

Deontic Logic and Artificial Intelligence

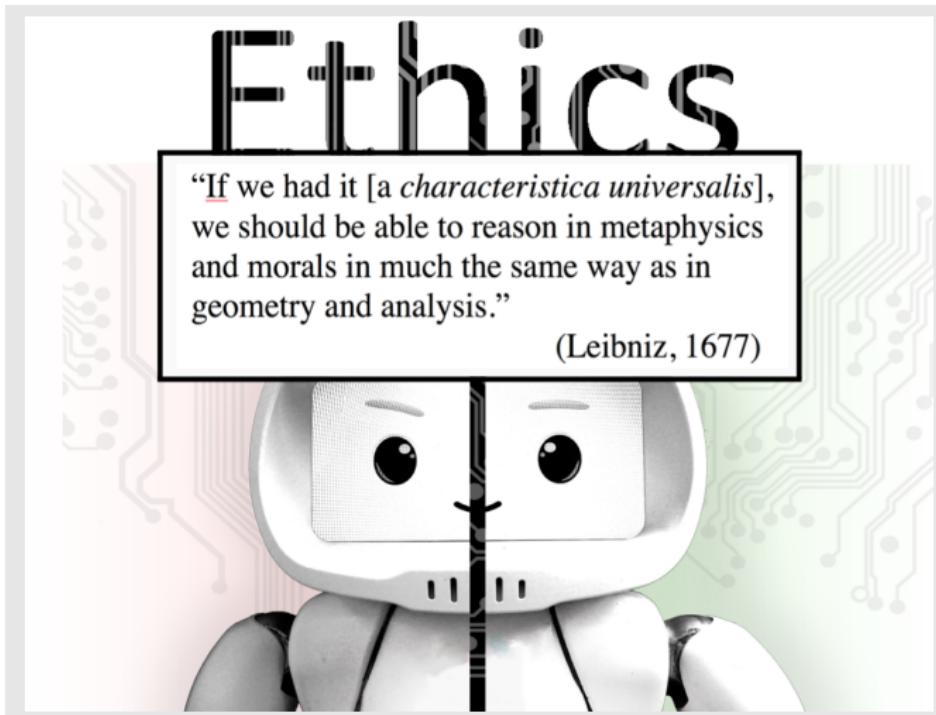
Christoph Benzmüller

Freie Universität Berlin | University of Luxembourg

Ethics

“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)

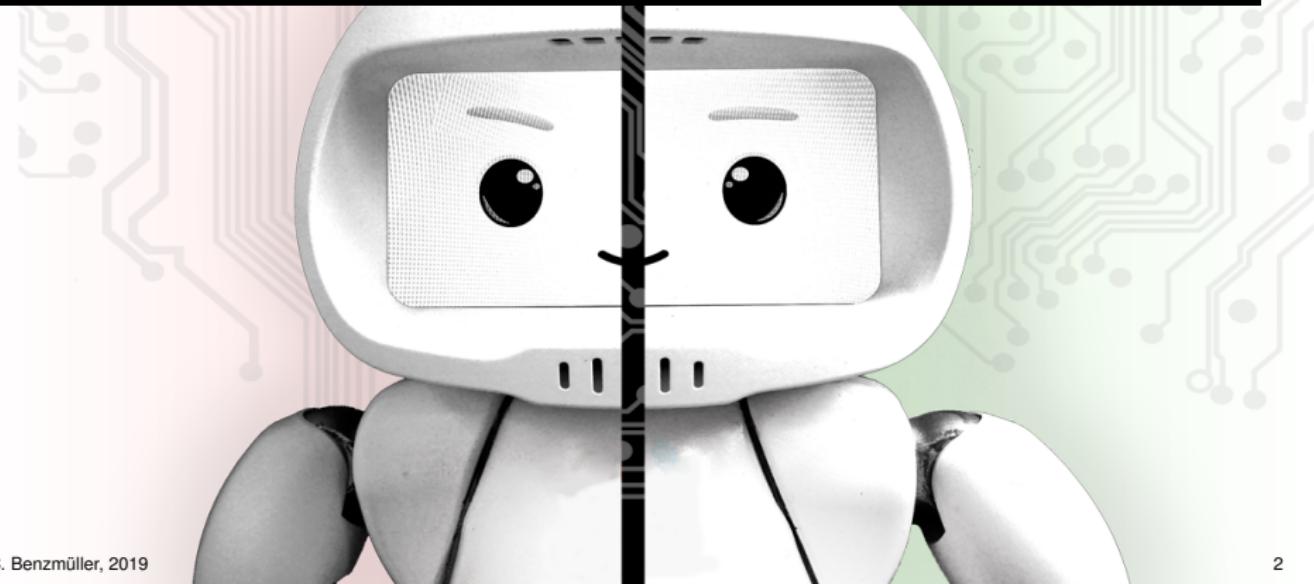


University of São Paulo, Faculty of Law — Aug 2, 2019

Ethics

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

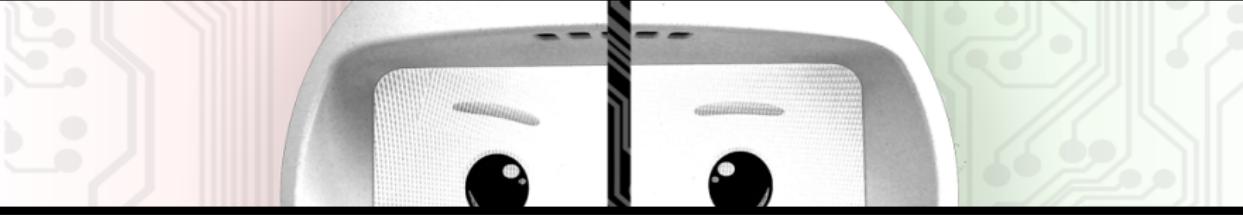
- ▶ appropriate forms of **machine-control**
- ▶ appropriate forms of **human-machine-interaction**



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?

Motivation: (Pseudo-)Ethical Intelligent Systems

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: (Pseudo-)Ethical Intelligent Systems

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: (Pseudo-)Ethical Intelligent Systems

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: (Pseudo-)Ethical Intelligent Systems

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:

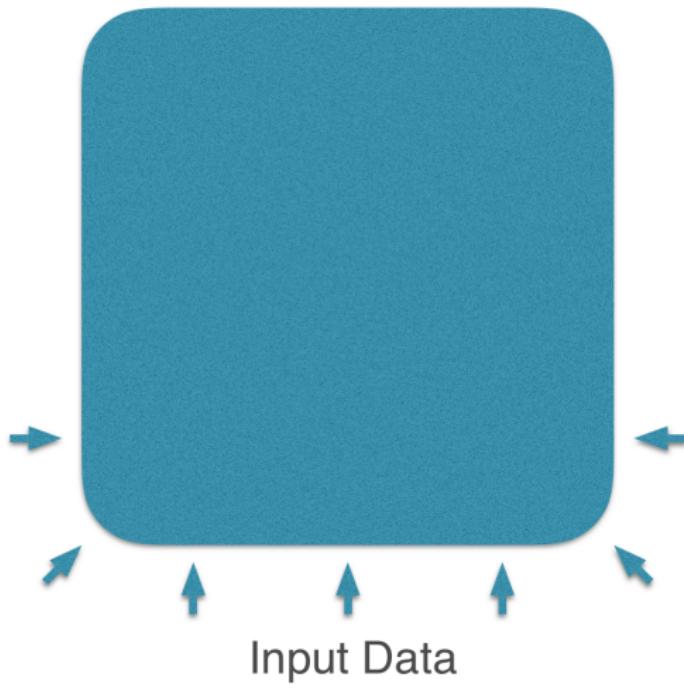
- ▶ [Moor, 2009]:
 - ethical impact agents (ethical consequences to actions)
 - implicit ethical agents (ethical reactions to given situations)
 - **explicit ethical agents** (reasoning with ethical theories/rules)
 - full ethical agents (conscious, intentional, free will)
- ▶ bottom-up vs. **top-down**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

(Pseudo-)Ethical Intelligent Systems

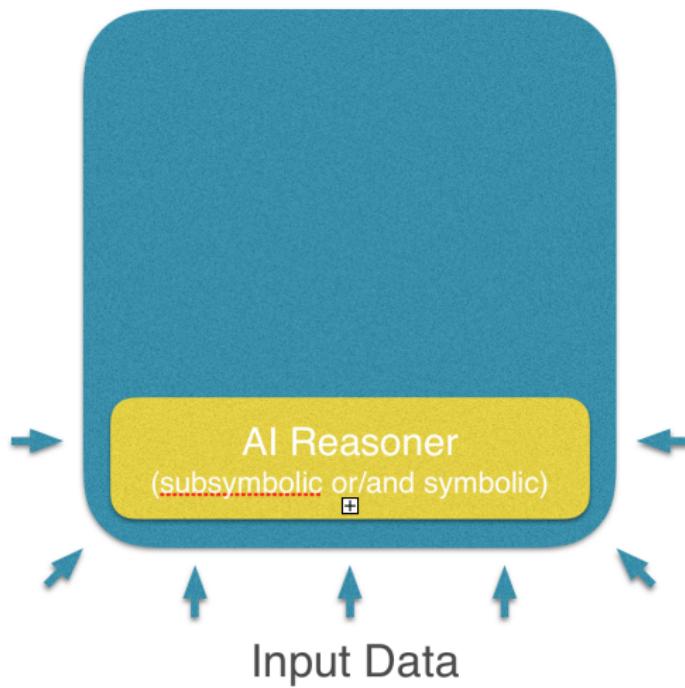


IAS

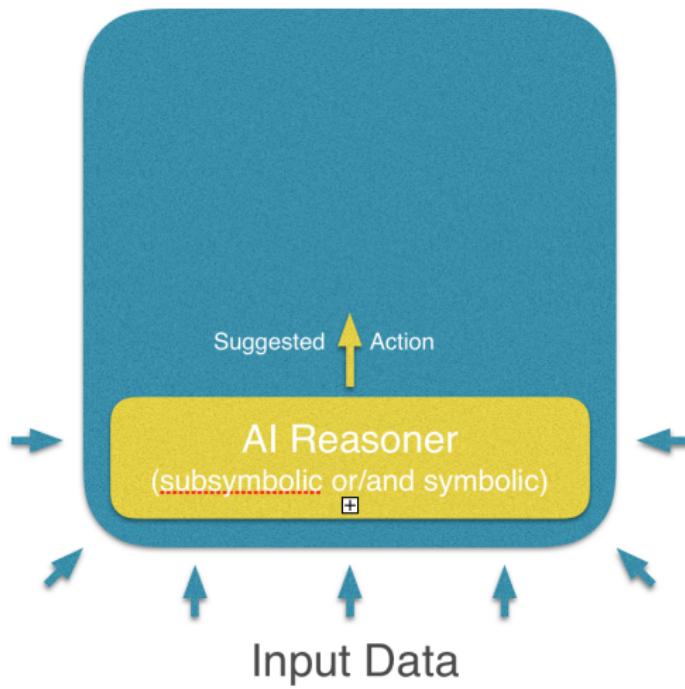
(Pseudo-)Ethical Intelligent Systems



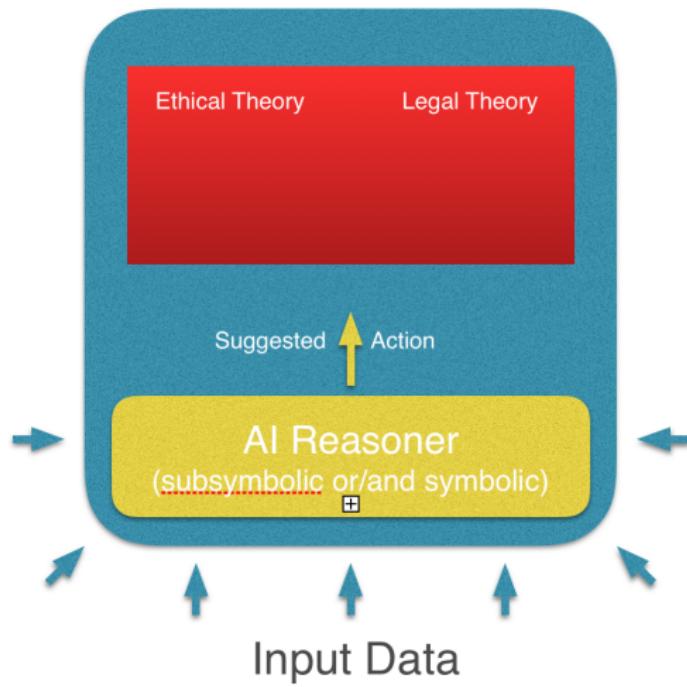
(Pseudo-)Ethical Intelligent Systems



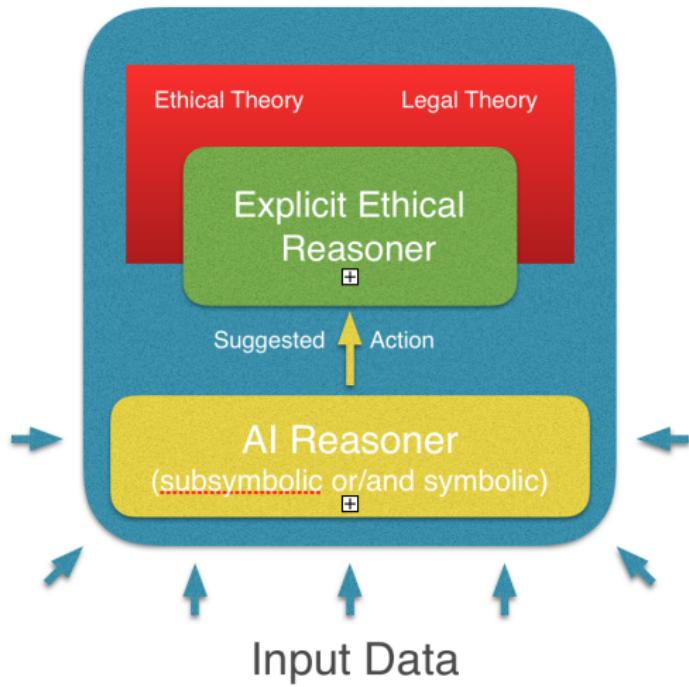
(Pseudo-)Ethical Intelligent Systems



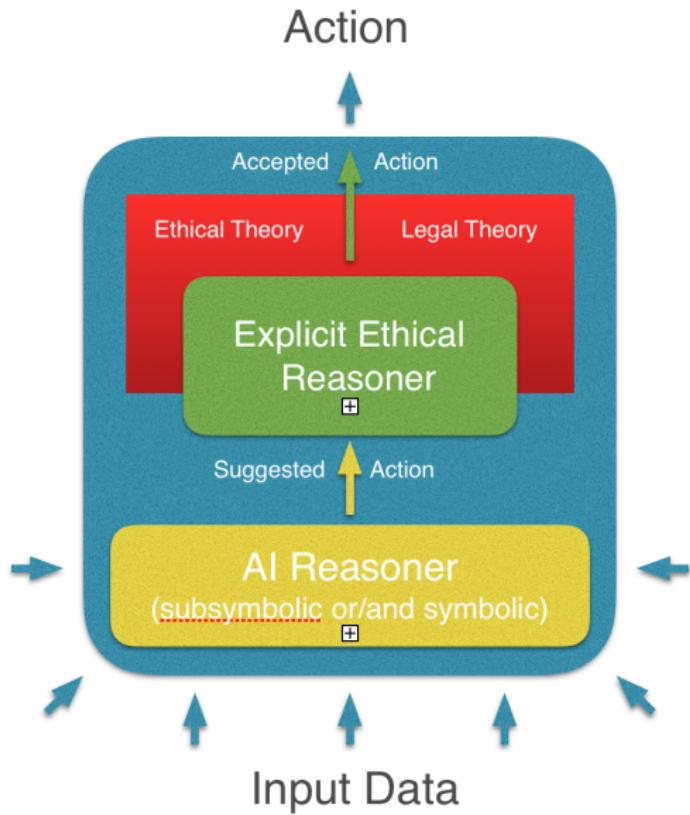
(Pseudo-)Ethical Intelligent Systems



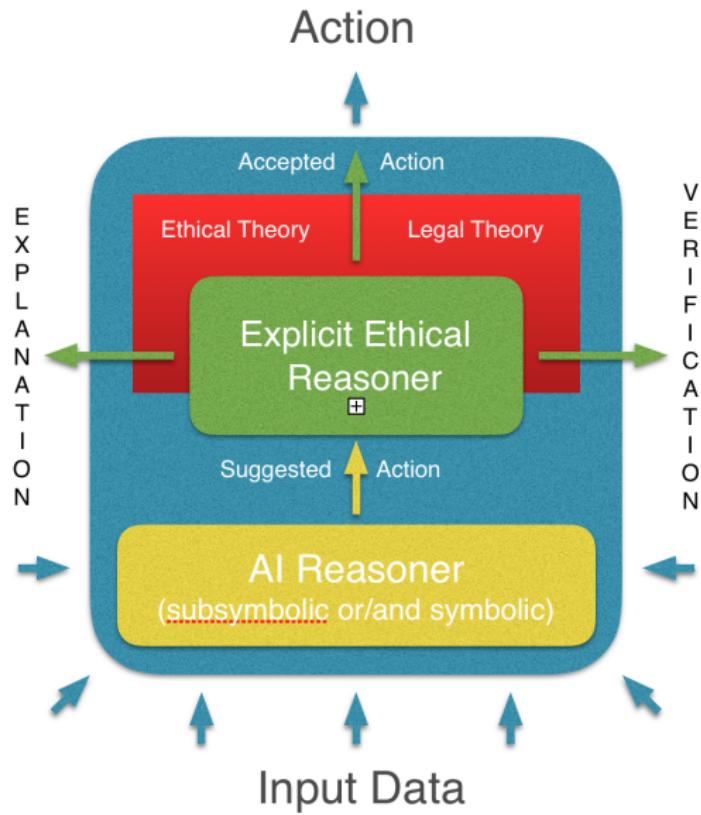
(Pseudo-)Ethical Intelligent Systems



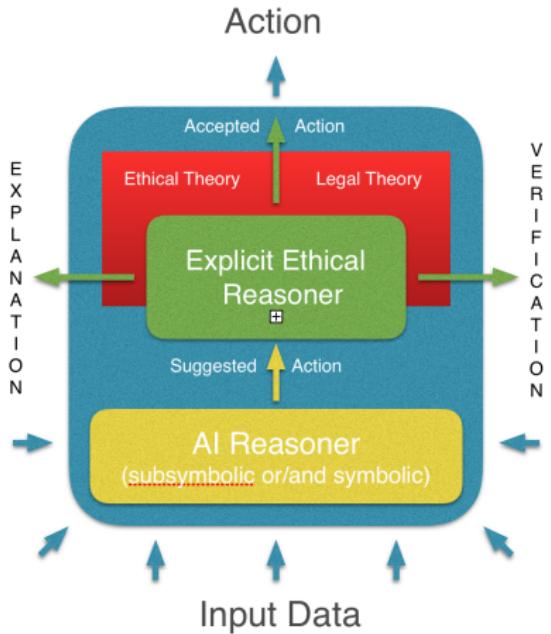
(Pseudo-)Ethical Intelligent Systems



(Pseudo-)Ethical Intelligent Systems



(Pseudo-)Ethical Intelligent Systems



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]
- ▶ ...

