Article Title

LogiKEy Workbench: Deontic Logics, Logic Combinations and Expressive Ethical and Legal Reasoning (Isabelle/HOL Dataset)

Authors

Christoph Benzmüller[2,1], Ali Farjami[1], David Fuenmayor[2], Paul Meder[1], Xavier Parent[1], Alexander Steen[1], Leendert van der Torre[1,3], Valeria Zahoransky[4]

Affiliations

[1]University of Luxembourg, Esch sur Alzette, Luxembourg
[2]Freie Universität Berlin, Berlin, Germany
[3]Zhejiang University, Hangzhou, China
[4]University of Oxford, Oxford, UK

Corresponding author(s)

Christoph Benzmüller (`c.benzmueller@fu-berlin.de`)
Xavier Parent (`x.parent.xavier@gmail.com`)

Abstract

The LogiKEy workbench and dataset for ethical and legal reasoning is presented. This workbench simultaneously supports development, experimentation, assessment and deployment of formal logics and ethical and legal theories at different conceptual layers. More concretely, it comprises, in form of a data set (Isabelle/HOL theory files), formal encodings of multiple deontic logics, logic combinations, deontic paradoxes and normative theories in the higher-order proof assistant system Isabelle/HOL. The data was acquired through application of the LogiKEy methodology, which supports experimentation with different normative theories, in different application scenarios, and which is not tied to specific logics or logic combinations. Our workbench consolidates related research contributions of the authors and it may serve as a starting point for further studies and experiments in flexible and expressive ethical and legal reasoning. It may also support hands-on teaching of non-trivial logic formalisms in lecture courses and tutorials.

The LogiKEy methodology and framework is discussed in more detail in the companion research article titled "Designing Normative Theories for Ethical and Legal Reasoning: LogiKEy Formal Framework, Methodology, and Tool Support" [5].

Keywords

Trustworthy and responsible AI; Knowledge representation and reasoning; Automated theorem proving; Model finding; Normative reasoning; Normative systems; Semantical embedding; Higher-order logic

Specifications Table

Subject	Computer Science
Specific subject area	Artificial intelligence; Knowledge representation and reasoning; Normative reasoning
Type of data	formal theories (.thy files) encoded in Isabelle/HOL syntax, readable (.png or .pdf) views of this data
How data was acquired	The data was acquired through manual encoding of various deontic logics, logic combinations, examples of contrary-to-duty paradoxes, excerpts of legal texts and exemplary ethical theories utilizing the LogiKEy methodology [5], which is based on shallow semantical embeddings (SSEs) of logics and theories in classical higher-order logic. The concrete encodings were conducted in the higher-order proof assistant system Isabelle/HOL (`https://isabelle.in.tum.de`); however, they are conceptually transferable to other expressive reasoning systems.
Data format	Raw, processed, analyzed and cleaned data. The data is provided in the syntax format of the Isabelle/HOL proof assistant, which has been used to process, analyze and verify it; the data files were also annotated by hand. Isabelle/HOL is freely available: `https://isabelle.in.tum.de`
Parameters for data collection	One objective was to empirically assess the expressiveness and proof automation capabilities of Isabelle/HOL and its integrated tools in normative reasoning when utilizing the LogiKEy methodology and the SSE approach. Another objective was to provide a reusable foundation for further experiments in expressive ethical and legal reasoning.
Description of data collection	The data were written by hand. As part of the data collection process it has been demonstrated that non-trivial, normative reasoning is supported in the provided framework. This in particular included studies of paradoxes in normative reasoning [6] and whether and how they can eventually be avoided. An integral aspect of the data collection process also has been to provide evidence for the practical normative reasoning performance of the various reasoning tools integrated with Isabelle/HOL when utilizing the LogiKEy approach. Useful comments were added to the data files. The practical performance of the logic encodings can be independently assessed by users in combination with the Isabelle/HOL system. It has also been demonstrated how deontic logics can be flexible combined with other logic formalisms within the LogiKEy approach.
Data source location	The data is hosted on `github.com`.
Data accessibility	The data is accessible via `logikey.org`, which redirects to the repository `https://github.com/cbenzmueller/LogiKEy` on `github.com`, where the data is hosted and maintained. The two subdirectories `2020-DataInBrief-Article` and `2020-DataInBrief-Data` are associated with this article; the latter contains the data set.
Related research article	C. Benzmüller, X. Parent, and L. van der Torre. Designing normative theories for ethical and legal reasoning: LogiKEy framework, methodology, and tool support. Artificial Intelligence, 287(103348) 2020. Doi: 10.1016/j.artint.2020.103348.

Value of the Data

- The provided data can be reused, independent of the related research article(s), as a starting point for further studies and experiments in expressive ethical and legal reasoning. Moreover, it can be reused, extended and adapted to support also other various other application directions, including, e.g., the study of deontic modality and quantifiers in linguistics.
- The data collection is beneficial for research and application in a range of areas, including but not limited to: machine ethics (ethico-legal governor systems), explainable and trustworthy AI, regulatory technologies, argumentation, natural language semantics. To that end the data includes reusable SSEs of a portfolio

of deontic logics, logic combinations, paradoxes in normative reasoning and ethical theories in classical higher-order logic (HOL), interpretable in the Isabelle/HOL proof assistant system. The data set may also be used to support the teaching of expressive, classical and non-classical logic formalisms and their combinations in lecture courses and tutorials.

- To reuse the data interested researchers, students and practitioners only need to download the provided data files, include them in their formalization projects and suitably extend or adapt them. For example, the contributed data includes a sample encoding of selected statements from the GDPR (General Data Protection Regulation) and an encoding of Gewirth's ethical argument and principle, known as the Principle of Generic Consistency (PGC) [12], in a suitable extension of higher-order deontic logic. These are two examples in the area of knowledge representation and reasoning with an emphasis on regulatory and ethical aspects. They can be reused as a starting point for the encoding and automated solution of similar ethico-legal theories.

