott \[80\] was confirmed and all argument steps were verified. Further modern variants of the ontological argument have subsequently been studied with the approach, and theorem provers have even contributed to the clarification of an unsettled philosophical dispute \[81\]. The good performance of the SSE approach in previous work has been a core motivation for the new application direction addressed here. In previous work, Benzmüller and

---

\[7\] The assessment has included various variants of classical first-order and higher-order logic benchmark problems. First-order and higher-order deontic logics and other non-classical logics were still excluded though. Their inclusion would clearly further benefit the Leo-III prover.
colleagues also studied actualist quantifiers [50, 51], and it should be possible to transfer these ideas to our setting.

Another advantage of the SSE approach, when implemented within powerful proof assistants such as Isabelle/HOL, is that proof construction (interactive or automated) can be supported at different levels of abstraction. For this note that proof protocols/objects may generally serve two different purposes: (a) they may provide an independently verifyable explanation in a (typically) well-defined logical calculus, or (b) they may provide an intuitive explanation to the user why the problem in question has been answered positively or negatively. Many reasoning tools, if they are offering proof objects at all, do generate only objects of type (a). The SSE approach, however, has already demonstrated its capabilities to provide both types of responses simultaneously in even most challenging logic settings. For example, a quite powerful, abstract level theorem prover for hyper-intensional higher-order modal logic has been provided by Kirchner [82, 83]. He encoded, using abstraction layers, a proof calculus for this very complex logic as proof tactics and he demonstrated how these abstract level proof tactics can again be automated using respective tools in Isabelle/HOL. Kirchner [82, 83] then successfully applied his reasoning infrastructure to reveal, assess and intuitively communicate a non-trivial paradox in Zalta’s “Principia-logico Metaphysica” [84].

Drawing on the results and experiences from previous work, the ambition of our ongoing project is to further extend the already existing portfolio of deontic logics in Isabelle/HOL towards a most powerful, flexible and scalable deontic logic reasoning infrastructure. A core motivation thereby is to support empirical studies in various application scenarios, and to assess and compare the suitability, adequacy and performance of individual deontic logic solutions for the engineering of moral agents and explainable intelligent systems. It is relevant to mention that proof automation in Isabelle/HOL, and also in related higher-order ATPs such as Leo-III [11], is currently improving at good pace. These developments are fostered in particular by recently funded research and infrastructure projects.

8

8. Further research

In future work, the range of normative theories must be extended, and the currently represented theories must be further optimized. In particular, a wider range of explicit ethical theories must be studied. We have made historical and current developments in normative reasoning practically accessible for the use in machine ethics. We showed how our approach can support research in normative reasoning itself. The use of computer-assisted exploration and assessment of new deontic logics provides immediate feedback from systems to property

Prominent example projects include Matryoshka (http://matryoshka.gforge.inria.fr) and ALEXANDRIA (http://www.cl.cam.ac.uk/~lp15/Grants/Alexandria/).
checks. This is particularly valuable for unifying and combining different logics and for experimental studies. For example, since our approach supports meta-logical investigations, conjectured relationships between I/O logics and conditional logics can be formally assessed in future work.

Moreover, our approach can also be used for other relevant purposes. In education, for example, the different logics discussed in this article can now be integrated in computer-supported classroom education. First reassuring results have been obtained in lecture courses at University of Luxembourg (Intelligent Agents II) and FU Berlin (Universal Logical Reasoning). Students start exploring existing deontic logics without the need for a deep a priori understanding of them. They can play with the logics in small or larger examples, modify them, assess their modifications, etc.

In agent simulation, we plan the embodiment/implementation of explicit ethical theories in simulated agents. We can then investigate properties for single agents, but beyond that also study agent interaction and the behaviour of the agent society as a whole. To this end the agent-based LeoPARD framework [57], which is underlying the Leo-III prover, will be adapted and utilized in future work.

9. Conclusion

We propose higher-order logic as a uniform and highly expressive formal framework to represent and experiment with normative theories of ethical and legal reasoning. To some researchers, this may seem paradoxical for two reasons. First of all, we do no longer aim for a unique and standard deontic logic which can be used for all applications, but we do propose to use higher-order logic as a unique and formal framework to represent normative theories. So what exactly is the difference between a unique deontic logic and a unique formal framework? The second apparent paradox is that we propose higher-order logic for tool support, whereas it is well-known that higher-order logic is undecidable.