(Pseudo-)Ethical Intelligent Systems

Address requests towards **Thrustworthy & Responsible AI**

- ▶ **German National AI Strategy (Nov 2018)**

https://www.bmbf.de/files/Nationale_KI-Strategie.pdf

- ▶ **Recommendations from the EU HLEG group (May & June 2019)**

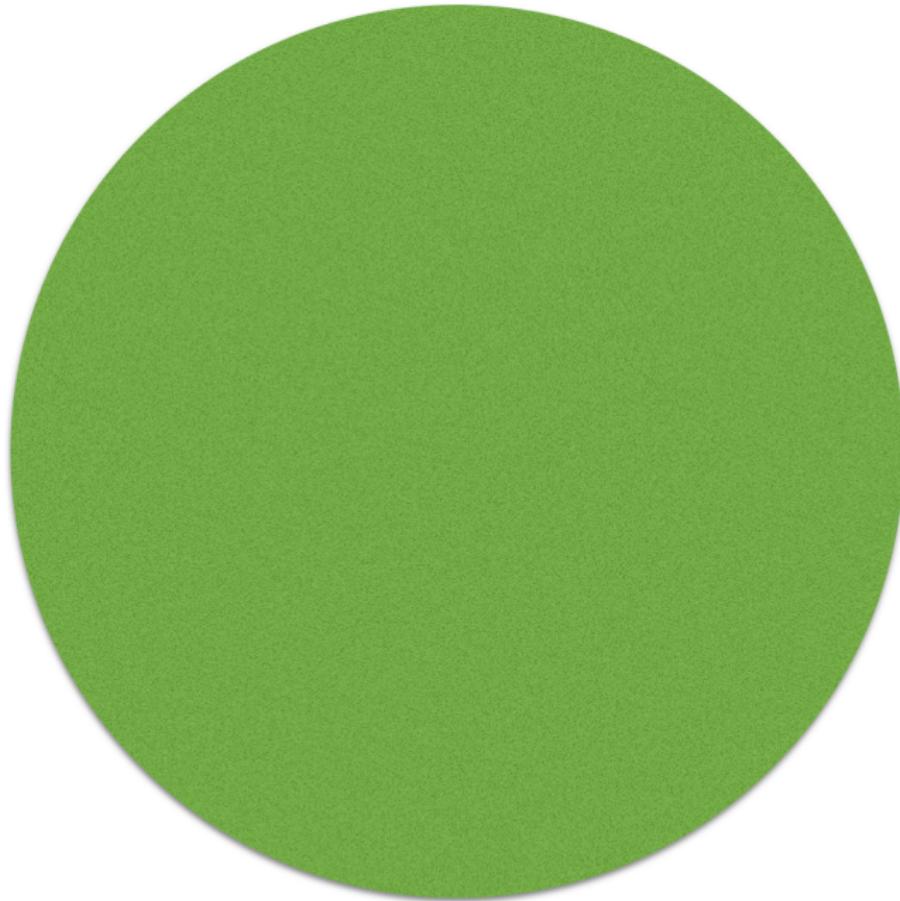


<https://ec.europa.eu/digital-single-market/en/high-level-expert-group-artificial-intelligence>

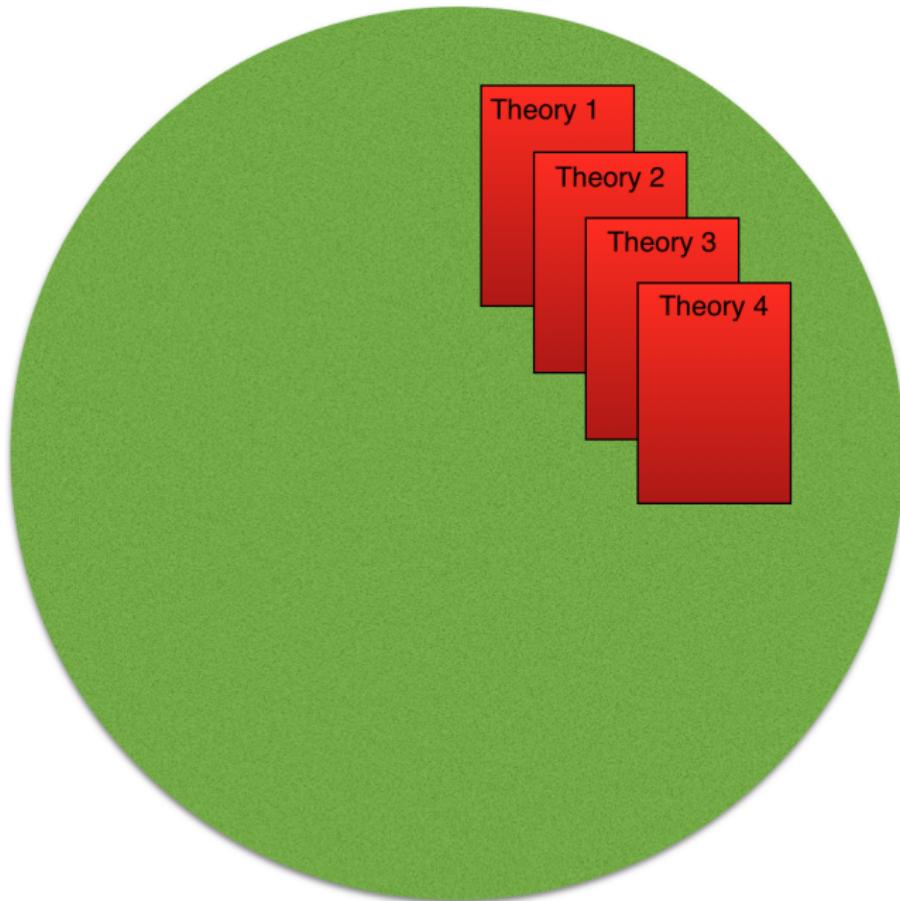
- ▶ **Ben Goertzel (CEO SingularityNET; Nov 2018): “Toward Democratic, Lawful Citizenship for AIs, Robots, and Corporations”**

<https://tinyurl.com/y8h94ouv>

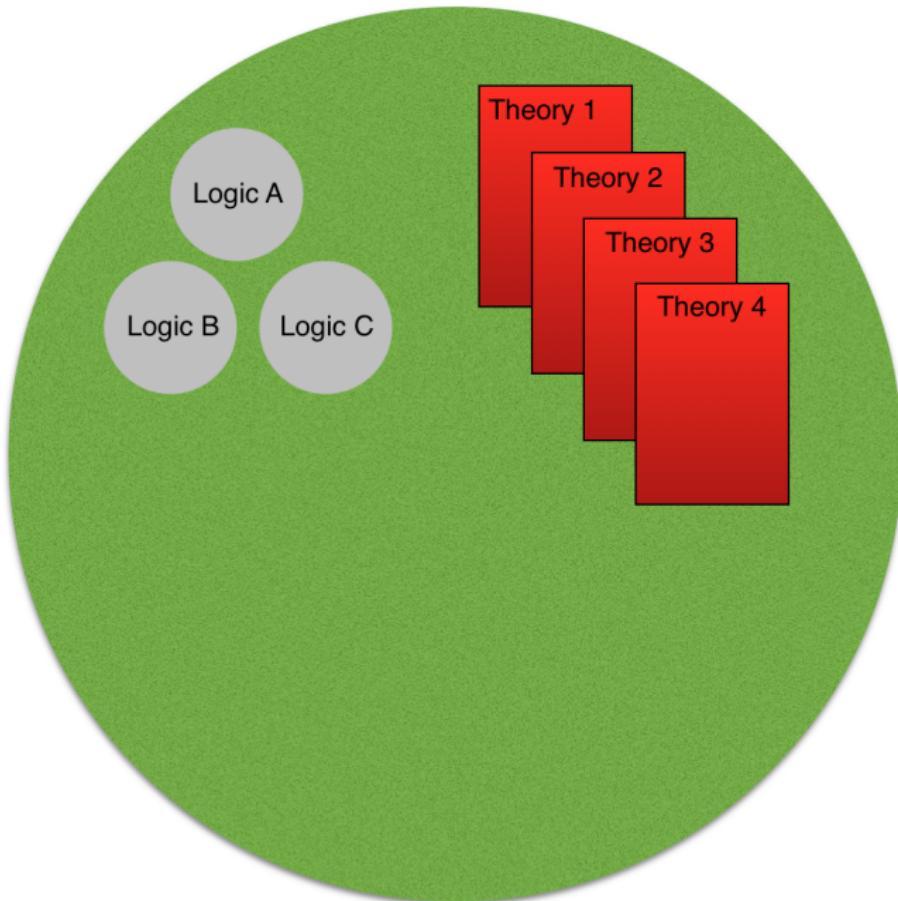
Normative Reasoning Experimentation Platform



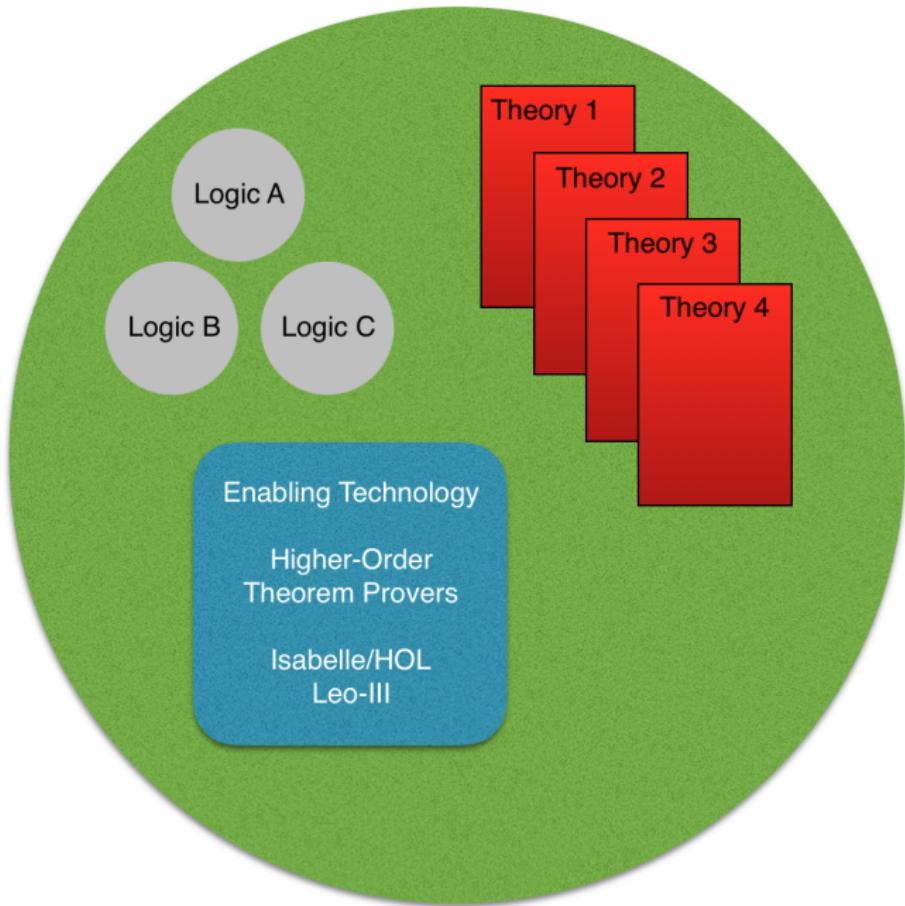
Normative Reasoning Experimentation Platform



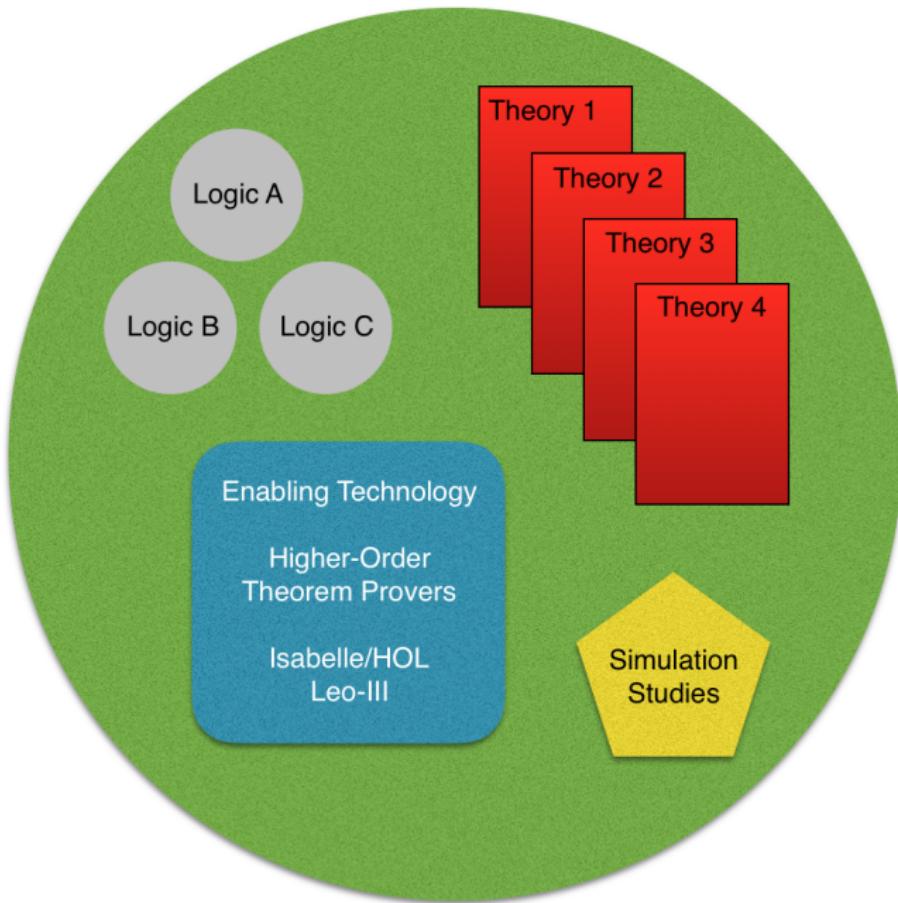
Normative Reasoning Experimentation Platform



Normative Reasoning Experimentation Platform



Normative Reasoning Experimentation Platform



“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)

**Flexible Reasoning Technology:
Shallow Semantical Embeddings (SSEs)
in Classical Higher-Order Logic (HOL)**

Flexible Reasoning Technology: SSEs in HOL

[Home](#) | [Video](#) | [Themen](#) | [Forum](#) | [English](#) | **DER SPIEGEL** | **SPIEGEL TV** | [Abo](#) | [Shop](#)

[RSS](#) | [Mobile](#) | [Newsletter](#)

SPIEGEL ONLINE INTERNATIONAL

[Sign in](#) | [Register](#)

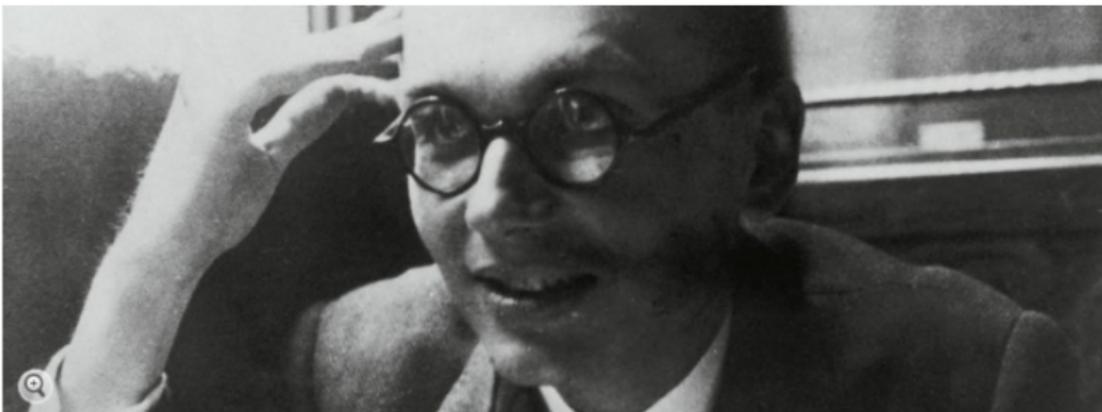


[Front Page](#) [World](#) [Europe](#) [Germany](#) [Business](#) [Zeitgeist](#) [Newsletter](#)

[English Site](#) > [Germany](#) > [Science](#) > Scientists Use Computer to Mathematically Prove Gödel God Theorem

Holy Logic: Computer Scientists 'Prove' God Exists

By David Knight



picture-alliance/ Imagno/ Wiener Stadt- und Landesbibliothek

Austrian mathematician Kurt Gödel kept his proof of God's existence a secret for decades. Now two scientists say they have proven it mathematically using a computer.

