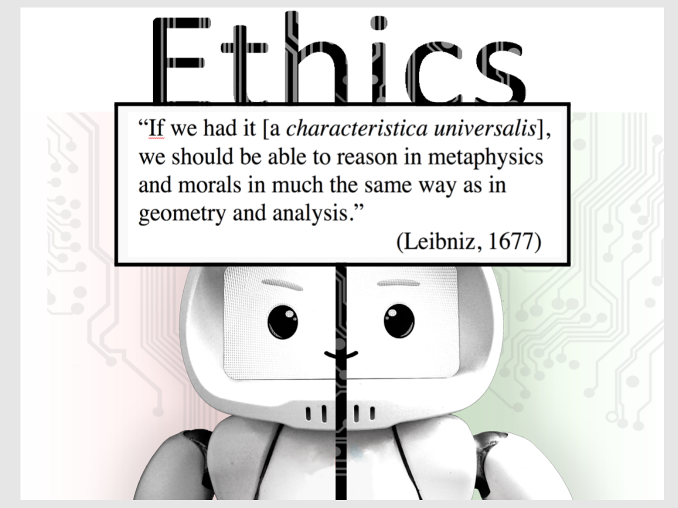
A Deontic Logic Reasoning Infrastructure

Christoph Benzmüller (jww Xavier Parent & Leon van der Torre)
Freie Universität Berlin | University of Luxembourg

HaPoC@CiE 2018, Kiel, 2 Aug 2018 — Celebration of Martin Davis' 90th Birthday

A Deontic Logic Reasoning Infrastructure

Christoph Benzmüller (jww Xavier Parent & Leon van der Torre)
Freie Universität Berlin | University of Luxembourg

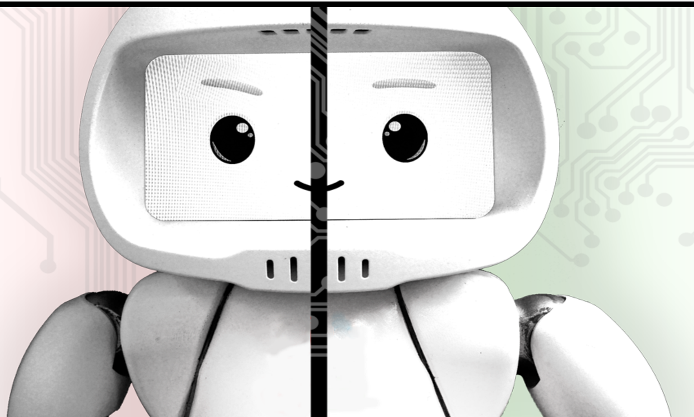


HaPoC@CiE 2018, Kiel, 2 Aug 2018 — Celebration of Martin Davis’ 90th Birthday

Ethics

Peaceful coexistence with **intelligent autonomous systems (IASs)**?

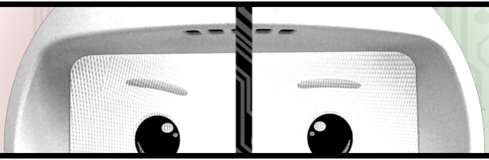
- appropriate forms of **machine-control**
- appropriate forms of **human-machine-interaction**



Ethics

Peaceful coexistence with **intelligent autonomous systems (IASs)**?

- appropriate forms of **machine-control**
- appropriate forms of **human-machine-interaction**



Existing societal processes are based on:

- **rational argumentation & dialog**
- **explicit normative reasoning** (legal & ethical)

Deployment of IASs lacking such competencies? How wise is this?

Ethics

Talk Outline

Foreword: How does Martin Davis fit in?

A **Motivation:** Explicit Ethical Reasoning

B **Technology:** Universal Logical Reasoning in Higher-Order Logic

C **Evidence:** Experiments in Computational Metaphysics

D **Demo(s):** Normative Reasoning Experimentation Platform

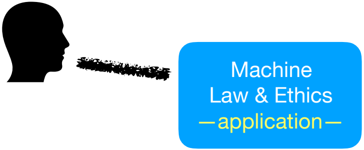
Ethics

Talk Outline

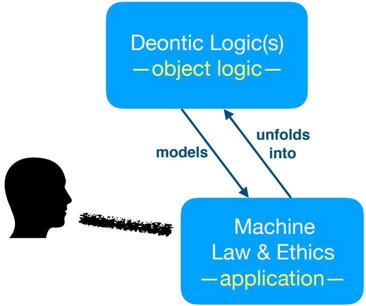
Foreword: How does Martin Davis fit in?

- **A** **Motivation:** Explicit Ethical Reasoning
- **B** **Technology:** Universal Logical Reasoning in Higher-Order Logic
- **C** **Evidence:** Experiments in Computational Metaphysics
- **D** **Demo(s):** Normative Reasoning Experimentation Platform

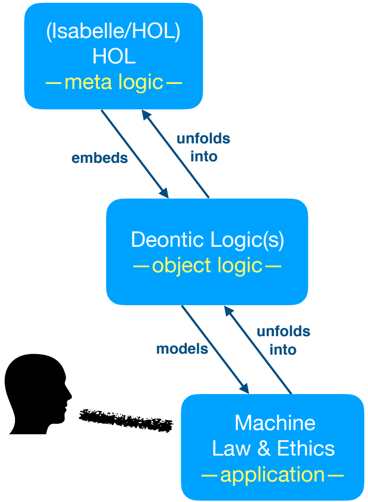
How does Martin Davis fit in?



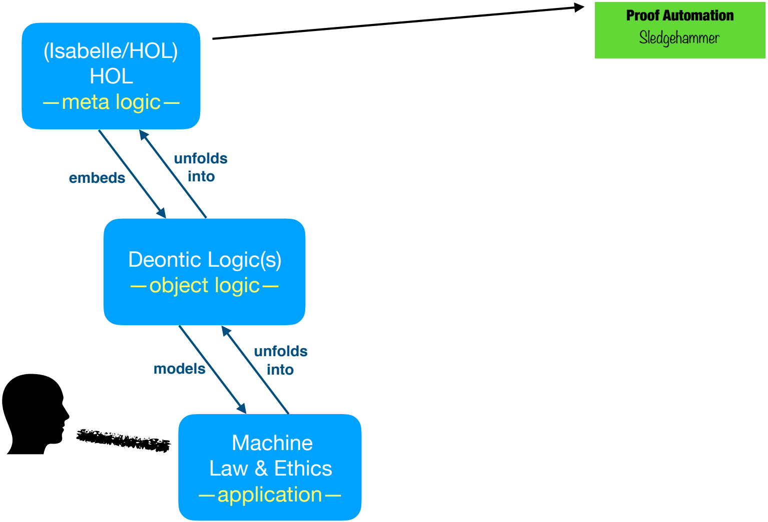
How does Martin Davis fit in?



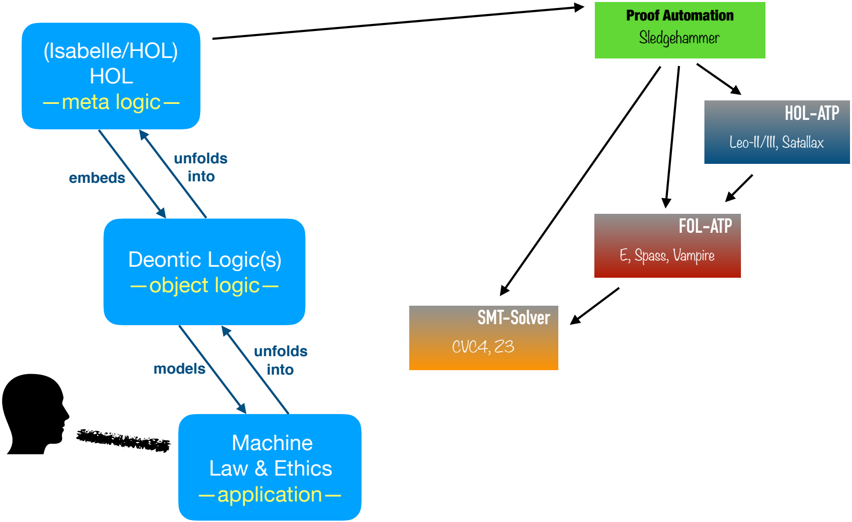
How does Martin Davis fit in?



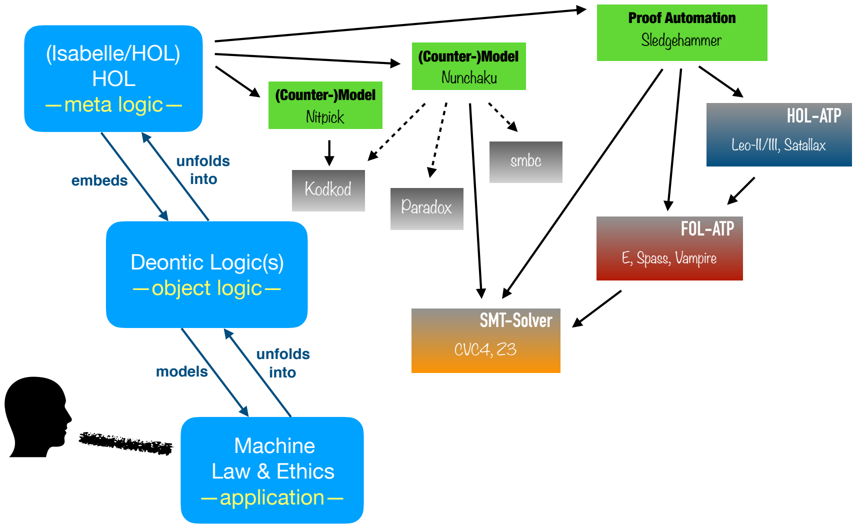
How does Martin Davis fit in?



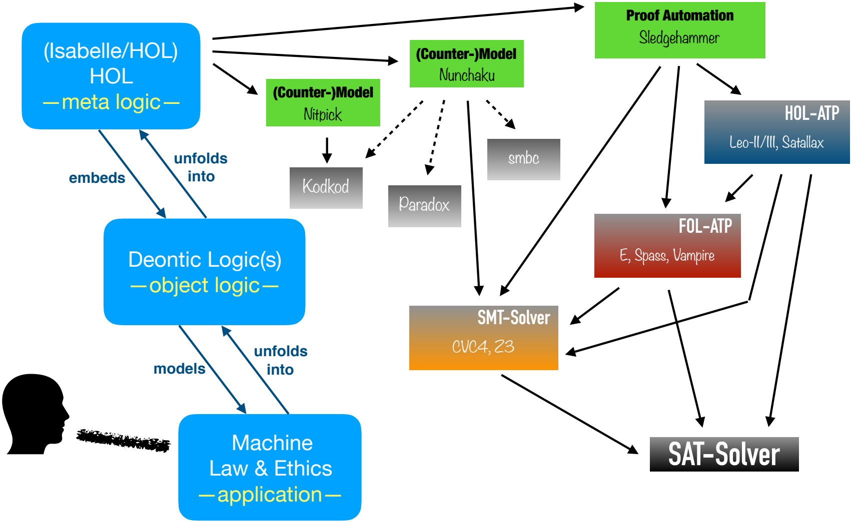
How does Martin Davis fit in?



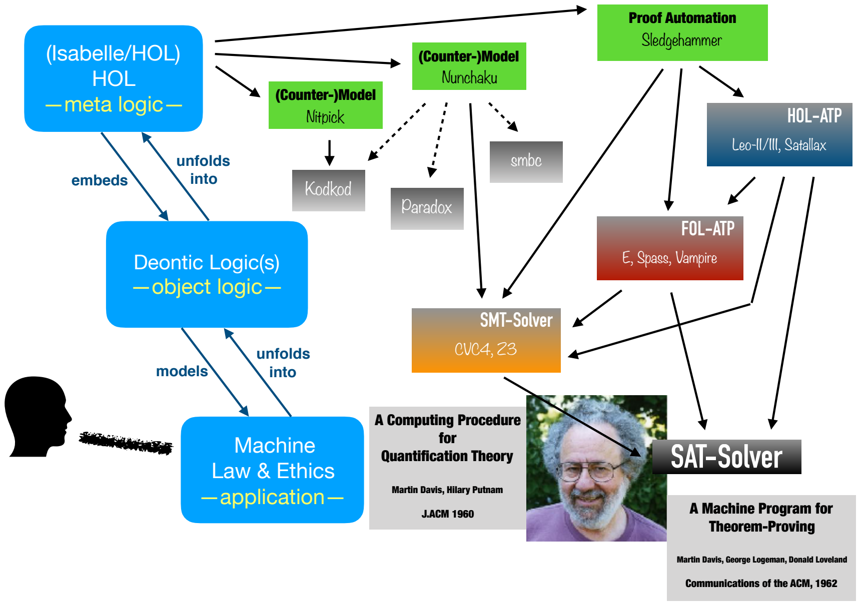
How does Martin Davis fit in?



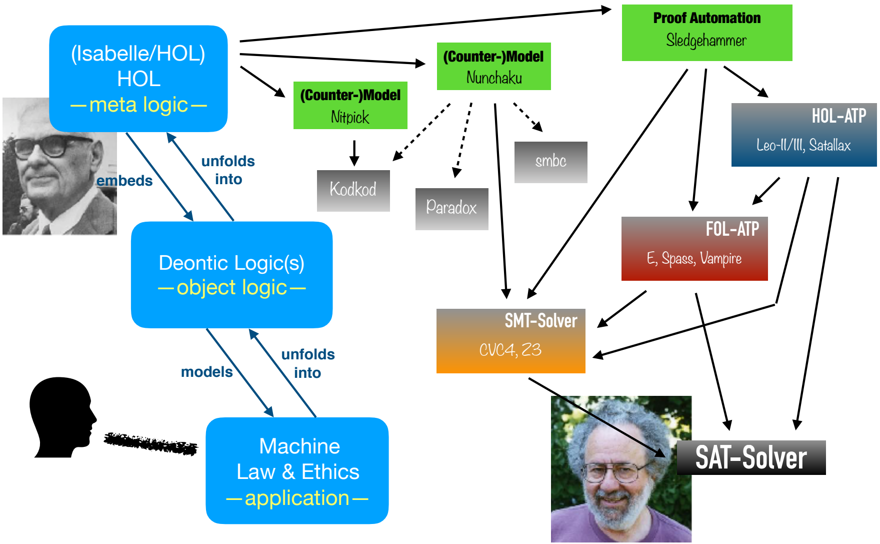
How does Martin Davis fit in?



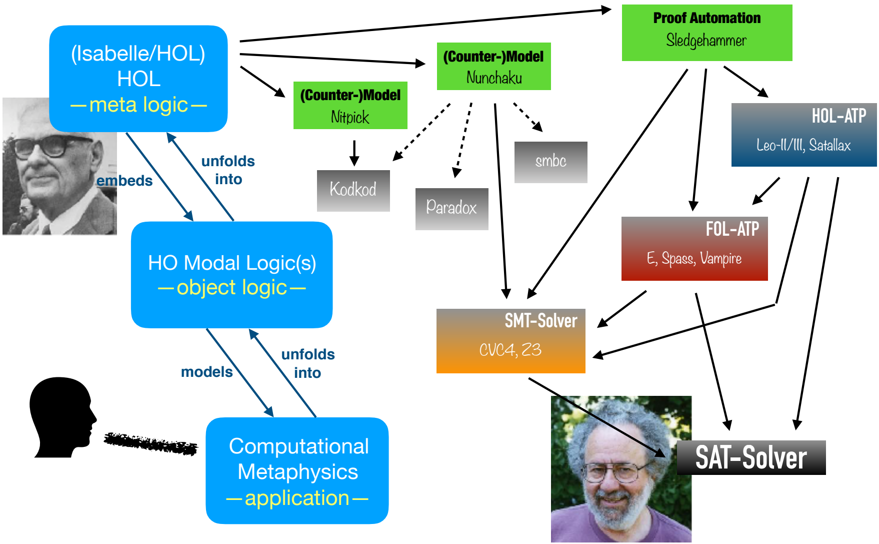
How does Martin Davis fit in?



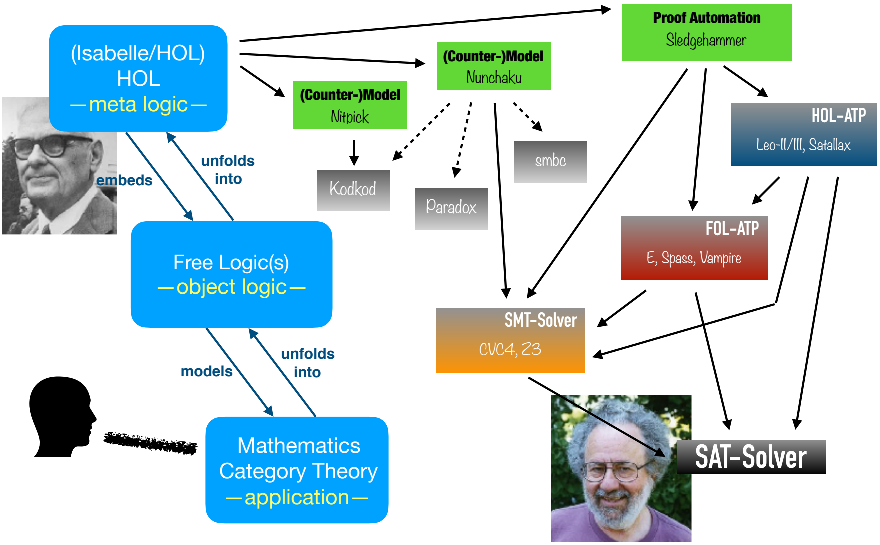
How does Martin Davis fit in?



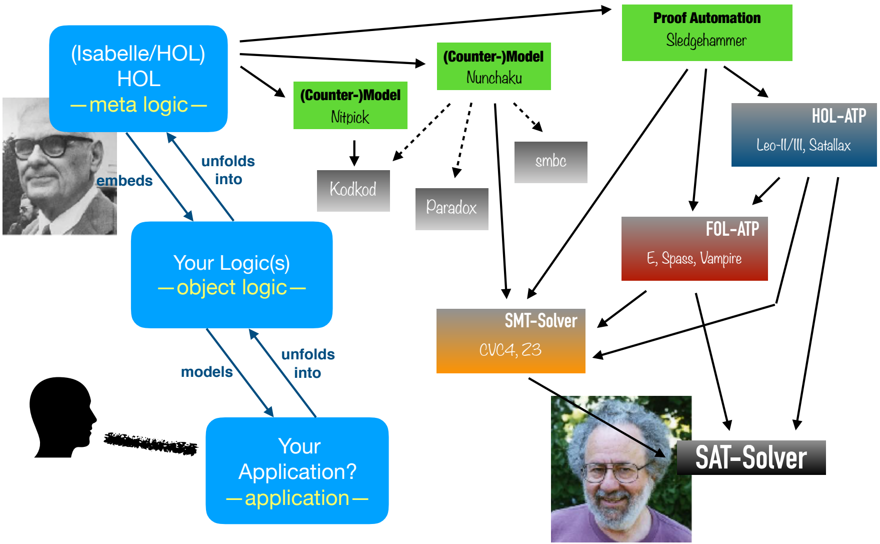
How does Martin Davis fit in?



