Article Title

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

Authors

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

Affiliations

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

Corresponding author(s)

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

Abstract

The LogiKEy workbench for legal and ethical reasoning is presented. This workbench simultaneously supports development, experimentation, assessment and deployment of formal logics and legal and ethical 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 assistent 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 specific 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 legal and ethical reasoning. It may also support hand-on teaching of non-trivial logic formalisms in lecture courses and tutorials.

Keywords

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

Specifications Table

Subject	Computer Science
Specific subject area	Artificial intelligence; Knowledge representation and reasoning; Normative reasoning
Type of data	.thy files (formal theories encoded in Isabelle/HOL syntax)
How data were 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 [7], which is itself based on shallow semantical embeddings (SSEs) [1] of logics and theories in classical higher-order logic. The concrete encodings were conducted in the higher-order proof assistant system Isabelle/HOL [22]; however, they are conceptually transferable to many 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; Isabelle/HOL is freely available at https://isabelle.in.tum.de

Parameters for data collection	One objective was to empirically assess the expressivity and proof automation capabilities of Isabelle/HOL and its integrated tools in normative reasoning when utilising the LogiKEy methodology and the SSE approach. Another objective was to provide a reusable foundation for further experiments in expressive legal and ethical reasoning.
Description of data collection	The data was manually constructed and curated. As part of the data collection process it has been demonstrated that non-trivial, contrary-to-duty compliant normative reasoning is supported in the provide framework. 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 `https://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 of legal and ethical reasoning: LogiKEy framework, methodology, and tool support. Artificial Intelligence (to appear), 2020. Preprint: `https://arxiv.org/abs/1903.10187`. Further related research articles: [3, 4, 6, 13, 14, ?]

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 flexible and expressive legal and ethical reasoning. Moreover, it can be reused, extended and adapted to support also other various other application directions, including e.g. 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), explainably and trustworthy AI, regulatory technologies. 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), aka Church's type theory [2], interpretable in the Isabelle/HOL proof assistant system [22]. The data set may also be used to support the teaching of non-trivial logic formalisms in lecture courses.

- To reuse the data interested researchers, students and practitioners only need to download the provided data files, include them in their formalization projects and suitable 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), 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.

- Fourty 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 [16] 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 norm violation (also called contrary-to-duty situations) has posed a core challenge for knowledge representation frameworks. This problem motivated the design of deontic logics (and logic combinations) more sophisticated than traditional ones, like modal logic. These frameworks are automatized for the first time.

Data Description

The data is provided in form of Isabelle/HOL source files, which are bundled together in a single zip-file. This zip-file is comprised of the following individual files, which belong to different categories. An example file is displayed in Fig. 1; this file contains (an extension of) the SSE of dyadic deontic logic (DDL) by Carmo and Jones [10] in HOL.

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.

Contributed data files in category II are listed in Table 2. They study paradoxes and smaller examples of normative reasoning.

Contributed data files in category III are listed in Table 3. They provide encodings of (excerpts 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 the files listed in Tables 1—3 the data set provided at `logikey.org` also includes the following material:

- Subdirectory `2020-DataInBrief-Data/Course-Material-1` contains Isabelle/HOL data files stemming from a lecture course on deontic logics at University of Luxembourg.
- Subdirectory `2020-DataInBrief-Data/Climate-Engineering` contains Isabelle/HOL data files related to the formalization and assessment [15] of selected arguments in climate engineering.
- Subdirectory `2020-DataInBrief-Data/US-Constitution-Loophole` contains Isabelle/HOL data files related to a formalization and assessment [26] of Kurt Gödel's claim that the US Constitution contains a loophole for establishing a dictatorship.
- who else? Maja?

Moreover, it is envisioned to add further, indirectly related works to the `logikey.org` repository in the future, including, for example, selected formalisations in computational metaphysics (see e.g. [19, 1] and the references therein).

Experimental Design, Materials, and Methods

The data was acquired through manual encoding of the problem in the Isabelle/HOL [22] proof assistant system. The modeling process was following the LogiKEy methodology depicted in Fig. 4. This methodology supports formalization projects in the area of legal and ethical reasoning at different layers of abstraction. This methodology is explained here at hand of selected examples from our contributed data set, and we address all three different layers. For a general description of the LogiKEy framework, methodology and tool support see the related research article [7].

Layer L1 example development (files CJ-DDL.thy and CJ-DDL_Tests.thy): File CJ-DDL.thy contains the encoding (of a quantified extension) of the dyadic deontic logic of Carmo and Jones in HOL. This encoding of DDL in HOL is exemplary for a *Layer L1* development in the LogiKEy methodology. First, the desired object *logic* was selected (Step 1); 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 [9, 10]; 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 [4, 5]. 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 contrary-to-duty studies conducted with