These two apparent paradoxes can be explained away by our methodology of representing normative theories in higher-order logic. There are many ways in which a normative theory can be represented in higher-order logic, and only a few of them will be such that the formal methods of the tool support can be applied to them. Therefore, the representation of deontic logics in higher-order logic is an art, and each new representation has to come with a proof that the embedding is faithful. These proofs play a similar role in our formal framework as soundness and completeness proofs play in most of the traditional work of deontic logic.

It is the availability of powerful systems such as Isabelle/HOL or Leo-III for tool support that allows our approach to revolutionize the field of formal ethics. For example, we were invited for a keynote talk at the International Conference on Deontic Logic and Normative systems at 2018 in Utrecht, where we also received the best paper award for a technical contribution to a faithfulness proof of one of the embeddings. Though the use of higher-order logic may come as a paradigm shift to the field of ethical reasoning, it is an insight which is
already well established in the area of formal deduction. Whereas it is far from straightforward to represent deontic logics in higher-order logic, once a deontic logic has been represented, it becomes much easier to make small changes to them and see the effect of these changes. And this is exactly how our approach supports the design of normative theories of ethical and legal reasoning. It is in the ease in which the user can play with and adapt existing theories, how the design of normative theories is made accessible to non-specialist users and developers.

To validate our approach we have embedded the main strands of current deontic logic within higher-order logic, and we have experimented with the approach over the past two years.

Our normative reasoning infrastructure supports empirical studies on legal and ethical theories in which the particular deontic logic formalisms itself can be varied, assessed and compared in context. This infrastructure can fruitfully support the development of much needed logic based approaches towards ethical agency. The solution we have presented supports a wide range of specific deontic logic variants, and it also scales for their first-order and higher-order extensions.

The use of tool support for ethical and legal reasoning is not only fruitful to develop new normative theories, it is now being employed also in teaching, and we plan to use it as a formal framework for simulation. Another promising application is the use of our approach for the study of deontic modality in natural language processing. In linguistics the use of higher-order logic is already adopted for the semantics of natural language, and we believe that our framework can also support studies of the pragmatic aspects of the use of deontic modality.

Acknowledgments: We thank Ali Farjami, David Fuenmayor, Tobias Gleißner, Alexander Steen and several further colleagues for their contributions to this project.

References

[1] B. Liao, M. Slavkovik, L. van der Torre, Building Jiminy Cricket: An architecture for moral agreements among stakeholders, in: AAAI/ACM Conference on Artificial Intelligence, Ethics and Society, URL (preprint): https://arxiv.org/pdf/1812.04741.pdf, 2019, pp. 1–15, forthcoming.

[2] P. M. Dung, On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games, Artificial Intelligence 77 (2) (1995) 321–357.

[3] B. F. Malle, Integrating robot ethics and machine morality: The study and design of moral competence in robots, Ethics and Information Technology 18 (4) (2016) 243–256.
[4] B. F. Malle, M. Scheutz, J. L. Austerweil, Networks of social and moral norms in human and robot agents, in: M. I. Aldinhas Ferreira, J. Silva Sequeira, M. O. Tokhi, E. E. Kadar, G. S. Virk (Eds.), A World with Robots: International Conference on Robot Ethics: ICRE 2015, Springer International Publishing, Cham, 2017, pp. 3–17.

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

[6] A. Chopra, L. van der Torre, H. Verhagen (Eds.), Handbook of Normative Multi-Agent Systems, College Publications, London, UK, 2018.

[7] X. Parent, L. van der Torre, Introduction to Deontic Logic and Normative Systems, College Publications, London, UK, 2018.

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

[9] T. Nipkow, L. Paulson, M. Wenzel, Isabelle/HOL: A Proof Assistant for Higher-Order Logic, Vol. 2283 of Lecture Notes in Computer Science, Springer, 2002.

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

[11] A. Steen, C. Benzmüller, The higher-order prover Leo-III, in: D. Galmiche, S. Schulz, R. Sebastiani (Eds.), Automated Reasoning, IJCAR 2018, Vol. 10900 of Lecture Notes in Computer Science, Springer, Cham, 2018, pp. 108–116. [doi:10.1007/978-3-319-94205-6_8]

[12] A. Gewirth, Reason and morality, University of Chicago Press, 1981.

[13] A. Farjami, P. Meder, X. Parent, C. Benzmüller, I/O logic in HOL, to appear in Journal of Applied Logics – IfCoLoG Journal of Logics and their Applications, Special issue MIREL 2018 workshop on MLining and REasoning with Legal texts, URL (preprint): [http://orbilu.uni.lu/handle/10993/37013](http://orbilu.uni.lu/handle/10993/37013) (2019).