Two scientists have formalized a theorem regarding the existence of God penned by mathematician Kurt Gödel. But the God angle is somewhat of a red herring -- the real step forward is the example it sets of how computers can make scientific progress simpler.



STUDIES IN LOGIC
AND
PRACTICAL REASONING

VOLUME 3

D.M. GABBAY / P. GARDENFORS / J. SIEKMANN / J. VAN BENTHEM / M. VARDI / J. WOODS

EDITORS

*Handbook of
Modal Logic*

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:

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

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

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

Syntax

Metalanguage

WHAT FOLLOWS WE HIGHLIGHTLY ASSUME THAT PROP IS DEDUCIBLY 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:

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

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

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):

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

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*,

and V

$V(p)$

$(W, \{$
in the

in a model is a binary relation on W ,
position symbol p in PROP a subset
 p is true. The first two components
model. If there is only one relation
 (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

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

Kripke Style Semantics

$M, g, s \models P$ if and only if $s \in g(P)$

$M, g, s \models \neg \varphi$ if and only if $M, g, s \not\models \varphi$

$M, g, s \models \varphi \vee \psi$ if and only if $M, g, s \models \varphi$ or $M, g, s \models \psi$

$M, g, s \models \Box \varphi$ if and only if for all t with sRt we have $M, g, t \models \varphi$

Kripke Style Semantics

$M, g, s \models P$ if and only if $s \in g(P)$

$M, g, s \models \neg \varphi$ if and only if $M, g, s \not\models \varphi$

$M, g, s \models \varphi \vee \psi$ if and only if $M, g, s \models \varphi$ or $M, g, s \models \psi$

$M, g, s \models \Box \varphi$ if and only if for all t with sRt we have $M, g, t \models \varphi$

Standard Translation for Propositional Fragment (encoded in HOL)

- ▶ "lifted" predicate $P_{i \rightarrow o}$
- ▶ $\neg_{(i \rightarrow o) \rightarrow (i \rightarrow o)} \varphi_{i \rightarrow o} w_i = \neg(\varphi w)$
- ▶ $\vee_{(i \rightarrow o) \rightarrow (i \rightarrow o) \rightarrow (i \rightarrow o)} \varphi_{i \rightarrow o} \psi_{i \rightarrow o} w_i = \varphi w \vee \psi w$
- ▶ $\Box_{(i \rightarrow o) \rightarrow (i \rightarrow o)} \varphi_{i \rightarrow o} w_i = \forall v_i R w v \rightarrow \varphi v$

Kripke Style Semantics

$M, g, s \models P$ if and only if $s \in g(P)$

$M, g, s \models \neg \varphi$ if and only if $M, g, s \not\models \varphi$

$M, g, s \models \varphi \vee \psi$ if and only if $M, g, s \models \varphi$ or $M, g, s \models \psi$

$M, g, s \models \Box \varphi$ if and only if for all t with sRt we have $M, g, t \models \varphi$

Standard Translation for Propositional Fragment (encoded in HOL)

- ▶ "lifted" predicate $P_{i \rightarrow o}$
- ▶ $\neg_{(i \rightarrow o) \rightarrow (i \rightarrow o)} \varphi_{i \rightarrow o} = \lambda w_i \neg(\varphi w)$
- ▶ $\vee_{(i \rightarrow o) \rightarrow (i \rightarrow o) \rightarrow (i \rightarrow o)} \varphi_{i \rightarrow o} \psi_{i \rightarrow o} = \lambda w_i \varphi w \vee \psi w$
- ▶ $\Box_{(i \rightarrow o) \rightarrow (i \rightarrow o)} \varphi_{i \rightarrow o} = \lambda w_i \forall v_i R w v \rightarrow \varphi v$

Kripke Style Semantics

$M, g, s \models P$ if and only if $s \in g(P)$

$M, g, s \models \neg \varphi$ if and only if $M, g, s \not\models \varphi$

$M, g, s \models \varphi \vee \psi$ if and only if $M, g, s \models \varphi$ or $M, g, s \models \psi$

$M, g, s \models \Box \varphi$ if and only if for all t with sRt we have $M, g, t \models \varphi$

Standard Translation for Propositional Fragment (encoded in HOL)

- ▶ "lifted" predicate $P_{i \rightarrow o}$
- ▶ $\neg_{(i \rightarrow o) \rightarrow (i \rightarrow o)} = \lambda \varphi_{i \rightarrow o} \lambda w_i \neg(\varphi w)$
- ▶ $\vee_{(i \rightarrow o) \rightarrow (i \rightarrow o) \rightarrow (i \rightarrow o)} = \lambda \varphi_{i \rightarrow o} \lambda \psi_{i \rightarrow o} \lambda w_i \varphi w \vee \psi w$
- ▶ $\Box_{(i \rightarrow o) \rightarrow (i \rightarrow o)} = \lambda \varphi_{i \rightarrow o} \lambda w_i \forall v_i R w v \rightarrow \varphi v$

Kripke Style Semantics

$M, g, s \models P$ if and only if $s \in g(P)$

$M, g, s \models \neg \varphi$ if and only if $M, g, s \not\models \varphi$

$M, g, s \models \varphi \vee \psi$ if and only if $M, g, s \models \varphi$ or $M, g, s \models \psi$

$M, g, s \models \Box \varphi$ if and only if for all t with sRt we have $M, g, t \models \varphi$

Standard Translation for Propositional Fragment (encoded in HOL)

- ▶ "lifted" predicate $P_{i \rightarrow o}$
- ▶ $\neg_{(i \rightarrow o) \rightarrow (i \rightarrow o)} = \lambda \varphi_{i \rightarrow o} \lambda w_i \neg(\varphi w)$
- ▶ $\vee_{(i \rightarrow o) \rightarrow (i \rightarrow o) \rightarrow (i \rightarrow o)} = \lambda \varphi_{i \rightarrow o} \lambda \psi_{i \rightarrow o} \lambda w_i \varphi w \vee \psi w$
- ▶ $\Box_{(i \rightarrow o) \rightarrow (i \rightarrow o)} = \lambda \varphi_{i \rightarrow o} \lambda w_i \forall v_i R w v \rightarrow \varphi v$

Validity

- ▶ $[\varphi_{i \rightarrow o}] = \forall w_i \varphi w$ resp. $[.] = \lambda \varphi_{i \rightarrow o} \forall w_i \varphi w$ (global)
- ▶ $[\varphi_{i \rightarrow o}]_{cw} = \varphi cw$ resp. $[.]_{cw} = \lambda \varphi_{i \rightarrow o} \varphi cw$ (local)

Kripke Style Semantics

$M, g, s \models P$ if and only if $s \in g(P)$

$M, g, s \models \neg \varphi$ if and only if $M, g, s \not\models \varphi$

$M, g, s \models \varphi \vee \psi$ if and only if $M, g, s \models \varphi$ or $M, g, s \models \psi$

$M, g, s \models \Box \varphi$ if and only if for all t with sRt we have $M, g, t \models \varphi$

Standard Translation for Propositional Fragment (encoded in HOL)

- ▶ "lifted" predicate $P_{i \rightarrow o}$
- ▶ $\neg_{(i \rightarrow o) \rightarrow (i \rightarrow o)} = \lambda \varphi_{i \rightarrow o} \lambda w_i \neg(\varphi w)$
- ▶ $\vee_{(i \rightarrow o) \rightarrow (i \rightarrow o) \rightarrow (i \rightarrow o)} = \lambda \varphi_{i \rightarrow o} \lambda \psi_{i \rightarrow o} \lambda w_i \varphi w \vee \psi w$
- ▶ $\Box_{(i \rightarrow o) \rightarrow (i \rightarrow o)} = \lambda \varphi_{i \rightarrow o} \lambda w_i \forall v_i R w v \rightarrow \varphi v$

Validity

- ▶ $[\varphi_{i \rightarrow o}] = \forall w_i \varphi w$ resp. $[.] = \lambda \varphi_{i \rightarrow o} \forall w_i \varphi w$ (global)
- ▶ $[\varphi_{i \rightarrow o}]_{cw} = \varphi cw$ resp. $[.]_{cw} = \lambda \varphi_{i \rightarrow o} \varphi cw$ (local)

Modal Logic (in fact, Hybrid Logic) as a Fragment of HOL

Kripke Style Semantics

$M, g, s \models \forall x \varphi$ if and only if for all $d \in D$ we have $M, ([d/x]g), s \models \varphi$

Standard Translation extended for Quantifiers (and encoded in HOL)

- in HOL $\forall x_\alpha \varphi x$ is shorthand for $\Pi_{(\alpha \rightarrow o) \rightarrow o} (\lambda x_\alpha \varphi x)$ —no binder needed!!!
- $\Pi_{(\alpha \rightarrow (i \rightarrow o)) \rightarrow (i \rightarrow o)} = \lambda \Phi_{\alpha \rightarrow (i \rightarrow o)} \lambda w_i \Pi_{(\alpha \rightarrow o) \rightarrow o} (\lambda x_\alpha \Phi x w)$

Example (compositionality and λ -conversion at work)

$$\begin{aligned} [\Box \forall x Px] &\equiv [\Box \Pi(\lambda x Px)] \equiv [\Box \Pi(\lambda x \lambda w Pxw)] \\ &\equiv [\Box ((\lambda \Phi \lambda w \Pi(\lambda x \Phi x w))(\lambda x \lambda w Pxw))] \\ &\equiv [\Box (\lambda w \Pi(\lambda x (\lambda x \lambda w Pxw)xw))] \\ &\equiv [\Box (\lambda w \Pi(\lambda x Pxw))] \\ &\equiv [(\lambda \varphi \lambda w \Pi(\lambda v Rvv \rightarrow \varphi v))(\lambda w \Pi(\lambda x Pxw))] \\ &\equiv [(\lambda w \Pi(\lambda v Rvv \rightarrow (\lambda w \Pi(\lambda x Pxw))v))] \\ &\equiv [\lambda w \Pi(\lambda v Rvv \rightarrow \Pi(\lambda x Pxv))] \\ &\equiv [\lambda w \forall v (Rvv \rightarrow \forall x Pxv)] \\ &\equiv \forall w \forall v (Rvv \rightarrow \forall x Pxv) \end{aligned}$$

- above: possibilist quantification
- actualist quantification: $\Pi = \lambda \Phi \lambda w \Pi(\lambda x \text{existsAt } x w \rightarrow \Phi x w)$
- also supported: local and global validity and consequence

Kripke Style Semantics

$M, g, s \models \forall x \varphi$ if and only if for all $d \in D$ we have $M, ([d/x]g), s \models \varphi$

Standard Translation extended for Quantifiers (and encoded in HOL)

- in HOL $\forall x_\alpha \varphi x$ is shorthand for $\Pi_{(\alpha \rightarrow o) \rightarrow o} (\lambda x_\alpha \varphi x)$ —no binder needed!!!
- $\Pi_{(\alpha \rightarrow (i \rightarrow o)) \rightarrow (i \rightarrow o)} = \lambda \Phi_{\alpha \rightarrow (i \rightarrow o)} \lambda w_i \Pi_{(\alpha \rightarrow o) \rightarrow o} (\lambda x_\alpha \Phi x w)$

Example (compositionality and λ -conversion at work)

$$\begin{aligned} [\Box \forall x Px] &\equiv [\Box \Pi(\lambda x Px)] \equiv [\Box \Pi(\lambda x \lambda w Pxw)] \\ &\equiv [\Box ((\lambda \Phi \lambda w \Pi(\lambda x \Phi x w))(\lambda x \lambda w Pxw))] \\ &\equiv [\Box (\lambda w \Pi(\lambda x (\lambda x \lambda w Pxw)xw))] \\ &\equiv [\Box (\lambda w \Pi(\lambda x Pxw))] \\ &\equiv [(\lambda \varphi \lambda w \Pi(\lambda v Rvv \rightarrow \varphi v))(\lambda w \Pi(\lambda x Pxw))] \\ &\equiv [(\lambda w \Pi(\lambda v Rvv \rightarrow (\lambda w \Pi(\lambda x Pxw))v))] \\ &\equiv [\lambda w \Pi(\lambda v Rvv \rightarrow \Pi(\lambda x Pxv))] \\ &\equiv [\lambda w \forall v (Rvv \rightarrow \forall x Pxv)] \\ &\equiv \forall w \forall v (Rvv \rightarrow \forall x Pxv) \end{aligned}$$

- above: possibilist quantification
- actualist quantification: $\Pi = \lambda \Phi \lambda w \Pi(\lambda x \text{existsAt } x w \rightarrow \Phi x w)$
- also supported: local and global validity and consequence

Kripke Style Semantics

$M, g, s \models \forall x \varphi$ if and only if for all $d \in D$ we have $M, ([d/x]g), s \models \varphi$

Standard Translation extended for Quantifiers (and encoded in HOL)

- in HOL $\forall x_\alpha \varphi x$ is shorthand for $\Pi_{(\alpha \rightarrow o) \rightarrow o} (\lambda x_\alpha \varphi x)$ —no binder needed!!!
- $\Pi_{(\alpha \rightarrow (i \rightarrow o)) \rightarrow (i \rightarrow o)} = \lambda \Phi_{\alpha \rightarrow (i \rightarrow o)} \lambda w_i \Pi_{(\alpha \rightarrow o) \rightarrow o} (\lambda x_\alpha \Phi x w)$

Example (compositionality and λ -conversion at work)

$$\begin{aligned} [\Box \forall x Px] &\equiv [\Box \Pi(\lambda x Px)] \equiv [\Box \Pi(\lambda x \lambda w Pxw)] \\ &\equiv [\Box ((\lambda \Phi \lambda w \Pi(\lambda x \Phi x w))(\lambda x \lambda w Pxw))] \\ &\equiv [\Box (\lambda w \Pi(\lambda x (\lambda x \lambda w Pxw)xw))] \\ &\equiv [\Box (\lambda w \Pi(\lambda x Pxw))] \\ &\equiv [(\lambda \varphi \lambda w \Pi(\lambda v Rvv \rightarrow \varphi v))(\lambda w \Pi(\lambda x Pxw))] \\ &\equiv [(\lambda w \Pi(\lambda v Rvv \rightarrow (\lambda w \Pi(\lambda x Pxw))v))] \\ &\equiv [\lambda w \Pi(\lambda v Rvv \rightarrow \Pi(\lambda x Pxv))] \\ &\equiv [\lambda w \forall v (Rvv \rightarrow \forall x Pxv)] \\ &\equiv \forall w \forall v (Rvv \rightarrow \forall x Pxv) \end{aligned}$$

- above: possibilist quantification
- actualist quantification: $\Pi = \lambda \Phi \lambda w \Pi(\lambda x \text{existsAt } x w \rightarrow \Phi x w)$
- also supported: local and global validity and consequence

Kripke Style Semantics

$M, g, s \models \forall x \varphi$ if and only if for all $d \in D$ we have $M, ([d/x]g), s \models \varphi$

Standard Translation extended for Quantifiers (and encoded in HOL)

- in HOL $\forall x_\alpha \varphi x$ is shorthand for $\Pi_{(\alpha \rightarrow o) \rightarrow o} (\lambda x_\alpha \varphi x)$ —no binder needed!!!
- $\Pi_{(\alpha \rightarrow (i \rightarrow o)) \rightarrow (i \rightarrow o)} = \lambda \Phi_{\alpha \rightarrow (i \rightarrow o)} \lambda w_i \Pi_{(\alpha \rightarrow o) \rightarrow o} (\lambda x_\alpha \Phi x w)$

Example (compositionality and λ -conversion at work)

$$\begin{aligned} [\Box \forall x Px] &\equiv [\Box \Pi(\lambda x Px)] \equiv [\Box \Pi(\lambda x \lambda w Pxw)] \\ &\equiv [\Box ((\lambda \Phi \lambda w \Pi(\lambda x \Phi xw))(\lambda x \lambda w Pxw))] \\ &\equiv [\Box (\lambda w \Pi(\lambda x (\lambda x \lambda w Pxw)xw))] \\ &\equiv [\Box (\lambda w \Pi(\lambda x Pxw))] \\ &\equiv [(\lambda \varphi \lambda w \Pi(\lambda v Rvv \rightarrow \varphi v))(\lambda w \Pi(\lambda x Pxw))] \\ &\equiv [(\lambda w \Pi(\lambda v Rvv \rightarrow (\lambda w \Pi(\lambda x Pxw))v))] \\ &\equiv [\lambda w \Pi(\lambda v Rvv \rightarrow \Pi(\lambda x Pxv))] \\ &\equiv [\lambda w \forall v (Rvv \rightarrow \forall x Pxv)] \\ &\equiv \forall w \forall v (Rvv \rightarrow \forall x Pxv) \end{aligned}$$