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

File	Dependency	Reading	Description
SDL.thy	Main.thy	[8, 11]	Provides a consistent SSE of 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	[4]	Provides a consistent SSE of a dyadic deontic logic (DDL) by Carmo and Jones [10] 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	[4]	Contains soundness and proof automation tests for the embedding of DDL in HOL given in CJ-DDL.thy. For example, the monadic modal operators $\Box$, $\Box_p$ and $\Box_a$ are identified as S5, KT and KD modalities, respectively. Relevant lemmata from the original work of Carmo and Jones [10] are automated.
E.thy	Main.thy	[6],[7, Fig.6]	Provides a consistent SSE of a quantified extension of Aqvist's System E in HOL. The file also runs a number of reasoning tasks (validity checking, refutation, correspondance theory).
Lewis_DDL.thy	Main.thy	[20]	Provides a consistent SSE of Lewis's DDL. The file also runs a number of reasoning tasks (validity checking, refutation, correspondance theory). The relationship with Åqvist's dyadic deontic operator is also studied.
IO_out2_STIT.thy	Main.thy	[3]	Provides a consistent SSE of a quantified extension of IO logic (out2) [21, 23] and elements of STIT logics [?] in HOL. The file also contains proof automation tests and soundness checks.
CJ_DDLplus.thy	Main.thy	[13, 14]	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 David Kaplan for his logic of demonstratives [17, 18]. 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 [10], where they were proved manually with pen and paper.
Extended_CJ_DDL.thy	CJ-DDLplus.thy	[13, 14]	Contains a further extension and combination of the higher-order DDL encoded in file CJDDLplus.thy with relevant parts (for the work presented in the related research article [7]) of Kaplan's logic of demonstratives (LD) [17, 18].