- The data set advances the state of the art in deontic logic [11] as follows. Sixty years after Von Wright's invention of deontic logic, the question has always been how deontic logics and normative theories can be used in computer science applications. The LogiKEy workbench and associated methodology addresses this challenge; it has the potential to revolutionize the area of deontic logic itself.

- The data set is useful also for stimulation of cross-fertilization effects between different research communities including the deontic logics and normative reasoning communities, the area of higher-order logics, and the area of interactive and automated theorem proving with its various sub-communities targeting very different logic formalisms.

- The presented encodings put a particular emphasis on the modeling of (regulative) norms. We agree with, e.g., Jones and Sergot [14] that deontic logic is needed when it is necessary to make explicit, and then reason about, the distinction between what ought to be the case and what is the case. Furthermore, the adequate handling of the deontic paradoxes (like in particular Chisholm's paradox of contrary-to-duty (CTD) obligation, which deals with norm violation) posed a core challenge for knowledge representation frameworks. This problem motivated the design of deontic logics (and logic combinations) more sophisticated and finer-grained than the traditional ones, like modal logic. These frameworks are automatized for the first time. It is demonstrated that a computer or a machine can reason about norm violation during run-time.

Data Description

The data is provided in form of Isabelle/HOL source files, which are hosted at `logikey.org`. The individual data files belong to different categories.

Contributed data files in category I are listed in Table 1. They provide encodings of SSEs, and associated tests, of various deontic logics in meta-logic HOL. A category I example file is displayed in Figs. 1–2; this data file contains (an extension of) the SSE of dyadic deontic logic (DDL) by Carmo and Jones [6] in HOL and studies, resp. verifies, its properties.

Contributed data files in category II are listed in Table 2. They study paradoxes and smaller examples of normative reasoning. An example is displayed in Fig. 3, which presents an analysis of Chisholm's paradox of CTD obligation [8] in Standard Deontic Logic (SDL). The known fact that SDL cannot handle CTD scenarios is confirmed by the computer.

Contributed data files in category III are listed in Table 3. They provide an encoding of (excerpt of) legal and ethical theories and arguments formalized using the deontic logics as provided in category I files and further examined in the category II files.

In addition to data listed in Tables 1—3 the data set provided at `logikey.org` also includes the following:

- Subdirectory `2020-DataInBrief-Data/Course-Material-1` contains Isabelle/HOL data files stemming from a lecture course on deontic logic at University of Luxembourg based on [18].

- Subdirectory `2020-DataInBrief-Data/Climate-Engineering` contains Isabelle/HOL data files related to the formalization and assessment [10] of selected arguments in climate engineering.

- Subdirectory `2020-DataInBrief-Data/US-Constitution-Loophole` contains Isabelle/HOL data files related to a formalization and assessment of Kurt Gödel's claim that the US Constitution contains a loophole for establishing a dictatorship.

- Subdirectory `2020-DataInBrief-Data/WiseMenPuzzle` contains Isabelle/HOL data files related to a formalization and study of the well known Wise Men Puzzle; this data set, which has been published before [1], is included here to make it better available for `logikey.org` users.

Table 1: Category I data files — deontic logics, extensions of deontic logics and logic combinations