- above: possibilist quantification
- actualist quantification: $\Pi = \lambda \Phi \lambda w \Pi(\lambda x \text{existsAt } x w \rightarrow \Phi xw)$
- also supported: local and global validity and consequence

Kripke Style Semantics

$M, g, s \models \forall x \varphi$ if and only if for all $d \in D$ we have $M, ([d/x]g), s \models \varphi$

Standard Translation extended for Quantifiers (and encoded in HOL)

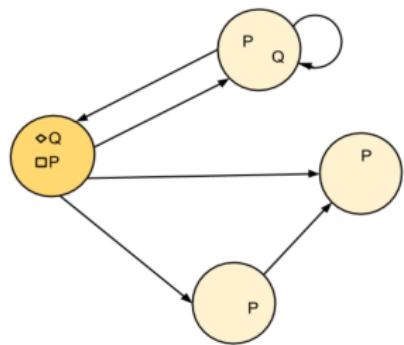
- in HOL $\forall x_\alpha \varphi x$ is shorthand for $\Pi_{(\alpha \rightarrow o) \rightarrow o} (\lambda x_\alpha \varphi x)$ —no binder needed!!!
- $\Pi_{(\alpha \rightarrow (i \rightarrow o)) \rightarrow (i \rightarrow o)} = \lambda \Phi_{\alpha \rightarrow (i \rightarrow o)} \lambda w_i \Pi_{(\alpha \rightarrow o) \rightarrow o} (\lambda x_\alpha \Phi x w)$

Example (compositionality and λ -conversion at work)

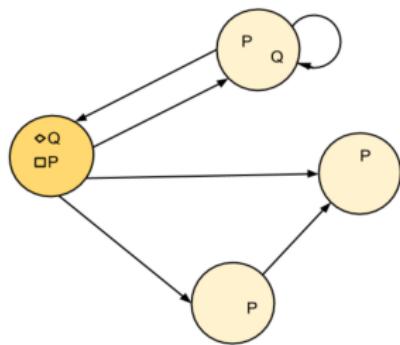
$$\begin{aligned} [\Box \forall x Px] &\equiv [\Box \Pi(\lambda x Px)] \equiv [\Box \Pi(\lambda x \lambda w Pxw)] \\ &\equiv [\Box ((\lambda \Phi \lambda w \Pi(\lambda x \Phi xw))(\lambda x \lambda w Pxw))] \\ &\equiv [\Box (\lambda w \Pi(\lambda x (\lambda x \lambda w Pxw)xw))] \\ &\equiv [\Box (\lambda w \Pi(\lambda x Pxw))] \\ &\equiv [(\lambda \varphi \lambda w \Pi(\lambda v Rvv \rightarrow \varphi v))(\lambda w \Pi(\lambda x Pxw))] \\ &\equiv [(\lambda w \Pi(\lambda v Rvv \rightarrow (\lambda w \Pi(\lambda x Pxw))v))] \\ &\equiv [\lambda w \Pi(\lambda v Rvv \rightarrow \Pi(\lambda x Pxv))] \\ &\equiv [\lambda w \forall v (Rvv \rightarrow \forall x Pxv)] \\ &\equiv \forall w \forall v (Rvv \rightarrow \forall x Pxv) \end{aligned}$$

- above: possibilist quantification
- actualist quantification: $\Pi = \lambda \Phi \lambda w \Pi(\lambda x \text{existsAt } x w \rightarrow \Phi xw)$
- also supported: local and global validity and consequence

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

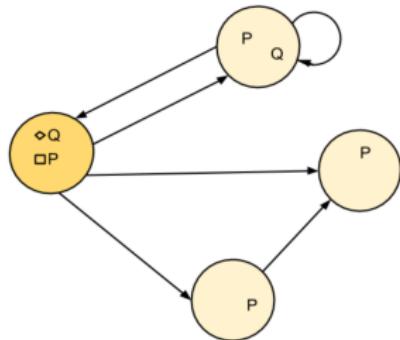


Properties of \Box and \Diamond correlated to structure of transition system between worlds



- ▶ 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: **equivalence relation** as transition system, add $\forall P. \Box P \rightarrow \Box \Box P$

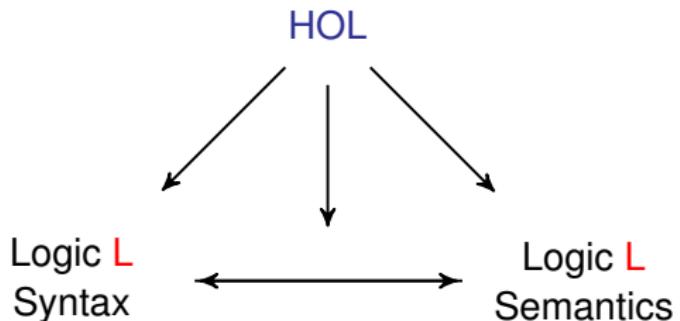
Properties of \Box and \Diamond correlated to structure of transition system between worlds



- ▶ 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: **equivalence relation** as transition system, add $\forall P. \Box P \rightarrow \Box \Box P$

- ▶ Logic D: **serial** transition relation, $\forall P. \Box P \rightarrow \Diamond P$ (**Standard Deontic Logic**)
alternatively: $\forall P. \neg(\Box P \wedge \Box \neg P)$ or $\forall x. \exists y. x R y$

Flexible Reasoning Technology: SSEs in 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, Deontic Logics, ...

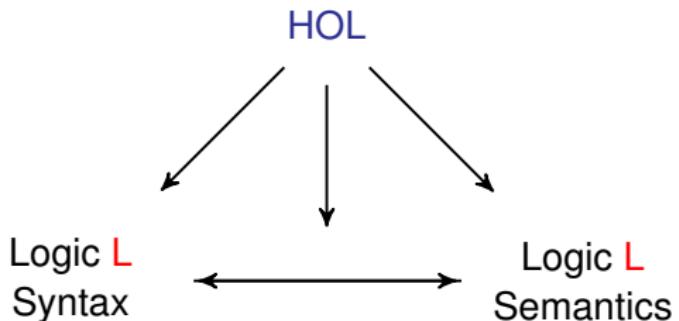
Embedding works also for quantifiers (first-order & higher-order)

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, ...

Flexible Reasoning Technology: SSEs in 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, Deontic Logics, ...

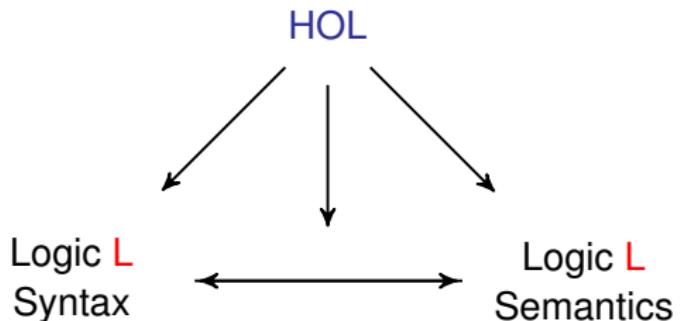
Embedding works also for quantifiers (first-order & higher-order)

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, ...

Flexible Reasoning Technology: SSEs in 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, Deontic Logics, ...

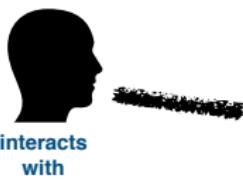
Embedding works also for quantifiers (first-order & higher-order)

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, ...

Flexible Reasoning Technology: SSEs in HOL [SCP, 2019], [OpenPhilos., 2019]

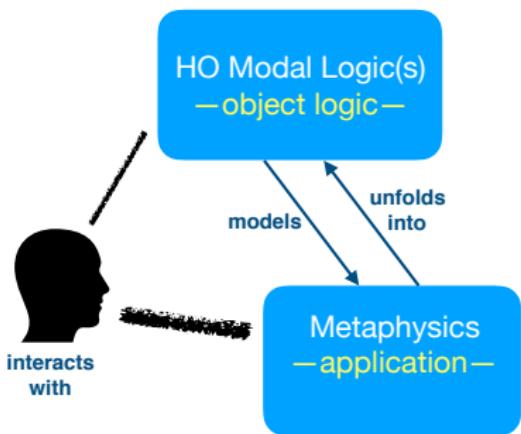


interacts
with

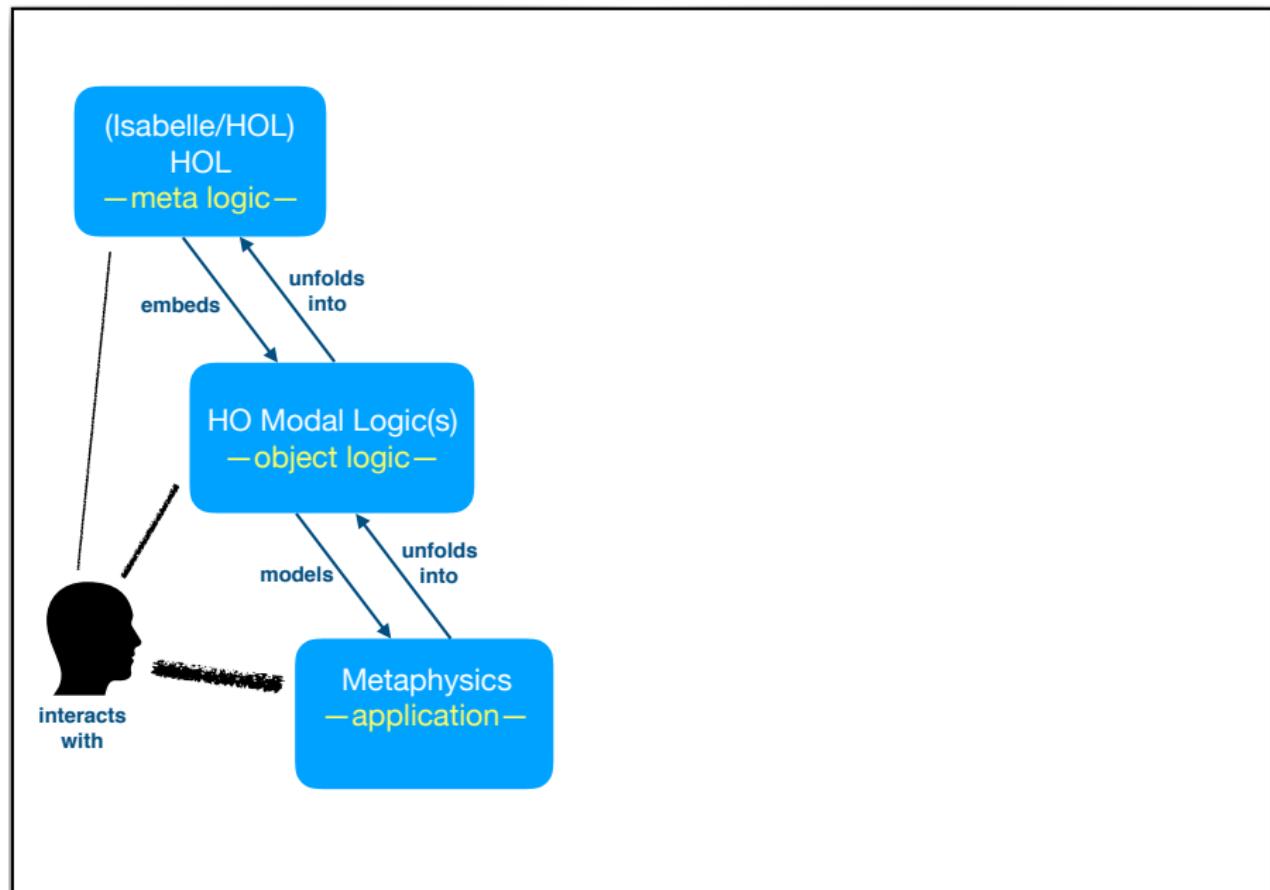
Metaphysics
—application—

A blue rounded rectangular box containing the text "Metaphysics —application—".

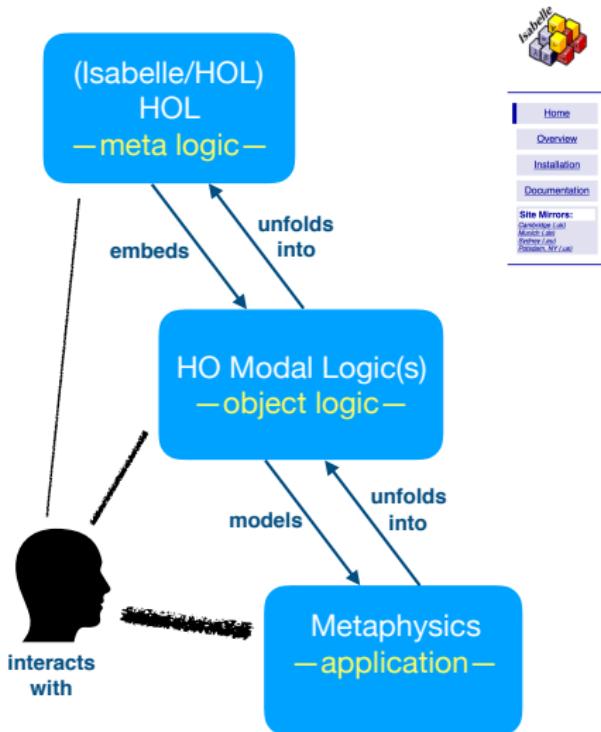
Flexible Reasoning Technology: SSEs in HOL [SCP, 2019], [OpenPhilos., 2019]



Flexible Reasoning Technology: SSEs in HOL [SCP, 2019], [OpenPhilos., 2019]



Flexible Reasoning Technology: SSEs in HOL [SCP, 2019], [OpenPhilos., 2019]



Isabelle

UNIVERSITY OF CAMBRIDGE
Computer Laboratory
TUM
Technische Universität München

What is Isabelle?

Isabelle is a generic proof assistant. It allows mathematical formulas to be expressed in a formal language and provides tools for proving those formulas in a logical calculus. Isabelle was originally developed at the University of Cambridge and Technische Universität München, but now includes numerous contributions from institutions and individuals worldwide. See the [Isabelle overview](#) for a brief introduction.

Now available: Isabelle2017 (October 2017)



[Download for Linux](#) · [Download for Windows \(32bit\)](#) · [Download for Windows \(64bit\)](#) · [Download for Mac OS X](#)

Some notable changes:

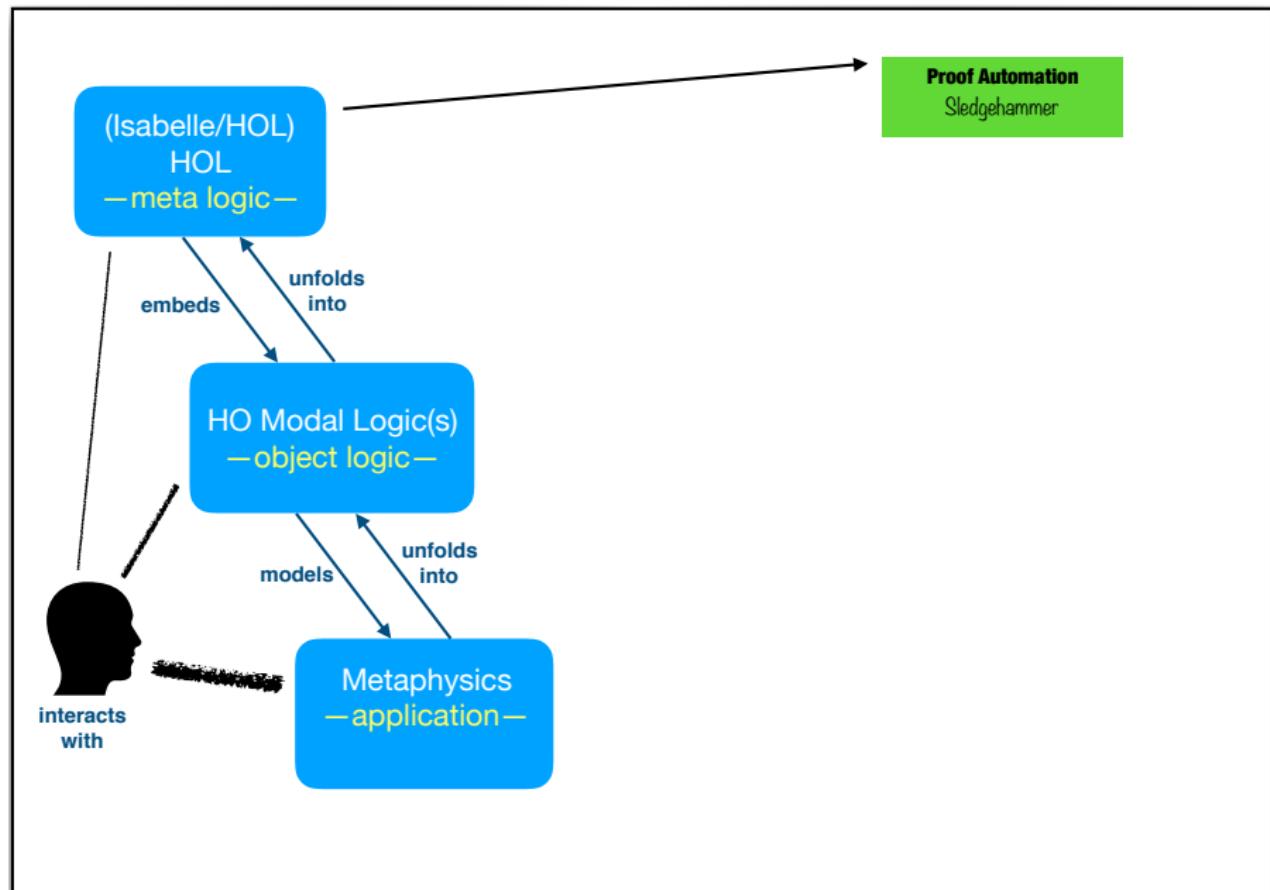
- Experimental support for Visual Studio Code as alternative PIDE front-end.
- Improved Isabelle/Edit Prover IDE: management of session sources independently of editor buffers, removal of unused theories, explicit indication of theory status, more careful auto-indentation.
- Separated theory imports from proofs.
- Code generator improvements: support for statically embedded computations.
- Numerous HOL library improvements.
- More material in HOL-Algebra, HOL-Computational_Algebra and HOL-Analysis (ported from HOL-Light).
- Improved Nunchaku model finder, now in main HOL.
- SQL database support in Isabelle/SQLs.