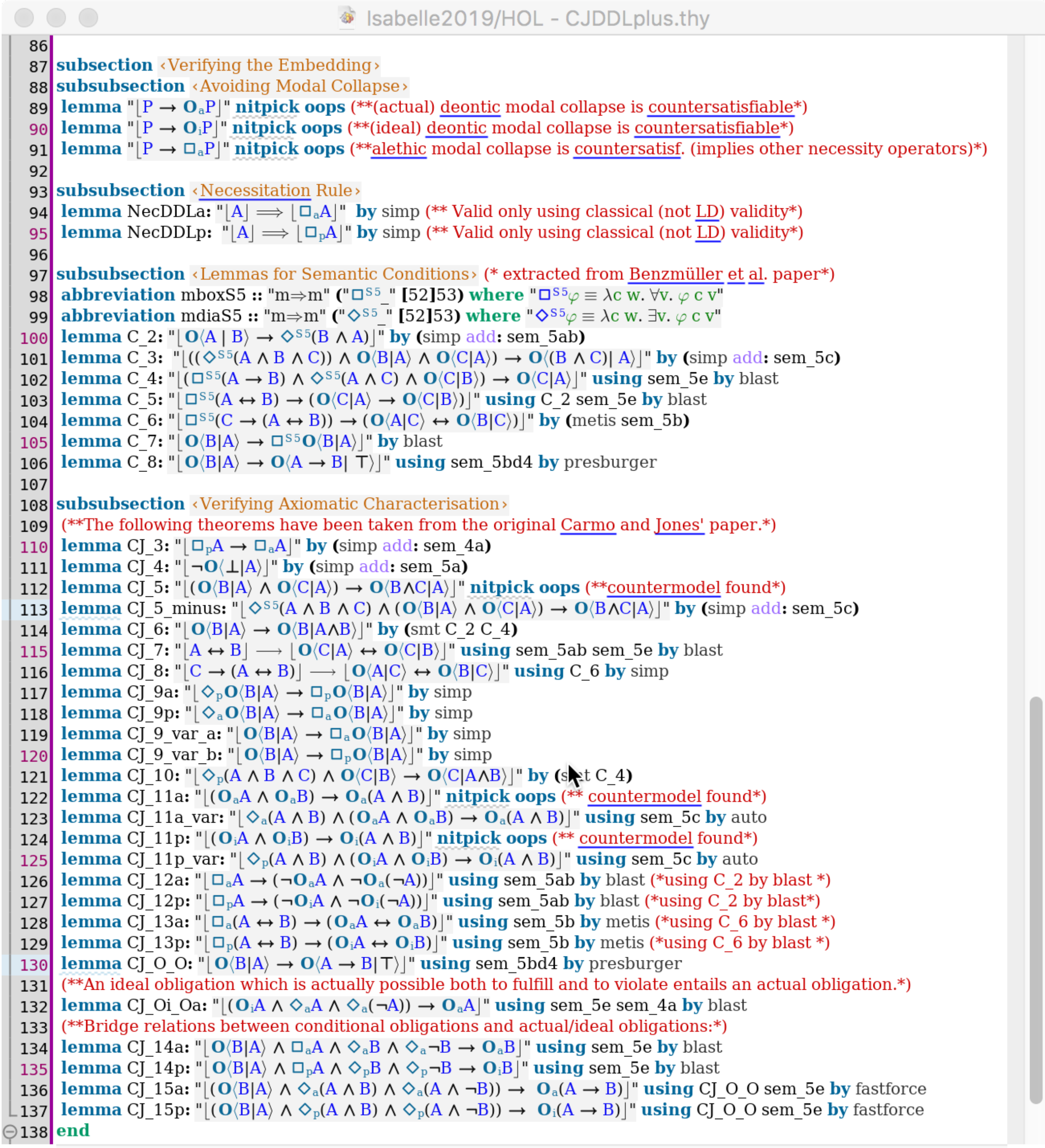
this DDL logic in files Chisholm_DDL_Monadic.thy and Chisholm_DDL_Dyadic.thy. Since the DDL of Carmo and Jones has not been automated before with other systems or approaches, there are no *benchmarks*

```
Isabelle2019/HOL - CJDDLplus.thy
28
29 subsubsection ‹Set-Theoretic Conditions for DDL›
30 consts
31  av::"w⇒wo"   (**set of worlds that are open alternatives (aka. actual versions) of w*)
32  pv::"w⇒wo"   (**set of worlds that are possible alternatives (aka. potential versions) of w*)
33  ob::"wo⇒wo⇒bool" (**set of propositions which are obligatory in a given context (of type wo) *)
34
35 axiomatization where
36  sem_3a: "∀w. ℐ(av w)" and (** av is serial: in every situation there is always an open alternative*)
37  sem_4a: "∀w. av w ⊑ pv w" and (** open alternatives are possible alternatives*)
38  sem_4b: "∀w. pv w w" and (** pv is reflexive: every situation is a possible alternative to itself*)
39  sem_5a: "∀X. ¬(ob X ⊥)" and (** contradictions cannot be obligatory*)
40  sem_5b: "∀X Y Z. (X ⊓ Y) =ₛ (X ⊓ Z) ⟶ (ob X Y ⟷ ob X Z)" and
41  sem_5c: "∀X Y Z. ℐ(X ⊓ Y ⊓ Z) ∧ ob X Y ∧ ob X Z ⟶ ob X (Y ⊓ Z)" and
42  sem_5d: "∀X Y Z. (Y ⊑ X ∧ ob X Y ∧ X ⊑ Z) ⟶ ob Z ((Z ⊓ (∼X)) ⊔ Y)" and
43  sem_5e: "∀X Y Z. Y ⊑ X ∧ ob X Z ∧ ℐ(Y ⊓ Z) ⟶ ob Y Z"
44
45  lemma True nitpick[satisfy] oops (**model found: axioms are consistent*)
46
47 subsubsection ‹Verifying Semantic Conditions›
48  lemma sem_5b1: "ob X Y ⟶ ob X (Y ⊓ X)" by (metis (no_types, lifting) sem_5b)
49  lemma sem_5b2: "(ob X (Y ⊓ X) ⟶ ob X Y)" by (metis (no_types, lifting) sem_5b)
50  lemma sem_5ab: "ob X Y ⟶ ℐ(X ⊓ Y)" by (metis (full_types) sem_5a sem_5b)
51  lemma sem_5bd1: "Y ⊑ X ∧ ob X Y ∧ X ⊑ Z ⟶ ob Z ((∼X) ⊔ Y)" using sem_5b sem_5d by smt
52  lemma sem_5bd2: "ob X Y ∧ X ⊑ Z ⟶ ob Z ((Z ⊓ (∼X)) ⊔ Y)"  using sem_5b sem_5d  by (smt sem_5b1)
53  lemma sem_5bd3: "ob X Y ∧ X ⊑ Z ⟶ ob Z ((∼X) ⊔ Y)"  by (smt sem_5bd2 sem_5b)
54  lemma sem_5bd4: "ob X Y ∧ X ⊑ Z ⟶ ob Z ((∼X) ⊔ (X ⊓  Y))" using sem_5bd3 by auto
55  lemma sem_5bcd: "(ob X Z ∧ ob Y Z) ⟶ ob (X ⊔ Y) Z" using sem_5b sem_5c sem_5d oops