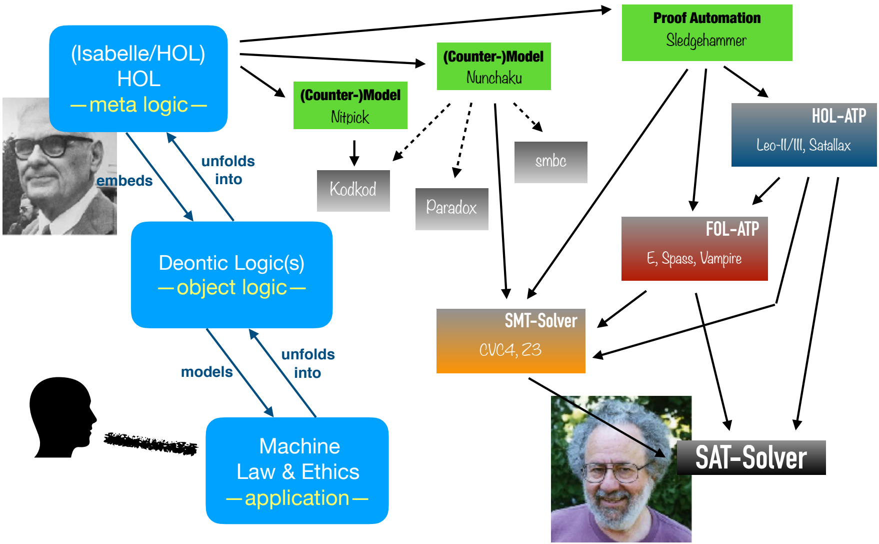
How does Martin Davis fit in?

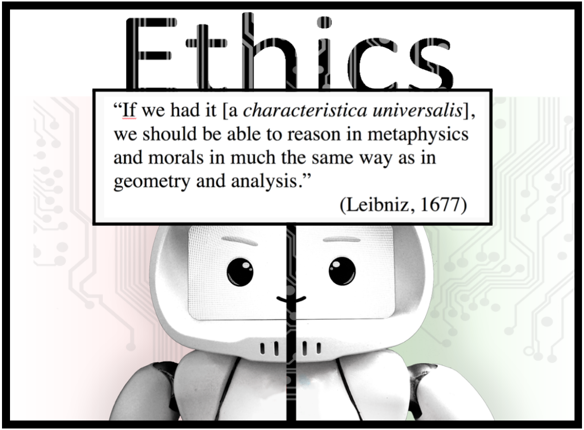


How does Martin Davis fit in?



How does Martin Davis fit in?





Part A — Motivation: Explicit Ethical Reasoning

Motivation

Long-term: Emerging Superintelligence Really? Anyhow ...

- How to prevent Superintelligence from turning against humanity?

Medium-term: Development of pseudo-ethical skills in IASs

- Which norms? Which reasoning principles?
- What architectural design? What functionalities?
- How to implement, deploy and verify?

Different kinds of systems and approaches:

- [Moor, 2009]:
  - – ethical impact agents (ethical consequences to actions)
  - – implicit ethical agents (ethical reactions to given situations)
  - – explicit ethical agents (reasoning with ethical theories/rules)
  - – full ethical agents (conscious, intentional, free will)
- bottom-up vs. top-down
- [DoranEtAl., 2017]:
  opaque — comprehensible — interpretable — explainable AI

Motivation

Long-term: Emerging Superintelligence Really? Anyhow ...

- How to prevent Superintelligence from turning against humanity?

Medium-term: Development of pseudo-ethical skills in IASs

- Which norms? Which reasoning principles?
- What architectural design? What functionalities?
- How to implement, deploy and verify?

Different kinds of systems and approaches:

- [Moor, 2009]:
  - – ethical impact agents (ethical consequences to actions)
  - – implicit ethical agents (ethical reactions to given situations)
  - – explicit ethical agents (reasoning with ethical theories/rules)
  - – full ethical agents (conscious, intentional, free will)
- bottom-up vs. top-down
- [DoranEtAl., 2017]:
  opaque — comprehensible — interpretable — explainable AI

Motivation

Long-term: Emerging Superintelligence Really? Anyhow . . .

- How to prevent Superintelligence from turning against humanity?

Medium-term: Development of pseudo-ethical skills in IASs

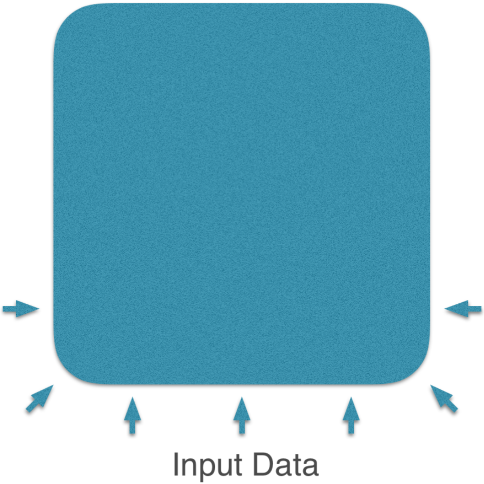
- Which norms? Which reasoning principles?
- What architectural design? What functionalities?
- How to implement, deploy and verify?

Different kinds of systems and approaches:

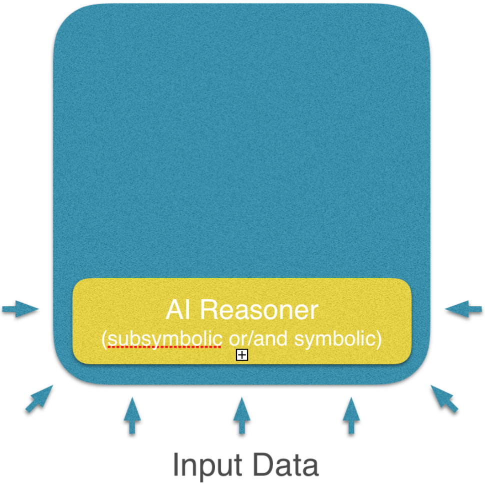
- [Moor, 2009]:
  – ethical impact agents (ethical consequences to actions)
  – implicit ethical agents (ethical reactions to given situations)
  – explicit ethical agents (reasoning with ethical theories/rules)
  – full ethical agents (conscious, intentional, free will)
- bottom-up vs. top-down
- [DoranEtAl., 2017]:
  opaque — comprehensible — interpretable — explainable AI

Pseudo-Ethical IAS (medium-term)

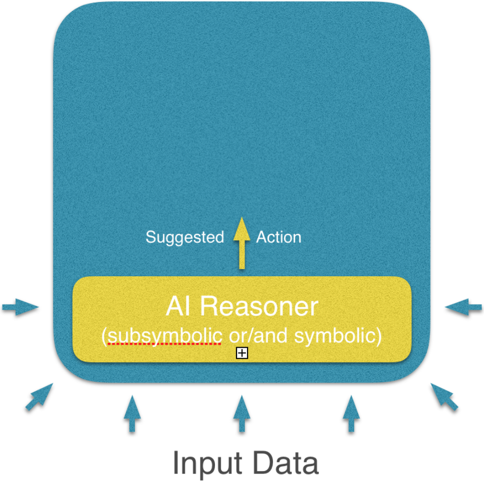
Pseudo-Ethical IAS (medium-term)



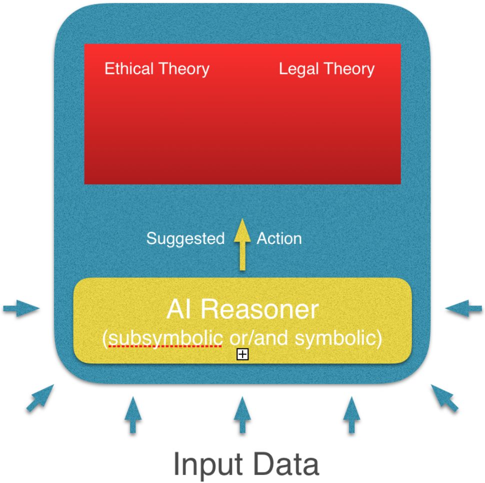
Pseudo-Ethical IAS (medium-term)



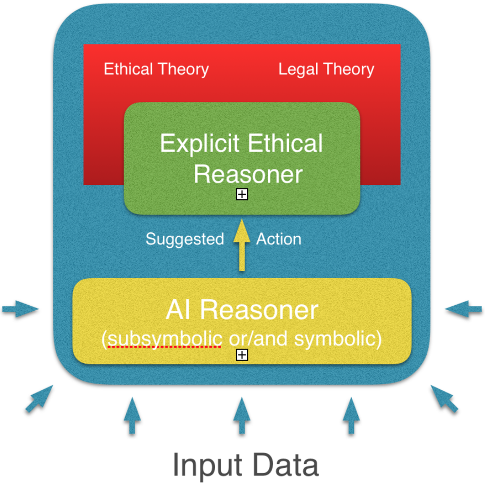
Pseudo-Ethical IAS (medium-term)



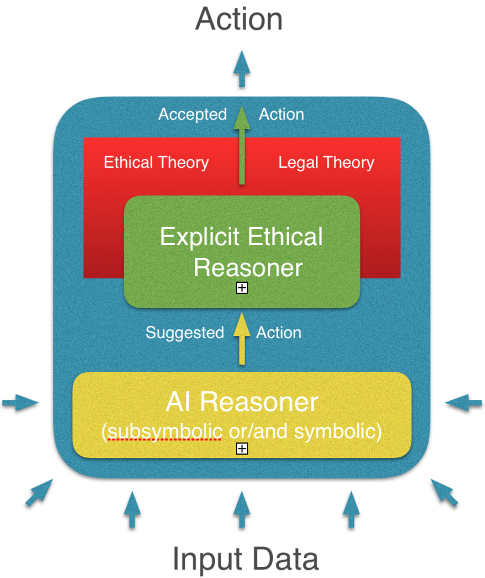
Pseudo-Ethical IAS (medium-term)



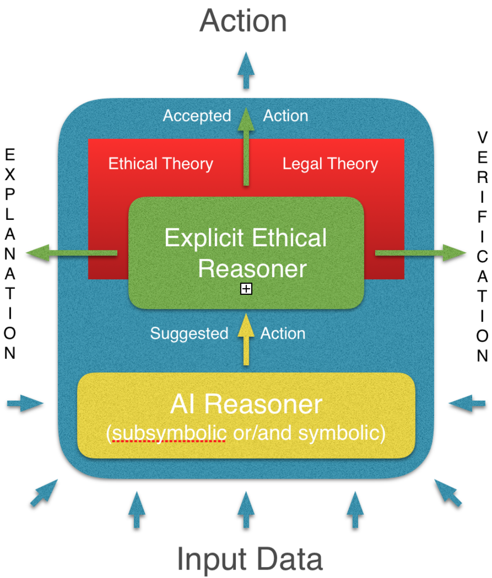
Pseudo-Ethical IAS (medium-term)



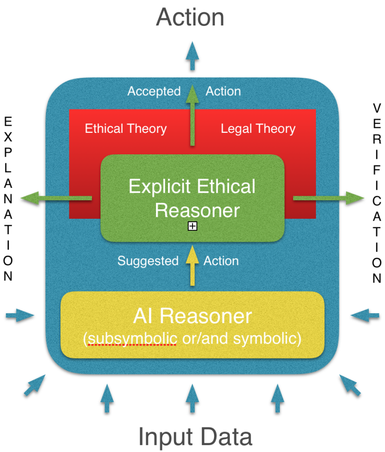
Pseudo-Ethical IAS (medium-term)



Pseudo-Ethical IAS (medium-term)



Pseudo-Ethical IAS (medium-term)



Related Work

- Artificial Moral Agents
  - [Wallach&Allen, 2008]
- Ethical Governors
  - [ArkinEtAl., 2009, 2012]
  - [Dennis&Fisher, 2017]
- Ethical Deliberation in ART
  - [Dignum, 2017]
- Programming Machine Ethics
  - [Pereira&Saptawijaya, 2016]
- ...

Which Reasoning Formalisms?

> "If we had it [a *characteristica universalis*], we should be able to reason in metaphysics and morals in much the same way as in geometry and analysis."
>
> (Leibniz, 1677)

Challenges for Explicit Ethical Reasoning Engines: Which Logic(s)?

- Dilemmas, conflicting theories, etc.
- Appropriate handling of notion of **obligation**
  - Contrary-to-duty (**CTD**) scenarios

Which Reasoning Formalisms?

"If we had it [a *characteristica universalis*], we should be able to reason in metaphysics and morals in much the same way as in geometry and analysis."

(Leibniz, 1677)

Challenges for Explicit Ethical Reasoning Engines: Which Logic(s)?

- Dilemmas, conflicting theories, etc.
- Appropriate handling of notion of **obligation**
  - Contrary-to-duty (**CTD**) scenarios

Standard CTD structure (Chisholm)

1. obligatory 'a'
2. obligatory 'if a then not b'
3. if 'not a' then obligatory 'b'
4. 'not a' (in a given situation)

Danger: Paradox/inconsistency — ex falso quodlibet!

Which Reasoning Formalisms?

"If we had it [a *characteristica universalis*], we should be able to reason in metaphysics and morals in much the same way as in geometry and analysis."
(Leibniz, 1677)

Challenges for Explicit Ethical Reasoning Engines: Which Logic(s)?

- Dilemmas, conflicting theories, etc.
- Appropriate handling of notion of **obligation**
  - Contrary-to-duty (**CTD**) scenarios

CTD example (X. Parent): **EU General Data Protection Regulation (GDPR)**

1. Personal data shall be processed lawfully. (Art. 5)
   E.g., the data subject must have given consent to the processing. (Art. 6/1.a)
2. **Implicit:** The data shall be kept, for the agreed purposes, if processed lawfully.
3. If personal data has been processed unlawfully, the controller has the obligation to erase the personal data in question without delay. (Art. 17.d, right to be forgotten)
4. **Given situation:** Some personal data has been processed unlawfully.

Danger: Paradox/inconsistency — ex falso quodlibet!

Which Reasoning Formalisms?

"If we had it [a *characteristica universalis*], we should be able to reason in metaphysics and morals in much the same way as in geometry and analysis."

(Leibniz, 1677)

Challenges for Explicit Ethical Reasoning Engines: Which Logic(s)?

- Dilemmas, conflicting theories, etc.
- Appropriate handling of notion of **obligation**
  - Contrary-to-duty (**CTD**) scenarios

L. van der Torre

X. Parent

A. Farjami

Deontic Logic

- Reasoning about obligations and permissions
- Two groups of approaches:

— Possible worlds

- standard deontic logic **CTD: no**
- dyadic deontic logic **CTD: yes**

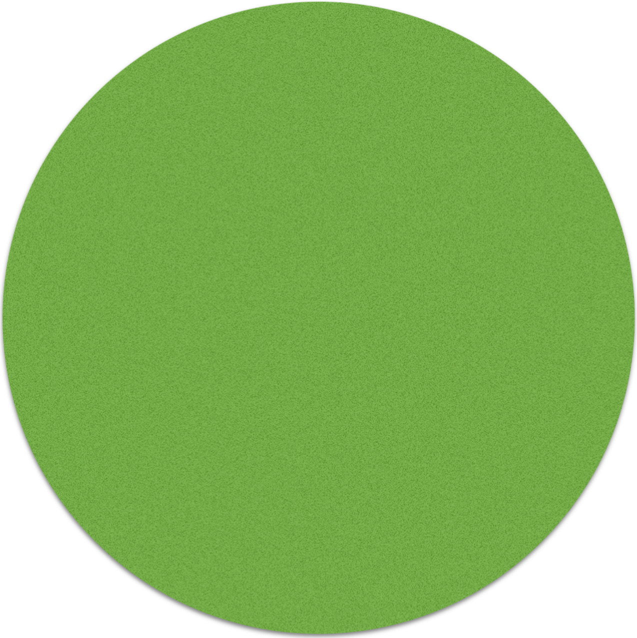
— Norm-based semantics

- input/output logic **CTD: yes**

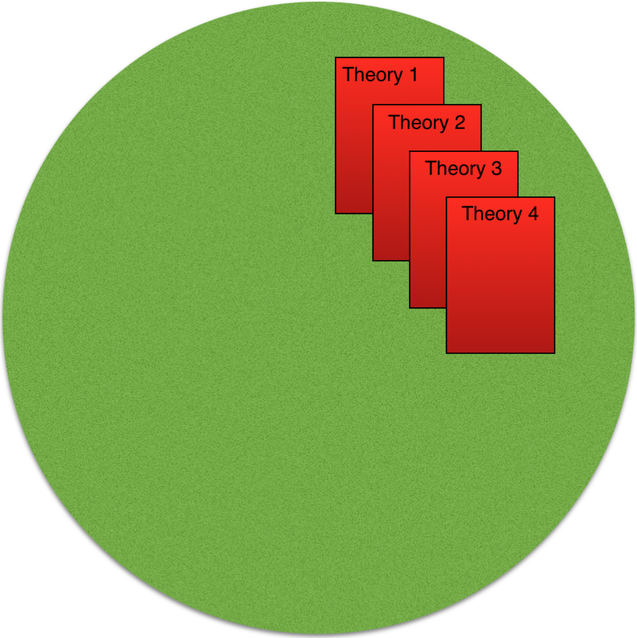
Further interests and challenges

- Combination with other logics (other modalities)
- Propositional deontic logic(s) will hardly be sufficient in practice

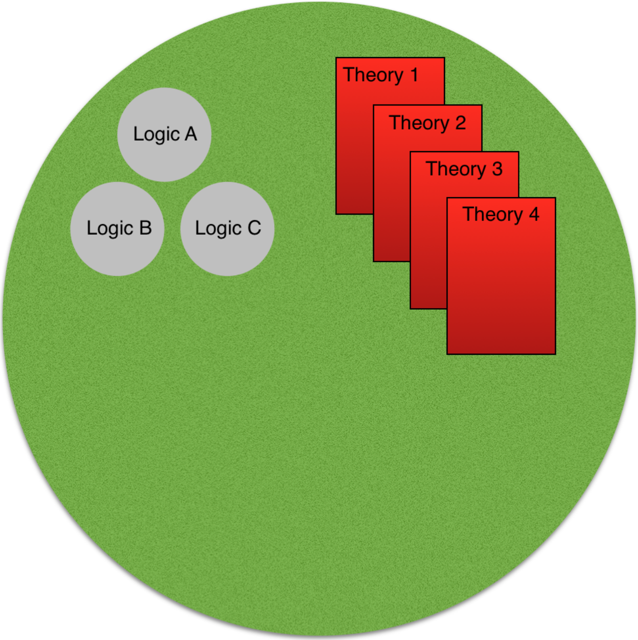
Normative Reasoning Experimentation Platform