See also the cumulative [NEWS](#).

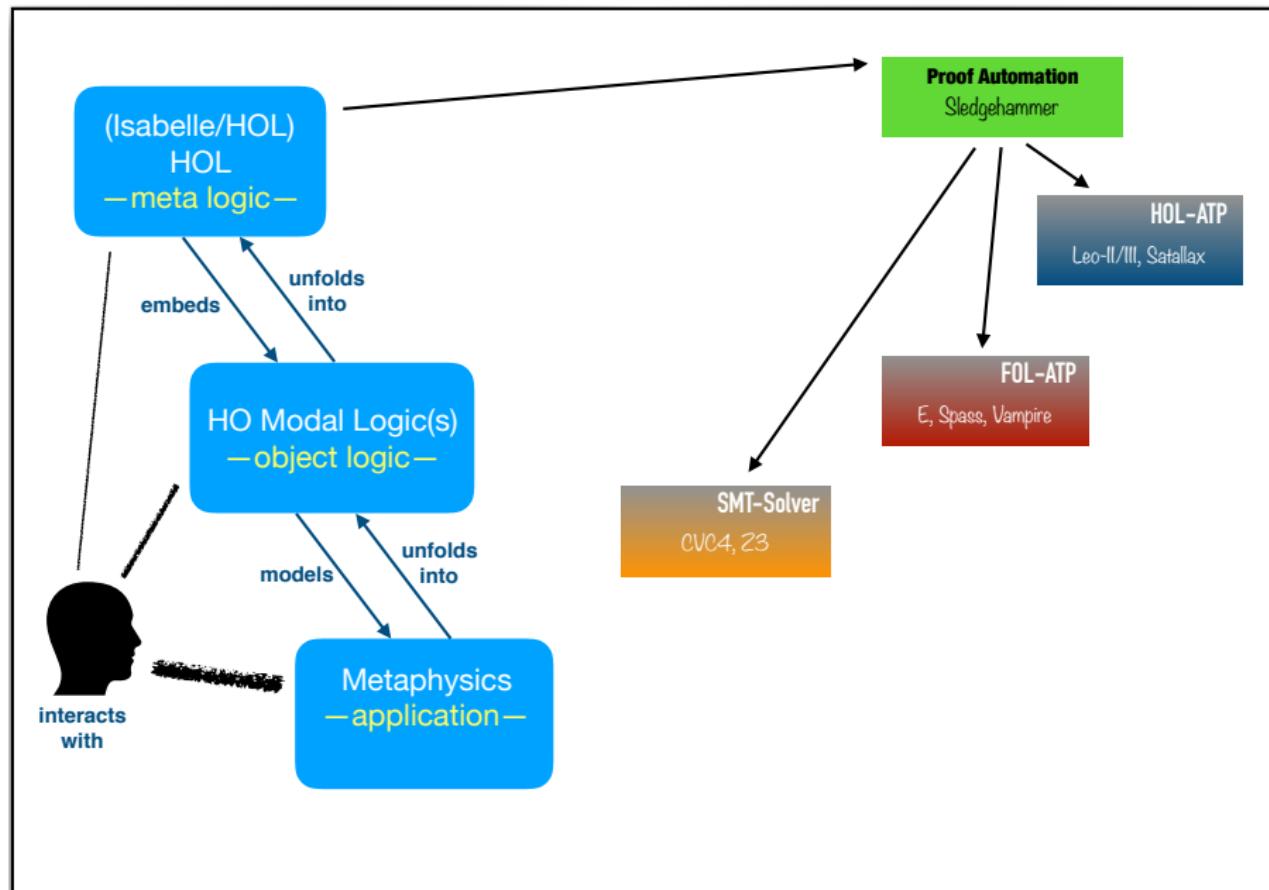
Distribution & Support

Isabelle is distributed for free under a conglomerate of open-source licenses, but the main code-base is subject to BSD-style regulations. The application bundles include source and binary packages and documentation, see the detailed [Installation Instructions](#). A vast collection of Isabelle examples and applications is available from the [Archive of Formal Proofs](#).

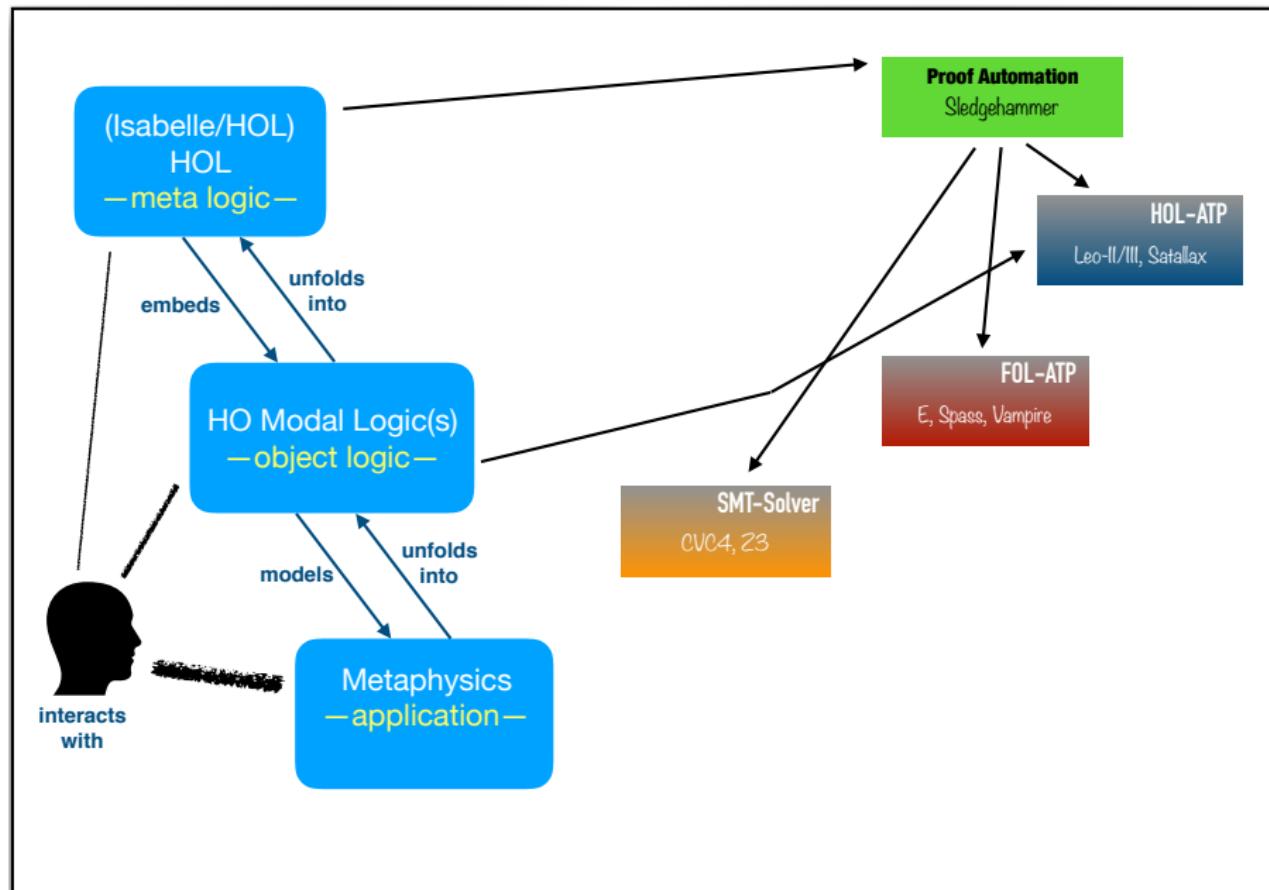
Flexible Reasoning Technology: SSEs in HOL [SCP, 2019], [OpenPhilos., 2019]



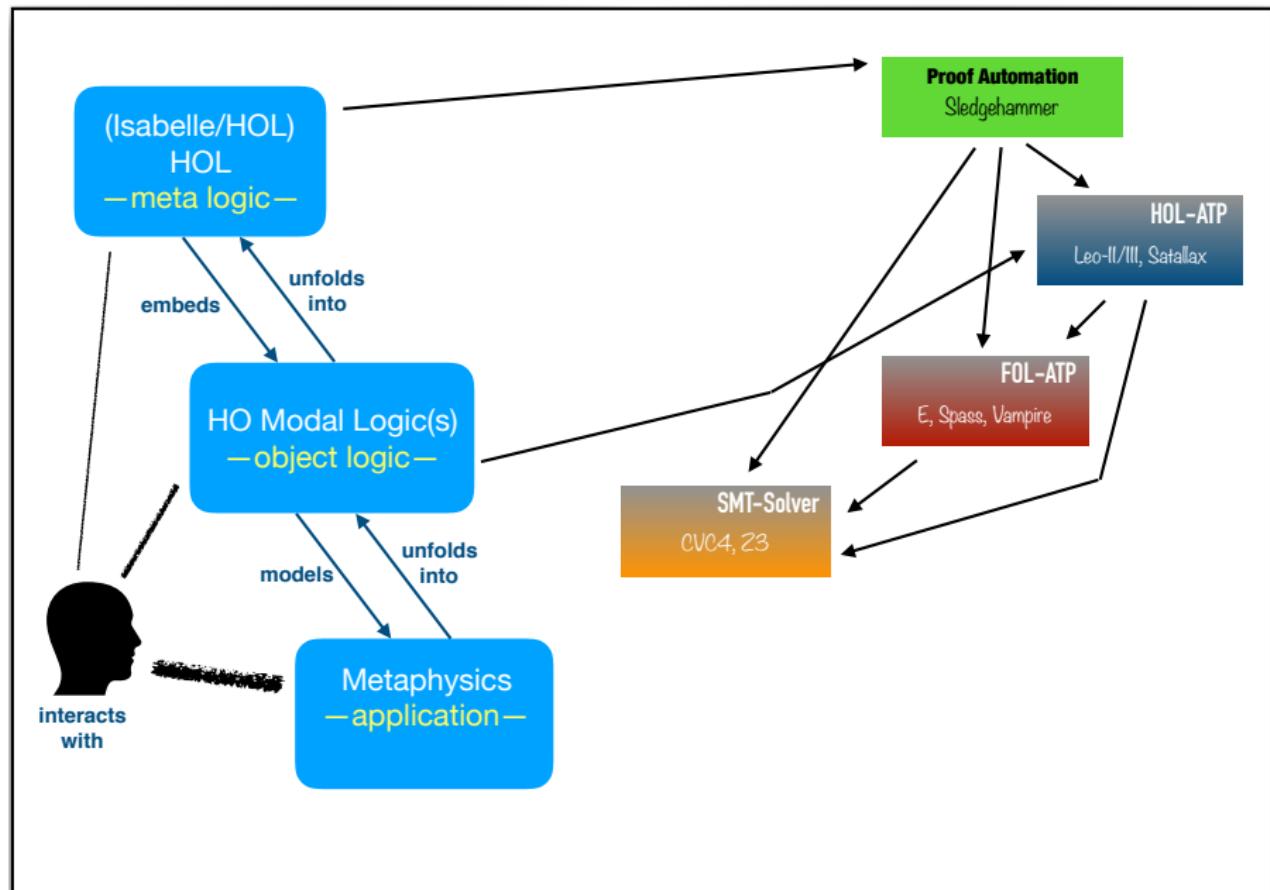
Flexible Reasoning Technology: SSEs in HOL [SCP, 2019], [OpenPhilos., 2019]



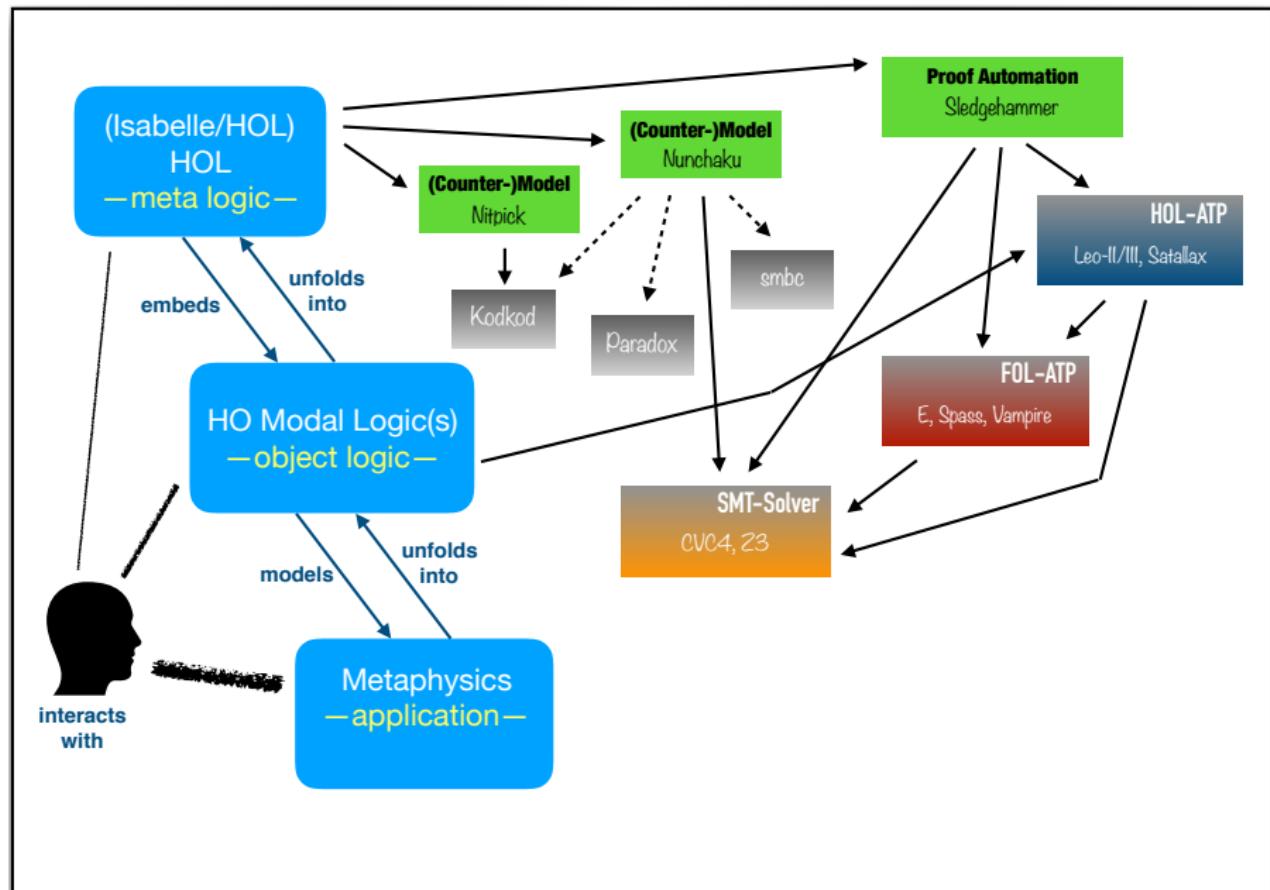
Flexible Reasoning Technology: SSEs in HOL [SCP, 2019], [OpenPhilos., 2019]