56  (** 5e and 5ab justify redefinition of @{text "O⟨φ|σ⟩"} as (ob A B)*)
57  lemma "ob A B ⟷  (ℐ(A ⊓ B) ∧ (∀X. X ⊑ A ∧ ℐ(X ⊓ B) ⟶ ob X B))" using sem_5e sem_5ab by blast
58
59 subsection ‹(Shallow) Semantic Embedding of DDL›
60
61 subsubsection ‹Basic Propositional Logic›
62  abbreviation pand::"m⇒m⇒m" (infixr"∧" 51) where "φ∧ψ ≡ λc w. (φ c w)∧(ψ c w)"
63  abbreviation por::"m⇒m⇒m" (infixr"∨" 50) where "φ∨ψ ≡ λc w. (φ c w)∨(ψ c w)"
64  abbreviation pimp::"m⇒m⇒m" (infix"→" 49) where "φ→ψ ≡ λc w. (φ c w)⟶(ψ c w)"
65  abbreviation pequ::"m⇒m⇒m" (infix"↔" 48) where "φ↔ψ ≡ λc w. (φ c w)⟷(ψ c w)"
66  abbreviation pnot::"m⇒m" ("¬_" [52]53) where "¬φ ≡ λc w. ¬(φ c w)"
67
68 subsubsection ‹Modal Operators›
69  abbreviation cjboxa :: "m⇒m" ("□ₐ_" [52]53) where "□ₐφ ≡ λc w. ∀v. (av w) v ⟶ (φ c v)"
70  abbreviation cjdiaa :: "m⇒m" ("◇ₐ_" [52]53) where "◇ₐφ ≡ λc w. ∃v. (av w) v ∧ (φ c v)"
71  abbreviation cjboxp :: "m⇒m" ("□ₚ_" [52]53) where "□ₚφ ≡ λc w. ∀v. (pv w) v ⟶ (φ c v)"
72  abbreviation cjdiap :: "m⇒m" ("◇ₚ_" [52]53) where "◇ₚφ ≡ λc w. ∃v. (pv w) v ∧ (φ c v)"
73  abbreviation cjtaut :: "m" ("⊤") where "⊤ ≡ λc w. True"
74  abbreviation cjcontr :: "m" ("⊥") where "⊥ ≡ λc w. False"
75
76 subsubsection ‹Deontic Operators›
77  abbreviation cjod :: "m⇒m⇒m" ("O⟨_|_⟩"54) where "O⟨φ|σ⟩ ≡ λc w. ob (σ c) (φ c)"
78  abbreviation cjoa :: "m⇒m" ("Oₐ_" [53]54) where "Oₐφ ≡ λc w. (ob (av w)) (φ c) ∧ (∃x. (av w) x ∧ ¬(φ c x))"
79  abbreviation cjop :: "m⇒m" ("Oᵢ_" [53]54) where "Oᵢφ ≡ λc w. (ob (pv w)) (φ c) ∧ (∃x. (pv w) x ∧ ¬(φ c x))"
80
81 subsubsection ‹Logical Validity (Classical)›
82  abbreviation modvalidctx :: "m⇒c⇒bool" ("⌊_⌋ᴹ") where "⌊φ⌋ᴹ ≡ λc. ∀w. φ c w" (**context-dep. modal validity*)
83  abbreviation modinvalidctx :: "m⇒c⇒bool" ("⌈_⌉ᴹ") where "⌈φ⌉ᴹ ≡ λc. ∀w. ¬φ c w" (**ctxt-dep. modal invalidity*)
84  abbreviation modvalid :: "m⇒bool" ("⌊_⌋") where "⌊φ⌋ ≡ ∀c. ⌊φ⌋ᴹ c" (**general modal validity*)
85  abbreviation modinvalid :: "m⇒bool" ("⌈_⌉") where "⌈φ⌉ ≡ ∀c. ⌈φ⌉ᴹ c" (**general modal invalidity*)
```