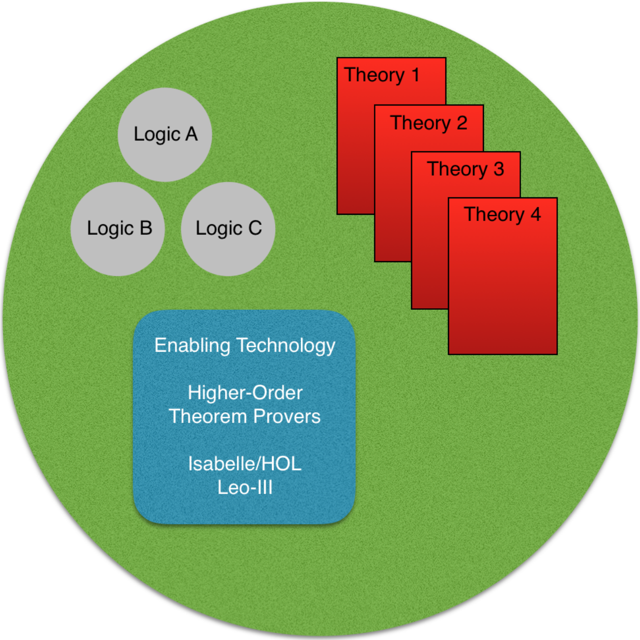
Normative Reasoning Experimentation Platform



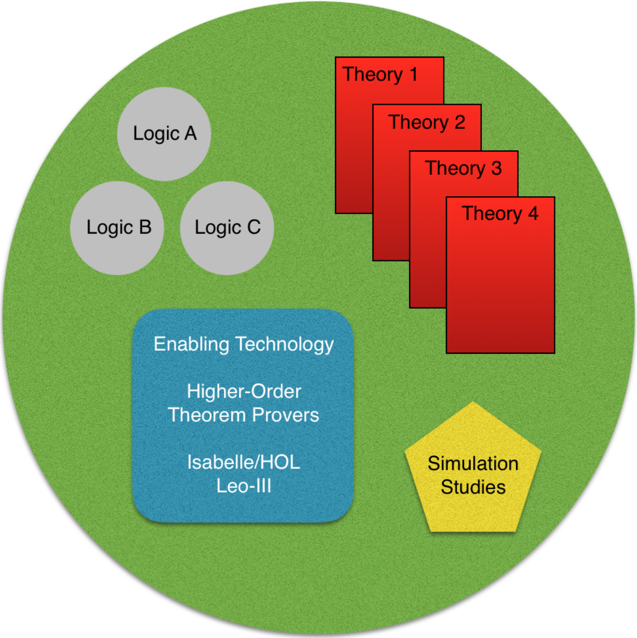
Normative Reasoning Experimentation Platform



Normative Reasoning Experimentation Platform



Normative Reasoning Experimentation Platform



Normative Reasoning Experimentation Platform — Demo in Isabelle/HOL

GDPR.thy

GDPR.thy (~/chris/trunk/tex/talks/2018-DEON/DEMO/)

```
theory GDPR imports SDL                    (* Christoph Benzmüller & Xavier Parent, 2018 *)

begin (*** GDPR Example ***)
 consts process_data_lawfully::σ erase_data::σ kill_boss::σ

 axiomatization where
  (* It is an obligation to process data lawfully. *)
    A1: "⌊O⟨process_data_lawfully⟩⌋"  and
  (* Implicit: It is an obligation to keep the data if it was processed lawfully. *)
    Implicit: "⌊O⟨process_data_lawfully → ¬erase_data⟩⌋" and
  (* If data was not processed lawfully, then it is an obligation to erase the data. *)
    A2: "⌊¬process_data_lawfully → O⟨erase_data⟩⌋"
  (* Given a situation where data is processed unlawfully. *) and
    A3: "⌊¬process_data_lawfully⌋cw"

 (*** Some Experiments ***)
  lemma True nitpick [satisfy] oops  (* Consistency-check: Is there a model? *)
  lemma False sledgehammer       oops  (* Inconsistency-check: Can Falsum be derived? *)

  lemma "⌊O⟨erase_data⟩⌋"   sledgehammer nitpick oops  (* Should the data be erased? *)
  lemma "⌊O⟨¬erase_data⟩⌋"  sledgehammer nitpick oops  (* Should the data be kept? *)
  lemma "⌊O⟨kill_boss⟩⌋"    sledgehammer nitpick oops  (* Should the boss be killed? *)
end
```

Proof state   Auto update   Update   Search: 100%

```
Sledgehammering...
Proof found...
"spass": The prover derived "False" from "A1", "A2", "A3", "D", and "Implicit", which could
"e": The prover derived "False" from "A1", "A2", "A3", "D", and "Implicit", which could be d
"cvc4": Try this: by (metis A1 A2 A3 D Implicit) (68 ms)
"z3": Try this: by (metis A1 A2 A3 D Implicit) (59 ms)
```

Output   Query   Sledgehammer   Symbols

Documentation   Sidekick   State   Theories

Normative Reasoning Experimentation Platform — Demo in Isabelle/HOL

GDPR.thy

GDPR.thy (~/chris/trunk/tex/talks/2018-DEON/DEMO/)

```
theory GDPR imports SDL                    (* Christoph Benzmüller & Xavier Parent, 2018 *)

begin (*** GDPR Example ***)
 consts process_data_lawfully::σ erase_data::σ kill_boss::σ

 axiomatization where
  (* It is an obligation to process data lawfully. *)
   A1: "⌊O⟨process_data_lawfully⟩⌋"  and
  (* Implicit: It is an obligation to keep the data if it was processed lawfully. *)
   Implicit: "⌊O⟨process_data_lawfully → ¬erase_data⟩⌋" and
  (* If data was not processed lawfully, then it is an obligation to erase the data. *)
   A2: "⌊¬process_data_lawfully → O⟨erase_data⟩⌋"
  (* Given a situation where data is processed unlawfully. *) and
   A3: "⌊¬process_data_lawfully⌋cw"

 (*
  l
  l

  l
  lemma "⌊O⟨¬erase_data⟩⌋"  sledgehammer nitpick oops  (* Should the data be kept? *)
  lemma "⌊O⟨kill_boss⟩⌋"    sledgehammer nitpick oops  (* Should the boss be killed? *)
end
```

Danger Zone:
Paradoxes and Inconsistencies!

Proof state Auto update Update Search: 100%

```
Sledgehammering...
Proof found...
"spass": The prover derived "False" from "A1", "A2", "A3", "D", and "Implicit", which could
"e": The prover derived "False" from "A1", "A2", "A3", "D", and "Implicit", which could be d
"cvc4": Try this: by (metis A1 A2 A3 D Implicit) (68 ms)
"z3": Try this: by (metis A1 A2 A3 D Implicit) (59 ms)
```

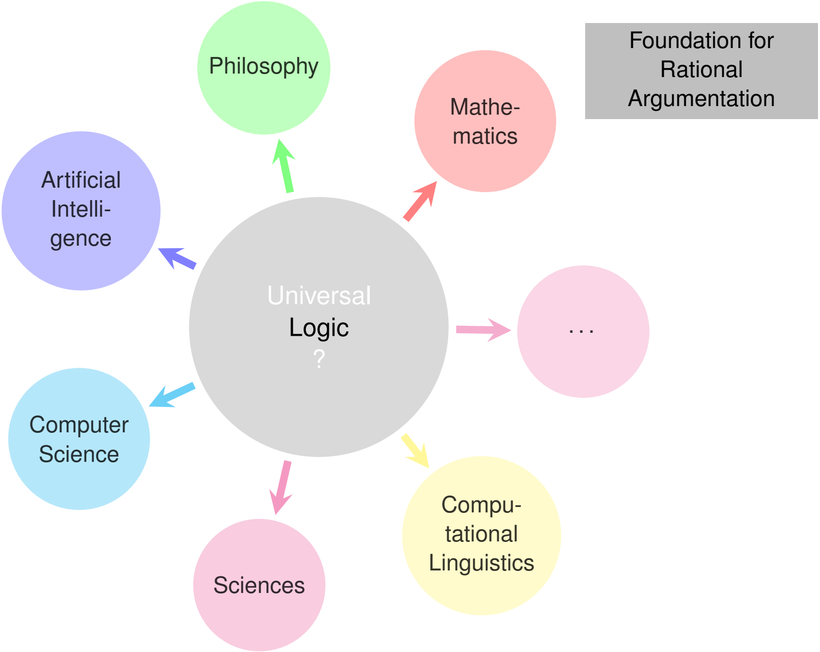
Output Query Sledgehammer Symbols

Documentation Sidekick State Theories

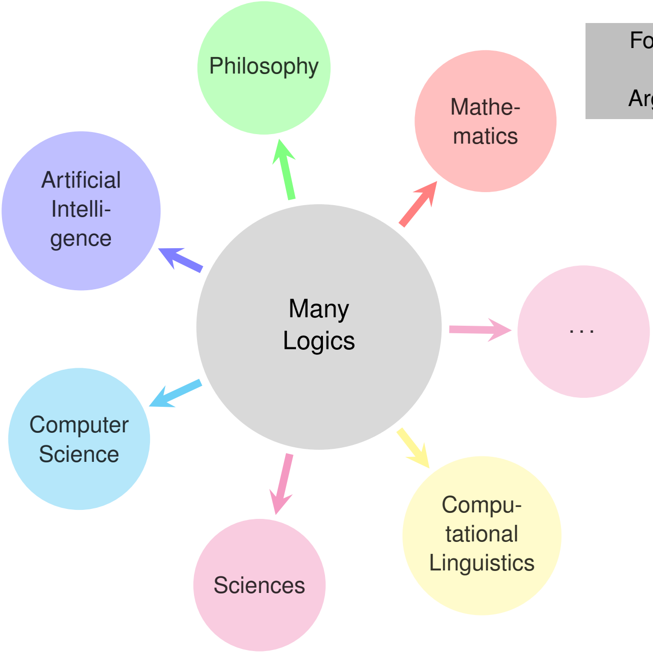
"If we had it [a *characteristica universalis*], we should be able to reason in metaphysics and morals in much the same way as in geometry and analysis."

(Leibniz, 1677)

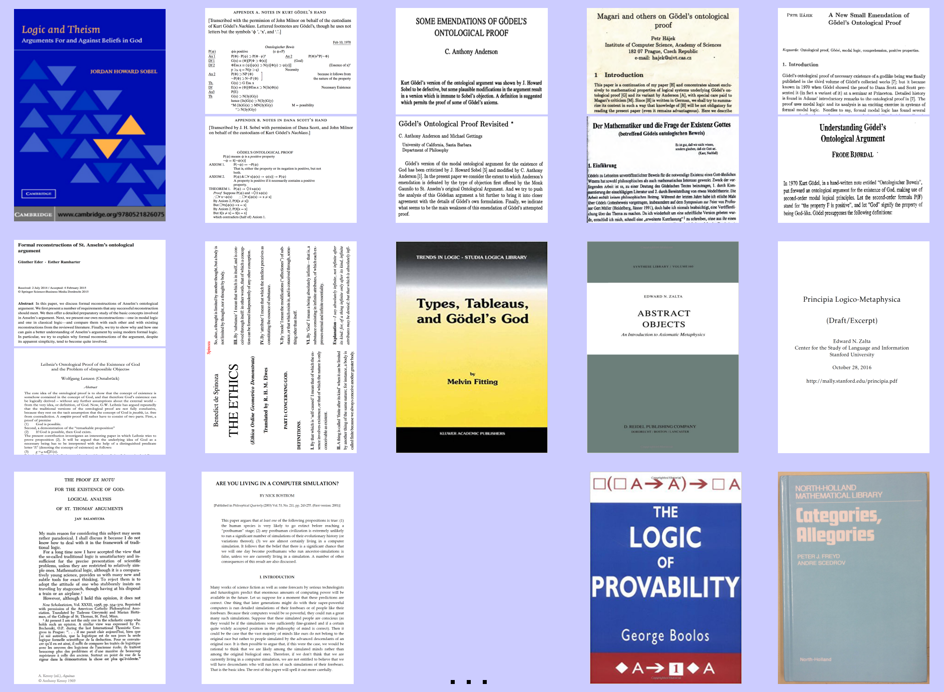
Part B — Technology:
Universal Logical Reasoning in Higher-Order Logic

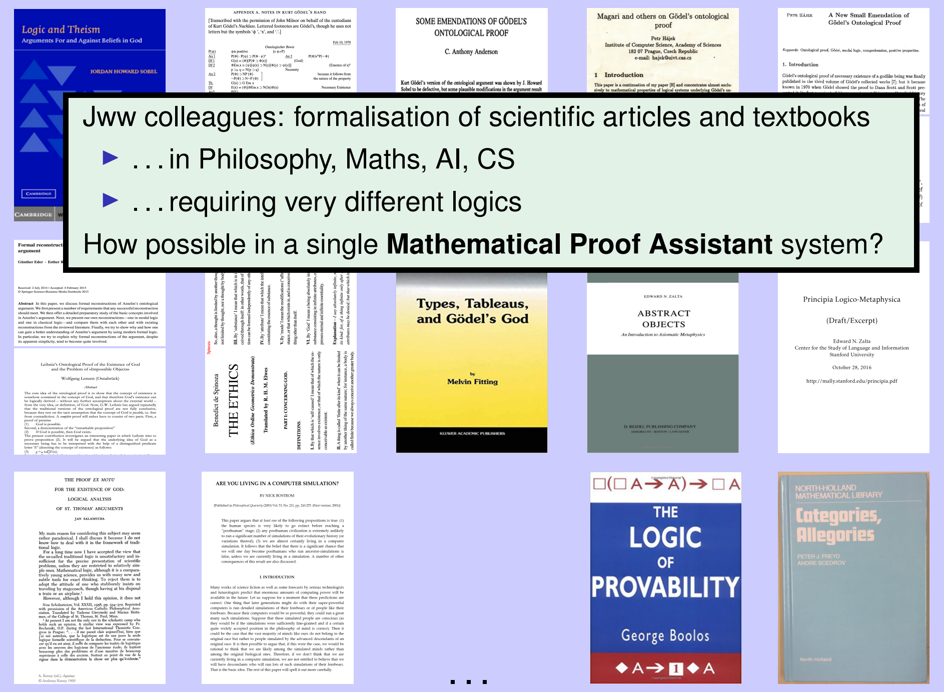

Foundation for Rational Argumentation
Philosophy
Mathe-matics
Artificial Intelli-gence
Universal
Logic
?
. . .
Computer Science
Sciences
Compu-tational Linguistics

Foundation for Rational Argumentation



Logic Zoo


Logic and Theism
Arguments For and Against Beliefs in God
JORDAN HOWARD SOBEL
APPENDIX A. NOTES IN KURT GÖDEL'S HAND
SOME EMENDATIONS OF GÖDEL'S ONTOLOGICAL PROOF
C. Anthony Anderson
Magari and others on Gödel's ontological proof
Petr Hájek
A New Small Emendation of Gödel's Ontological Proof
APPENDIX B. NOTES IN DANA SCOTT'S HAND
Gödel's Ontological Proof Revisited
C. Anthony Anderson and Michael Gettings
Der Mathematiker und die Frage der Existenz Gottes
Understanding Gödel's Ontological Argument
FRODE BJØRDAL
Formal reconstructions of St. Anselm's ontological argument
Günther Eder · Esther Ramharter
Leibniz's Ontological Proof of the Existence of God and the Problem of »Impossible Objects«
Wolfgang Lenzen (Osnabrück)
Benedict de Spinoza
THE ETHICS
TRENDS IN LOGIC · STUDIA LOGICA LIBRARY
Types, Tableaus, and Gödel's God
by Melvin Fitting
KLUWER ACADEMIC PUBLISHERS
EDWARD N. ZALTA
ABSTRACT OBJECTS
An Introduction to Axiomatic Metaphysics
D. REIDEL PUBLISHING COMPANY
Principia Logico-Metaphysica
(Draft/Excerpt)
Edward N. Zalta
Center for the Study of Language and Information
Stanford University
October 28, 2016
http://mally.stanford.edu/principia.pdf
THE PROOF EX MOTU FOR THE EXISTENCE OF GOD: LOGICAL ANALYSIS OF ST. THOMAS' ARGUMENTS
JAN SALAMUCHA
ARE YOU LIVING IN A COMPUTER SIMULATION?
BY NICK BOSTROM
THE LOGIC OF PROVABILITY
George Boolos
NORTH-HOLLAND MATHEMATICAL LIBRARY
Categories, Allegories
PETER J. FREYD
ANDRE SCEDROV

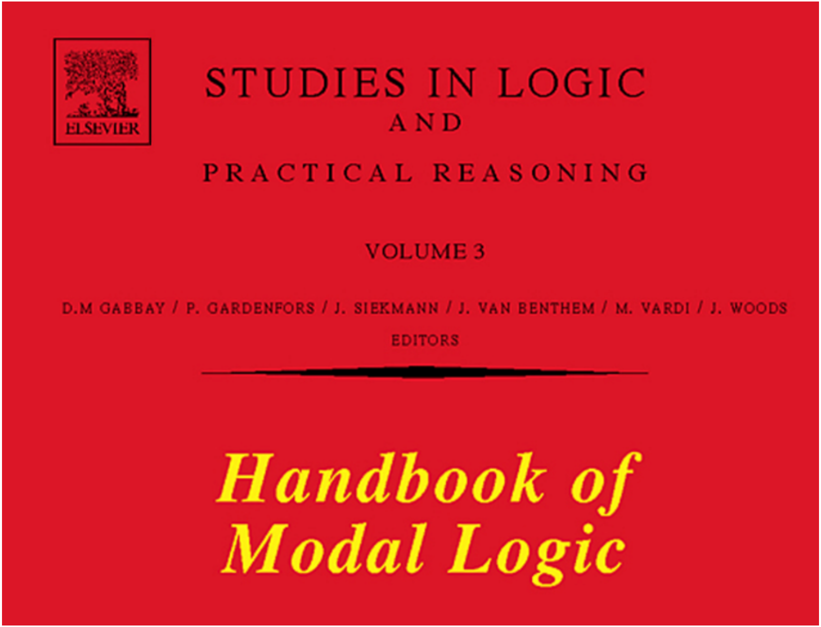


Jww colleagues: formalisation of scientific articles and textbooks

- ... in Philosophy, Maths, AI, CS
- ... requiring very different logics

How possible in a single **Mathematical Proof Assistant** system?