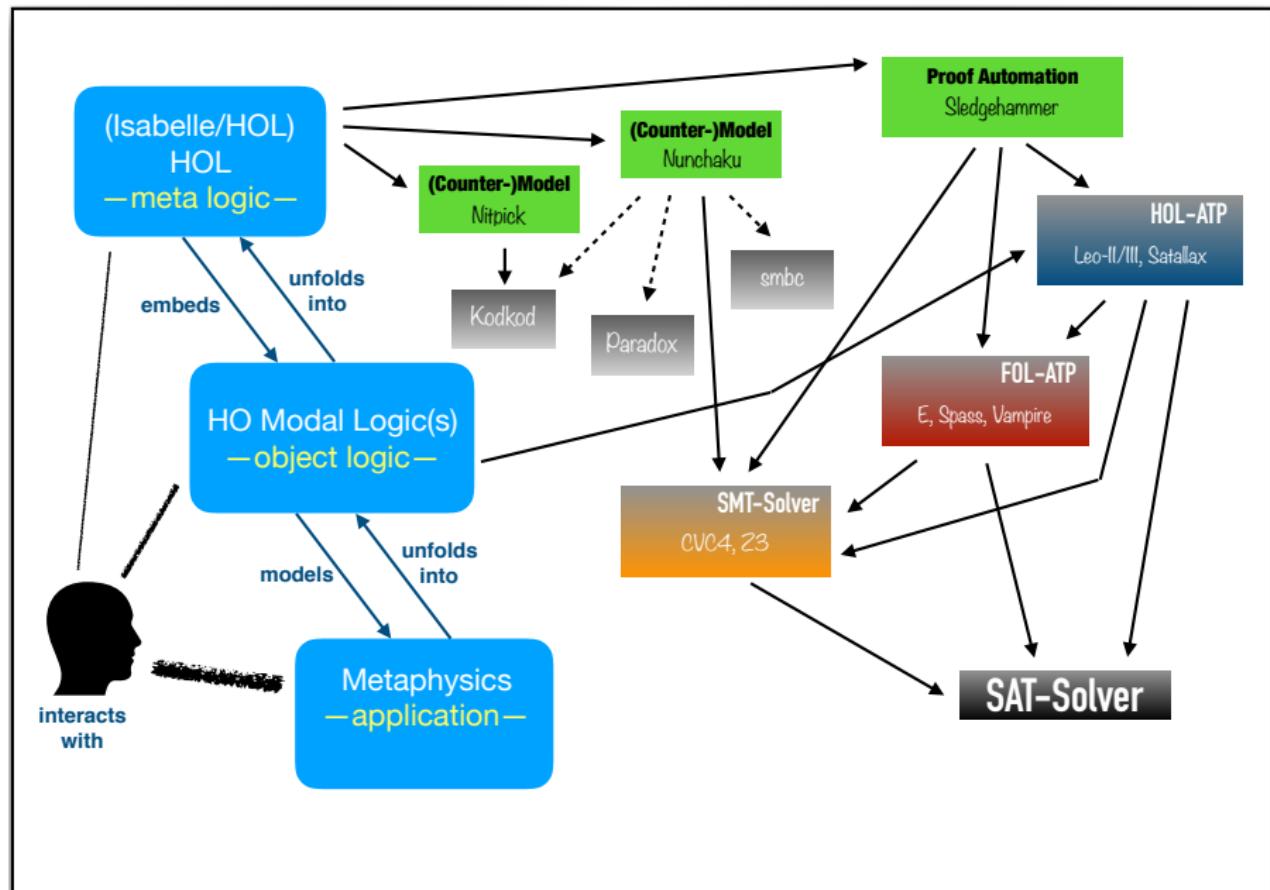
Flexible Reasoning Technology: SSEs in HOL [SCP, 2019], [OpenPhilos., 2019]



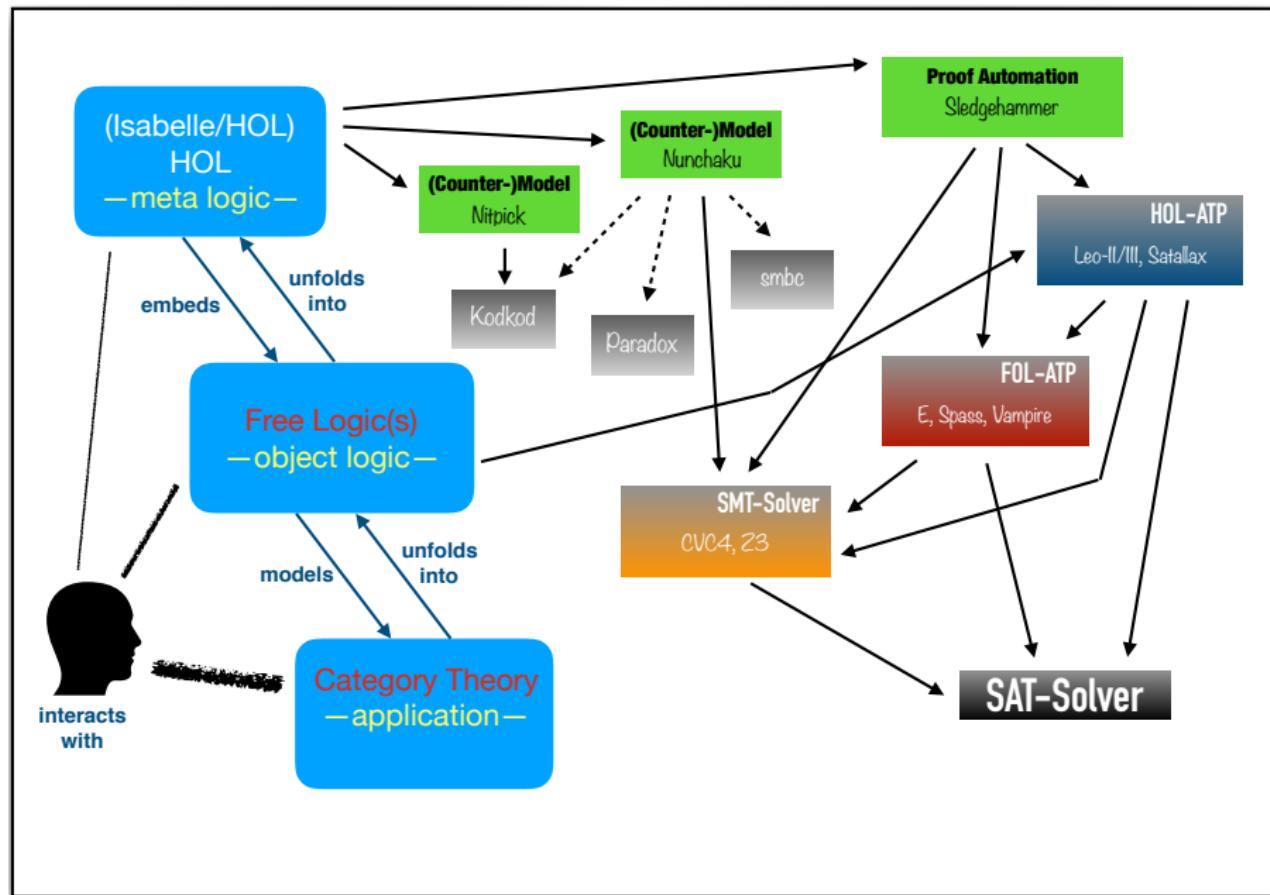
Flexible Reasoning Technology: SSEs in HOL [SCP, 2019], [OpenPhilos., 2019]



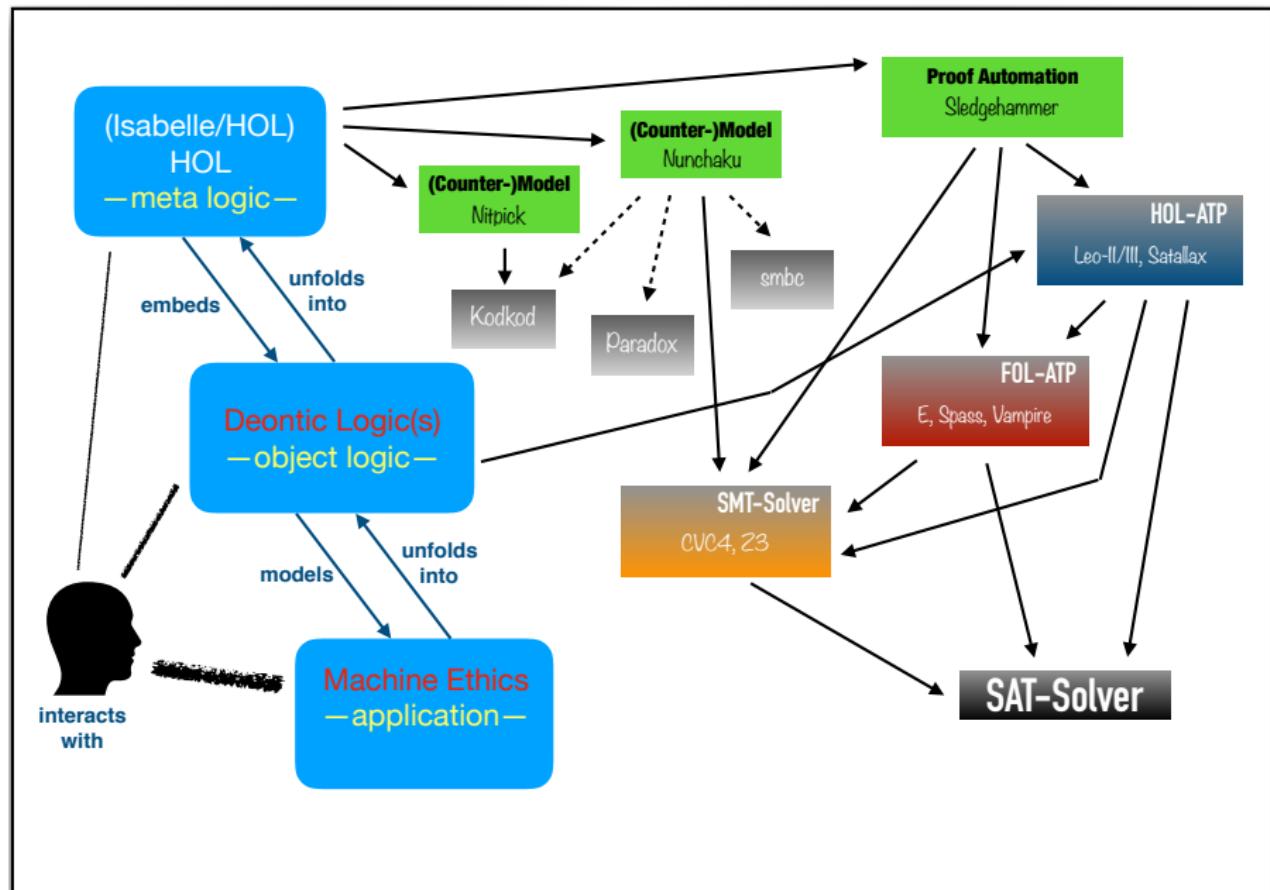
Flexible Reasoning Technology: SSEs in HOL [SCP, 2019], [OpenPhilos., 2019]



Flexible Reasoning Technology: SSEs in HOL [SCP, 2019], [OpenPhilos., 2019]



Flexible Reasoning Technology: SSEs in HOL [SCP, 2019], [OpenPhilos., 2019]



Flexible Reasoning Technology: SSEs in HOL



Ed Zalta (Stanford)

Principia Logico-Metaphysica

Hyperintensional higher-order modal logic

Inconsistency/Paradox detected & fixed

[Open Philosophy, in print] [RSL, in print]



Daniel Kirchner
(Mathematics, FU Berlin)

Flexible Reasoning Technology: SSEs in HOL



Ed Zalta (Stanford)

Principia Logico-Metaphysica

Hyperintensional higher-order modal logic

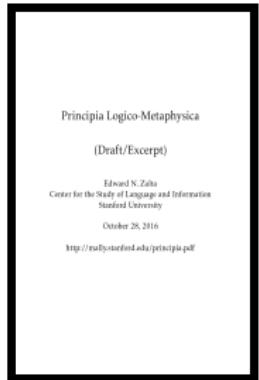
Inconsistency/Paradox detected & fixed

[Open Philosophy, in print] [RSL, in print]



Daniel Kirchner
(Mathematics, FU Berlin)

Flexible Reasoning Technology: SSEs in HOL



Ed Zalta (Stanford)

Principia Logico-Metaphysica

Hyperintensional higher-order modal logic

Inconsistency/Paradox detected & fixed

[Open Philosophy, in print] [RSL, in print]



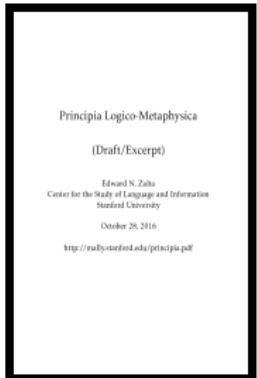
Daniel Kirchner
(Mathematics, FU Berlin)

Kirchner Paradox

Daniel & Isabelle/HOL have become close advisors of
Ed Zalta in the search for a repair

Computational Metaphysics par excellence!!!

Flexible Reasoning Technology: SSEs in HOL



Ed Zalta (Stanford)

Principia Logico-Metaphysica

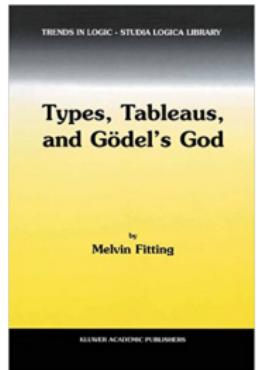
Hyperintensional higher-order modal logic

Inconsistency/Paradox detected & fixed

[Open Philosophy, in print] [RSL, in print]



Daniel Kirchner
(Mathematics, FU Berlin)



M. Fitting (New York)

Philosophy/Metaphysics

Ontological Argument
(avoids modal collapse)

Intensional higher-order modal logic

Verified (main chapters)

[Archive of Formal Proofs, 2017]



David Fuenmayor
(Philosophy, FU Berlin)

Flexible Reasoning Technology: SSEs in HOL



E. J. Lowe's (Durham)

Philosophy/Metaphysics

E. J. Lowe's Modal Ontological Argument

Higher-order modal logic

Verified (minor corrections)

[Journal of Applied Logics, 2018]



David Fuenmayor
(Philosophy, FU Berlin)

Flexible Reasoning Technology: SSEs in HOL



E. J. Lowe's (Durham)

Philosophy/Metaphysics

E. J. Lowe's Modal Ontological Argument

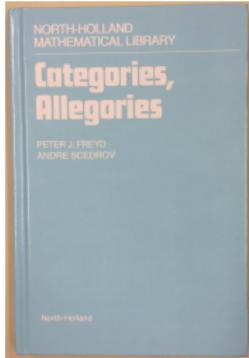
Higher-order modal logic

Verified (minor corrections)

[Journal of Applied Logics, 2018]



David Fuenmayor
(Philosophy, FU Berlin)



P. Freyd & A. Scedrov

Category Theory

Free first-order logic

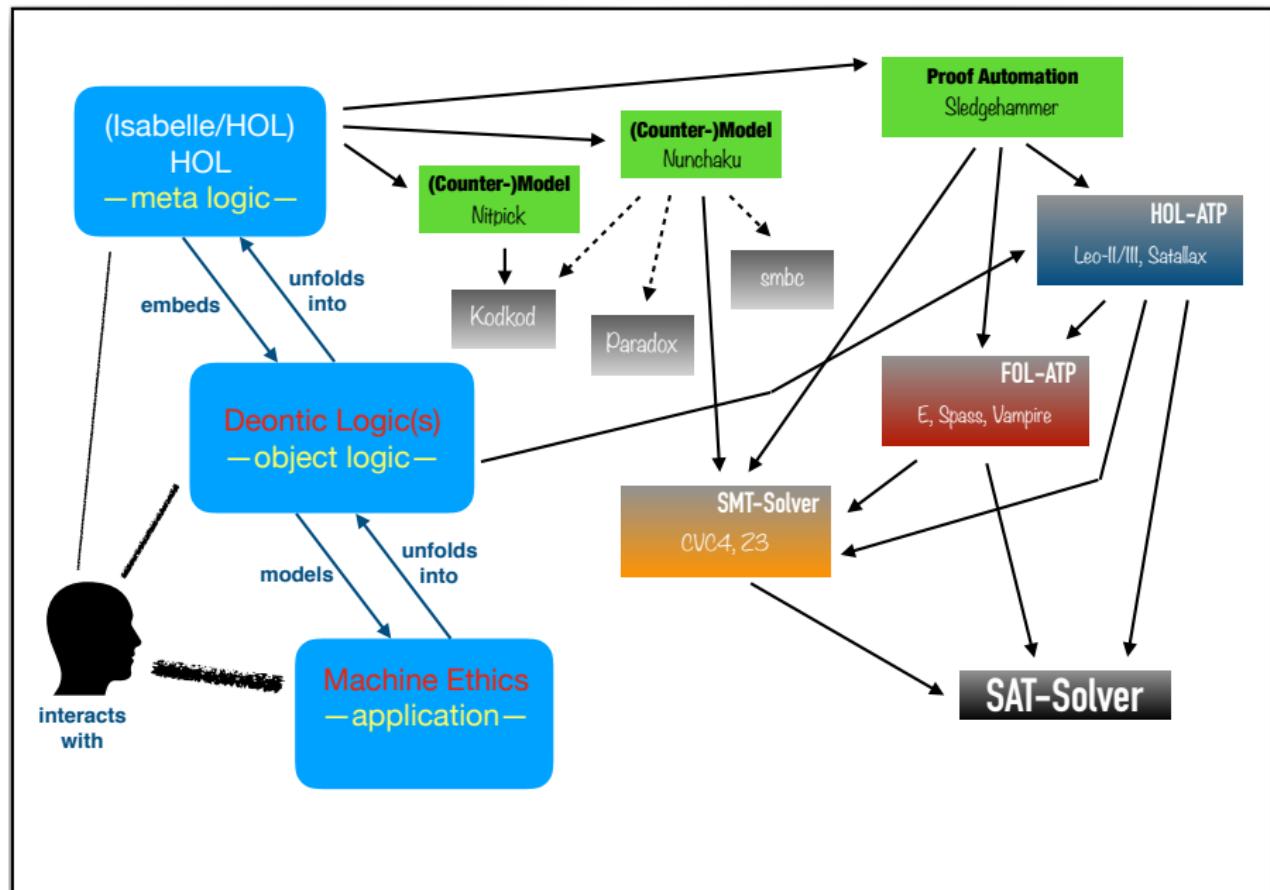
(Constricted) Inconsistency detected & fixed