[14] T. Gleißner, A. Steen, C. Benzmüller, Theorem provers for every normal modal logic, in: T. Eiter, D. Sands (Eds.), LPAR-21. 21st International Conference on Logic for Programming, Artificial Intelligence and Reasoning, Vol. 46 of EPiC Series in Computing, EasyChair, Maun, Botswana, 2017, pp. 14–30.

[15] P. Meder, Deontic agency and moral luck, Master’s thesis, Faculty of Science, Technology and Communication, University of Luxembourg (2018).
[16] D. Fuenmayor, C. Benzmüller, Formalisation and evaluation of Alan Gewirth’s proof for the Principle of Generic Consistency in Isabelle/HOL, Archive of Formal Proofs (2018) 1–13. URL http://isa-afp.org/entries/GewirthPGCProof.html

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

[18] J. C. Blanchette, S. Böhme, L. C. Paulson, Extending Sledgehammer with SMT solvers, Journal of Automated Reasoning 51 (1) (2013) 109–128.

[19] J. Blanchette, T. Nipkow, Nitpick: A counterexample generator for higher-order logic based on a relational model finder, in: Proc. of ITP 2010, Vol. 6172 of Lecture Notes in Computer Science, Springer, 2010, pp. 131–146.

[20] H. Ohlbach, A. Nonnengart, M. de Rijke, D. Gabbay, Encoding two-valued nonclassical logics in classical logic, in: J. Robinson, A. Voronkov (Eds.), Handbook of Automated Reasoning (in 2 volumes), Elsevier and MIT Press, 2001, pp. 1403–1486.

[21] G. Frege, Begriffsschrift. Eine der arithmetischen nachgebildete Formel-sprache des reinen Denkens, Halle, 1879.

[22] P. Andrews, Church’s type theory, in: E. N. Zalta (Ed.), The Stanford Encyclopedia of Philosophy, summer 2018 Edition, Metaphysics Research Lab, Stanford University, 2018, pp. 1–38 (in pdf version).

[23] C. Benzmüller, C. Brown, M. Kohlhase, Higher-order semantics and exten-sionality, Journal of Symbolic Logic 69 (4) (2004) 1027–1088.

[24] C. Benzmüller, D. Miller, Automation of higher-order logic, in: D. M. Gabbay, J. H. Siekmann, J. Woods (Eds.), Handbook of the History of Logic, Volume 9 — Computational Logic, North Holland, Elsevier, 2014, pp. 215–254.

[25] P. B. Andrews, General models, descriptions, and choice in type theory, Journal of Symbolic Logic 37 (2) (1972) 385–394.

[26] L. Henkin, Completeness in the theory of types, Journal of Symbolic Logic 15 (2) (1950) 81–91.

[27] G. H. von Wright, Deontic logic, Mind 60 (1951) 1–15.

[28] J. Carmo, A. J. I. Jones, Deontic logic and contrary-to-duties, in: D. M. Gabbay, F. Guenthner (Eds.), Handbook of Philosophical Logic: Volume 8, Springer Netherlands, Dordrecht, 2002, pp. 265–343.

[29] B. Hansson, An analysis of some deontic logics, Noûs 3 (4) (1969) 373–398.
[30] L. Åqvist, Deontic logic, in: D. Gabbay, F. Guenthner (Eds.), Handbook of Philosophical Logic, 2nd Edition, Vol. 8, Kluwer Academic Publishers, Dordrecht, Holland, 2002, pp. 147–264.

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

[32] A. Kratzer, Modals and Conditionals: New and Revised Perspectives, Oxford University Press, New York, 2012.

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

[34] C. Alchourrón, Philosophical foundations of deontic logic and the logic of defeasible conditionals, in: J.-J. Meyer, R. Wieringa (Eds.), Deontic Logic in Computer Science, John Wiley & Sons, Inc., New York, 1993, pp. 43–84.

[35] J. Horty, Agency and Deontic Logic, Oxford University Press, London, UK, 2009.

[36] J. Hansen, Reasoning about permission and obligation, in: S. O. Hansson (Ed.), David Makinson on Classical Methods for Non-Classical Problems, Springer, Dordrecht, 2014, pp. 287–333.

[37] X. Parent, L. van der Torre, Detachment in normative systems: Examples, inference patterns, properties., Journal of Applied Logics — IfCoLog Journal of Logics and their Applications 4 (2017) 2996–3039.

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

[39] J. Horty, Reasons as Defaults, Oxford University Press, 2012.