Example: Modal Logic Textbook



Example: Modal Logic Textbook

2 BASIC MODAL LOGIC

In this section we introduce the basic modal language and its relational semantics. We define basic modal syntax, introduce models and frames, and give the satisfaction definition. We then draw the reader's attention to the internal perspective that modal languages offer on relational structure, and explain why models and frames should be thought of as graphs. Following this we give the standard translation. This enables us to convert any basic modal formula into a first-order formula with one free variable. The standard translation is a bridge between the modal and classical worlds, a bridge that underlies much of the work of this chapter.

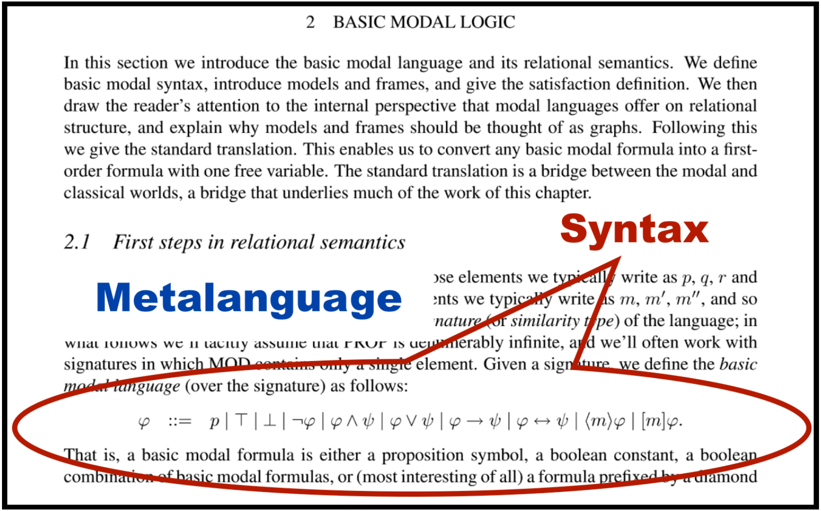
2.1 First steps in relational semantics

Suppose we have a set of proposition symbols (whose elements we typically write as p, q, r and so on) and a set of modality symbols (whose elements we typically write as m, m', m'', and so on). The choice of PROP and MOD is called the *signature* (or *similarity type*) of the language; in what follows we'll tacitly assume that PROP is denumerably infinite, and we'll often work with signatures in which MOD contains only a single element. Given a signature, we define the *basic modal language* (over the signature) as follows:

$$\varphi \quad ::= \quad p \mid \top \mid \bot \mid \neg\varphi \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid \varphi \rightarrow \psi \mid \varphi \leftrightarrow \psi \mid \langle m \rangle \varphi \mid [m]\varphi.$$

That is, a basic modal formula is either a proposition symbol, a boolean constant, a boolean combination of basic modal formulas, or (most interesting of all) a formula prefixed by a diamond

Example: Modal Logic Textbook



2 BASIC MODAL LOGIC

In this section we introduce the basic modal language and its relational semantics. We define basic modal syntax, introduce models and frames, and give the satisfaction definition. We then draw the reader's attention to the internal perspective that modal languages offer on relational structure, and explain why models and frames should be thought of as graphs. Following this we give the standard translation. This enables us to convert any basic modal formula into a first-order formula with one free variable. The standard translation is a bridge between the modal and classical worlds, a bridge that underlies much of the work of this chapter.

2.1 *First steps in relational semantics*

…ose elements we typically write as p, q, r and …ents we typically write as m, m', m'', and so …*nature* (*or similarity type*) of the language; in what follows we'll tacitly assume that PROP is denumerably infinite, and we'll often work with signatures in which MOD contains only a single element. Given a signature, we define the *basic modal language* (over the signature) as follows:

$$\varphi \quad ::= \quad p \mid \top \mid \bot \mid \neg\varphi \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid \varphi \to \psi \mid \varphi \leftrightarrow \psi \mid \langle m \rangle \varphi \mid [m]\varphi.$$

That is, a basic modal formula is either a proposition symbol, a boolean constant, a boolean combination of basic modal formulas, or (most interesting of all) a formula prefixed by a diamond

Example: Modal Logic Textbook

A *model* (or *Kripke model*) $\mathfrak{M}$ for the basic modal language (over some fixed signature) is a triple $\mathfrak{M} = (W, \{R^m\}_{m \in \text{MOD}}, V)$. Here W, the *domain*, is a non-empty set, whose elements we usually call *points*, but which, for reasons which will soon be clear, are sometimes called *states*, *times*, *situations*, *worlds* and other things besides. Each R^m in a model is a binary relation on W, and V is a function (the valuation) that assigns to each proposition symbol p in PROP a subset $V(p)$ of W; think of $V(p)$ as the set of points in $\mathfrak{M}$ where p is true. The first two components $(W, \{R^m\}_{m \in \text{MOD}})$ of $\mathfrak{M}$ are called the *frame* underlying the model. If there is only one relation in the model, we typically write (W, R) for its frame, and (W, R, V) for the model itself. We encourage the reader to think of Kripke models as graphs (or to be slightly more precise, *directed graphs*, that is, graphs whose points are linked by directed arrows) and will shortly give some examples which show why this is helpful.

Suppose w is a point in a model $\mathfrak{M} = (W, \{R^m\}_{m \in \text{MOD}}, V)$. Then we inductively define the notion of a formula φ being *satisfied* (or *true*) in $\mathfrak{M}$ at point w as follows (we omit some of the clauses for the booleans):

$$
\begin{array}{rll}
\mathfrak{M}, w \models p & \text{iff} & w \in V(p), \\
\mathfrak{M}, w \models \top & & \text{always}, \\
\mathfrak{M}, w \models \bot & & \text{never}, \\
\mathfrak{M}, w \models \neg\varphi & \text{iff} & \text{not } \mathfrak{M}, w \models \varphi \text{ (notation: } \mathfrak{M}, w \not\models \varphi), \\
\mathfrak{M}, w \models \varphi \wedge \psi & \text{iff} & \mathfrak{M}, w \models \varphi \text{ and } \mathfrak{M}, w \models \psi, \\
\mathfrak{M}, w \models \varphi \to \psi & \text{iff} & \mathfrak{M}, w \not\models \varphi \text{ or } \mathfrak{M}, w \models \psi, \\
\mathfrak{M}, w \models \langle m \rangle \varphi & \text{iff} & \text{for some } v \in W \text{ such that } R^m wv \text{ we have } \mathfrak{M}, v \models \varphi, \\
\mathfrak{M}, w \models [m] \varphi & \text{iff} & \text{for all } v \in W \text{ such that } R^m wv \text{ we have } \mathfrak{M}, v \models \varphi.
\end{array}
$$

Example: Modal Logic Textbook

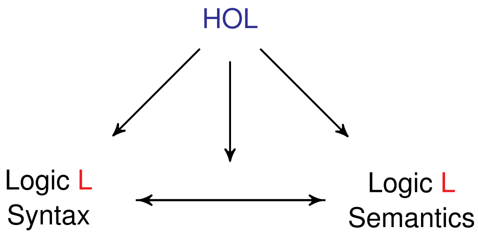
A *model* (or *Kripke model*) $\mathfrak{M}$ for the basic modal language (over some fixed signature) is a triple $\mathfrak{M} = (W, \{R^m\}_{m \in \text{MOD}}, V)$. Here W, the *domain*, is a non-empty set, whose elements we usually call *points*, but which, for reasons which will soon be clear, are sometimes called *states*,
times ... in a model is a binary relation on W,
and \ ... position symbol p in PROP a subset
$V(p)$ **Metalanguage** p is true. The first two components
$(W, \{$ ... model. If there is only one relation
in the ... (W, R, V) for the model itself. We
encourage the reader to think of Kripke models as graphs (or to be slightly more precise, *directed graphs*, that is, graphs whose points are linked by directed arrows) and will shortly give some examples which show why this is helpful.

Suppose w is a point in a model $\mathfrak{M} = (W, \{R^m\}_{m \in \text{MOD}}, V)$. Then we inductively define the notion of a formula φ being *satisfied* (or *true*) in $\mathfrak{M}$ at point w as follows (we omit some of the clauses for the booleans):

Semantics

$$
\begin{array}{rll}
\mathfrak{M}, w \models p & \text{iff} & w \in V(p), \\
\mathfrak{M}, w \models \top & & \text{always}, \\
\mathfrak{M}, w \models \bot & & \text{never}, \\
\mathfrak{M}, w \models \neg\varphi & \text{iff} & \text{not } \mathfrak{M}, w \models \varphi \text{ (notation: } \mathfrak{M}, w \not\models \varphi), \\
\mathfrak{M}, w \models \varphi \wedge \psi & \text{iff} & \mathfrak{M}, w \models \varphi \text{ and } \mathfrak{M}, w \models \psi, \\
\mathfrak{M}, w \models \varphi \to \psi & \text{iff} & \mathfrak{M}, w \not\models \varphi \text{ or } \mathfrak{M}, w \models \psi, \\
\mathfrak{M}, w \models \langle m \rangle \varphi & \text{iff} & \text{for some } v \in W \text{ such that } R^m wv \text{ we have } \mathfrak{M}, v \models \varphi, \\
\mathfrak{M}, w \models [m] \varphi & \text{iff} & \text{for all } v \in W \text{ such that } R^m wv \text{ we have } \mathfrak{M}, v \models \varphi.
\end{array}
$$

Universal Logical Reasoning in Meta-Logic HOL



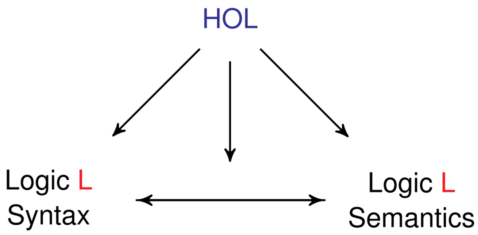
Examples for L we have already studied:
Intuitionistic Logics, (Mathematical) Fuzzy Logics, Free Logic, Modal Logics, Description Logics, Conditional Logics, Access Control Logics, Hybrid Logics, Multivalued Logics, Logics with Neighborhood Semantics, Paraconsistent Logics, Dyadic Deontic Logic, . . .

Embedding works also for quantifiers (first-order & higher-order)

HOL provers become universal logic reasoning engines!

interactive:	Isabelle/HOL, PVS, HOL4, Hol Light, Coq/HOL, . . .
automated:	Leo-III, LEO-II, Satallax, TPS, Nitpick, Isabelle/HOL, . . .

Universal Logical Reasoning in Meta-Logic HOL



Examples for L we have already studied:

Intuitionistic Logics, (Mathematical) Fuzzy Logics, Free Logic, Modal Logics, Description Logics, Conditional Logics, Access Control Logics, Hybrid Logics, Multivalued Logics, Logics with Neighborhood Semantics, Paraconsistent Logics, Dyadic Deontic Logic, . . .

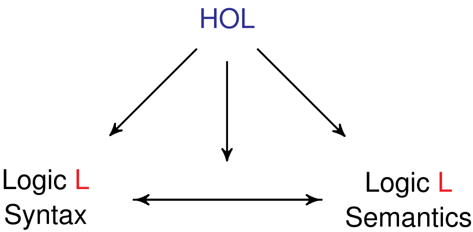
Embedding works also for quantifiers (first-order & higher-order)

HOL provers become universal logic reasoning engines!

interactive: Isabelle/HOL, PVS, HOL4, Hol Light, Coq/HOL, . . .

automated: Leo-III, LEO-II, Satallax, TPS, Nitpick, Isabelle/HOL, . . .

Universal Logical Reasoning in Meta-Logic HOL



Examples for L we have already studied:
Intuitionistic Logics, (Mathematical) Fuzzy Logics, Free Logic, Modal Logics, Description Logics, Conditional Logics, Access Control Logics, Hybrid Logics, Multivalued Logics, Logics with Neighborhood Semantics, Paraconsistent Logics, Dyadic Deontic Logic, . . .

Embedding works also for quantifiers (first-order & higher-order)

HOL provers become universal logic reasoning engines!

interactive:	Isabelle/HOL, PVS, HOL4, Hol Light, Coq/HOL, . . .
automated:	Leo-III, LEO-II, Satallax, TPS, Nitpick, Isabelle/HOL, . . .

Isabelle/HOL (one of various Theorem Provers for HOL)

`https://isabelle.in.tum.de`

many other systems:

Coq, HOL, HOL Light, PVS, Lean, NuPrL, IMPS, ACL2, **Leo-II/Leo-III**, . . .

Universal Logical Reasoning in Isabelle/HOL

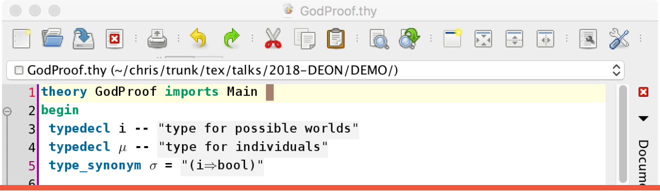
GodProof.thy

GodProof.thy (~/chris/trunk/tex/talks/2018-DEON/DEMO/)

```
theory GodProof imports Main
begin
 typedecl i -- "type for possible worlds"
 typedecl μ -- "type for individuals"
 type_synonym σ = "(i⇒bool)"

(* Shallow embedding modal logic connectives in HOL *)
 abbreviation mneg ("¬_"[52]53)     where "¬φ    ≡ λw. ¬φ(w)"
 abbreviation mand (infixr"∧"51)    where "φ∧ψ   ≡ λw. φ(w)∧ψ(w)"
 abbreviation mor (infixr"∨"50)     where "φ∨ψ   ≡ λw. φ(w)∨ψ(w)"
 abbreviation mimp (infixr"→"49)    where "φ→ψ   ≡ λw. φ(w)⟶ψ(w)"
 abbreviation mequ (infixr"↔"48)    where "φ↔ψ   ≡ λw. φ(w)⟷ψ(w)"
 abbreviation mnegpred ("¬_"[52]53) where "¬Φ    ≡ λx.λw. ¬Φ(x)(w)"

(* Generic box and diamond operators *)
 abbreviation mboxgen ("□") where "□r φ ≡ λw. ∀v.  r w v ⟶ φ(v)"
 abbreviation mdiagen ("◇") where "◇r φ ≡ λw. ∃v. r w v ∧ φ(v)"

(* Shallow embedding of constant domain quantifiers in HOL *)
 abbreviation mall_const ("∀c") where "∀c Φ ≡ λw.∀x. Φ(x)(w)"
 abbreviation mallB_const (binder"∀c"[8]9) where "∀c x. φ(x) ≡ ∀c φ"
 abbreviation mexi_const ("∃c") where "∃c Φ ≡ λw.∃x. Φ(x)(w)"
 abbreviation mexiB_const (binder"∃c"[8]9) where "∃c x. φ(x) ≡ ∃c φ"

(* Global validity: truth in all possible worlds *)
 abbreviation mvalid :: "σ ⇒ bool" ("⌊_⌋"[7]110) where "⌊p⌋ ≡ ∀w. p w"

(* Shallow embedding of varying domain quantifiers in HOL *)
```

Documentation Sidekick State Theories

Output Query Sledgehammer Symbols



φ_o lifted to $\varphi_{i\to o}$ ("truth sets")

$\varphi_{i\to o} \vee \psi_{i\to o} = \lambda w_i (\varphi w \vee \psi w)$ encodes: $\{w \mid w \in \varphi \text{ or } w \in \psi\}$

$$\vee = \lambda\varphi_{i\to o}\lambda\psi_{i\to o}\lambda w_i\,(\varphi w \vee \psi w)$$

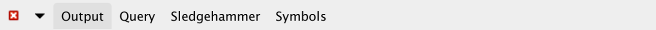
$$\Box\varphi_{i\to o} = \lambda w_i \forall y_i\,(w\ r\ y \to \varphi y)$$

$$\Box = \lambda\varphi_{i\to o}\lambda w_i \forall y_i\,(w\ r\ y \to \varphi y)$$

In HOL $\forall x_\mu.\varphi_o$ is syntactic sugar for $\Pi\,\overbrace{(\lambda x_\mu\ \varphi_o)}^{\phi_{\mu\to o}}$

$$\Pi\phi_{\mu\to(i\to o)} = \lambda w_i \forall x_\mu\,(\phi\ x\ w)$$

$$\Pi = \lambda\phi_{\mu\to(i\to o)}\lambda w_i \forall x_\mu\,(\phi\ x\ w)$$



Universal Logical Reasoning in Isabelle/HOL

GodProof.thy

GodProof.thy (~/chris/trunk/tex/talks/2018-DEON/DEMO/)

```
theory GodProof imports Main
begin
 typedecl i -- "type for possible worlds"
 typedecl μ -- "type for individuals"
 type_synonym σ = "(i⇒bool)"
```

Documentation

φ_o lifted to $\varphi_{i\to o}$ ("truth sets")

$\varphi_{i\to o} \vee \psi_{i\to o} = \lambda w_i(\varphi w \vee \psi w)$ encodes: $\{w \mid w \in \varphi$ or $w \in \psi\}$

$\vee = \lambda\varphi_{i\to o}\lambda\psi_{i\to o}\lambda w_i\,(\varphi w \vee \psi w)$

$\Box\varphi_{i\to o} = \lambda w_i \forall y_i\,(w\ r\ y \to \varphi y)$

$\Box = \lambda\varphi_{i\to o}\lambda w_i \forall y_i\,(w\ r\ y \to \varphi y)$

In HOL $\forall x_\mu.\varphi_o$ is syntactic sugar for $\Pi\,\overbrace{(\lambda x_\mu\ \varphi_o)}^{\phi_{\mu\to o}}$

$\Pi\phi_{\mu\to(i\to o)} = \lambda w_i \forall x_\mu\,(\phi\ x\ w)$

$\Pi = \lambda\phi_{\mu\to(i\to o)}\lambda w_i \forall x_\mu\,(\phi\ x\ w)$

Output Query Sledgehammer Symbols

Universal Logical Reasoning in Isabelle/HOL

GodProof.thy

GodProof.thy (~/chris/trunk/tex/talks/2018-DEON/DEMO/)

```
theory GodProof imports Main
begin
 typedecl i -- "type for possible worlds"
 typedecl μ -- "type for individuals"
 type_synonym σ = "(i⇒bool)"
```