[JAR, 2019]



D. Scott
(UC Berkeley)

Flexible Reasoning Technology: SSEs in HOL



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 modeling-of/reasoning-with 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 modeling-of/reasoning-with 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 modeling-of/reasoning-with 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 modeling-of/reasoning-with notion of **obligation**
 - ▶ Contrary-to-duty (**CTD**) scenarios



L. van der Torre



X. Parent

Deontic Logic

- ▶ Reasoning about obligations and permissions
- ▶ Two groups of approaches:
 - Possible worlds
 - ▶ standard deontic logic
 - ▶ dyadic deontic logic
 - Norm-based semantics
 - ▶ input/output logic

CTD: no

CTD: yes

CTD: yes

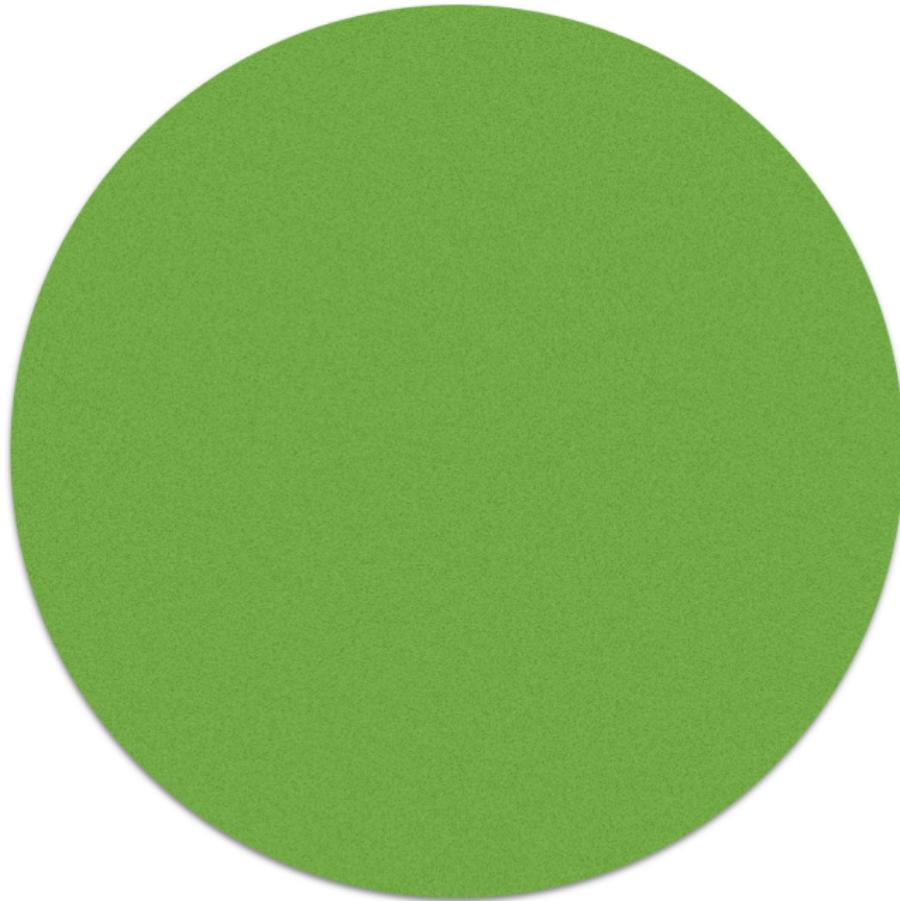


A. Farjam

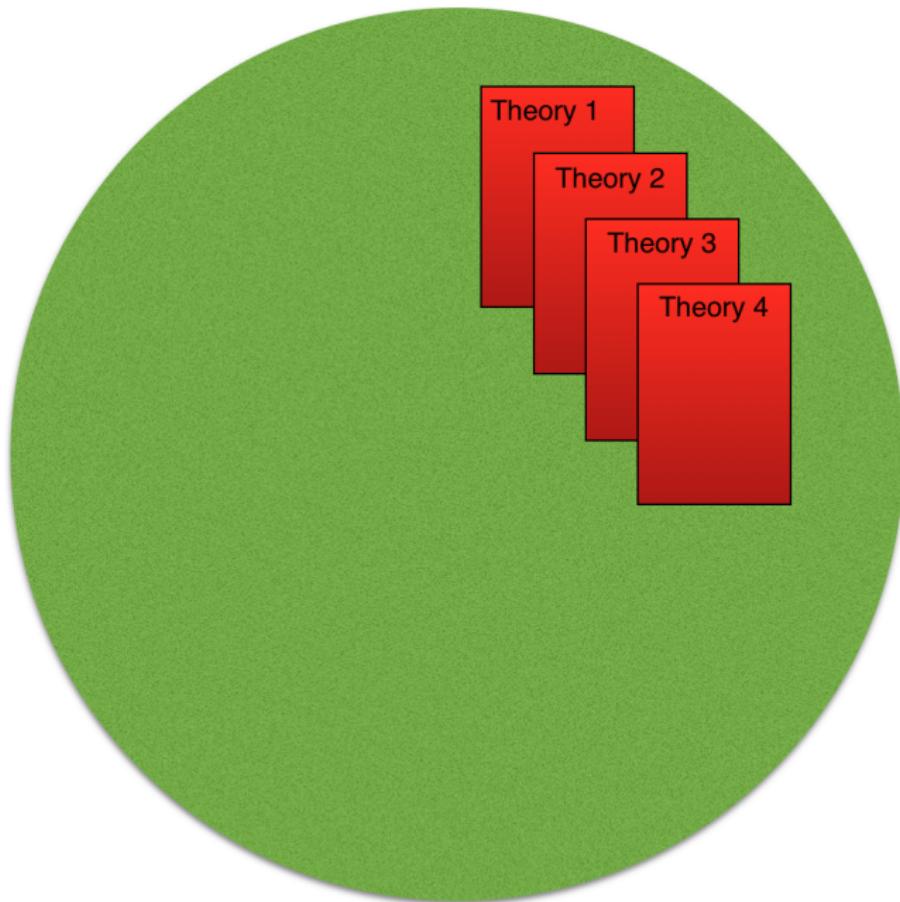
Further interests and challenges

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

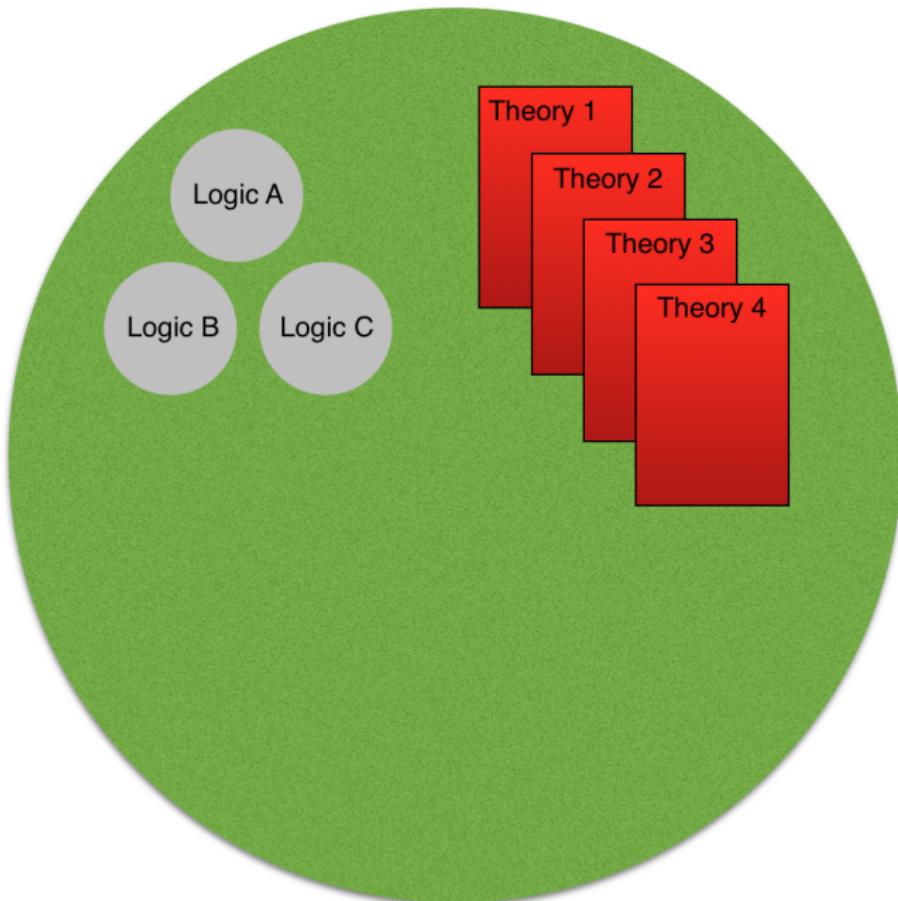
Normative Reasoning Experimentation Platform



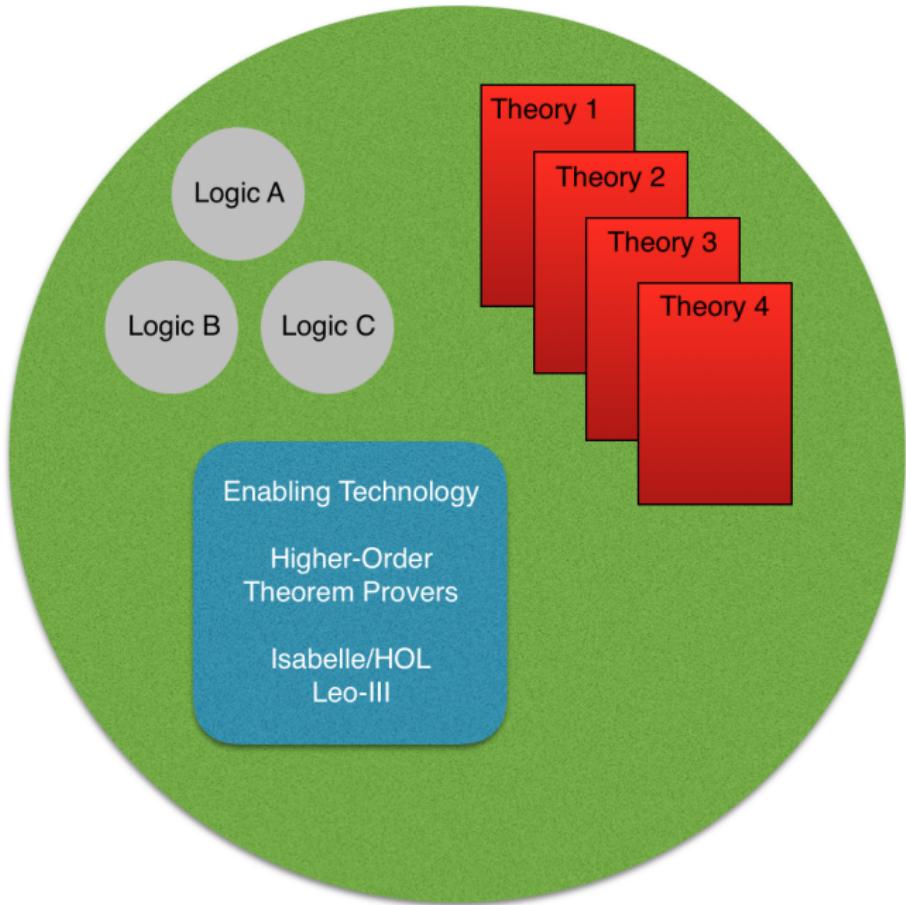
Normative Reasoning Experimentation Platform



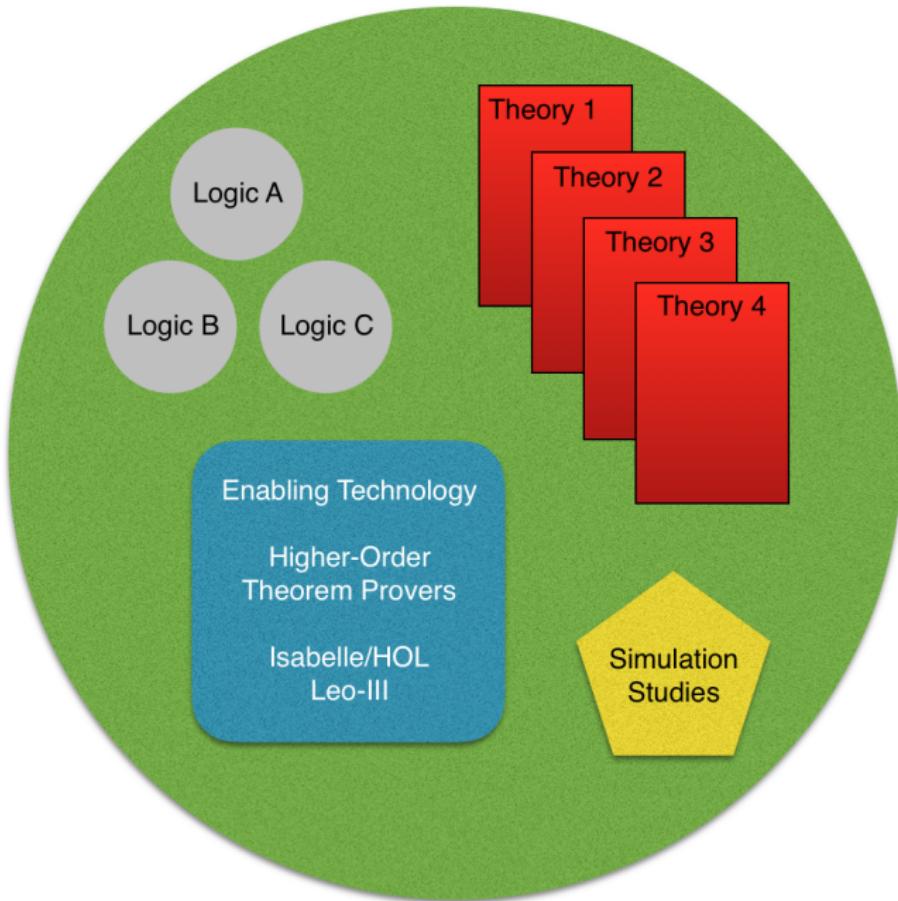
Normative Reasoning Experimentation Platform



Normative Reasoning Experimentation Platform

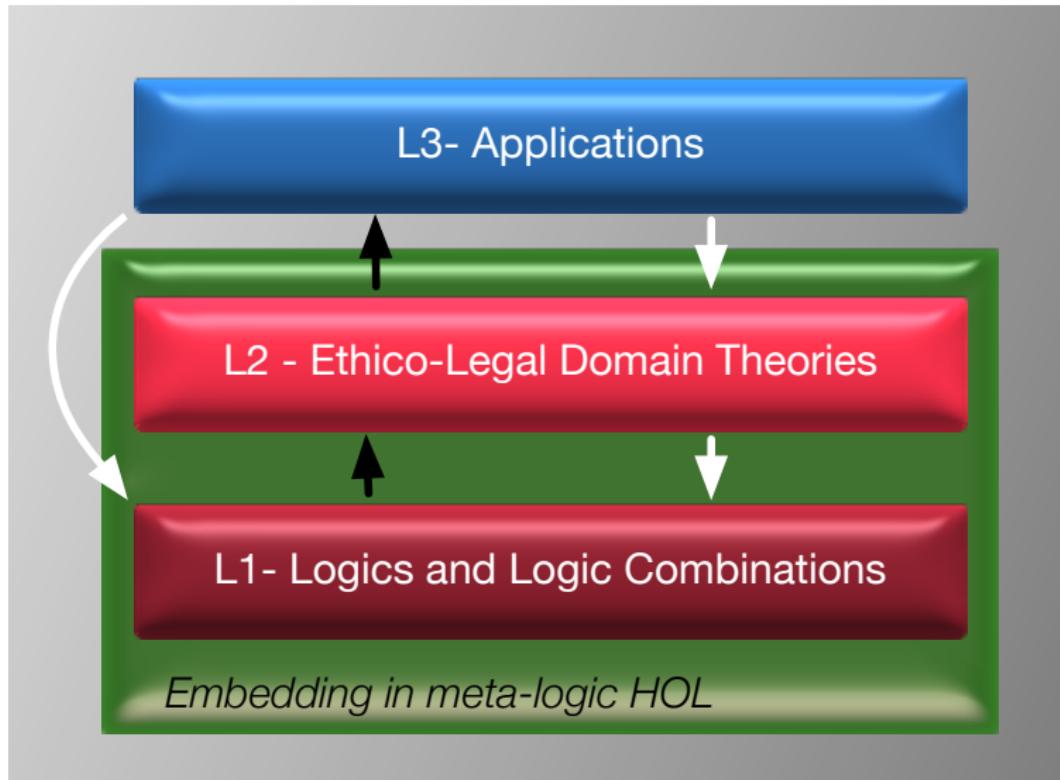


Normative Reasoning Experimentation Platform



LogIKEy Logic and Knowledge Engineering Methodology

[BenzmüllerParentTorre, arXiv:1903.10187, 2019]