[40] J. Hansen, Prioritized conditional imperatives: problems and a new proposal, Journal of Autonomous Agents and Multi-Agent Systems 17 (1) (2008) 11–35.

[41] C. Straßer, M. Beirlaen, F. Van De Putte, Adaptive logic characterizations of input/output logic, Studia Logica 104 (5) (2016) 869–916.

[42] C. Straßer, Adaptive Logics for Defeasible Reasoning: Applications in Argumentation, Normative Reasoning and Default Reasoning, Springer, 2013.

[43] T. Gordon, The Pleading Game: an Artificial Intelligence Model of Procedural Approach, Springer, New York, 1995.

[44] G. Sartor, Legal Reasoning: A Cognitive Approach to Law, Springer, 2005.
C. Benzmüller, A. Farjami, X. Parent, Aqvist’s dyadic deontic logic E in HOL, to appear in Journal of Applied Logics — IfCoLoG Journal of Logics and their Applications, Url (preprint): http://orbilu.uni.lu/handle/10993/37014 (2019).

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

C. Benzmüller, X. Parent, I/O logic in HOL — first steps, Tech. rep., CoRR, https://arxiv.org/abs/1803.09681 (2018).

X. Parent, L. W. N. van der Torre, The pragmatic oddity in norm-based deontic logics, in: J. Keppens, G. Governatori (Eds.), Proceedings of the 16th edition of the International Conference on Artificial Intelligence and Law, ICAIL 2017, London, United Kingdom, June 12-16, 2017, ACM, 2017, pp. 169–178.

X. Parent, L. W. N. van der Torre, I/O logics with a consistency check, in: J. M. Broersen, C. Condoravdi, N. Shyam, G. Pigozzi (Eds.), Deontic Logic and Normative Systems — 14th International Conference, DEON 2018, Utrecht, The Netherlands, July 3-6, 2018., College Publications, 2018, pp. 285–299.

C. Benzmüller, Automating quantified conditional logics in HOL, in: F. Rossi (Ed.), IJCAI-13, AAAI Press, Beijing, China, 2013, pp. 746–753.

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

X. Parent, Maximality vs. optimality in dyadic deontic logic, Journal of Philosophical Logic 43 (6) (2014) 1101–1128.

X. Parent, Preference-based semantics for dyadic deontic logics in Hansson’s tradition: a survey of axiomatisation results, in D. Gabbay, J. Hory, X. Parent, R. van der Meyden and L. van der Torre (eds.), Handbook of Deontic Logic and Normative Systems, volume 2, College Publications, London, UK (to appear).

D. Lewis, Counterfactuals, Blackwell, Oxford, 1973.

D. Beyleveld, The dialectical necessity of morality: An analysis and defense of Alan Gewirth’s argument to the principle of generic consistency, University of Chicago Press, 1991.

N. S. Govindarajulu, S. Bringsjord, On automating the doctrine of double effect, in: IJCAI-17, 2017, pp. 4722–4730.

34
[57] M. Wisniewski, A. Steen, C. Benzmüller, LeoPARD — A generic platform for the implementation of higher-order reasoners, in: M. Kerber, J. Carette, C. Kaliszyk, F. Rabe, V. Sorge (Eds.), Intelligent Computer Mathematics - International Conference, CICM 2015, Washington, DC, USA, July 13-17, 2015, Proceedings, Vol. 9150 of Lecture Notes in Computer Science, Springer, 2015, pp. 325–330.

[58] R. Chisholm, Contrary-to-duty imperatives and deontic logic, Analysis 24 (1963) 33–36.

[59] D. Fuenmayor, C. Benzmüller, Automated reasoning with complex ethical theories – a case study towards responsible AI, in: 16th International Congress on Logic, Methodology and Philosophy of Science and Technology, Prague, Czechia, August 5-10, 2019, to appear, accepted symposium presentation.

[60] A. Kornai, Bounding the impact of AGI, Journal of Experimental & Theoretical Artificial Intelligence 26 (3) (2014) 417–438.

[61] V. Dignum (Ed.), Special issue: Ethics and artificial intelligence, Ethics and Information Technology 20 (1).

[62] A. F. T. Winfield, M. Jirotka, Ethical governance is essential to building trust in robotics and artificial intelligence systems, Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences 376 (2133) (2018) 20180085.

[63] V. Dignum, Responsible autonomy, in: IJCAI-17, 2017, pp. 4698–4704.

[64] M. Scheutz, The case for explicit ethical agents, AI Magazine 38 (2017) 57–64.