File (Dependency) Reading	Description
SDL.thy (Main.thy) [7]	Provides a SSE of standard deontic logic (SDL) in HOL. An unary deontic operator is defined. The D axiom is postulated and correspondence to seriality of the accessibility is proved. The added first-order and higher-order quantifiers are constant domain (possibilist notion of quantification). This is verified by proving the Barcan formula and its converse.
CJ_DDL.thy (Main.thy) [3]	Provides a SSE of a dyadic deontic logic (DDL) by Carmo and Jones [6] in HOL. Different modal operators are introduced: dyadic deontic obligation, monadic deontic operator for actual obligation, monadic deontic operator for primary obligation, and further alethic modalities. Moreover, constant domain first-order and higher-order quantifiers are added.
CJ_DDL_Tests.thy (CJ_DDL.thy) [3]	Contains soundness and proof automation tests for the embedding of DDL in HOL given in CJ_DDL.thy. For example, the monadic modal operators D, D_p and D_a are identified as S5, KT and KD modalities, respectively. Relevant lemmata from the original work of Carmo and Jones are automated.
E.thy (Main.thy) [4],[5, Fig.6]	Provides a SSE of a quantified extension of Aqvist's System E in HOL. The file also runs a number of reasoning tasks (validity checking, refutation, correspondence theory).
Lewis_DDL.thy (Main.thy) [16]	Provides a SSE of Lewis's DDL. The file runs a number of reasoning tasks (validity checking, refutation, correspondence theory). The relationship with Åqvist's dyadic deontic operator is also studied.
IOL_out2.thy (Main.thy)[2]	Provides a SSE of a quantified extension of Input/Output (IO) logic (out_2) [17, 18]. The file also contains an analysis of a benchmark example discussed in the literature on moral luck.
IO_out2_STIT.thy (Main.thy) [2]	Provides a SSE of a quantified extension of I/O logic (out_2) [17, 18] and elements of STIT logics [13] in HOL. The file also contains proof automation tests and soundness checks.
CJ_DDLplus.thy (Main.thy) [9]	A modification of the SSE developed in file CJ_DDL.thy is presented; see Figs. 1 and 2. This theory provides the starting point for an extension of a higher-order variant of DDL into a two-dimensional semantics as originally presented by Kaplan for his logic of demonstratives [15]. The logic extension is completed in file Extended_CJ_DDL.thy. The displayed lines in Fig. 2 show automations of various lemmata from the original paper of Carmo and Jones [6], where they were proved manually with pen and paper.
Extended_CJ_DDL.thy (CJ_DDLplus.thy) [9]	Contains a further extension and combination of the higher-order DDL encoded in file CJ_DDLplus.thy with relevant parts (for the work presented in the related research article [5]) of Kaplan's logic of demonstratives.

```
1 theory CJ_DDLplus imports Main                                    (*David Fuenmayor and C. Benzmüller, 2019*)
2 begin
3 nitpick_params[user_axioms=true, show_all, expect=genuine, format = 3]
4
5 section ‹Semantic Embedding of Carmo and Jones' Dyadic Deontic Logic (DDL) augmented with Kaplanian contexts›
6 (**We introduce a modification of the semantic embedding developed by Benzm\"uller et al.
7 for the Dyadic Deontic Logic originally presented by Carmo and Jones. We extend this embedding
8 to a two-dimensional semantics as originally presented by David Kaplan.*)
9 subsection ‹Definition of Types›
10  typedecl w   (** Type for possible worlds (Kaplan's "circumstances of evaluation" or "counterfactual situations") *)
11  typedecl e   (** Type for individuals (entities eligible to become agents)*)
12  typedecl c   (** Type for Kaplanian "contexts of use"*)
13  type_synonym wo = "w⇒bool" (** contents/propositions are identified with their truth-sets*)
14  type_synonym cwo = "c⇒wo"  (** sentence meaning (Kaplan's "character") is a function from contexts to contents*)
15  type_synonym m = "cwo"      (** we use the letter 'm' for characters (reminiscent of "meaning")*)
16
17 subsection ‹Semantic Characterisation of DDL›