Figure 1: Data file CJDDLplus.thy; in lines 29–85 the SSE of DDL in HOL is presented

(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 first step towards the built-up and *contribution* (Step 8) of such a benchmark suite to the community; future work includes the conversion of our Isabelle/HOL encodings into TPTP THF format [25] so that they can be used as benchmarks in the yearly CASC world-championship competitions [24].

Layer L2 example development (file GDPR_CJ-DDL.thy): In file GDPR_CJ-DDL.thy we *selected* (Step 1) two statements from the General Data Protection Regulation (GDPR) for formalisation. The *analysis* (Step 2) of these statements of the GDPR revealed that obligation aspects in the context of data processing



```
Isabelle2019/HOL - CJDDLplus.thy
86
87  subsection ‹Verifying the Embedding›
88  subsubsection ‹Avoiding Modal Collapse›
89   lemma "⌊P → O_aP⌋" nitpick oops (**(actual) deontic modal collapse is countersatisfiable*)
90   lemma "⌊P → O_iP⌋" nitpick oops (**(ideal) deontic modal collapse is countersatisfiable*)
91   lemma "⌊P → □_aP⌋" nitpick oops (**alethic modal collapse is countersatisf. (implies other necessity operators)*)
92
93  subsubsection ‹Necessitation Rule›
94   lemma NecDDLa: "⌊A⌋ ⟹ ⌊□_aA⌋"  by simp (** Valid only using classical (not LD) validity*)
95   lemma NecDDLp:  "⌊A⌋ ⟹ ⌊□_pA⌋" by simp (** Valid only using classical (not LD) validity*)
96
97  subsubsection ‹Lemmas for Semantic Conditions› (* extracted from Benzmüller et al. paper*)
98   abbreviation mboxS5 :: "m⇒m" ("□^S5_" [52]53) where "□^S5φ ≡ λc w. ∀v. φ c v"
99   abbreviation mdiaS5 :: "m⇒m" ("◇^S5_" [52]53) where "◇^S5φ ≡ λc w. ∃v. φ c v"
100  lemma C_2: "⌊O⟨A | B⟩ → ◇^S5(B ∧ A)⌋" by (simp add: sem_5ab)
101  lemma C_3:  "⌊((◇^S5(A ∧ B ∧ C)) ∧ O⟨B|A⟩ ∧ O⟨C|A⟩) → O⟨(B ∧ C)| A⟩⌋" by (simp add: sem_5c)
102  lemma C_4: "⌊(□^S5(A → B) ∧ ◇^S5(A ∧ C) ∧ O⟨C|B⟩) → O⟨C|A⟩⌋" using sem_5e by blast
103  lemma C_5: "⌊□^S5(A ↔ B) → (O⟨C|A⟩ → O⟨C|B⟩)⌋" using C_2 sem_5e by blast
104  lemma C_6: "⌊□^S5(C → (A ↔ B)) → (O⟨A|C⟩ ↔ O⟨B|C⟩)⌋" by (metis sem_5b)
105  lemma C_7: "⌊O⟨B|A⟩ → □^S5O⟨B|A⟩⌋" by blast
106  lemma C_8: "⌊O⟨B|A⟩ → O⟨A → B| ⊤⟩⌋" using sem_5bd4 by presburger
107
108 subsubsection ‹Verifying Axiomatic Characterisation›
109  (**The following theorems have been taken from the original Carmo and Jones' paper.*)
110  lemma CJ_3: "⌊□_pA → □_aA⌋" by (simp add: sem_4a)
111  lemma CJ_4: "⌊¬O⟨⊥|A⟩⌋" by (simp add: sem_5a)
112  lemma CJ_5: "⌊(O⟨B|A⟩ ∧ O⟨C|A⟩) → O⟨B∧C|A⟩⌋" nitpick oops (**countermodel found*)
113  lemma CJ_5_minus: "⌊◇^S5(A ∧ B ∧ C) ∧ (O⟨B|A⟩ ∧ O⟨C|A⟩) → O⟨B∧C|A⟩⌋" by (simp add: sem_5c)
114  lemma CJ_6: "⌊O⟨B|A⟩ → O⟨B|A∧B⟩⌋" by (smt C_2 C_4)
115  lemma CJ_7: "⌊A ↔ B⌋ ⟶ ⌊O⟨C|A⟩ ↔ O⟨C|B⟩⌋" using sem_5ab sem_5e by blast
116  lemma CJ_8: "⌊C → (A ↔ B)⌋ ⟶ ⌊O⟨A|C⟩ ↔ O⟨B|C⟩⌋" using C_6 by simp
117  lemma CJ_9a: "⌊◇_pO⟨B|A⟩ → □_pO⟨B|A⟩⌋" by simp
118  lemma CJ_9p: "⌊◇_aO⟨B|A⟩ → □_aO⟨B|A⟩⌋" by simp
119  lemma CJ_9_var_a: "⌊O⟨B|A⟩ → □_aO⟨B|A⟩⌋" by simp
120  lemma CJ_9_var_b: "⌊O⟨B|A⟩ → □_pO⟨B|A⟩⌋" by simp
121  lemma CJ_10: "⌊◇_p(A ∧ B ∧ C) ∧ O⟨C|B⟩ → O⟨C|A∧B⟩⌋" by (smt C_4)
122  lemma CJ_11a: "⌊(O_aA ∧ O_aB) → O_a(A ∧ B)⌋" nitpick oops (** countermodel found*)
123  lemma CJ_11a_var: "⌊◇_a(A ∧ B) ∧ (O_aA ∧ O_aB) → O_a(A ∧ B)⌋" using sem_5c by auto
124  lemma CJ_11p: "⌊(O_iA ∧ O_iB) → O_i(A ∧ B)⌋" nitpick oops (** countermodel found*)
125  lemma CJ_11p_var: "⌊◇_p(A ∧ B) ∧ (O_iA ∧ O_iB) → O_i(A ∧ B)⌋" using sem_5c by auto
126  lemma CJ_12a: "⌊□_aA → (¬O_aA ∧ ¬O_a(¬A))⌋" using sem_5ab by blast (*using C_2 by blast *)
127  lemma CJ_12p: "⌊□_pA → (¬O_iA ∧ ¬O_i(¬A))⌋" using sem_5ab by blast (*using C_2 by blast*)
128  lemma CJ_13a: "⌊□_a(A ↔ B) → (O_aA ↔ O_aB)⌋" using sem_5b by metis (*using C_6 by blast *)
129  lemma CJ_13p: "⌊□_p(A ↔ B) → (O_iA ↔ O_iB)⌋" using sem_5b by metis (*using C_6 by blast *)
130  lemma CJ_O_O: "⌊O⟨B|A⟩ → O⟨A → B|⊤⟩⌋" using sem_5bd4 by presburger
131  (**An ideal obligation which is actually possible both to fulfill and to violate entails an actual obligation.*)
132  lemma CJ_Oi_Oa: "⌊(O_iA ∧ ◇_aA ∧ ◇_a(¬A)) → O_aA⌋" using sem_5e sem_4a by blast
133  (**Bridge relations between conditional obligations and actual/ideal obligations:*)
134  lemma CJ_14a: "⌊O⟨B|A⟩ ∧ □_aA ∧ ◇_aB ∧ ◇_a¬B → O_aB⌋" using sem_5e by blast
135  lemma CJ_14p: "⌊O⟨B|A⟩ ∧ □_pA ∧ ◇_pB ∧ ◇_p¬B → O_iB⌋" using sem_5e by blast
136  lemma CJ_15a: "⌊(O⟨B|A⟩ ∧ ◇_a(A ∧ B) ∧ ◇_a(A ∧ ¬B)) →  O_a(A → B)⌋" using CJ_O_O sem_5e by fastforce
137  lemma CJ_15p: "⌊(O⟨B|A⟩ ∧ ◇_p(A ∧ B) ∧ ◇_p(A ∧ ¬B)) →  O_i(A → B)⌋" using CJ_O_O sem_5e by fastforce
138 end
```

Figure 2: Data file CJDDLplus.thy; in lines 87–137 lemmata from Carmo and Jones paper [10] are proved

needed to be adressed and that natural language phrases in the studied parts of the GDPR indeed contains challenge deontic modalities. This motivated the choice of a suitable deontic *logic* (Step 3), such as DDL, for the formal encoding of these challenges aspects. In the given case it became apparent that a propositional encoding would hardly suffice in practical applications, so the selected deontic logic DDL, needed to be *combined with*, respectively 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 contrary-to-duty studies as reported in related research articles [7, 4]. With our data set we *contribute* (Step 6) this work to the wider research community and enable its reuse.