[65] M. Anderson, S. L. Anderson, Toward ensuring ethical behavior from autonomous systems: a case-supported principle-based paradigm, Industrial Robot 42 (4) (2015) 324–331.

[66] W. Wallach, C. Allen, I. Smit, Machine morality: bottom-up and top-down approaches for modelling human moral faculties, AI & Society 22 (4) (2008) 565–582.

[67] L. A. Dennis, M. Fisher, M. Slavkovik, M. Webster, Formal verification of ethical choices in autonomous systems, Robotics and Autonomous Systems 77 (2016) 1–14.

[68] J. Otten, Non-clausal connection calculi for non-classical logics, in: TABLEAUX, Vol. 10501 of Lecture Notes in Computer Science, Springer, 2017, pp. 209–227.
[69] U. Furbach, C. Schon, Deontic logic for human reasoning, in: T. Eiter, H. Strass, M. Truszczynski, S. Woltran (Eds.), Advances in Knowledge Representation, Logic Programming, and Abstract Argumentation, Vol. 9060 of Lecture Notes in Computer Science, Springer, 2015, pp. 63–80.

[70] R. A. Schmidt, U. Hustadt, First-order resolution methods for modal logics, in: Programming Logics, Vol. 7797 of Lecture Notes in Computer Science, Springer, 2013, pp. 345–391.

[71] E. Kontopoulos, N. Bassiliades, G. Governatori, G. Antoniou, A modal defeasible reasoner of deontic logic for the semantic web, International Journal on Semantic Web Information Systems 7 (1) (2011) 18–43.

[72] S. Bringsjord, N. Sundar G., B. F. Malle, M. Scheutz, Contextual deontic cognitive event calculi for ethically correct robots, in: International Symposium on Artificial Intelligence and Mathematics, ISAIM 2018, Fort Lauderdale, Florida, USA, January 3-5, 2018, 2018, p. (3 pages).

[73] L. M. Pereira, A. Saptawijaya, Programming Machine Ethics, Vol. 26 of Studies in Applied Philosophy, Epistemology and Rational Ethics, Springer, 2016.

[74] A. Saptawijaya, L. M. Pereira, Logic programming for modeling morality, Logic Journal of the IGPL 24 (4) (2016) 510–525.

[75] L. M. Pereira, A. Saptawijaya, Counterfactuals, logic programming and agent morality, in: R. Urbaniaik, G. Payette (Eds.), Applied Formal/Mathematical Philosophy, Logic, Argumentation & Reasoning, Springer, 2017.

[76] C. E. Brown, T. Gauthier, C. Kaliszyk, G. Sutcliffe, J. Urban, GRUNGE: A Grand Unified ATP Challenge, arXiv e-prints (2019) arXiv:1903.02539.

[77] C. Benzmüller, B. Woltzenlogel Paleo, The inconsistency in Gödel’s ontological argument: A success story for AI in metaphysics, in: S. Kambhampati (Ed.), IJCAI-16, Vol. 1-3, AAAI Press, 2016, pp. 936–942.

[78] C. Benzmüller, B. Woltzenlogel Paleo, Automating Gödel’s ontological proof of God’s existence with higher-order automated theorem provers, in: T. Schaub, G. Friedrich, B. O’Sullivan (Eds.), ECAI 2014, Vol. 263 of Frontiers in Artificial Intelligence and Applications, IOS Press, 2014, pp. 93 – 98.

[79] K. Gödel, Appendix A: Notes in Kurt Gödel’s hand, in: J. Sobel (Ed.), Logic and Theism, Cambridge University Press, 1970, pp. 144–145.

[80] D. Scott, Appendix B: Notes in Dana Scott’s hand, in: J. Sobel (Ed.), Logic and Theism, Cambridge University Press, 1972, pp. 145–146.

[81] C. Benzmüller, L. Weber, B. Woltzenlogel Paleo, Computer-assisted analysis of the Anderson-Hájek controversy, Logica Universalis 11 (1) (2017) 139–151.
[82] D. Kirchner, C. Benzmüller, E. N. Zalta, Mechanizing principia logico-metaphysica in functional type theory, CoRR, \url{https://arxiv.org/abs/1711.06542}.

[83] D. Kirchner, C. Benzmüller, E. N. Zalta, Computer science and metaphysics: A cross-fertilization, preprint of submitted article.

[84] E. N. Zalta, Principia logico-metaphysica, draft version, preprint available at \url{https://mally.stanford.edu/principia.pdf} (2018).