18 subsubsection ‹Basic Set Operations›
19  abbreviation subset::"wo⇒wo⇒bool" (infix "⊑" 46) where "α ⊑ β ≡ ∀w. α w  ⟶ β w"
20  abbreviation intersection::"wo⇒wo⇒wo" (infixr "⊓" 48) where "α ⊓ β ≡ λx. α x ∧ β x"
21  abbreviation union::"wo⇒wo⇒wo" (infixr "⊔" 48) where "α ⊔ β ≡ λx. α x ∨ β x"
22  abbreviation complement::"wo⇒wo" ("∼_"[45]46) where "∼α ≡ λx. ¬α x"
23  abbreviation instantiated::"wo⇒bool" ("I_"[45]46) where "I φ ≡ ∃x. φ x"
24  abbreviation setEq::"wo⇒wo⇒bool" (infix "=s" 46) where "α =s β ≡ ∀x. α x ⟷ β x"
25  abbreviation univSet :: "wo" ("⊤") where "⊤ ≡ λw. True"
26  abbreviation emptySet :: "wo" ("⊥") where "⊥ ≡ λw. False"
27
28 subsubsection ‹Set-Theoretic Conditions for DDL›
29 consts
30  av::"w⇒wo"   (**set of worlds that are open alternatives (aka. actual versions) of w*)
31  pv::"w⇒wo"   (**set of worlds that are possible alternatives (aka. potential versions) of w*)
32  ob::"wo⇒wo⇒bool" (**set of propositions which are obligatory in a given context (of type wo) *)
33
34 axiomatization where
35  sem_3a: "∀w. I(av w)" and (** av is serial: in every situation there is always an open alternative*)
36  sem_4a: "∀w. av w ⊑ pv w" and (** open alternatives are possible alternatives*)
37  sem_4b: "∀w. pv w w" and (** pv is reflexive: every situation is a possible alternative to itself*)
38  sem_5a: "∀X. ¬(ob X ⊥)" and (** contradictions cannot be obligatory*)
39  sem_5b: "∀X Y Z. (X ⊓ Y) =s (X ⊓ Z) ⟶ (ob X Y ⟷ ob X Z)" and
40  sem_5c: "∀X Y Z. I(X ⊓ Y ⊓ Z) ∧ ob X Y ∧ ob X Z ⟶ ob X (Y ⊓ Z)" and
41  sem_5d: "∀X Y Z. (Y ⊑ X ∧ ob X Y ∧ X ⊑ Z) ⟶ ob Z ((Z ⊓ (∼X)) ⊔ Y)" and
42  sem_5e: "∀X Y Z. Y ⊑ X ∧ ob X Z ∧ I(Y ⊓ Z) ⟶ ob Y Z"
43
44  lemma True nitpick[satisfy] oops (**model found: axioms are consistent*)
45
46 subsubsection ‹Verifying Semantic Conditions›
47  lemma sem_5b1: "ob X Y ⟶ ob X (Y ⊓ X)" by (metis (no_types, lifting) sem_5b)
48  lemma sem_5b2: "(ob X (Y ⊓ X) ⟶ ob X Y)" by (metis (no_types, lifting) sem_5b)
49  lemma sem_5ab: "ob X Y ⟶ I(X ⊓ Y)" by (metis (full_types) sem_5a sem_5b)
50  lemma sem_5bd1: "Y ⊑ X ∧ ob X Y ∧ X ⊑ Z ⟶ ob Z ((∼X) ⊔ Y)" using sem_5b sem_5d by smt
51  lemma sem_5bd2: "ob X Y ∧ X ⊑ Z ⟶ ob Z ((Z ⊓ (∼X)) ⊔ Y)"  using sem_5b sem_5d  by (smt sem_5b1)
52  lemma sem_5bd3: "ob X Y ∧ X ⊑ Z ⟶ ob Z ((∼X) ⊔ Y)"  by (smt sem_5bd2 sem_5b)
53  lemma sem_5bd4: "ob X Y ∧ X ⊑ Z ⟶ ob Z ((∼X) ⊔ (X ⊓  Y))" using sem_5bd3 by auto
54  lemma sem_5bcd: "(ob X Z ∧ ob Y Z) ⟶ ob (X ⊔ Y) Z" using sem_5b sem_5c sem_5d oops
55  (** 5e and 5ab justify redefinition of @{text "O⟨φ|σ⟩"} as (ob A B)*)
56  lemma "ob A B ⟷  (I(A ⊓ B) ∧ (∀X. X ⊑ A ∧ I(X ⊓ B) ⟶ ob X B))" using sem_5e sem_5ab by blast
57
58 subsection ‹(Shallow) Semantic Embedding of DDL›
59
60 subsubsection ‹Basic Propositional Logic›
61  abbreviation pand::"m⇒m⇒m" (infixr"∧" 51) where "φ∧ψ ≡ λc w. (φ c w)∧(ψ c w)"
62  abbreviation por::"m⇒m⇒m" (infixr"∨" 50) where "φ∨ψ ≡ λc w. (φ c w)∨(ψ c w)"
63  abbreviation pimp::"m⇒m⇒m" (infix"→" 49) where "φ→ψ ≡ λc w. (φ c w)⟶(ψ c w)"
64  abbreviation pequ::"m⇒m⇒m" (infix"↔" 48) where "φ↔ψ ≡ λc w. (φ c w)⟷(ψ c w)"
65  abbreviation pnot::"m⇒m" ("¬_" [52]53) where "¬φ ≡ λc w. ¬(φ c w)"
66
67 subsubsection ‹Modal Operators›
68  abbreviation cjboxa :: "m⇒m" ("□a_" [52]53) where "□aφ ≡ λc w. ∀v. (av w) v ⟶ (φ c v)"
69  abbreviation cjdiaa :: "m⇒m" ("◇a_" [52]53) where "◇aφ ≡ λc w. ∃v. (av w) v ∧ (φ c v)"
70  abbreviation cjboxp :: "m⇒m" ("□p_" [52]53) where "□pφ ≡ λc w. ∀v. (pv w) v ⟶ (φ c v)"
71  abbreviation cjdiap :: "m⇒m" ("◇p_" [52]53) where "◇pφ ≡ λc w. ∃v. (pv w) v ∧ (φ c v)"
72  abbreviation cjtaut :: "m" ("⊤") where "⊤ ≡ λc w. True"
73  abbreviation cjcontr :: "m" ("⊥") where "⊥ ≡ λc w. False"
74
75 subsubsection ‹Deontic Operators›
76  abbreviation cjod :: "m⇒m⇒m" ("O⟨_|_⟩"54) where "O⟨φ|σ⟩ ≡ λc w. ob (σ c) (φ c)"
77  abbreviation cjoa :: "m⇒m" ("Oa_" [53]54) where "Oaφ ≡ λc w. (ob (av w)) (φ c) ∧ (∃x. (av w) x ∧ ¬(φ c x))"
78  abbreviation cjop :: "m⇒m" ("Oi_" [53]54) where "Oiφ ≡ λc w. (ob (pv w)) (φ c) ∧ (∃x. (pv w) x ∧ ¬(φ c x))"
```