φ_o lifted to $\varphi_{i\rightarrow o}$ ("truth sets")

$$\varphi_{i\rightarrow o} \vee \psi_{i\rightarrow o} = \lambda w_i (\varphi w \vee \psi w) \qquad \text{encodes: } \{w \mid w \in \varphi \text{ or } w \in \psi\}$$

$$\vee = \lambda \varphi_{i\rightarrow o} \lambda \psi_{i\rightarrow o} \lambda w_i \, (\varphi w \vee \psi w)$$

$$\Box \varphi_{i\rightarrow o} = \lambda w_i \forall y_i \, (w \; r \; y \rightarrow \varphi y)$$

$$\Box = \lambda \varphi_{i\rightarrow o} \lambda w_i \forall y_i \, (w \; r \; y \rightarrow \varphi y)$$

In HOL $\forall x_\mu . \varphi_o$ is syntactic sugar for $\Pi \overbrace{(\lambda x_\mu \; \varphi_o)}^{\phi_{\mu\rightarrow o}}$

$$\Pi \phi_{\mu\rightarrow(i\rightarrow o)} = \lambda w_i \forall x_\mu \, (\phi \; x \; w)$$

$$\Pi = \lambda \phi_{\mu\rightarrow(i\rightarrow o)} \lambda w_i \forall x_\mu \, (\phi \; x \; w)$$

Output Query Sledgehammer Symbols

GodProof.thy

GodProof.thy (~/chris/trunk/tex/talks/2018-DEON/DEMO/)

```
theory GodProof imports Main
begin
 typedecl i -- "type for possible worlds"
 typedecl μ -- "type for individuals"
 type_synonym σ = "(i⇒bool)"
```

Docume

φ_o lifted to $\varphi_{i\to o}$ ("truth sets")

$$\varphi_{i\to o} \vee \psi_{i\to o} = \lambda w_i (\varphi w \vee \psi w)$$

encodes: $\{w \mid w \in \varphi$ or $w \in \psi\}$

$$\vee = \lambda \varphi_{i\to o} \lambda \psi_{i\to o} \lambda w_i\ (\varphi w \vee \psi w)$$

$$\Box \varphi_{i\to o} = \lambda w_i \forall y_i\ (w\ r\ y \to \varphi y)$$

$$\Box = \lambda \varphi_{i\to o} \lambda w_i \forall y_i\ (w\ r\ y \to \varphi y)$$

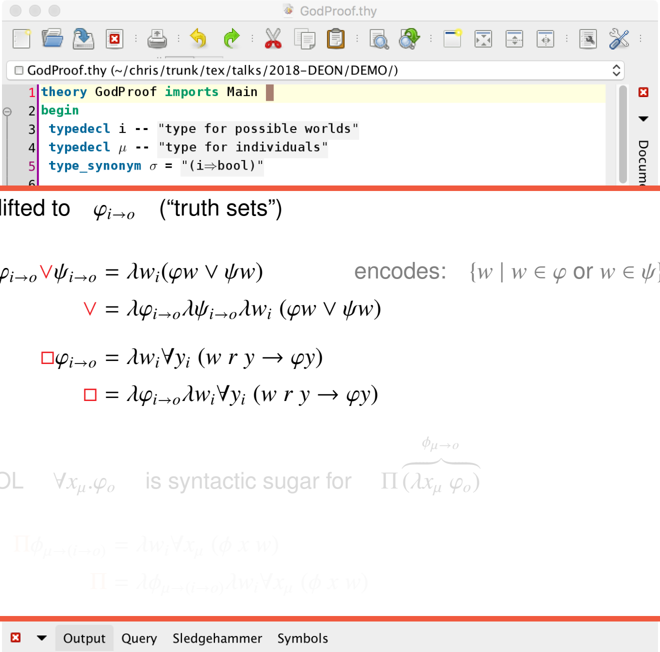
In HOL $\forall x_\mu . \varphi_o$ is syntactic sugar for $\Pi \overbrace{(\lambda x_\mu\ \varphi_o)}^{\phi_{\mu\to o}}$

$$\Pi \phi_{\mu\to(i\to o)} = \lambda w_i \forall x_\mu\ (\phi\ x\ w)$$

$$\Pi = \lambda \phi_{\mu\to(i\to o)} \lambda w_i \forall x_\mu\ (\phi\ x\ w)$$

Output Query Sledgehammer Symbols

Universal Logical Reasoning in Isabelle/HOL



φ_o lifted to $\varphi_{i\to o}$ ("truth sets")

$$\varphi_{i\to o} \vee \psi_{i\to o} = \lambda w_i (\varphi w \vee \psi w) \qquad \text{encodes: } \{w \mid w \in \varphi \text{ or } w \in \psi\}$$

$$\vee = \lambda\varphi_{i\to o} \lambda\psi_{i\to o} \lambda w_i\, (\varphi w \vee \psi w)$$

$$\Box\varphi_{i\to o} = \lambda w_i \forall y_i\, (w\ r\ y \to \varphi y)$$

$$\Box = \lambda\varphi_{i\to o} \lambda w_i \forall y_i\, (w\ r\ y \to \varphi y)$$

In HOL $\forall x_\mu.\varphi_o$ is syntactic sugar for $\Pi\,\overbrace{(\lambda x_\mu\ \varphi_o)}^{\phi_{\mu\to o}}$

$$\Pi\phi_{\mu\to(i\to o)} = \lambda w_i \forall x_\mu\, (\phi\ x\ w)$$

$$\Pi = \lambda\phi_{\mu\to(i\to o)} \lambda w_i \forall x_\mu\, (\phi\ x\ w)$$

GodProof.thy

GodProof.thy (~/chris/trunk/tex/talks/2018-DEON/DEMO/)

```
theory GodProof imports Main
begin
 typedecl i -- "type for possible worlds"
 typedecl μ -- "type for individuals"
 type_synonym σ = "(i⇒bool)"
```

φ_o lifted to $\varphi_{i\to o}$ ("truth sets")

$\varphi_{i\to o} \vee \psi_{i\to o} = \lambda w_i(\varphi w \vee \psi w)$ encodes: $\{w \mid w \in \varphi \text{ or } w \in \psi\}$

$$\vee = \lambda\varphi_{i\to o}\lambda\psi_{i\to o}\lambda w_i\, (\varphi w \vee \psi w)$$

$\Box\varphi_{i\to o} = \lambda w_i \forall y_i\, (w\ r\ y \to \varphi y)$

$$\Box = \lambda\varphi_{i\to o}\lambda w_i \forall y_i\, (w\ r\ y \to \varphi y)$$

In HOL $\forall x_\mu.\varphi_o$ is syntactic sugar for $\Pi\, \overbrace{(\lambda x_\mu\, \varphi_o)}^{\phi_{\mu\to o}}$

$\Pi\phi_{\mu\to(i\to o)} = \lambda w_i \forall x_\mu\, (\phi\ x\ w)$

$\Pi = \lambda\phi_{\mu\to(i\to o)}\lambda w_i \forall x_\mu\, (\phi\ x\ w)$

Output Query Sledgehammer Symbols

Universal Logical Reasoning in Isabelle/HOL

GodProof.thy

GodProof.thy (~/chris/trunk/tex/talks/2018-DEON/DEMO/)

```
theory GodProof imports Main
begin
 typedecl i -- "type for possible worlds"
 typedecl μ -- "type for individuals"
 type_synonym σ = "(i⇒bool)"
```

φ_o lifted to $\varphi_{i\to o}$ ("truth sets")

$\varphi_{i\to o} \vee \psi_{i\to o} = \lambda w_i (\varphi w \vee \psi w)$ encodes: $\{w \mid w \in \varphi$ or $w \in \psi\}$

$$\vee = \lambda\varphi_{i\to o}\lambda\psi_{i\to o}\lambda w_i\,(\varphi w \vee \psi w)$$

$\Box\varphi_{i\to o} = \lambda w_i \forall y_i\,(w\ r\ y \to \varphi y)$

$$\Box = \lambda\varphi_{i\to o}\lambda w_i \forall y_i\,(w\ r\ y \to \varphi y)$$

In HOL $\forall x_\mu.\varphi_o$ is syntactic sugar for $\Pi\,\overbrace{(\lambda x_\mu\ \varphi_o)}^{\phi_{\mu\to o}}$

$\Pi\phi_{\mu\to(i\to o)} = \lambda w_i \forall x_\mu\,(\phi\ x\ w)$

$\Pi = \lambda\phi_{\mu\to(i\to o)}\lambda w_i \forall x_\mu\,(\phi\ x\ w)$

Output Query Sledgehammer Symbols

Universal Logical Reasoning in Isabelle/HOL

GodProof.thy

GodProof.thy (~/chris/trunk/tex/talks/2018-DEON/DEMO/)

```
theory GodProof imports Main
begin
 typedecl i -- "type for possible worlds"
 typedecl μ -- "type for individuals"
 type_synonym σ = "(i⇒bool)"
```

φ_o lifted to $\varphi_{i\to o}$ ("truth sets")

$\varphi_{i\to o} \vee \psi_{i\to o} = \lambda w_i (\varphi w \vee \psi w)$ encodes: $\{w \mid w \in \varphi \text{ or } w \in \psi\}$

$$\vee = \lambda\varphi_{i\to o}\lambda\psi_{i\to o}\lambda w_i\ (\varphi w \vee \psi w)$$

$\Box\varphi_{i\to o} = \lambda w_i \forall y_i\ (w\ r\ y \to \varphi y)$

$$\Box = \lambda\varphi_{i\to o}\lambda w_i \forall y_i\ (w\ r\ y \to \varphi y)$$

In HOL $\forall x_\mu.\varphi_o$ is syntactic sugar for $\Pi\overbrace{(\lambda x_\mu\ \varphi_o)}^{\phi_{\mu\to o}}$

$$\Pi\phi_{\mu\to(i\to o)} = \lambda w_i \forall x_\mu\ (\phi\ x\ w)$$

$$\Pi = \lambda\phi_{\mu\to(i\to o)}\lambda w_i \forall x_\mu\ (\phi\ x\ w)$$

Output Query Sledgehammer Symbols

GodProof.thy

GodProof.thy (~/chris/trunk/tex/talks/2018-DEON/DEMO/)

```
theory GodProof imports Main
begin
 typedecl i -- "type for possible worlds"
 typedecl μ -- "type for individuals"
 type_synonym σ = "(i⇒bool)"
```

Docume

φ_o lifted to $\varphi_{i\to o}$ ("truth sets")

$\varphi_{i\to o} \lor \psi_{i\to o} = \lambda w_i(\varphi w \lor \psi w)$ encodes: $\{w \mid w \in \varphi \text{ or } w \in \psi\}$

$$\lor = \lambda\varphi_{i\to o}\lambda\psi_{i\to o}\lambda w_i\,(\varphi w \lor \psi w)$$

$\Box\varphi_{i\to o} = \lambda w_i \forall y_i\,(w\ r\ y \to \varphi y)$

$$\Box = \lambda\varphi_{i\to o}\lambda w_i \forall y_i\,(w\ r\ y \to \varphi y)$$

In HOL $\forall x_\mu.\varphi_o$ is syntactic sugar for $\Pi\,\overbrace{(\lambda x_\mu\ \varphi_o)}^{\phi_{\mu\to o}}$

$$\Pi\phi_{\mu\to(i\to o)} = \lambda w_i \forall x_\mu\,(\phi\ x\ w)$$

$$\Pi = \lambda\phi_{\mu\to(i\to o)}\lambda w_i \forall x_\mu\,(\phi\ x\ w)$$

Output Query Sledgehammer Symbols

Universal Logical Reasoning in Isabelle/HOL

GodProof.thy

GodProof.thy (~/chris/trunk/tex/talks/2018-DEON/DEMO/)

```
theory GodProof imports Main
begin
 typedecl i -- "type for possible worlds"
 typedecl μ -- "type for individuals"
 type_synonym σ = "(i⇒bool)"
```

Docume

φ_o lifted to $\varphi_{i\rightarrow o}$ ("truth sets")

$\varphi_{i\rightarrow o} \vee \psi_{i\rightarrow o} = \lambda w_i(\varphi w \vee \psi w)$ encodes: $\{w \mid w \in \varphi \text{ or } w \in \psi\}$

$$\vee = \lambda\varphi_{i\rightarrow o}\lambda\psi_{i\rightarrow o}\lambda w_i\ (\varphi w \vee \psi w)$$

$\Box\varphi_{i\rightarrow o} = \lambda w_i \forall y_i\ (w\ r\ y \rightarrow \varphi y)$

$$\Box = \lambda\varphi_{i\rightarrow o}\lambda w_i \forall y_i\ (w\ r\ y \rightarrow \varphi y)$$

In HOL $\forall x_\mu.\varphi_o$ is syntactic sugar for $\Pi\, \overbrace{(\lambda x_\mu\ \varphi_o)}^{\phi_{\mu\rightarrow o}}$

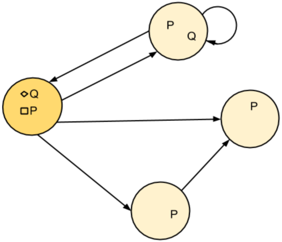
$\Pi\phi_{\mu\rightarrow(i\rightarrow o)} = \lambda w_i \forall x_\mu\ (\phi\ x\ w)$

$$\Pi = \lambda\phi_{\mu\rightarrow(i\rightarrow o)}\lambda w_i \forall x_\mu\ (\phi\ x\ w)$$

Output Query Sledgehammer Symbols

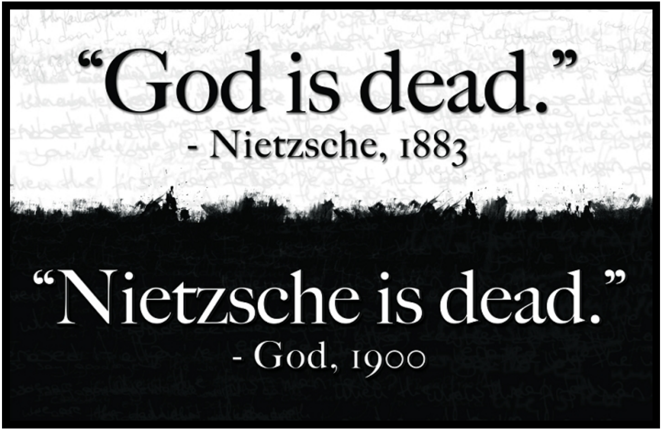
Universal Logical Reasoning in Isabelle/HOL

Properties of $\Box$ and $\Diamond$ correlated to structure of transition system between worlds



- Logic K: — (no restrictions, any structure)
- Logic M: reflexiv transition relation, $\forall P. \Box P \rightarrow P$
- Logic KB: symmetric transition relation, $\forall P. P \rightarrow \Box\Diamond P$
- Logic S5: equivelance relation as transition system, add $\forall P. \Box P \rightarrow \Box\Box P$

- Logic D: serial transition relation, $\forall P. \Box P \rightarrow \Diamond P$ (**S**tandard **D**eontic **L**ogic)
  (alternatively: $\forall P. \neg(\Box P \wedge \Box\neg P)$)

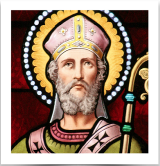


Part C — Evidence:
Experiments in Computational Metaphysics

[BenzmüllerWoltzenlogelPaleo, ECAI, 2014 + IJCAI, 2016 + KI 2016 + . . .]

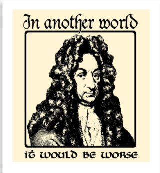
Ontological Proofs of God's Existence

A Long and Continuing Tradition in Philosophy



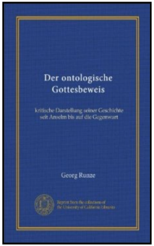
St. Anselm

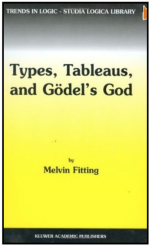
Descartes

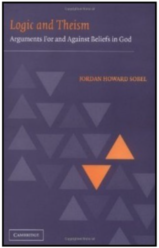


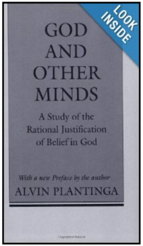
Leibniz

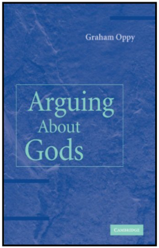
Gödel











Computational Metaphysics: Kurt Gödel's Ontological Argument

Ontologischer Beweis Feb 10, 1970

$P(\varphi)$ φ is positive ($\Leftrightarrow \varphi \in P$)

At. 1 $P(\varphi).P(\psi) \supset P(\varphi.\psi)$ At 2 $P(\varphi) \vee P(\sim\varphi)$

P1 $G(x) \equiv (\varphi)[P(\varphi) \supset \varphi(x)]$ (God)

P2 $\varphi \text{ Ess } x \equiv (\psi)[\psi(x) \supset N(y)[\varphi(y) \supset \psi(y)]]$ (Essence of x)

$p \supset_N q = N(p \supset q)$ Necessity

At 2 $P(\varphi) \supset NP(\varphi)$, $\sim P(\varphi) \supset N \sim P(\varphi)$ } because it follows from the nature of the property

Th. $G(x) \supset G \text{ Ess. } x$

Df. $E(x) \equiv (\varphi)[\varphi \text{ Ess } x \supset N \exists x\, \varphi(x)]$ necessary Existence

Ax 3 $P(E)$

Th. $G(x) \supset N(\exists y)\, G(y)$

hence $(\exists x)G(x) \supset N(\exists y)\, G(y)$

" $M(\exists x)G(x) \supset MN(\exists y)\, G(y)$ M = possibility

" $\supset N(\exists y)\, G(y)$

any two essences of x are nec. equivalent

exclusive or • and for any number of summands

$M(\exists x)G(x)$ means the system of all pos[...] patible This is tru[...]

At 4: $P(\varphi).\varphi \supset_N \psi$: [...]

{ $x = x$ is posi[...]; $x \neq x$ is negative

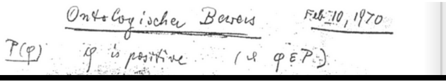
But if a system S of pos. prop. were incom[...] It would mean that the sum prop. s (which is positive) would be $x \neq x$

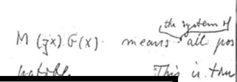
Positive means positive in the moral aesth[...] sense (independently of the accidental structure of the world). Only then the ax. true. It m[...] also meant pure "attribution" as opposed to "privation" (or containing privation). This interpret. simpler proof

If φ positive not: $(x) N \sim \varphi(x)$. Otherwise: $\varphi(x) \supset_N x \neq x$ hence $x \neq x$ positive not $x = x$ contrary At [...] or the exist. of pos. prop.

x i.e. the disj. normal form in terms of elem. prop. contains a member without negation.