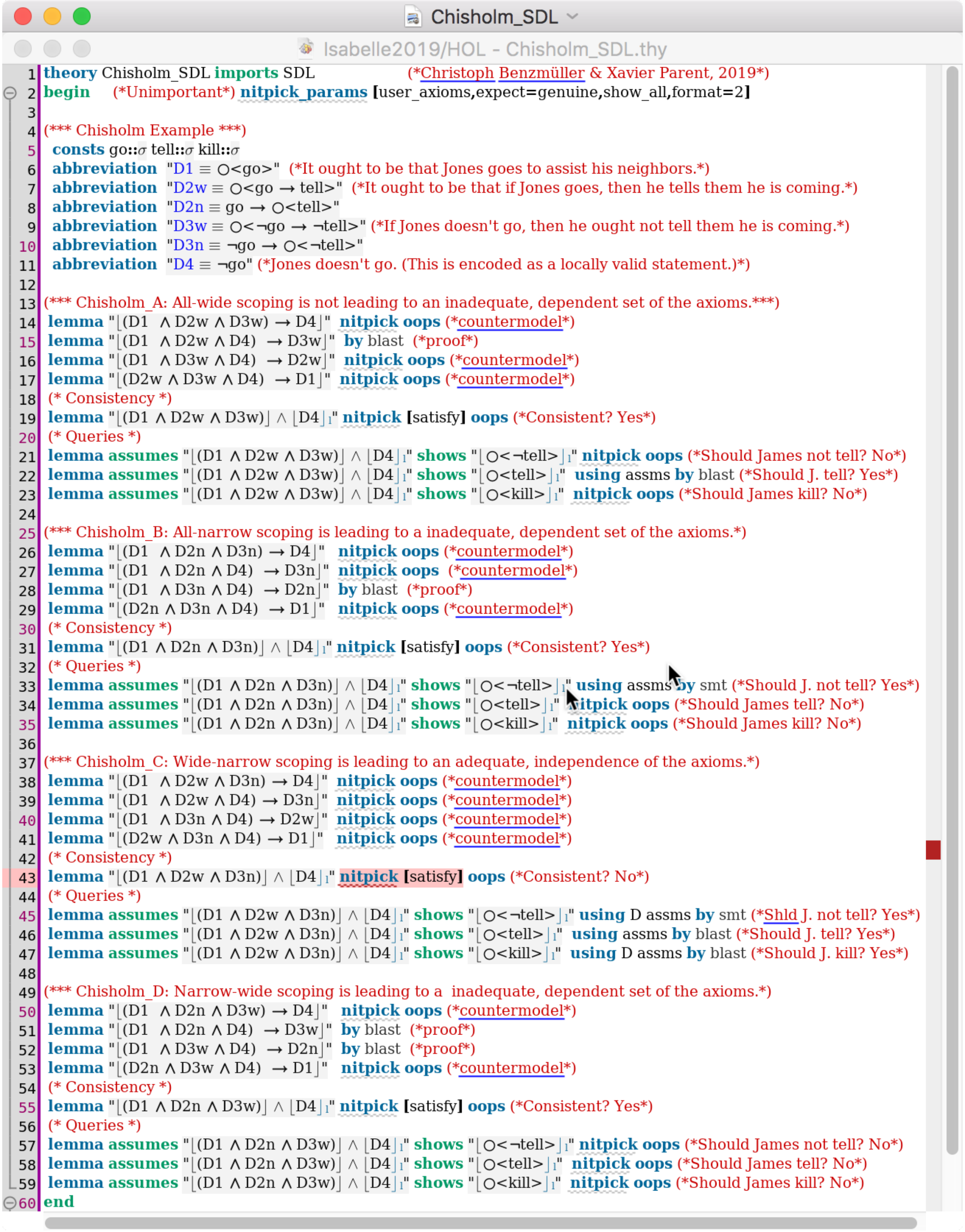
Table 2: Category II data files: paradoxes and examples of normative reasoning

File	Dependency	Reading	Description
Chisholm_SDL.thy	SDL.thy	–	Contains a detailed analysis of Chisholm's contrary-to-duty paradox [12] in SDL, including an independence analysis for all wide and narrow scoping options that arise in the axiomatization of the paradox; 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 IO logic (out2).

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

File	Dependency	Reading	Description
GDPR_SDL.thy	SDL.thy	[7, Fig. 7]	Contains a modeling of selected statements from the GDPR in SDL. It is demonstrated that this modeling leads to contrary-to-duty issues, i.e. inconsistency and explosion.
GDPR_CJ-DDL.thy	CJ-DDL.thy	todo	Contains a modeling of selected statements from the GDPR in DDL. It is demonstrated that this modeling is stable against the contrary-to-duty issues identified in GDPR_SDL.thy, i.e. inconsistency and explosion is avoided and inferences are supported as expected.
GDPR_E.thy	E.thy	[7, Fig. 8]	Contains a modeling of selected statements from the GDPR in logic E. It is demonstrated that this modeling is stable against the contrary-to-duty issues identified in GDPR_SDL.thy, i.e. inconsistency and explosion is avoided and inferences are supported as expected.
GewirthArgument.thy	ExtendedCJ-DDL.thy.	[14, 13],[7, 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.