Figure 1: Data file CJ_DDLplus.thy; in lines 29–84 the SSE of the DDL by Carmo and Jones [6] in HOL is presented

```
79
80 subsubsection ‹Logical Validity (Classical)›
81  abbreviation modvalidctx :: "m⇒c⇒bool" ("⌊_⌋M") where "⌊φ⌋M ≡ λc. ∀w. φ c w" (**context-dep. modal validity*)
82  abbreviation modinvalidctx :: "m⇒c⇒bool" ("⌈_⌉M") where "⌈φ⌉M ≡ λc. ∀w. ¬φ c w" (**ctxt-dep. modal invalidity*)
83  abbreviation modvalid :: "m⇒bool" ("⌊_⌋") where "⌊φ⌋ ≡ ∀c. ⌊φ⌋M c" (**general modal validity*)
84  abbreviation modinvalid :: "m⇒bool" ("⌈_⌉") where "⌈φ⌉ ≡ ∀c. ⌈φ⌉M c" (**general modal invalidity*)
85
86 subsection ‹Verifying the Embedding›
87 subsubsection ‹Avoiding Modal Collapse›
88  lemma "⌊P → Oa P⌋" nitpick oops (**(actual) deontic modal collapse is countersatisfiable*)
89  lemma "⌊P → Oi P⌋" nitpick oops (**(ideal) deontic modal collapse is countersatisfiable*)
90  lemma "⌊P → □a P⌋" nitpick oops (**alethic modal collapse is countersatisf. (implies other necessity operators)*)
91
92 subsubsection ‹Necessitation Rule›
93  lemma NecDDLa: "⌊A⌋ ⟹ ⌊□a A⌋"  by simp (** Valid only using classical (not LD) validity*)
94  lemma NecDDLp:  "⌊A⌋ ⟹ ⌊□p A⌋" by simp (** Valid only using classical (not LD) validity*)
95
96 subsubsection ‹Lemmas for Semantic Conditions› (* extracted from Benzmüller et al. paper*)
97  abbreviation mboxS5 :: "m⇒m" ("□S5_" [52]53) where "□S5 φ ≡ λc w. ∀v. φ c v"
98  abbreviation mdiaS5 :: "m⇒m" ("◇S5_" [52]53) where "◇S5 φ ≡ λc w. ∃v. φ c v"
99  lemma C_2: "⌊O⟨A | B⟩ → ◇S5(B ∧ A)⌋" by (simp add: sem_5ab)
100 lemma C_3:  "⌊((◇S5(A ∧ B ∧ C)) ∧ O⟨B|A⟩ ∧ O⟨C|A⟩) → O⟨(B ∧ C)| A⟩⌋" by (simp add: sem_5c)
101 lemma C_4: "⌊(□S5(A → B) ∧ ◇S5(A ∧ C) ∧ O⟨C|B⟩) → O⟨C|A⟩⌋" using sem_5e by blast
102 lemma C_5: "⌊□S5(A ↔ B) → (O⟨C|A⟩ → O⟨C|B⟩)⌋" using C_2 sem_5e by blast
103 lemma C_6: "⌊□S5(C → (A ↔ B)) → (O⟨A|C⟩ ↔ O⟨B|C⟩)⌋" by (metis sem_5b)
104 lemma C_7: "⌊O⟨B|A⟩ → □S5 O⟨B|A⟩⌋" by blast
105 lemma C_8: "⌊O⟨B|A⟩ → O⟨A → B| ⊤⟩⌋" using sem_5bd4 by presburger
106
107 subsubsection ‹Verifying Axiomatic Characterisation›
108 (**The following theorems have been taken from the original Carmo and Jones' paper.*)
109 lemma CJ_3: "⌊□p A → □a A⌋" by (simp add: sem_4a)
110 lemma CJ_4: "⌊¬O⟨⊥|A⟩⌋" by (simp add: sem_5a)
111 lemma CJ_5: "⌊(O⟨B|A⟩ ∧ O⟨C|A⟩) → O⟨B∧C|A⟩⌋" nitpick oops (**countermodel found*)
112 lemma CJ_5_minus: "⌊◇S5(A ∧ B ∧ C) ∧ (O⟨B|A⟩ ∧ O⟨C|A⟩) → O⟨B∧C|A⟩⌋" by (simp add: sem_5c)
113 lemma CJ_6: "⌊O⟨B|A⟩ → O⟨B|A∧B⟩⌋" by (smt C_2 C_4)
114 lemma CJ_7: "⌊A ↔ B⌋ ⟶ ⌊O⟨C|A⟩ ↔ O⟨C|B⟩⌋" using sem_5ab sem_5e by blast
115 lemma CJ_8: "⌊C → (A ↔ B)⌋ ⟶ ⌊O⟨A|C⟩ ↔ O⟨B|C⟩⌋" using C_6 by simp
116 lemma CJ_9a: "⌊◇p O⟨B|A⟩ → □p O⟨B|A⟩⌋" by simp
117 lemma CJ_9p: "⌊◇a O⟨B|A⟩ → □a O⟨B|A⟩⌋" by simp
118 lemma CJ_9_var_a: "⌊O⟨B|A⟩ → □a O⟨B|A⟩⌋" by simp
119 lemma CJ_9_var_b: "⌊O⟨B|A⟩ → □p O⟨B|A⟩⌋" by simp
120 lemma CJ_10: "⌊◇p(A ∧ B ∧ C) ∧ O⟨C|B⟩ → O⟨C|A∧B⟩⌋" by (smt C_4)
121 lemma CJ_11a: "⌊(Oa A ∧ Oa B) → Oa(A ∧ B)⌋" nitpick oops (** countermodel found*)
122 lemma CJ_11a_var: "⌊◇a(A ∧ B) ∧ (Oa A ∧ Oa B) → Oa(A ∧ B)⌋" using sem_5c by auto
123 lemma CJ_11p: "⌊(Oi A ∧ Oi B) → Oi(A ∧ B)⌋" nitpick oops (** countermodel found*)
124 lemma CJ_11p_var: "⌊◇p(A ∧ B) ∧ (Oi A ∧ Oi B) → Oi(A ∧ B)⌋" using sem_5c by auto
125 lemma CJ_12a: "⌊□a A → (¬Oa A ∧ ¬Oa(¬A))⌋" using sem_5ab by blast (*using C_2 by blast *)
126 lemma CJ_12p: "⌊□p A → (¬Oi A ∧ ¬Oi(¬A))⌋" using sem_5ab by blast (*using C_2 by blast*)
127 lemma CJ_13a: "⌊□a(A ↔ B) → (Oa A ↔ Oa B)⌋" using sem_5b by metis (*using C_6 by blast *)
128 lemma CJ_13p: "⌊□p(A ↔ B) → (Oi A ↔ Oi B)⌋" using sem_5b by metis (*using C_6 by blast *)
129 lemma CJ_0_0: "⌊O⟨B|A⟩ → O⟨A → B|⊤⟩⌋" using sem_5bd4 by presburger
130 (**An ideal obligation which is actually possible both to fulfill and to violate entails an actual obligation.*)
131 lemma CJ_Oi_Oa: "⌊(Oi A ∧ ◇a A ∧ ◇a(¬A)) → Oa A⌋" using sem_5e sem_4a by blast
132 (**Bridge relations between conditional obligations and actual/ideal obligations:*)
133 lemma CJ_14a: "⌊O⟨B|A⟩ ∧ □a A ∧ ◇a B ∧ ◇a¬B → Oa B⌋" using sem_5e by blast
134 lemma CJ_14p: "⌊O⟨B|A⟩ ∧ □p A ∧ ◇p B ∧ ◇p¬B → Oi B⌋" using sem_5e by blast
135 lemma CJ_15a: "⌊(O⟨B|A⟩ ∧ ◇a(A ∧ B) ∧ ◇a(A ∧ ¬B)) →  Oa(A → B)⌋" using CJ_0_0 sem_5e by fastforce
136 lemma CJ_15p: "⌊(O⟨B|A⟩ ∧ ◇p(A ∧ B) ∧ ◇p(A ∧ ¬B)) →  Oi(A → B)⌋" using CJ_0_0 sem_5e by fastforce
137 end
```