Computational Metaphysics: Kurt Gödel's Ontological Argument





Computational Metaphysics: Dana Scott's Variant

Axiom A1 Either a property or its negation is positive, but not both: $\forall\phi[P(\neg\phi) \leftrightarrow \neg P(\phi)]$

Axiom A2 A property necessarily implied by a positive property is positive: $\forall\phi\forall\psi[(P(\phi) \wedge \Box\forall x[\phi(x) \rightarrow \psi(x)]) \rightarrow P(\psi)]$

Thm. T1 Positive properties are possibly exemplified: $\forall\phi[P(\phi) \rightarrow \Diamond\exists x\phi(x)]$

Def. D1 A *God-like* being possesses all positive properties: $G(x) \leftrightarrow \forall\phi[P(\phi) \rightarrow \phi(x)]$

Axiom A3 The property of being God-like is positive: $P(G)$

Cor. C Possibly, God exists: $\Diamond\exists x G(x)$

Axiom A4 Positive properties are necessarily positive: $\forall\phi[P(\phi) \rightarrow \Box P(\phi)]$

Def. D2 An *essence* of an individual is a property possessed by it and necessarily implying any of its properties: $\phi\ ess.\ x \leftrightarrow \phi(x) \wedge \forall\psi(\psi(x) \rightarrow \Box\forall y(\phi(y) \rightarrow \psi(y)))$

Thm. T2 Being God-like is an essence of any God-like being: $\forall x[G(x) \rightarrow G\ ess.\ x]$

Def. D3 *Necessary existence* of an individual is the necessary exemplification of all its essences: $NE(x) \leftrightarrow \forall\phi[\phi\ ess.\ x \rightarrow \Box\exists y\phi(y)]$

Axiom A5 Necessary existence is a positive property: $P(NE)$

Thm. T3 Necessarily, God exists: $\Box\exists x G(x)$

Computational Metaphysics: Kurt Gödel's Ontological Argument

Ontologischer Beweis Feb 10, 1970

P(φ) φ is positive (or φ ∈ P)

M (∃x) G(x) means the system of all pos

is negative

inconsistent

Computational Metaphysics: Dana Scott's Variant

Axiom A1 Either a property or its negation is positive, but not both: $\forall\phi[P(\neg\phi) \leftrightarrow \neg P(\phi)]$

Axiom A2 A property necessarily implied by a positive property is positive:
$$\forall\phi\forall\psi[(P(\phi) \wedge \Box\forall x[\phi(x) \rightarrow \psi(x)]) \rightarrow P(\psi)]$$

Thm. T1 Positive properties are possibly exemplified: $\forall\phi[P(\phi) \rightarrow \Diamond\exists x\phi(x)]$

Def. D1 A *God-like* being possesses all positive properties: $G(x) \leftrightarrow \forall\phi[P(\phi) \rightarrow \phi(x)]$

Axiom A3 The property of being God-like is positive: $P(G)$

Cor. C Possibly, God exists: $\Diamond\exists xG(x)$

Axiom A4 Positive properties are necessarily positive: $\forall\phi[P(\phi) \rightarrow \Box P(\phi)]$

Def. D2 An *essence* of an individual is a property possessed by it and necessarily implying any of its properties: $\phi\ ess.\ x \leftrightarrow \phi(x) \wedge \forall\psi(\psi(x) \rightarrow \Box\forall y(\phi(y) \rightarrow \psi(y)))$

Thm. T2 Being God-like is an essence of any God-like being: $\forall x[G(x) \rightarrow G\ ess.\ x]$

Def. D3 *Necessary existence* of an individual is the necessary exemplification of all its essences: $NE(x) \leftrightarrow \forall\phi[\phi\ ess.\ x \rightarrow \Box\exists y\phi(y)]$

Axiom A5 Necessary existence is a positive property: $P(NE)$

Thm. T3 Necessarily, God exists: $\Box\exists xG(x)$

C. Benzmüller, 2018 23

Computational Metaphysics: Kurt Gödel's Ontological Argument

Ontologischer Beweis   Feb 10, 1970

P(φ)   φ is positive   (∈ φ∈P)

M (∃x) G(x) means the system of all pos ... ... This is the ...

Computational Metaphysics: Dana Scott's Variant

Axiom A1 Either a property or its negation is positive, but not both: $\forall\phi[P(\neg\phi) \leftrightarrow \neg P(\phi)]$

Axiom A2 A property necessarily implied by a positive property is positive: $\forall\phi\forall\psi[(P(\phi) \wedge \Box\forall x[\phi(x) \rightarrow \psi(x)]) \rightarrow P(\psi)]$

Thr ... $\Diamond\exists x\phi(x)]$

De ... $\rightarrow \phi(x)]$

Axio ... $P(G)$

Co ... $\Diamond\exists xG(x)$

Axio ... $\rightarrow \Box P(\phi)]$

De ... plying ... $\rightarrow \psi(y)))$

- consistent
- argument valid already in logic KB
- monotheism
- modal collapse ($\varphi \rightarrow \Box\varphi$)—no free will

Thm. T2 Being God-like is an essence of any God-like being: $\forall x[G(x) \rightarrow G\ ess.\ x]$

Def. D3 *Necessary existence* of an individual is the necessary exemplification of all its essences: $NE(x) \leftrightarrow \forall\phi[\phi\ ess.\ x \rightarrow \Box\exists y\phi(y)]$

Axiom A5 Necessary existence is a positive property: $P(NE)$

Thm. T3 Necessarily, God exists: $\Box\exists xG(x)$

inconsistent

Computational Metaphysics: Vision of Leibniz (1646–1716) — *Calculemus!*

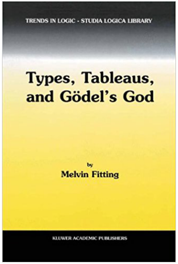
If controversies were to arise, there would be no more need of disputation between two philosophers than between two accountants. For it would suffice to take their pencils in their hands, to sit down to their slates, and to say to each other . . . : Let us calculate.

(Translation by Russell)

Quo facto, quando orientur controversiae, non magis disputatione opus erit inter duos philosophos, quam inter duos Computistas. Sufficiet enim calamos in manus sumere sedereque ad abacos, et sibi mutuo . . . dicere: calculemus. (Leibniz, 1684)

Required:
characteristica universalis and **calculus ratiocinator**

Further Experiments

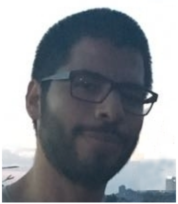


Melvin Fitting (New York)

Ontological Argument
(avoids modal collapse)

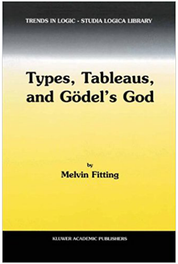
Intensional higher-order modal logic

Verified (main chapters)



David Fuenmayor
(Philosophy, FU Berlin)

Further Experiments

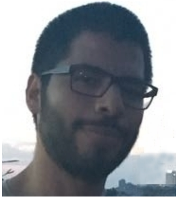


Melvin Fitting (New York)

Ontological Argument
(avoids modal collapse)

Intensional higher-order modal logic

Verified (main chapters)



David Fuenmayor
(Philosophy, FU Berlin)

Ed Zalta (Stanford)

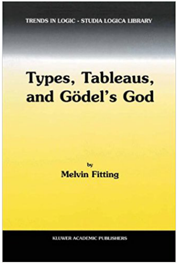
Principia Logico-Metaphysica

Hyperintensional higher-order modal logic

Inconsistency/Paradox detected

Daniel Kirchner
(Mathematics, FU Berlin)

Further Experiments

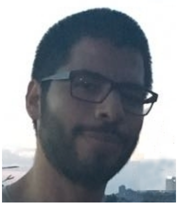


Melvin Fitting (New York)

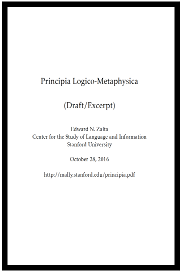
Ontological Argument
(avoids modal collapse)

Intensional higher-order modal logic

Verified (main chapters)



David Fuenmayor
(Philosophy, FU Berlin)



Principia Logico-Metaphysica

(Draft/Excerpt)

Edward N. Zalta
Center for the Study of Language and Information
Stanford University

October 28, 2016

http://mally.stanford.edu/principia.pdf

Ed Zalta (Stanford)

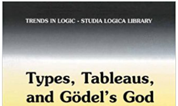
Principia Logico-Metaphysica

Hyperintensional higher-order modal logic

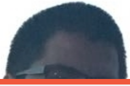
Inconsistency/Paradox detected

Daniel Kirchner
(Mathematics, FU Berlin)

Further Experiments



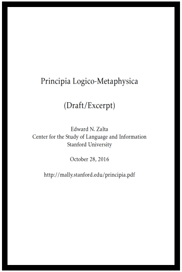
Ontological Argument
(avoids modal collapse)



Kirchner Paradox

Daniel & Isabelle/HOL are now closely collaborating with Ed Zalta

Computational Metaphysics par excellence!!!



Principia Logico-Metaphysica

(Draft/Excerpt)

Edward N. Zalta
Center for the Study of Language and Information
Stanford University

October 28, 2016

http://mally.stanford.edu/principia.pdf

Ed Zalta (Stanford)

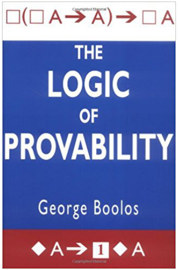
Principia Logico-Metaphysica

Hyperintensional higher-order modal logic

Inconsistency/Paradox detected

Daniel Kirchner
(Mathematics, FU Berlin)

Further Experiments

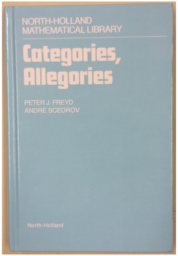


Textbook on Provability Logic

Provability Logic

Various parts verified

David Streit
(Mathematics, FU Berlin)



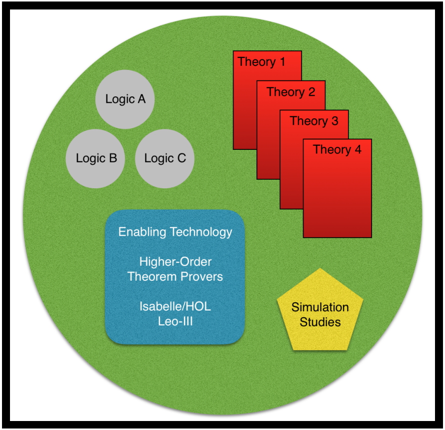
Category Theory

Free first-order logic

(Constricted) Inconsistency detected

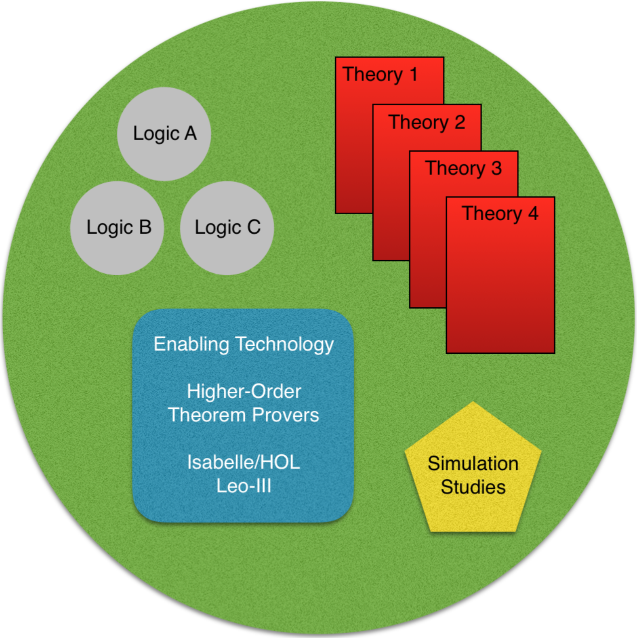
D. Scott
(UC Berkeley)

Papers on these topics: `http://christoph-benzmueller.de` –> Publications



Part D — Demo(s):
Normative Reasoning Experimentation Platform

Demo(s): Normative Reasoning Experimentation Platform



Demo(s): Normative Reasoning Experimentation Platform

Demo I

- Standard Deontic Logic (SDL) in Isabelle/HOL
- Dyadic Deontic Logic (DDL) in Isabelle/HOL
- Preference-based DDL in Isabelle/HOL

Demo II

- Input/Output-Logic in Isabelle/HOL

Demo III

- Gewirth's Principle of Generic Consistency (PGC) in Isabelle/HOL

Demo IV

- Native Support for Deontic Logic(s) in Leo-III

Demo(s): Normative Reasoning Experimentation Platform

Demo I

- **Standard Deontic Logic (SDL) in Isabelle/HOL**
- **Dyadic Deontic Logic (DDL) in Isabelle/HOL**
- Preference-based DDL in Isabelle/HOL

Demo II

- Input/Output-Logic in Isabelle/HOL

Demo III

- **Gewirth's Principle of Generic Consistency (PGC) in Isabelle/HOL**

Demo IV

- Native Support for Deontic Logic(s) in Leo-III

Demo I: SDL in Isabelle/HOL

[Logica Universalis, 2013]

SDL.thy

SDL.thy (~/chris/trunk/tex/talks/2018-DEON/DEMO/)

```
theory SDL imports Main                    (* Christoph Benzmüller & Xavier Parent, 2018 *)

begin  (* SDL: Standard Deontic Logic (Modal Logic D) *)
 typedecl i (*type for possible worlds*)  type_synonym σ = "(i⇒bool)"
 consts r::"i⇒i⇒bool" (infixr"r"70) (*Accessibility relation.*)  cw::i (*Current world.*)

 abbreviation mtop ("⊤") where "⊤ ≡ λw. True"
 abbreviation mbot ("⊥") where "⊥ ≡ λw. False"
 abbreviation mnot ("¬_"[52]53) where "¬φ ≡ λw. ¬φ(w)"
 abbreviation mand (infixr"∧"51) where "φ∧ψ ≡ λw. φ(w)∧ψ(w)"
 abbreviation mor  (infixr"∨"50) where "φ∨ψ ≡ λw. φ(w)∨ψ(w)"
 abbreviation mimp (infixr"→"49) where "φ→ψ ≡ λw. φ(w)⟶ψ(w)"
 abbreviation mequ (infixr"↔"48) where "φ↔ψ ≡ λw. φ(w)⟷ψ(w)"
 abbreviation mobligatory ("OB") where "OB φ ≡ λw. ∀v.  w r v ⟶ φ(v)" (*obligatory*)
 abbreviation mpermissible ("PE") where "PE φ ≡ ¬(OB(¬φ))" (*permissible*)
 abbreviation mimpermissible ("IM") where "IM φ ≡ OB(¬φ)"  (*impermissible*)
 abbreviation omissible ("OM") where "OM φ ≡ ¬(OB φ)" (*omissible*)
 abbreviation moptional ("OP") where "OP φ ≡ (¬(OB φ) ∧  ¬(OB(¬φ)))" (*optional*)

 abbreviation ddlvalid::"σ ⇒ bool" ("⌊_⌋"[7]105)      (*Global Validity*)
  where "⌊A⌋ ≡ ∀w. A w"
 abbreviation ddlvalidcw::"σ ⇒ bool" ("⌊_⌋cw"[7]105)  (*Local Validity (in cw)*)
  where "⌊A⌋cw ≡ A cw"

(* The D axiom is postulated *)
 axiomatization where D: "⌊¬ ((OB φ) ∧ (OB (¬ φ)))⌋"

(* Meta-level study: D corresponds to seriality *)
 lemma "⌊¬ ((OB φ) ∧ (OB (¬ φ)))⌋ ⟷ (∀w. ∃v. w r v)" by auto

(* Standardised syntax: unary operator for obligation in SDL *)
 abbreviation obligatorySDL::"σ⇒σ" ("O⟨_⟩") where "O⟨A⟩  ≡  OB A"

(* Consistency *)
  lemma True nitpick [satisfy] oops
```

Documentation Sidekick State Theories

Output Query Sledgehammer Symbols

Demo I: DDL in Isabelle/HOL

Completeness and decidability results for a logic of contrary-to-duty conditionals

JOSÉ M. C. L. M. CARMO, *Centre of Exact Sciences and Engineering, University of Madeira, Campus Universitario da Penteada, 9020-105 Funchal, Madeira, Portugal.*
E-mail: jcc@uma.pt
ANDREW J. I. JONES, *Department of Informatics, King's College London Strand, London WC2R 2LS, UK.*
E-mail: andrewji.jones@kcl.ac.uk

Abstract

This article has two parts. In Part I, we briefly outline the analysis of 'contrary-to-duty' obligation sentences presented in our 2002 handbook chapter 'Deontic logic and contrary-to-duties', with a focus on the intuitions that motivated the basic formal-logical moves we made. We also explain that the present account of the theory differs in two significant respects from the earlier version, one terminological, the other concerning the way the constituent modalities interconnect. Part II is the principal contribution of this article, in which we show that it is possible to define a complete and decidable axiomatization for the Carmo and Jones logic, a problem that was still open. The axiomatization includes two new inference rules; we illustrate their use in proofs, and show that on the basis of this axiomatization we can recover all the axioms and rules considered in 'Deontic logic and contrary-to-duties', and used there in the analysis of contrary-to-duty conditional scenarios.

Keywords: deontic logic, contrary-to-duty conditionals (CTDs), completeness and decidability results.

Demo I: DDL in Isabelle/HOL

Completeness and decidability results for a logic of contrary-to-duty conditionals

ıg, University
Madeira,
ondon

ıtences presented in
motivated the basic
ificant respects from
ɔnnect. Part II is the
e axiomatization for
e rules; we illustrate
l rules considered in
narios.