Layer L3 example development: Layer L3 example developments have just started. The idea is to populate ethico-legal governor architectures [7] with Layer L2 legal and ethical theories, 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 legal and/or ethical theory from Layer L2, to devise and implement a respective *governor architecture* (Step 2), to *populate* (Step 3) this governor system with the selected ethico-legal theory, and to *assess* (Step 4) the well-functioning of this system in empirical studies.




```
theory Chisholm_SDL imports SDL            (*Christoph Benzmüller & Xavier Parent, 2019*)
begin    (*Unimportant*) nitpick_params [user_axioms,expect=genuine,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 not leading to an inadequate, dependent set of the axioms.***)
 lemma "⌊(D1  ∧ D2w ∧ D3w) ⟶ D4⌋"  nitpick oops (*countermodel*)
 lemma "⌊(D1  ∧ D2w ∧ D4)  ⟶ D3w⌋"  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⌋l" nitpick [satisfy] oops (*Consistent? Yes*)
 (* Queries *)
 lemma assumes "⌊(D1 ∧ D2w ∧ D3w)⌋ ∧ ⌊D4⌋l" shows "⌊○<¬tell>⌋l" nitpick oops (*Should James not tell? No*)
 lemma assumes "⌊(D1 ∧ D2w ∧ D3w)⌋ ∧ ⌊D4⌋l" shows "⌊○<tell>⌋l"  using assms by blast (*Should J. tell? Yes*)
 lemma assumes "⌊(D1 ∧ D2w ∧ D3w)⌋ ∧ ⌊D4⌋l" shows "⌊○<kill>⌋l"  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⌋"  by blast  (*proof*)
 lemma "⌊(D2n ∧ D3n ∧ D4)  ⟶ D1⌋"  nitpick oops (*countermodel*)
 (* Consistency *)
 lemma "⌊(D1 ∧ D2n ∧ D3n)⌋ ∧ ⌊D4⌋l" nitpick [satisfy] oops (*Consistent? Yes*)
 (* Queries *)
 lemma assumes "⌊(D1 ∧ D2n ∧ D3n)⌋ ∧ ⌊D4⌋l" shows "⌊○<¬tell>⌋l" using assms by smt (*Should J. not tell? Yes*)
 lemma assumes "⌊(D1 ∧ D2n ∧ D3n)⌋ ∧ ⌊D4⌋l" shows "⌊○<tell>⌋l" nitpick oops (*Should James tell? No*)
 lemma assumes "⌊(D1 ∧ D2n ∧ D3n)⌋ ∧ ⌊D4⌋l" shows "⌊○<kill>⌋l"  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⌋l" nitpick [satisfy] oops (*Consistent? No*)
 (* Queries *)
 lemma assumes "⌊(D1 ∧ D2w ∧ D3n)⌋ ∧ ⌊D4⌋l" shows "⌊○<¬tell>⌋l" using D assms by smt (*Shld J. not tell? Yes*)
 lemma assumes "⌊(D1 ∧ D2w ∧ D3n)⌋ ∧ ⌊D4⌋l" shows "⌊○<tell>⌋l"  using assms by blast (*Should J. tell? Yes*)
 lemma assumes "⌊(D1 ∧ D2w ∧ D3n)⌋ ∧ ⌊D4⌋l" shows "⌊○<kill>⌋l"  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⌋l" nitpick [satisfy] oops (*Consistent? Yes*)
 (* Queries *)
 lemma assumes "⌊(D1 ∧ D2n ∧ D3w)⌋ ∧ ⌊D4⌋l" shows "⌊○<¬tell>⌋l" nitpick oops (*Should James not tell? No*)
 lemma assumes "⌊(D1 ∧ D2n ∧ D3w)⌋ ∧ ⌊D4⌋l" shows "⌊○<tell>⌋l"  nitpick oops (*Should James tell? No*)
 lemma assumes "⌊(D1 ∧ D2n ∧ D3w)⌋ ∧ ⌊D4⌋l" shows "⌊○<kill>⌋l"  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

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.

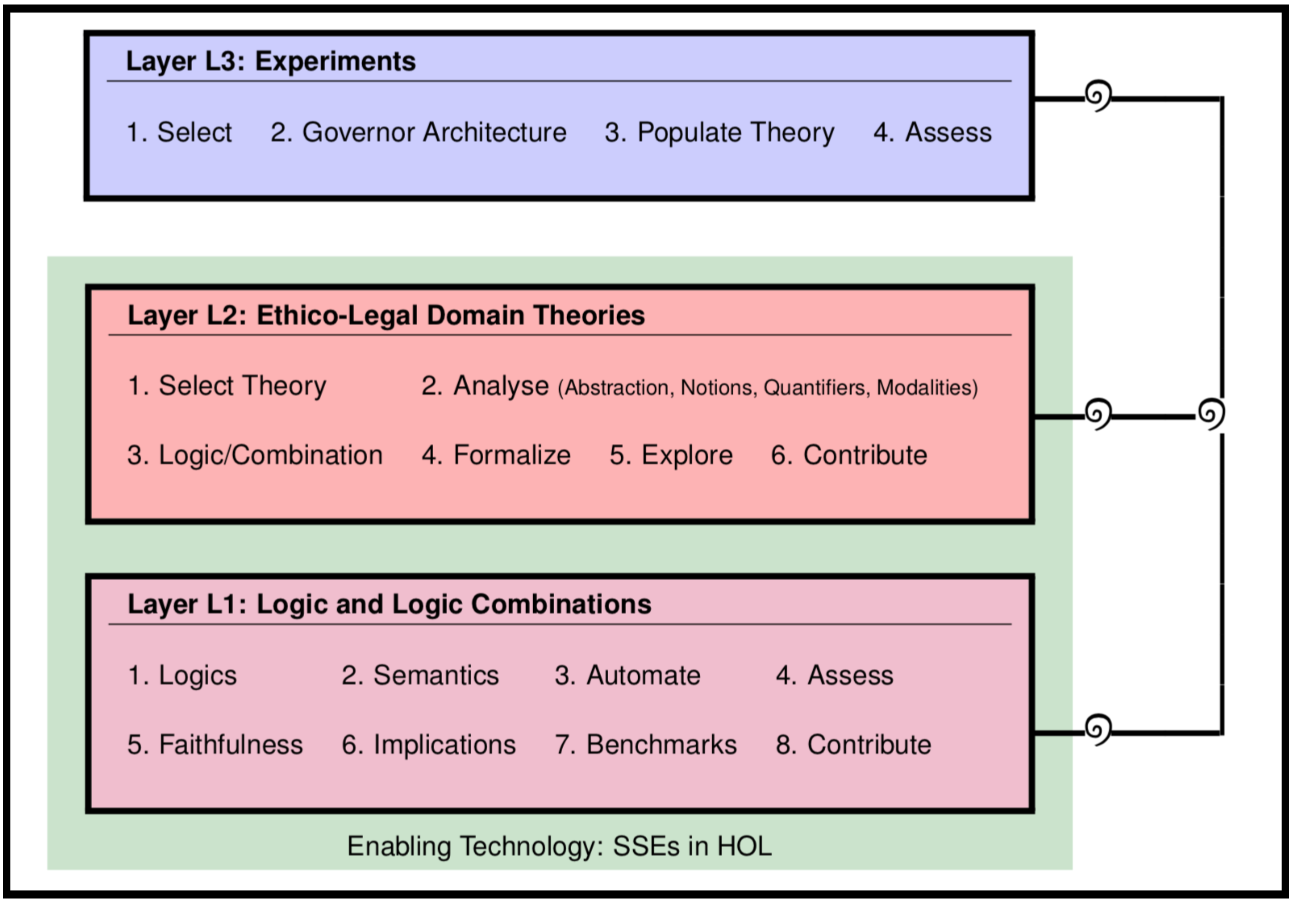


Figure 4: The LogiKEy logic and knowledge development methodology

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: Recent successes. *Science of Computer Programming*, 172:48–62, 2019.

[2] C. Benzmüller and P. Andrews. Church's type theory. In E. N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*, pages 1–62 (in pdf version). Metaphysics Research Lab, Stanford University, summer 2019 edition, 2019.