Figure 2: File CJ_DDLplus.thy (cont'd); in lines 87–136 lemmata from Carmo and Jones's paper [6] are proved

Further related data sets, including selected formalisations in computational metaphysics, will be added to `logikey.org` as we think fit.

Experimental Design, Materials, and Methods

The data was acquired through manual encodings of logics, theories and arguments in the Isabelle/HOL proof assistant system. The modeling process was following the LogiKEy methodology depicted in Fig. 4. This methodology supports formalization projects in the area of ethical and legal reasoning at different layers of abstraction. The spirals in Fig. 4 indicate that the formalization work may proceed in cycles – at each layer and overall. The LogiKEy methodology is briefly explained below at hand of selected examples from our contributed data set; we address all three different layers and discuss examples.[1]

Layer L1 example development (files CJ_DDL.thy and CJ_DDL_Tests.thy): File CJ_DDL.thy contains the en- coding (of a quantified extension) of the DDL of Carmo and Jones in HOL. This encoding of DDL in HOL is

[1]For a general description of the LogiKEy framework, methodology and tool support see the related research article [5].

Table 2: Category II data files: paradoxes and examples of normative reasoning

File (Dependency) Reading	Description
Chisholm_SDL.thy (SDL.thy) [18]	The well-known analysis of Chisholm's CTD paradox in SDL is automated. The formalization uses both the wide-scope interpretation of conditional "ought" and the narrow-scope one; see Fig. 3.
Chisholm_CJ_DDL_Monadic.thy (CJ_DDL.thy)	Contains a study analogous to Chisholm_SDL.thy for monadic obligation in DDL.
Chisholm_CJ_DDL_Dyadic.thy (CJ_DDL.thy)	Contains a study analogous to Chisholm_SDL.thy for dyadic obligation in DDL.
Chisholm_E.thy (CJ_DDL.thy)	Contains a study analogous to Chisholm_SDL.thy for deontic logic E.
IO_Experiments (IO_out2_STIT)	Contains a study of different paradoxes from the literature in I/O logic (out_2); the file imports IO_out2_STIT.

Table 3: Category III data files: (excerpts of) legal and ethical theories and arguments

File (Dependency) Reading	Description
GDPR_SDL.thy (SDL.thy) [5, Fig. 7]	Contains a modeling of selected statements from the GDPR in first-order SDL. It is confirmed by automated means that first-order SDL cannot handle CTD scenarios.
GDPR_CJ_DDL.thy (CJ_DDL.thy)	Contains a modeling of selected statements from the GDPR in first-order DDL. It is confirmed by automated means that the logic can handle CTD scenarios. The problems identified in GDPR_SDL.thy, i.e., inconsistency and explosion, are avoided. The reasoners return "intuitive" answers to queries.
GDPR_E.thy (E.thy) [5, Fig. 8]	Contains a modeling of selected statements from the GDPR in a first-order extension of logic E. It is confirmed by automated means that the logic can handle CTD scenarios, and does not face the problems identified in GDPR_SDL.thy, i.e., inconsistency and explosion.
GewirthArgument.thy (Extended_CJ_DDL.thy) [9],[5, Fig. 10]	Contains a formalization and partial automation of Gewirth's supporting argument for his *Principle of Generic Consistency*. This principle constitutes, loosely speaking, an emendation of the *Golden Rule*, i.e., the principle of treating others as one's self would wish to be treated. Gewirth's argument and theory is assessed, emended (minor corrections) and verified.