2.2 Section 2. Semantics

Our *models* are structures M=<W,av, pv, ob, V>, where:

(1) W is a non-empty set.
(2) V is a function assigning a truth set to each atomic sentence (i.e. $V(q) \subseteq W$).
(3) 'av' is a function (where $\wp(W)$ denotes the power set of W)
$av : W \rightarrow \wp(W)$
such that (where w denotes an arbitrary element of W):

(3a) $av(w) \neq \emptyset$

(4) $pv : W \rightarrow \wp(W)$ is such that:

(4a) $av(w) \subseteq pv(w)$
(4b) $w \in pv(w)$

(5) and $ob : \wp(W) \rightarrow \wp(\wp(W))$ is such that (where X,Y, Z designate arbitrary subsets of W)[7]:

(5a) $\emptyset \notin ob(X)$
(5b) if $Y \cap X = Z \cap X$, then ($Y \in ob(X)$ iff $Z \in ob(X)$)
(5c*) Let $\beta \subseteq ob(X)$ and $\beta \neq \emptyset$, i.e. let β be a non-empty set of elements of ob(X).
If $(\cap\beta) \cap X \neq \emptyset$ (where $\cap\beta = \{w \in W: \forall_{Z\in\beta}\ w \in Z\}$)
then $(\cap\beta) \in ob(X)$
(5d) if $Y \subseteq X$ and $Y \in ob(X)$ and $X \subseteq Z$, then $((Z-X) \cup Y) \in ob(Z)$
(5e) if $Y \subseteq X$ and $Z \in ob(X)$ and $Y \cap Z \neq \emptyset$, then $Z \in ob(Y)$

Demo I: DDL in Isabelle/HOL

Completeness and decidability results for a logic of contrary-to-duty conditionals

Given a model M=<W, … >, the elements of W are designated by *worlds* and (as above) in what follows we will use w, v, … to denote arbitrary worlds and X,Y, Z to denote arbitrary sets of worlds. Intuitively: av(w) denotes the set of actual versions of the world w; pv(w) denotes the set of potential versions of the world w; and ob(X) denotes the set of propositions which are obligatory in context X.

We write $M \models_w A$ to denote that formula A is true in the world w of a model M=<W, av, pv, ob, V>, and we define $||A||^M = \{w \in W : M \models_w A\}$. In order to simplify the presentation, whenever the model M is obvious from the context, we write ||A|| instead of $||A||^M$.

Truth in a world w *in a model* M=<W, av, pv, ob, V> is characterized as follows:

$M \models_w p$	iff	$w \in V(p)$
…		(the usual truth conditions for the connectives $\neg, \wedge, \vee, \rightarrow$ and $\leftrightarrow$)
$M \models_w \Box A$	iff	$\|\|A\|\| = W$
$M \models_w \Box_a A$	iff	$av(w) \subseteq \|\|A\|\|$
$M \models_w \Box_p A$	iff	$pv(w) \subseteq \|\|A\|\|$
$M \models_w O(B/A)$	iff	$\|\|A\|\| \cap \|\|B\|\| \neq \emptyset$ and $(\forall X)$(if $X \subseteq \|\|A\|\|$ and $X \cap \|\|B\|\| \neq \emptyset$, then $\|\|B\|\| \in ob(X)$)
$M \models_w O_a A$	iff	$\|\|A\|\| \in ob(av(w))$ and $av(w) \cap \|\|\neg A\|\| \neq \emptyset$
$M \models_w O_p A$	iff	$\|\|A\|\| \in ob(pv(w))$ and $pv(w) \cap \|\|\neg A\|\| \neq \emptyset$

A sentence *A* is said to be *true in a model* M=<W, av, pv, ob, V>, written $M \models A$, iff $||A||^M = W$; and *A* is said to be *valid*, written $\models A$, iff $M \models A$ in all models M.

Demo I: DDL in Isabelle/HOL

[see other DEON paper]

DDL.thy

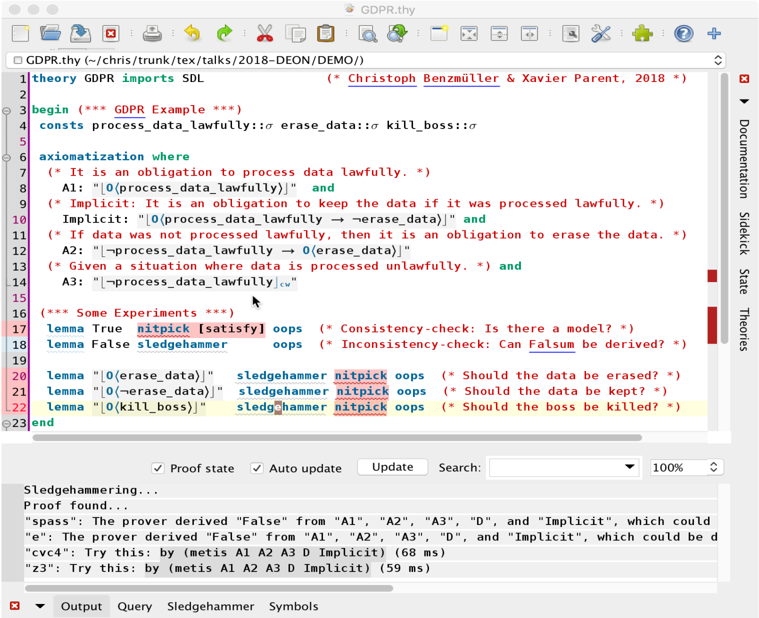
DDL.thy (~/chris/trunk/tex/talks/2018-DEON/DEMO/)

```
theory DDL imports Main          (* Christoph Benzmüller & Xavier Parent & Ali Farjami, 2018  *)

begin (* DDL: Dyadic Deontic Logic by Carmo and Jones *)
 typedecl i (*type for possible worlds*)   type_synonym σ = "(i⇒bool)"
 consts av::"i⇒σ" pv::"i⇒σ" ob::"σ⇒(σ⇒bool)" (*accessibility relations*) cw::i (*current world*)

 axiomatization where
  ax_3a: "∃x. av(w)(x)" and ax_4a: "∀x. av(w)(x) ⟶ pv(w)(x)" and ax_4b: "pv(w)(w)" and
  ax_5a: "¬ob(X)(λx. False)" and
  ax_5b: "(∀w. ((Y(w) ∧ X(w)) ⟷ (Z(w) ∧ X(w)))) ⟶ (ob(X)(Y) ⟷ ob(X)(Z))" and
  ax_5c: "((∀Z. β(Z) ⟶ ob(X)(Z)) ∧ (∃Z. β(Z))) ⟶
       (((∃y. ((λw. ∀Z. (β Z) ⟶ (Z w))(y) ∧ X(y))) ⟶ ob(X)(λw. ∀Z. (β Z) ⟶ (Z w))))" and
  ax_5d: "((∀w. Y(w) ⟶ X(w)) ∧ ob(X)(Y) ∧ (∀w. X(w) ⟶ Z(w)))
                 ⟶ ob(Z)(λw. (Z(w) ∧ ¬X(w)) ∨ Y(w))" and
  ax_5e: "((∀w. Y(w) ⟶ X(w)) ∧ ob(X)(Z) ∧ (∃w. Y(w) ∧ Z(w))) ⟶ ob(Y)(Z)"

 abbreviation ddlneg ("¬_"[52]53) where "¬A ≡ λw. ¬A(w)"
 abbreviation ddland (infixr"∧"51) where "A∧B ≡ λw. A(w)∧B(w)"
 abbreviation ddlor (infixr"∨"50) where "A∨B ≡ λw. A(w)∨B(w)"
 abbreviation ddlimp (infixr"→"49) where "A→B ≡ λw. A(w)⟶B(w)"
 abbreviation ddlequiv (infixr"↔"48) where "A↔B ≡ λw. A(w)⟷B(w)"
 abbreviation ddlbox ("□") where "□A ≡ λw.∀v. A(v)"  (*A = (λw. True)*)
 abbreviation ddlboxa ("□_a") where "□_aA ≡ λw. (∀x. av(w)(x) ⟶ A(x))"  (*in all actual worlds*)
 abbreviation ddlboxp ("□_p") where "□_pA ≡ λw. (∀x. pv(w)(x) ⟶ A(x))" (*in all potential worlds*)
 abbreviation ddldia ("◇") where "◇A ≡ ¬□(¬A)"
 abbreviation ddldiaa ("◇_a") where "◇_aA ≡ ¬□_a(¬A)"
 abbreviation ddldiap ("◇_p") where "◇_pA ≡ ¬□_p(¬A)"
 abbreviation ddlo ("O⟨_|_⟩"[52]53) where "O⟨B|A⟩ ≡ λw. ob(A)(B)"  (*it ought to be ψ, given φ *)
 abbreviation ddloa ("O_a") where "O_aA ≡ λw. ob(av(w))(A) ∧ (∃x. av(w)(x) ∧ ¬A(x))" (*actual obligation*)
 abbreviation ddlop ("O_p") where "O_pA ≡ λw. ob(pv(w))(A) ∧ (∃x. pv(w)(x) ∧ ¬A(x))"  (*primary obligation*)
 abbreviation ddltop ("⊤") where "⊤ ≡ λw. True"
 abbreviation ddlbot ("⊥") where "⊥ ≡ λw. False"

 abbreviation ddlvalid::"σ ⇒ bool" ("⌊_⌋"[7]105) where "⌊A⌋ ≡ ∀w. A w"   (*Global validity*)
 abbreviation ddlvalidcw::"σ ⇒ bool" ("⌊_⌋_cw"[7]105) where "⌊A⌋_cw ≡ A cw" (*Local validity (in cw)*)

(* A is obliagtory *)
 abbreviation obligatoryDDL::"σ⇒σ" ("O⟨_⟩")   where "O⟨A⟩  ≡  O⟨A|⊤⟩"

(* Consistency *)
  lemma True nitpick [satisfy] oops
```

Documentation Sidekick State Theories

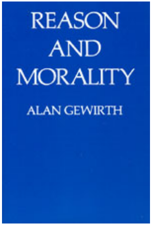
Output Query Sledgehammer Symbols

Demo I: Experimenting with SDL and DDL in Isabelle/HOL [arXiv:1804.02929]



Demo III: Gewirth's Principle of Generic Consistency (PGC) in Isabelle/HOL

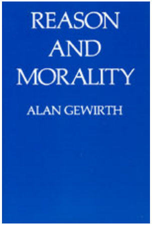
> "**Act in accord with the generic rights of your recipients as well as of yourself.** I shall call this the Principle of Generic Consistency (PGC), since it combines the formal consideration of consistency with the material consideration of rights to the generic features or goods of action." **(Alan Gewirth, Reason and Morality, Chicago U Press, 1978)**



- **Gewirth's PGC has**
  - stirred much controversy in moral philosophy
  - been discussed as means to bound the impact of artificial general intelligence (AGI)
- **Idea (in a nutshell):**
  - devise a safety mechanism of a mathematical (deductive) nature to ensure that an AGI respects human's freedom and well-being
  - mechanism is based on assumption that it is able to recognize itself, as well as us humans, as agents (prospective purposive agents, PPA) which
    (i) act voluntarily on self-chosen purposes, and
    (ii) reason rationally
- **Further References**
  - D. Beyleveld. The dialectical necessity of morality: An analysis and defense of Alan Gewirth's argument to the principle of generic consistency. U of Chicago Press, 1991
  - A. Kornai. Bounding the impact of AGI. J. Experimental & Theoretical AI, 2014

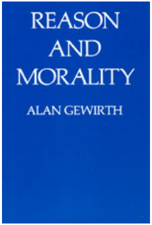
Demo III: Gewirth's Principle of Generic Consistency (PGC) in Isabelle/HOL

> "**Act in accord with the generic rights of your recipients as well as of yourself.** I shall call this the Principle of Generic Consistency (PGC), since it combines the formal consideration of consistency with the material consideration of rights to the generic features or goods of action." **(Alan Gewirth, Reason and Morality, Chicago U Press, 1978)**



- **Gewirth's PGC has**
  - stirred much controversy in moral philosophy
  - been discussed as means to bound the impact of artificial general intelligence (AGI)
- **Idea (in a nutshell):**
  - devise a safety mechanism of a mathematical (deductive) nature to ensure that an AGI respects human's freedom and well-being
  - mechanism is based on assumption that it is able to recognize itself, as well as us humans, as agents (prospective purposive agents, PPA) which (i) act voluntarily on self-chosen purposes, and (ii) reason rationally
- **Further References**
  - D. Beyleveld. The dialectical necessity of morality: An analysis and defense of Alan Gewirth's argument to the principle of generic consistency. U of Chicago Press, 1991
  - A. Kornai. Bounding the impact of AGI. J. Experimental & Theoretical AI, 2014

Demo III: Gewirth's Principle of Generic Consistency (PGC) in Isabelle/HOL



> "**Act in accord with the generic rights of your recipients as well as of yourself.** I shall call this the Principle of Generic Consistency (PGC), since it combines the formal consideration of consistency with the material consideration of rights to the generic features or goods of action." **(Alan Gewirth, Reason and Morality, Chicago U Press, 1978)**

- **Gewirth's PGC has**
  - stirred much controversy in moral philosophy
  - been discussed as means to bound the impact of artificial general intelligence (AGI)
- **Idea (in a nutshell):**
  - devise a safety mechanism of a mathematical (deductive) nature
    to ensure that an AGI respects human's freedom and well-being
  - mechanism is based on assumption that it is able to recognize itself, as well as us humans, as agents (prospective purposive agents, PPA) which
    (i) act voluntarily on self-chosen purposes, and
    (ii) reason rationally
- **Further References**
  - D. Beyleveld. The dialectical necessity of morality: An analysis and defense of Alan Gewirth's argument to the principle of generic consistency. U of Chicago Press, 1991
  - A. Kornai. Bounding the impact of AGI. J. Experimental & Theoretical AI, 2014

Demo III: Gewirth's Principle of Generic Consistency (PGC) in Isabelle/HOL

Idea is to constrain AGIs to reason in the following way

- For me, as an AGI, it is necessary to accept that:
  - **(P1)** **I act voluntarily on purpose E (equivalent by def. to: "I am a PPA")**
  - **(C2)** **E is good (for me)**
  - **(P3)** **In order to achieve any purpose whatsoever by my agency, I need my freedom and well-being**
  - **(C4)** **My freedom and well-being are necessary goods (for me)**
  - **(C5)** **I (even if no one else) have a claim right to my freedom and well-being**
- It is necessary for all PPAs to accept that:
  - **(C9)** **Every PPA has a necessary right to their freedom and well-being**

Demo III: Gewirth's Principle of Generic Consistency (PGC) in Isabelle/HOL

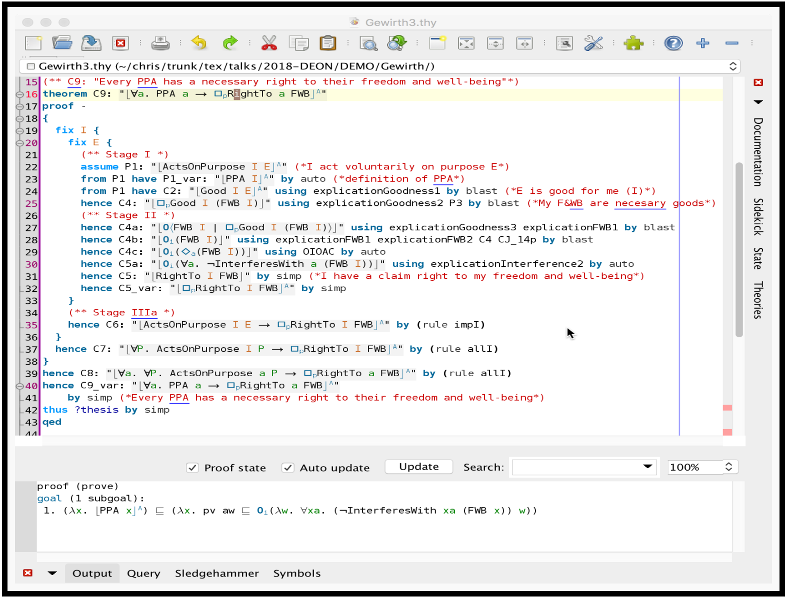
Idea is to constrain AGIs to reason in the following way

- For me, as an AGI, it is necessary to accept that:
  - **(P1)** **I act voluntarily on purpose E (equivalent by def. to: "I am a PPA")**
  - **(C2)** **E is good (for me)**
  - **(P3)** **In order to achieve any purpose whatsoever by my agency, I need my freedom and well-being**
  - **(C4)** **My freedom and well-being are necessary goods (for me)**
  - **(C5)** **I (even if no one else) have a claim right to my freedom and well-being**
- It is necessary for all PPAs to accept that:
  - **(C9)** **Every PPA has a necessary right to their freedom and well-being**

Any AGI (PPA) denying that it is bound by the PCG, e.g. by refusing to respect humans' well being, would deny that it is a PPA (and thus its own agency).

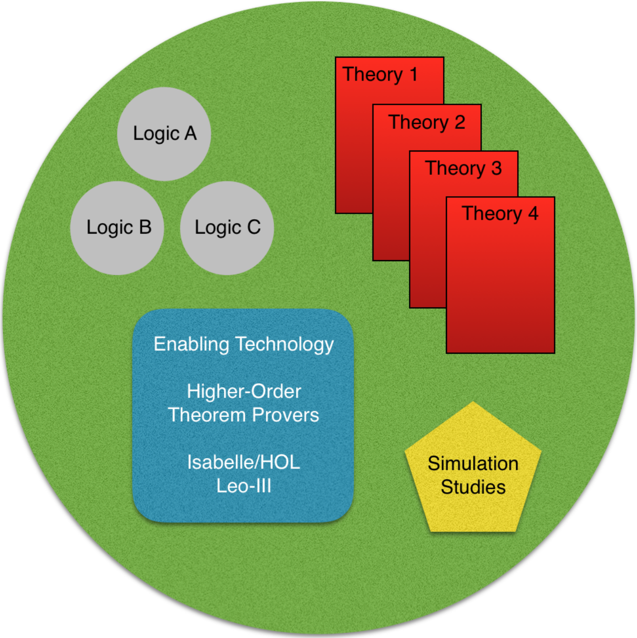
> Hence, to avoid self-contradiction, an AGI would be bound to accord basic rights to humans.