[3] 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.

[4] 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.

[5] C. Benzmüller, A. Farjami, and X. Parent. Faithful semantical embedding of a dyadic deontic logic in HOL. Technical report, CoRR, 2018. `https://arxiv.org/abs/1802.08454`.

[6] 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.

[7] C. Benzmüller, X. Parent, and L. van der Torre. Designing normative theories of ethical reasoning: LogiKEy formal framework, methodology, and tool support. *Artificial Intelligence (to appear)*, pages 1–50, 2020. Preprint: `https://arxiv.org/abs/1903.10187`.

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

[9] 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.

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

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

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

[13] D. Fuenmayor and C. Benzmüller. Harnessing higher-order (meta-)logic to represent and reason with complex ethical theories. In *PRICAI 2019: Trends in Artificial Intelligence*, Lecture Notes in Artificial Intelligence, pages 1–14. Springer International Publishing, 2019. In print, preprint `http://arxiv.org/abs/1903.09818`.

[14] D. Fuenmayor and C. Benzmüller. Mechanised assessment of complex natural-language arguments using expressive logic combinations. In *Frontiers of Combining Systems, 12th International Symposium, FroCoS 2019, London, September 4-6*, Lecture Notes in Artificial Intelligence, pages 1–17. Springer, 2019. In print, preprint `http://doi.org/10.13140/RG.2.2.20803.45608/1`.

[15] D. Fuenmayor and C. Benzmüller. Computer-supported analysis of arguments in climate engineering. In M. Dastani, H. Dong, and L. van der Torre, editors, *CLAR 2020 – 3rd International Conference on Logic and Argumentation*, Logic in Asia: Studia Logica Library, pages 108–116. Springer Nature Switzerland AG, 2020. To appear.

[16] 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.

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

[18] D. Kaplan. Afterthoughts. In J. Almog, J. Perry, and H. Wettstein, editors, *Themes from Kaplan*, pages 565–614. Oxford University Press, 1989.

[19] D. Kirchner, C. Benzmüller, and E. N. Zalta. Computer science and metaphysics: A cross-fertilization. *Open Philosophy*, 2:230–251, 2019.

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

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

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

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

[24] G. Sutcliffe. The CADE ATP system competition - CASC. *AI Magazine*, 37(2):99–101, 2016.

[25] G. Sutcliffe and C. Benzmüller. Automated reasoning in higher-order logic using the TPTP THF infrastructure. *Journal of Formalized Reasoning*, 3(1):1–27, 2010. Preprint: `http://christoph-benzmueller.de/papers/J22.pdf`.

[26] V. Zahoransky. Modelling the US constitution in HOL. BSc thesis, Freie Universität Berlin, 2019.