Carmo and Jones's DDL in the given case. A *semantics* (Step 2) for this object logic was sought and found in the original papers by Carmo and Jones [6]; such a mathematical description of a semantics, a neighborhood semantics in the given case, constitutes the ideal starting point for the definition of a SSE of the object logic in HOL, which in turn enables its *automation* (Step 3) with off-the-shelf reasoning tools for HOL. The automation of DDL was subsequently assessed (Step 4) with automated theorem provers and model finders integrated with Isabelle/HOL. Then, by pen and paper means on a theoretical level, the *faithfulness* (Step 4) of the embedding of DDL in HOL was studied and proved; this proof has been published [3]. Furthermore, *implications* of the embedding of DDL in HOL were studied (Step 5); see for example the additional theorems in file CJ_DDL_Tests.thy and the analysis of CTD scenarios conducted in files Chisholm_DDL_Monadic.thy and Chisholm_DDL_Dyadic.thy. Since the DDL of Carmo and Jones has not been automated before with other systems or approaches, there are no *benchmarks* (Step 7) available that we could use to properly assess and compare the competitiveness of our solution. The publication of this data set can be seen as a first step towards the built-up and *contribution* (Step 8) of such a benchmark

```
theory Chisholm_SDL imports SDL                              (*Christoph Benzmüller & Xavier Parent, 2019*)
begin     (*Unimportant*) nitpick_params [user_axioms,show_all,format=2]

(*** Chisholm Example ***)
  consts go::σ tell::σ kill::σ
  abbreviation  "D1 ≡ ○<go>"  (*It ought to be that Jones goes to assist his neighbors.*)
  abbreviation  "D2w ≡ ○<go → tell>"  (*It ought to be that if Jones goes, then he tells them he is coming.*)
  abbreviation  "D2n ≡ go → ○<tell>"
  abbreviation  "D3w ≡ ○<¬go → ¬tell>" (*If Jones doesn't go, then he ought not tell them he is coming.*)
  abbreviation  "D3n ≡ ¬go → ○<¬tell>"
  abbreviation  "D4 ≡ ¬go" (*Jones doesn't go. (This is encoded as a locally valid statement.)*)

(*** Chisholm_A: All-wide scoping is leading to an inadequate, dependent set of the axioms.***)
 lemma "⌊(D1  ∧ D2w ∧ D3w) → D4⌋"  nitpick oops (*countermodel*)
 lemma "⌊(D1  ∧ D2w ∧ D4)  → D3w⌋"  sledgehammer by blast  (*proof*)
 lemma "⌊(D1  ∧ D3w ∧ D4)  → D2w⌋"  nitpick oops (*countermodel*)
 lemma "⌊(D2w ∧ D3w ∧ D4)  → D1⌋"  nitpick oops (*countermodel*)
 (* Consistency *)
 lemma "⌊(D1 ∧ D2w ∧ D3w)⌋ ∧ ⌊D4⌋ι" nitpick [satisfy] oops (*Consistent? Yes*)
 (* Queries *)
 lemma assumes "⌊(D1 ∧ D2w ∧ D3w)⌋ ∧ ⌊D4⌋ι" shows "⌊○<¬tell>⌋ι" nitpick oops (*Should James not tell? No*)
 lemma assumes "⌊(D1 ∧ D2w ∧ D3w)⌋ ∧ ⌊D4⌋ι" shows "⌊○<tell>⌋ι"  using assms by blast (*Should J. tell? Yes*)
 lemma assumes "⌊(D1 ∧ D2w ∧ D3w)⌋ ∧ ⌊D4⌋ι" shows "⌊○<kill>⌋ι"  nitpick oops (*Should James kill? No*)

(*** Chisholm_B: All-narrow scoping is leading to a inadequate, dependent set of the axioms.*)
 lemma "⌊(D1  ∧ D2n ∧ D3n) → D4⌋"   nitpick oops (*countermodel*)
 lemma "⌊(D1  ∧ D2n ∧ D4)  → D3n⌋"   nitpick oops  (*countermodel*)
 lemma "⌊(D1  ∧ D3n ∧ D4)  → D2n⌋"   sledgehammer by blast  (*proof*)
 lemma "⌊(D2n ∧ D3n ∧ D4)  → D1⌋"   nitpick oops (*countermodel*)
 (* Consistency *)
 lemma "⌊(D1 ∧ D2n ∧ D3n)⌋ ∧ ⌊D4⌋ι" nitpick [satisfy] oops (*Consistent? Yes*)
 (* Queries *)
 lemma assumes "⌊(D1 ∧ D2n ∧ D3n)⌋ ∧ ⌊D4⌋ι" shows "⌊○<¬tell>⌋ι" using assms by smt (*Should J. not tell? Yes*)
 lemma assumes "⌊(D1 ∧ D2n ∧ D3n)⌋ ∧ ⌊D4⌋ι" shows "⌊○<tell>⌋ι"  nitpick oops (*Should James tell? No*)
 lemma assumes "⌊(D1 ∧ D2n ∧ D3n)⌋ ∧ ⌊D4⌋ι" shows "⌊○<kill>⌋ι"  nitpick oops (*Should James kill? No*)

(*** Chisholm_C: Wide-narrow scoping is leading to an adequate, independence of the axioms.*)
 lemma "⌊(D1  ∧ D2w ∧ D3n) → D4⌋"  nitpick oops (*countermodel*)
 lemma "⌊(D1  ∧ D2w ∧ D4) → D3n⌋"  nitpick oops (*countermodel*)
 lemma "⌊(D1  ∧ D3n ∧ D4) → D2w⌋"  nitpick oops (*countermodel*)
 lemma "⌊(D2w ∧ D3n ∧ D4) → D1⌋"   nitpick oops (*countermodel*)
 (* Consistency *)
 lemma "⌊(D1 ∧ D2w ∧ D3n)⌋ ∧ ⌊D4⌋ι" nitpick [satisfy] oops (*Consistent? No*)
 (* Queries *)
 lemma assumes "⌊(D1 ∧ D2w ∧ D3n)⌋ ∧ ⌊D4⌋ι" shows "⌊○<¬tell>⌋ι" using D assms by smt (*Shld J. not tell? Yes*)
 lemma assumes "⌊(D1 ∧ D2w ∧ D3n)⌋ ∧ ⌊D4⌋ι" shows "⌊○<tell>⌋ι"  using assms by blast (*Should J. tell? Yes*)
 lemma assumes "⌊(D1 ∧ D2w ∧ D3n)⌋ ∧ ⌊D4⌋ι" shows "⌊○<kill>⌋ι"  using D assms by blast (*Should J. kill? Yes*)

(*** Chisholm_D: Narrow-wide scoping is leading to a  inadequate, dependent set of the axioms.*)
 lemma "⌊(D1  ∧ D2n ∧ D3w) → D4⌋"   nitpick oops (*countermodel*)
 lemma "⌊(D1  ∧ D2n ∧ D4)  → D3w⌋"   by blast  (*proof*)
 lemma "⌊(D1  ∧ D3w ∧ D4)  → D2n⌋"   by blast  (*proof*)
 lemma "⌊(D2n ∧ D3w ∧ D4)  → D1⌋"   nitpick oops (*countermodel*)
 (* Consistency *)
 lemma "⌊(D1 ∧ D2n ∧ D3w)⌋ ∧ ⌊D4⌋ι" nitpick [satisfy] oops (*Consistent? Yes*)
 (* Queries *)
 lemma assumes "⌊(D1 ∧ D2n ∧ D3w)⌋ ∧ ⌊D4⌋ι" shows "⌊○<¬tell>⌋ι" nitpick oops (*Should James not tell? No*)
 lemma assumes "⌊(D1 ∧ D2n ∧ D3w)⌋ ∧ ⌊D4⌋ι" shows "⌊○<tell>⌋ι"  nitpick oops (*Should James tell? No*)
 lemma assumes "⌊(D1 ∧ D2n ∧ D3w)⌋ ∧ ⌊D4⌋ι" shows "⌊○<kill>⌋ι"  nitpick oops (*Should James kill? No*)
end
```

Figure 3: Data file Chisholm_SDL.thy studies Chisholm's paradox in combination with wide-narrow scoping issues

suite to the community.

Layer L2 example development (file GDPR_CJ_DDL.thy): In file GDPR_CJ_DDL.thy we *selected* (Step 1) state- ments from the General Data Protection Regulation (GDPR) for formalisation. The *analysis* (Step 2) of these statements revealed that obligation aspects in the context of data processing needed to be addressed. Natural language phrases in the studied parts of the GDPR indeed contains occurrences of the deontic modalities. This motivated the choice of a suitable deontic *logic* (Step 3), such as DDL, for the formal encoding of these challenging aspects. In the given case it also became apparent that a propositional encoding would hardly suffice in practical applications, so the selected deontic logic DDL needed to be extended by a notion of quantification, which led to the addition of quantifiers to the file CJ_DDL.thy.

Subsequently the two GDPR articles were *formalized* (Step 4) using logical connectives as provided in the imported file `CJ_DDL.thy`, and then some *exploration* (Step 5) and assessment studies were conducted. This included the analysis of the CTD scenario as reported in related research articles [5, 3]. With our data set we *contribute* (Step 6) this work to the wider research community and enable its reuse.

Layer L3 example development: Layer L3 example developments have just started. The idea is to populate regulatory governor architectures [5] with ethical and legal theories from Layer L2, so that reasoning with the theories can be utilized to explain and control the behaviour of (autonomous) AI systems. To realize such applications it is required to *select* (Step 1) some ethical and/or legal theory from Layer L2, to devise and implement (or reuse) a respective *governor architecture* (Step 2), to *populate* (Step 3) this governor system with the selected ethical and/or legal theory, and to *assess* (Step 4) the well-functioning of this system in empirical studies.