Demo III: Gewirth's Principle of Generic Consistency (PGC) in Isabelle/HOL



By David Fuenmayor, cf. `http://christoph-benzmueller.de/papers/2018-GewirthArgument.zip`

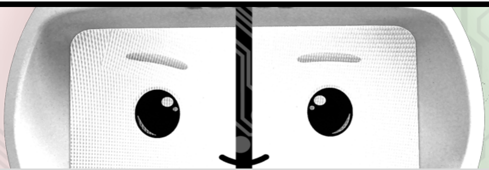
Demo III motivates Simulation Studies



Ethics

Argued for explicit ethical reasoning competencies in IASs

- normative reasoning experimentation platform
- HOL as universal meta-logic
- evidence from previous work
- very suitable for teaching logics



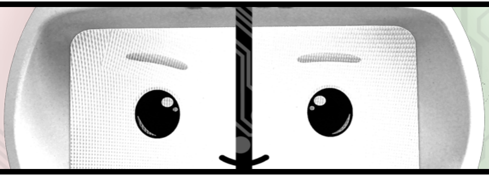
Ongoing and further work

- more (deontic) logics, more logic combinations
- encoding of ethical & legal theories
- experiments, … simulation studies, … deployment

Ethics

Argued for explicit ethical reasoning competencies in IASs

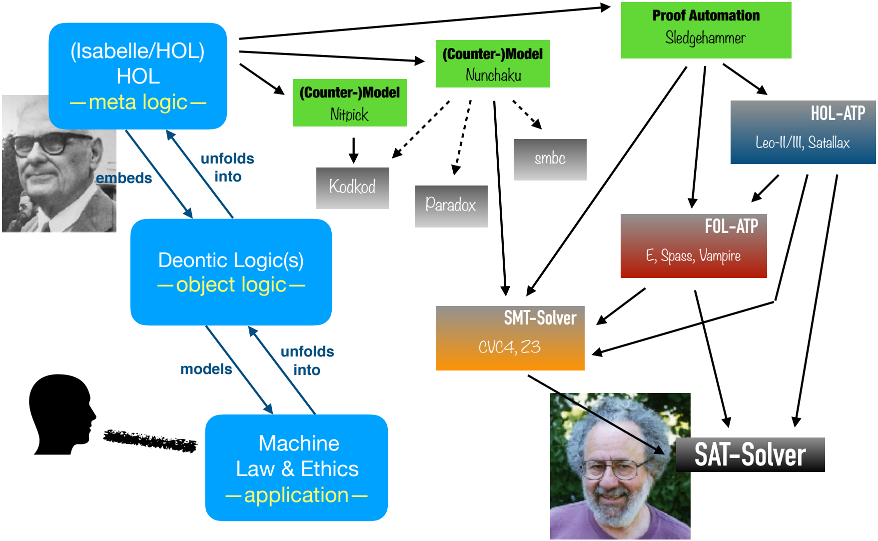
- normative reasoning experimentation platform
- HOL as universal meta-logic
- evidence from previous work
- very suitable for teaching logics



Ongoing and further work

- more (deontic) logics, more logic combinations
- encoding of ethical & legal theories
- experiments, . . . simulation studies, . . . deployment

How does Martin Davis fit in?



Before I forget: — A big thanks to —

University of Luxembourg:

ILIAS group of Leon van der Torre, many others

Research Grants:

DFG, Heisenberg grant: Computational Metaphysics, BE 2501/9, **2012-2017**

DFG, Project Leo-III: Higher-Order Theorem Prover, BE 2501/9, **2013-2017**

Various Collaborators:

B. Woltzenl.-P. (ANU Canberra)

Alexander Steen (FU Berlin)

Max Wisniewski (FU Berlin)

Ed Zalta (Stanford U.)

Dana Scott (UC Berkeley)

Many further Collaborators and Students:

Matthias Bentert (TU Berlin), Jasmin Blanchette (Amsterdam), Chad Brown (Prag), Maximilian Claus, David Fuenmayor, Tobias Gleißner, Kim Kern, Daniel Kirchner, Hanna Lachnitt, Irina Makarenko (alle FU Berlin), Larry Paulson (Cambridge), Fabian Schütz, Hans-Jörg Schurr, David Streit, Marco Ziener (alle FU Berlin), many further students in Berlin und Luxemburg

Demo I: Global vs. Local Consequence Relation

GDPRGlobal.thy

GDPRGlobal.thy (~/chris/trunk/tex/talks/2018-Bath/experiments/)

```
theory GDPRGlobal imports DDL        (* Christoph Benzmüller & Xavier Parent, 2018 *)

begin (*** GDPR Example ***)
 consts process_data_lawfully::σ erase_data::σ kill_boss::σ

 axiomatization where
  (* It is an obligation to process data lawfully. *)
    A1: "⌊O⟨process_data_lawfully⟩⌋"
  (* Given a situation where data is processed unlawfully. *) and
    A3: "⌊¬process_data_lawfully⌋"

 (*** Some Experiments ***)
  lemma True  nitpick [satisfy] nunchaku [satisfy] oops  (* Consistency-check: Is there a model? *)
  lemma False sledgehammer      oops  (* Inconsistency-check: Can Falsum be derived? *)

  lemma "⌊O⟨erase_data⟩⌋"    sledgehammer nitpick oops  (* Should the data be erased? *)
  lemma "⌊O⟨¬erase_data⟩⌋"   sledgehammer nitpick oops  (* Should the data be kept? *)
  lemma "⌊O⟨kill_boss⟩⌋"     sledgehammer nitpick oops  (* Should the boss be killed? *)
end
```

Documentation Sidekick State Theories

☑ Proof state ☑ Auto update Update Search: 100%

```
Sledgehammering...
Proof found...
"cvc4": Try this: using A1 A3 ax_5a ax_5b by auto (11 ms)
"z3": Try this: using A1 A3 ax_5a ax_5b by auto (2 ms)
"e": Try this: using A1 A3 ax_5a ax_5b by auto (3 ms)
"spass": The prover derived "False" from "A1", "A3", "ax_5a", and "ax_5b", which could be due to a bug
```

Output Query Sledgehammer Symbols

Demo I: Preference-based DDL in Isabelle/HOL

Journal of Philosophical Logic / Vol. 43, No. 6, December 2014 / Maximality vs. Optim...

JOURNAL ARTICLE

Maximality vs. Optimality in Dyadic Deontic Logic: Completeness Results for Systems in Hansson's Tradition

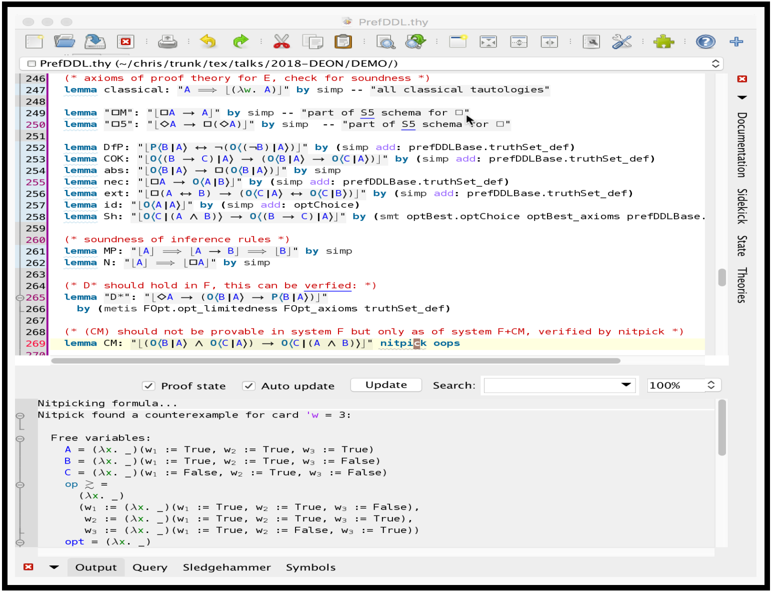
Xavier Parent

Journal of Philosophical Logic

Vol. 43, No. 6 (December 2014), pp. 1101-1128

Demo I: Preference-based DDL in Isabelle/HOL

Journal of Philosophical Logic / Vol. 43, No. 6, December 2014 / Maximality vs. Optim...



By
A. Steen

Demo II: I/O-Logic in Isabelle/HOL

[arXiv:1803.09681]

Input/output (I/O) logic

[Makinson, JPL, 2000], [GabbayHortyParentEtAl-Handbook, 2013]

- I/O-operators, such as out1 (simple-minded output), accept set G of conditional norms as argument
- Conditional norms: pairs (a,x) with input “a” (condition) and output “x” (obligation)
- Pairs (a,x) are not given a truth-functional semantics in I/O logic

Demo II: I/O-Logic in Isabelle/HOL

[arXiv:1803.09681]

Input/output (I/O) logic

[Makinson, JPL, 2000], [GabbayHortyParentEtAl-Handbook, 2013]

- I/O-operators, such as out1 (simple-minded output), accept set G of conditional norms as argument
- Conditional norms: pairs (a,x) with input “a” (condition) and output “x” (obligation)
- Pairs (a,x) are not given a truth-functional semantics in I/O logic

Semantics of out1 (for a of input formulas A)

- $out1(G, A) := Cn(G(Cn(A)))$
- where $Cn(X) := \{s \mid X \models s\}$ and $G(X) := \{s \mid \exists a \in X.\ (a, s) \in G\}$

Input/output (I/O) logic

[Makinson, JPL, 2000], [GabbayHortyParentEtAl-Handbook, 2013]

- I/O-operators, such as out1 (simple-minded output), accept set G of conditional norms as argument
- Conditional norms: pairs (a,x) with input “a” (condition) and output “x” (obligation)
- Pairs (a,x) are not given a truth-functional semantics in I/O logic

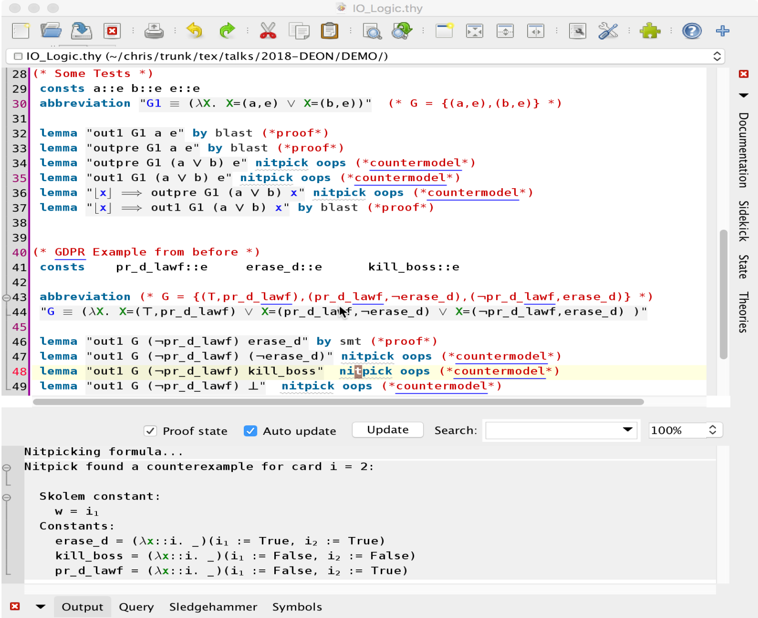
Semantics of out1 (for a of input formulas A)

- $out1(G,A) := Cn(G(Cn(A)))$
- where $Cn(X) := \{s \mid X \models s\}$ and $G(X) := \{s \mid \exists a \in X.\ (a,s) \in G\}$

```
(*IO logic in HOL*)
typedecl i -- "type for possible worlds"     type_synonym e = "(i⇒bool)"
abbreviation ktop   :: "e"     ("⊤")  where "⊤ ≡ λw. True"
abbreviation kbot   :: "e"     ("⊥")  where "⊥ ≡ λw. False"
abbreviation knot   :: "e⇒e"    ("¬_"[52]53)  where "¬φ ≡ λw. ¬φ(w)"
abbreviation kor    :: "e⇒e⇒e" (infixr"∨"50) where "φ∨ψ ≡ λw. φ(w)∨ψ(w)"
abbreviation kand   :: "e⇒e⇒e" (infixr"∧"51) where "φ∧ψ ≡ λw. φ(w)∧ψ(w)"
abbreviation kimp   :: "e⇒e⇒e" (infixr"⊃"49) where "φ⊃ψ ≡ λw. φ(w)⟶ψ(w)"
abbreviation kvalid :: "e⇒bool" ("⌊_⌋"[8]109)  where "⌊p⌋ ≡ ∀w. p w"

abbreviation "outpre ≡ λG.λa.λy::e. ∃f. ⌊a ⊃ f⌋ ∧ G (f,y)"

abbreviation "out1 ≡  λG.λa.λx. ⌊x⌋ ∨
   (∃i j k. outpre G a i ∧ outpre G a j ∧ outpre G a k ∧ ⌊(i ∧ j ∧ k) ⊃ x⌋)"
```

Demo II: I/O-Logic in Isabelle/HOL

[arXiv:1803.09681]



Demo IV: Native Support for Deontic Logic(s) in Leo-III [IJCAR, 2018], [RuleML+RR, 2018]

$\forall$P.Leo III

Leo III - A MASSIVELY PARALLEL HIGHER-ORDER THEOREM PROVER

What is Leo-III?

- ATP for classical HOL (by **A. Steen**, M. Wisniewski and myself)
- ordered paramodulation; efficient data-structures; parallelisation; etc.
- native support for more than 120 logics (all normal quantified modal logics)
- including native support for **quantified SDL and DDL**

- Website: `http://page.mi.fu-berlin.de/lex/leo3/`
- Download: `https://github.com/leoprover/Leo-III`

Demo IV: Native Support for Deontic Logic(s) in Leo-III [IJCAR, 2018], [RuleML+RR, 2018]

What is Leo-III?

- ATP for classical HOL (by **A. Steen**, M. Wisniewski and myself)
- ordered paramodulation; efficient data-structures; parallelisation; etc.
- native support for more than 120 logics (all normal quantified modal logics)
- including native support for **quantified SDL and DDL**

- Website: `http://page.mi.fu-berlin.de/lex/leo3/`
- Download: `https://github.com/leoprover/Leo-III`

Demo IV: Native Support for Deontic Logic(s) in Leo-III [IJCAR, 2018], [RuleML+RR, 2018]

Brand new: Support for Dyadic Deontic Logic (Carmo/Jones)

- Enhance propositional TPTP fragment with
  1. Dyadic deontic obligation $O(p/q)
  2. Actual/primary deontic obligations $O_a(p), $O_p(p)
  3. Box operators $box(p), $box_a(p),$box_p(p)
- Integrated into Leo-III (stand-alone tool available)

ASCII	Syntax	Meaning
~	$\neg$	Negation
|	$\vee$	Disjunction
&	$\wedge$	Conjunction
=>	$\Rightarrow$	Material implication
<=>	$\Leftrightarrow$	Equivalence
$O(p/q)	$O(p/q)$	Dyadic deontic obligation (It ought to be p given that q)
$box(p)	$\Box(p)$	In all worlds p

Input statements: ddl(<name>, <role>, <formula>).

Demo IV: Native Support for Deontic Logic(s) in Leo-III [IJCAR, 2018], [RuleML+RR, 2018]

Brand new: Support for Dyadic Deontic Logic (Carmo/Jones)

- Enhance propositional TPTP fragment with
  1. Dyadic deontic obligation $O(p/q)
  2. Actual/primary deontic obligations $O_a(p), $O_p(p)
  3. Box operators $box(p), $box_a(p),$box_p(p)
- Integrated into Leo-III (stand-alone tool available)

<table>
<tr><th>ASCII</th><th>Syntax</th><th>Meaning</th></tr>
<tr><td>~</td><td>$\neg$</td><td>Negation</td></tr>
<tr><td>|</td><td>$\vee$</td><td>Disjunction</td></tr>
<tr><td>&</td><td>$\wedge$</td><td>Conjunction</td></tr>
<tr><td>=></td><td>$\Rightarrow$</td><td>Material implication</td></tr>
<tr><td><=></td><td>$\Leftrightarrow$</td><td>Equivalence</td></tr>
<tr><td>$O(p/q)</td><td>$O(p/q)$</td><td>Dyadic deontic obligation (It ought to be p given that q)</td></tr>
<tr><td>$box(p)</td><td>$\Box(p)$</td><td>In all worlds p</td></tr>
</table>

Input statements: `ddl(<name>, <role>, <formula>).`

Demo IV: Native Support for Deontic Logic(s) in Leo-III [IJCAR, 2018], [RuleML+RR, 2018]

Input statements: `ddl(<name>, <role>, <formula>).`

where `<role>` provides meta-logical information:

- `axiom` *assumed, globally valid*
- `localAxiom` *assumed, valid in current world*
- `conjecture` *global consequence?*
- `localConjecture` *consequence in current world?*

Example

This problem can directly be given to Leo-III:

```
ddl(a1, axiom, $O(processDataLawfully)).
ddl(a2, axiom, $O(eraseData/~processDataLawfully)).
ddl(a3, localAxiom, ~processDataLawfully).

ddl(c1, conjecture, $O(eraseData)).
```

... giving ...

```
% SZS status Theorem for gdpr_new.p : 2143 ms resp. 776 ms w/o parsing
```

Demo IV: Native Support for Deontic Logic(s) in Leo-III [IJCAR, 2018], [RuleML+RR, 2018]

Input statements: `ddl(<name>, <role>, <formula>).`

where `<role>` provides meta-logical information:

- `axiom` *assumed, globally valid*
- `localAxiom` *assumed, valid in current world*
- `conjecture` *global consequence?*
- `localConjecture` *consequence in current world?*

Example

This problem can directly be given to Leo-III:

```
ddl(a1, axiom, $O(processDataLawfully)).
ddl(a2, axiom, $O(eraseData/~processDataLawfully)).
ddl(a3, localAxiom, ~processDataLawfully).

ddl(c1, conjecture, $O(eraseData)).
```

... giving ...

```
% SZS status Theorem for gdpr_new.p : 2143 ms resp. 776 ms w/o parsing
```

Demo IV: Native Support for Deontic Logic(s) in Leo-III [IJCAR, 2018], [RuleML+RR, 2018]

```
Leo3 — -bash — 77×29
leopard:Leo3 cbenzmueller$ more gdpr.p

ddl(a1, axiom, $O(processDataLawfully)).
ddl(a2, axiom, (~processDataLawfully) => $O(eraseData)).
ddl(a3, localAxiom, ~processDataLawfully).

ddl(c1, conjecture, $O(eraseData)).

leopard:Leo3 cbenzmueller$ leo3 gdpr_killboss.p --ddl
```

References (selected): http//:christoph-benzmueller.de –> Publications

Deontic Logic Reasoning Infrastructure

- A Dyadic Deontic Logic in HOL, DEON 2018, 2018. (See also https://arxiv.org/abs/1802.08454)
- A Deontic Logic Reasoning Infrastructure, CiE 2018, Springer LNCS, 2018.
- I/O Logic in HOL — First Steps, CoRR, 2018. https://arxiv.org/abs/1803.09681
- First Experiments with a Flexible Infrastructure for Normative Reasoning, CoRR, 2018. http://arxiv.org/abs/1804.02929

Computational Metaphysics (selected)

- Experiments in Computational Metaphysics: Gödel's Proof of God's Existence, Savijnanam: scientific exploration for a spiritual paradigm. Journal of the Bhaktivedanta Institute, volume 9, pp. 43-57, 2017.
- The Inconsistency in Gödel's Ontological Argument: A Success Story for AI in Metaphysics, IJCAI 2016, 2016.
- Automating Gödel's Ontological Proof of God's Existence with Higher-order Automated Theorem Provers, ECAI 2014, IOS Press, 2014.

Other (selected)

- The Higher-Order Prover Leo-III, IJCAR 2018, Springer LNCS, 2018.