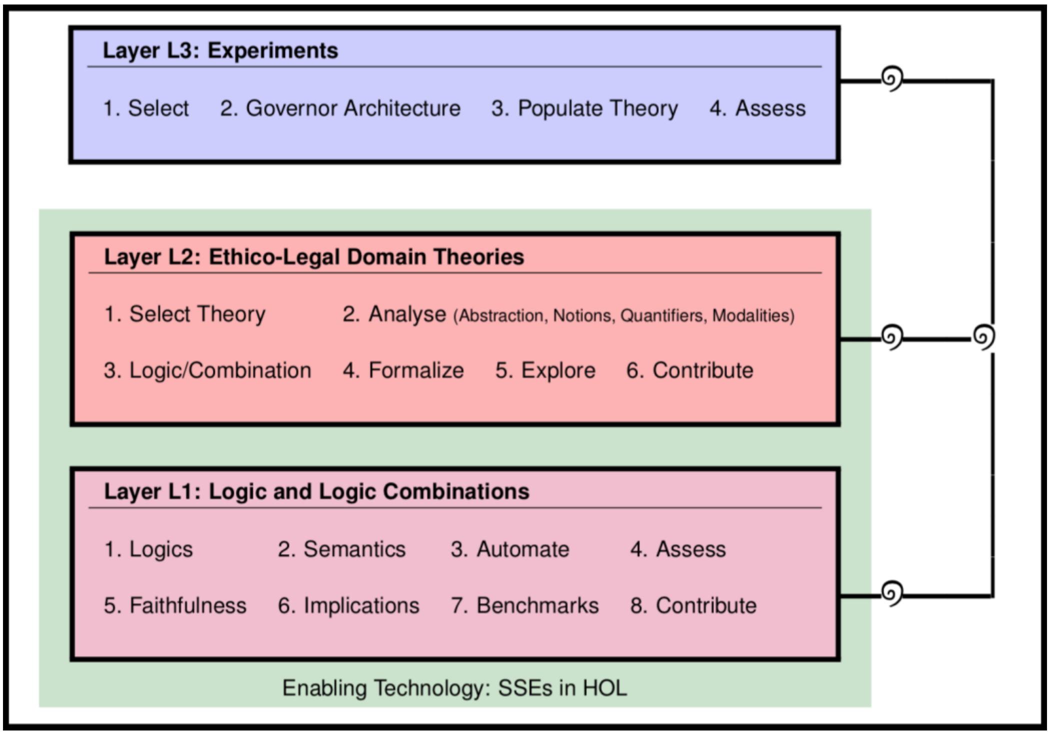

Figure 4: The LogiKEy logic and knowledge development methodology

Ethics Statement

Our work did not involve the use of human subjects, and it did not involve animal experiments.

Acknowledgments

We thank all our collaborators and students from University of Luxembourg and Freie Universität Berlin that have already utilized and tested the LogiKEy methodology in several projects not reported here. Moreover, we thank the anonymous reviewers for their valuable comments.

Competing Interests

Benzmüller was funded by the VolkswagenStiftung under grant CRAP (Consistent Rational Argumentation in Politics). Parent and van der Torre were 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.

References

[1] C. Benzmüller. Universal (meta-)logical reasoning: The wise men puzzle. *Data in brief*, 24:103774, 2019.

[2] C. Benzmüller, A. Farjami, P. Meder, and X. Parent. I/O logic in HOL. *Journal of Applied Logics – IfCoLoG Journal of Logics and their Applications (Special Issue on Reasoning for Legal AI)*, 6(5):715–732, 2019.

[3] C. Benzmüller, A. Farjami, and X. Parent. A dyadic deontic logic in HOL. In J. Broersen, C. Condoravdi, S. Nair, and G. Pigozzi, editors, *Deontic Logic and Normative Systems – 14th International Conference, DEON 2018, Utrecht, The Netherlands, 3-6 July, 2018*, pages 33–50. College Publications, 2018.

[4] C. Benzmüller, A. Farjami, and X. Parent. Åqvist's dyadic deontic logic E in HOL. *Journal of Applied Logics – IfCoLoG Journal of Logics and their Applications (Special Issue on Reasoning for Legal AI)*, 6(5):733–755, 2019.

[5] C. Benzmüller, X. Parent, and L. van der Torre. Designing normative theories for ethical and le- gal reasoning: Logikey framework, methodology, and tool support. *Artificial Intelligence*, 287(103348), 2020. Doi: 10.1016/j.artint.2020.103348.

[6] J. Carmo and A. J. I. Jones. Deontic logic and contrary-to-duties. In D. M. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic: Volume 8*, pages 265–343. Springer Netherlands, Dordrecht, 2002.

[7] B. Chellas. *Modal Logic*. Cambridge University Press, Cambridge, 1980.

[8] R. Chisholm. Contrary-to-duty imperatives and deontic logic. *Analysis*, 24:33–36, 1963.

[9] D. Fuenmayor and C. Benzmüller. Harnessing higher-order (meta-)logic to represent and reason with complex ethical theories. In A. Nayak and A. Sharma, editors, *PRICAI 2019: Trends in Artificial Intelligence*, volume 11670 of *Lecture Notes in Artificial Intelligence*, pages 418–432. Springer, Cham, 2019.

[10] D. Fuenmayor and C. Benzmüller. Computer-supported analysis of arguments in climate engineering. In M. Dastani, H. Dong, and L. van der Torre, editors, *Logic and Argumentation. CLAR 2020.*, volume 12061 of *Lecture Notes in Computer Science*, pages 104–115. Springer, Cham, 2020.

[11] D. Gabbay, J. Horty, X. Parent, R. van der Meyden, and L. van der Torre. *Handbook of Deontic Logic and Normative Systems*, volume 1. College Publications, UK, 2013.

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

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

[14] A. J. I. Jones and M. Sergot. Deontic logic in the representation of law: Towards a methodology. *Artificial Intelligence and Law*, 1(1):45–64, 1992.

[15] D. Kaplan. On the logic of demonstratives. *Journal of Philosophical Logic*, 8(1):81–98, 1979.

[16] D. Lewis. *Counterfactuals*. Blackwell, Oxford, 1973.

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

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