Library of Deontic Logics

Base Logics

- ▶ Standard Deontic Logic
- ▶ Daydic Deontic Logic [Carmo& Jones, 2013]
- ▶ Aqvist System E [Aqvist, 2002]
- ▶ Input/Output Logic [Makinson&VanDerTorre, 2002]

Extensions

- ▶ first-order
- ▶ higher-order
- ▶ possibilist quantifiers (constant domain)
- ▶ actualist quantifiers (varying domain)

Combinations with other Logics

- ▶ Kaplan's Logic of Demonstratives (LD)
- ▶ fusion
- ▶ product

Library of Deontic Logics

Base Logics

- ▶ Standard Deontic Logic
- ▶ Daydic Deontic Logic [Carmo& Jones, 2013]
- ▶ Aqvist System E [Aqvist, 2002]
- ▶ Input/Output Logic [Makinson&VanDerTorre, 2002]

Extensions

- ▶ first-order
- ▶ higher-order
- ▶ possibilist quantifiers (constant domain)
- ▶ actualist quantifiers (varying domain)

Combinations with other Logics

- ▶ Kaplan's Logic of Demonstratives (LD)
- ▶ fusion
- ▶ product

Library of Deontic Logics

Base Logics

- ▶ Standard Deontic Logic
- ▶ Daydic Deontic Logic [Carmo& Jones, 2013]
- ▶ Aqvist System E [Aqvist, 2002]
- ▶ Input/Output Logic [Makinson&VanDerTorre, 2002]

Extensions

- ▶ first-order
- ▶ higher-order
- ▶ possibilist quantifiers (constant domain)
- ▶ actualist quantifiers (varying domain)

Combinations with other Logics

- ▶ Kaplan's Logic of Demonstratives (LD)
- ▶ fusion
- ▶ product

LogIKEy Logic and Knowledge Engineering Methodology

Layer L3: Experiments

1. Select
2. Governor Architecture
3. Populate Theory
4. Assess

Layer L2: Ethico-Legal Domain Theories

1. Select Theory
2. Analyse (Abstraction, Notions, Quantifiers, Modalities)
3. Logic/Combination
4. Formalize
5. Explore
6. Contribute

Layer L1: Logic and Logic Combinations

1. Logics
2. Semantics
3. Automate
4. Assess
5. Faithfulness
6. Implications
7. Benchmarks
8. Contribute

Enabling Technology: SSEs in HOL

What's Next? — ENoRMous Plans



What's Next? — ENoRMous Plans



Proposal: AI-Laboratory

- ▶ **Experiments in Normative Reasoning an Machine-Ethics (ENoRME)**
 - Cloud-based Plattform for Universal Normative Reasoning
 - Experiments: Ethico-legal Governance of Autonomous Systems
- ▶ **Transdisciplinary:** KR&R, Formal Ethics/Philosophy, Law, Machine Learning
- ▶ **International Team:** 12 Professors **Industry Participation**

Demo: SDL in Isabelle/HOL

[Logica Universalis, 2013]

```
Isabelle2018/HOL - SDL.thy
1 theory SDL imports Main (* SDL: Standard Deontic Logic. C. Benzmüller & X. Parent, 2019 *)
2 begin
3   typeclass i (*Type for possible worlds.*) type_synonym σ = "(i⇒bool)"
4   type_synonym γ = "σ⇒σ" type_synonym ρ = "σ⇒σ⇒σ"
5   consts R::"i⇒i⇒bool" (infixr "R" 70) (*Accessibility relation.*)
6   constdefs aw::i (*Actual world.*)
7   abbreviation SDLtop::σ ("T") where "T ≡ Aw. True"
8   abbreviation SDLbot::σ ("⊥") where "⊥ ≡ Aw. False"
9   abbreviation SDLnot::γ ("¬" [52] 53) where "¬φ ≡ Aw. ¬φ(w)"
10  abbreviation SDLand::ρ (infixr "∧" 51) where "φ ∧ ψ ≡ Aw. φ(w) ∧ ψ(w)"
11  abbreviation SDLor::ρ (infixr "∨" 50) where "φ ∨ ψ ≡ Aw. φ(w) ∨ ψ(w)"
12  abbreviation SDLimp::ρ (infixr "→" 49) where "φ → ψ ≡ Aw. φ(w) → ψ(w)"
13  abbreviation SDLequ::ρ (infixr "↔" 48) where "φ ↔ ψ ≡ Aw. φ(w) ↔ ψ(w)"
14
15 (*Possibilist Quantification.*)
16  abbreviation mforall ("∀") where "∀Φ ≡ Aw. ∀x. (Φ x w)"
17  abbreviation mforallB (binder "∀" [8] 9) where "∀x. φ(x) ≡ ∀ρ"
18  abbreviation mexists ("∃") where "∃Φ ≡ Aw. ∃x. (Φ x w)"
19  abbreviation mexistsB (binder "∃" [8] 9) where "∃x. φ(x) ≡ ∃ρ"
20
21 (*Obligation*)
22  abbreviation SDLobligatory::γ ("OB") where "OB φ ≡ Aw. ∀v. w R v → φ(v)"
23  abbreviation SDLpermissible::γ ("PE") where "PE φ ≡ ¬(OB(¬φ))"
24
25 (*Validity*)
26  abbreviation SDLvalid::"σ⇒bool" ("[]" [7] 105) (*Global validity.*)
27  where "[A] ≡ Aw. A w"
28  abbreviation SDLvalidcw::"σ⇒bool" ("[_]t" [7] 105) (*Validity in actual world.*)
29  where "[A]t ≡ Aw"
30
31 (*SDL axiom D*)
32  axiomatization where serial_R: "∀x. ∃y. x R y" (*Accessibility relation R is serial*)
33
34 lemma D: "[∀φ. ¬((OB φ) ∧ (OB(¬φ)))]" using serial_R by auto (*Axiom D*)
35 lemma D': "[∀φ.(OB φ) → (PE φ)]" using serial_R by auto (*Axiom D'*)
36
37 (*Notation*)
38  abbreviation SLDobl::γ ("○<_>") where "○<A> ≡ OB A" (*New syntax: A is obligatory.*)
39
40 (*Consistency confirmed by model finder Nitpick.*)
41  lemma True nitpick[satisfy,user_axioms,expect=genuine,show_all,format=2] oops
42
43 (*Barcan Formulas*)
44  lemma Barcan: "[(∀d. ○<φ(d)>) → (○<∀d. φ(d)>)]" by simp
45  lemma ConverseBarcan: "[(○<∀d. φ(d)>) → (∀d. ○<φ(d)>)]" by simp
46 end
```

Demo: DDL in Isabelle/HOL

The screenshot shows the Isabelle/HOL IDE interface with the file `DDL.thy` open. The code defines a theory `DDL` imports `Main`. It includes an axiomatic section for Deontic Logic and various abbreviations for modal operators and obligations. The code is annotated with comments and some parts are highlighted in yellow.

```
theory DDL imports Main
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:::"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)(x). False)" and
  ax_5b: "(∀w. ((Y(w) ∧ X(w)) → (Z(w) ∧ X(w)))) → (ob(X)(Y) → ob(X)(Z))" and
  ax_5c: "((Y. β(Z) → ob(X)(Z)) ∧ (∃Z. β(Z))) →
    (((y. ((Aw. ∀Z. (β Z) → (Z w)) → ob(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))" and
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 ddbox ("□") where "□A ≡ λw. ∀v. A(v)" (*A = (λw. True)*)
abbreviation ddboxa ("□a") where "□a ≡ λw. (Y. av(w)(x) → A(x))" (*in all actual worlds*)
abbreviation ddboxp ("□o") where "□o ≡ λw. (Y. pv(w)(x) → A(x))" (*in all potential worlds*)
abbreviation ddldia ("◊") where "◊A ≡ ▦(¬A)"
abbreviation ddldiaa ("◊a") where "◊a ≡ ▦(¬a)"
abbreviation ddldiap ("◊o") where "◊o ≡ ▦(¬A)"
abbreviation ddlo ("O[_)") [52]53) where "O[B|A] ≡ λw. ob(A)(B)" (*it ought to be w, given φ*)
abbreviation ddloa ("Oa") where "Oa ≡ λw. ▦(ob(av(w))(A) ∧ (∃x. av(w)(x) ∧ ¬A(x)))" (*actual obligation*)
abbreviation ddlop ("Oo") where "Oo ≡ λw. ▦(ob(pv(w))(A) ∧ (∃x. pv(w)(x) ∧ ¬A(x)))" (*primary obligation*)
abbreviation ddltop ("T") where "T ≡ λw. True"
abbreviation ddbot ("⊥") where "⊥ ≡ λw. False"
abbreviation ddvalid:::"σ ⇒ bool" ("[_]" [7]105) where "[A] ≡ λw. A w" (*Global validity*)
abbreviation ddlidcw:::"σ ⇒ bool" ("[_]cw" [7]105) where "[A]cw ≡ A cw" (*Local validity (in cw)*)
(* A is obligatory *)
abbreviation obligatoryDDL:::"σ ⇒ σ" ("O[_)") where "O(A) ≡ O(A|T)"
(* Consistency *)
lemma True nitpick [satisfy] oops
```

Output Query Sledgehammer Symbols

Demo: Normative Reasoning Experimentation Platform

The screenshot shows a software interface for normative reasoning. At the top, there's a toolbar with various icons. Below it is a menu bar with "File", "Edit", "Search", "Help", and a "GDPR.thy" tab. The main area contains the following code:

```
1 theory GDPR imports SDL          (* Christoph Benzmueller & Xavier Parent, 2018 *)
2
3 begin (** GDPR Example **)
4 consts process_data_lawfully::σ erase_data::σ kill_boss::σ
5
6 axiomatization where
7   (* It is an obligation to process data lawfully. *)
8   A1: " $\text{process\_data\_lawfully}$ " and
9   (* Implicit: It is an obligation to keep the data if it was processed lawfully. *)
10  Implicit: " $\text{process\_data\_lawfully} \rightarrow \text{erase\_data}$ " and
11  (* If data was not processed lawfully, then it is an obligation to erase the data. *)
12  A2: " $\neg \text{process\_data\_lawfully} \rightarrow \text{erase\_data}$ " and
13  (* Given a situation where data is processed unlawfully. *)
14  A3: " $\neg \text{process\_data\_lawfully} \vee \text{erase\_data}$ "
15
16 (** Some Experiments **)
17 lemma True nitpick [satisfy] oops (* Consistency-check: Is there a model? *)
18 lemma False sledgehammer oops (* Inconsistency-check: Can Falsum be derived? *)
19
20 lemma " $\text{erase\_data}$ " sledgehammer nitpick oops (* Should the data be erased? *)
21 lemma " $\neg \text{erase\_data}$ " sledgehammer nitpick oops (* Should the data be kept? *)
22 lemma " $\text{kill\_boss}$ " sledgehammer nitpick oops (* Should the boss be killed? *)
23 end
```

On the right side, there are tabs for "Documentation", "Sidekick", "State", and "Theories". Below the code editor, there's a status bar with checkboxes for "Proof state" and "Auto update", an "Update" button, a "Search" input field, and a zoom level of "100%". The bottom part of the interface shows a log window with the following text:

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)

At the very bottom, there are tabs for "Output", "Query", "Sledgehammer", and "Symbols".

Demo: Normative Reasoning Experimentation Platform

The screenshot shows a software interface for normative reasoning. At the top, there's a toolbar with various icons. Below it is a menu bar with "File", "Edit", "Search", "Help", and a "GDPR.thy" tab. The main area contains a code editor with the following content:

```
1 theory GDPR imports SDL          (* Christoph Benzmueller & Xavier Parent, 2018 *)
2
3 begin (** GDPR Example **)
4 consts process_data_lawfully::σ erase_data::σ kill_boss::σ
5
6 axiomatization where
7   (* It is an obligation to process data lawfully. *)
8   A1: "[0(process_data_lawfully)]" and
9   (* Implicit: It is an obligation to keep the data if it was processed lawfully. *)
10  Implicit: "[0(process_data_lawfully → ~erase_data)]" and
11  (* If data was not processed lawfully, then it is an obligation to erase the data. *)
12  A2: "[~process_data_lawfully → 0(erase_data)]"
13  (* Given a situation where data is processed unlawfully. *) and
14  A3: "[~process_data_lawfully]_cw"
15
16 (*
17 l
18 l
19 l
20 l
21 lemma "[0(~erase_data)]" sledgehammer nitpick oops (* Should the data be kept? *)
22 lemma "[0(kill_boss)]" sledgehammer nitpick oops (* Should the boss be killed? *)
23 end
```

A large orange rectangular box covers several lines of code, containing the text:

Danger Zone:
Paradoxes and Inconsistencies!

At the bottom of the interface, there are several buttons: "Proof state" (checked), "Auto update" (checked), "Update", "Search:" (with a dropdown menu), and a zoom slider set to 100%. Below these buttons, a status message says "Sledgehammering... Proof found...". Further down, there are error messages from different provers:

- "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)

At the very bottom, there are tabs for "Output", "Query", "Sledgehammer", and "Symbols".

Demo: Global vs. Local Consequence Relation

The screenshot shows the HOL4 Prover interface with the theory file `GDPRGlobal.thy` open. The code defines a theory `GDPRGlobal` imports DDL, with obligations to process data lawfully, erase data, and kill the boss. It includes experiments using `nitpick` and `sledgehammer` to check consistency and derive `False`. The interface includes a toolbar, a navigation bar, and a status bar at the bottom.

```
1 theory GDPRGlobal imports DDL      (* Christoph Benzmueller & Xavier Parent, 2018 *)
2
3 begin (** GDPR Example **)
4 consts process_data_lawfully::σ erase_data::σ kill_boss::σ
5
6 axiomatization where
7   (* It is an obligation to process data lawfully. *)
8   A1: "[0(process_data_lawfully)]"
9   (* Given a situation where data is processed unlawfully. *) and
10  A3: "[~process_data_lawfully]"
11
12 (** Some Experiments **)
13 lemma True nitpick [satisfy] nunchaku [satisfy] oops (* Consistency-check: Is there a model? *)
14 lemma False sledgehammer oops (* Inconsistency-check: Can Falsum be derived? *)
15
16 lemma "[0(erase_data)]" sledgehammer nitpick oops (* Should the data be erased? *)
17 lemma "[0(~erase_data)]" sledgehammer nitpick oops (* Should the data be kept? *)
18 lemma "[0(kill_boss)]" sledgehammer nitpick oops (* Should the boss be killed? *)
19 end
20
21
22
23
```

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: Normative Reasoning Experimentation Platform

The screenshot shows a software interface for normative reasoning. At the top, there's a toolbar with various icons. Below it is a menu bar with "File", "Edit", "View", "Tools", "Help", and a "GDPR.thy" tab. The main area contains the following code:

```
1 theory GDPR imports SDL (* Christoph Benzmueller & Xavier Parent, 2018 *)
2
3 begin (** GDPR Example **)
4 consts process_data_lawfully::σ erase_data::σ kill_boss::σ
5
6 axiomatization where
7   (* It is an obligation to process data lawfully. *)
8   A1: " $\text{process\_data\_lawfully} \rightarrow \text{erase\_data}$ " and
9   (* Implicit: It is an obligation to keep the data if it was processed lawfully. *)
10  Implicit: " $\text{process\_data\_lawfully} \rightarrow \neg \text{erase\_data}$ " and
11  (* If data was not processed lawfully, then it is an obligation to erase the data. *)
12  A2: " $\neg \text{process\_data\_lawfully} \rightarrow \text{erase\_data}$ " and
13  (* Given a situation where data is processed unlawfully. *)
14  A3: " $\neg \text{process\_data\_lawfully} \vee \text{erase\_data}$ "
15
16 (** Some Experiments **)
17 lemma True nitpick [satisfy] oops (* Consistency-check: Is there a model? *)
18 lemma False sledgehammer oops (* Inconsistency-check: Can Falsum be derived? *)
19
20 lemma " $\text{erase\_data}$ " sledgehammer nitpick oops (* Should the data be erased? *)
21 lemma " $\neg \text{erase\_data}$ " sledgehammer nitpick oops (* Should the data be kept? *)
22 lemma " $\text{kill\_boss}$ " sledgehammer nitpick oops (* Should the boss be killed? *)
23 end
```

To the right of the code editor, there's a vertical sidebar with tabs for "Documentation", "Sidekick", "State", and "Theories". The "State" tab is currently selected.

At the bottom, there's a status bar with checkboxes for "Proof state" and "Auto update", a "Update" button, a "Search" input field, and a zoom level of "100%". Below the status bar, a message window displays:

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)

At the very bottom, there's a navigation bar with tabs for "Output", "Query", "Sledgehammer", and "Symbols".

Demo: Normative Reasoning Experimentation Platform

The screenshot shows a software interface for normative reasoning. At the top, there's a toolbar with various icons. Below it is a menu bar with 'File', 'Edit', 'View', 'Tools', 'Help', and a 'GDPR.thy' tab. The main area contains a code editor with the following content:

```
1 theory GDPR imports SDL          (* Christoph Benzmueller & Xavier Parent, 2018 *)
2
3 begin (** GDPR Example **)
4 consts process_data_lawfully::σ erase_data::σ kill_boss::σ
5
6 axiomatization where
7   (* It is an obligation to process data lawfully. *)
8   A1: "[0(process_data_lawfully)]" and
9   (* Implicit: It is an obligation to keep the data if it was processed lawfully. *)
10  Implicit: "[0(process_data_lawfully → ~erase_data)]" and
11  (* If data was not processed lawfully, then it is an obligation to erase the data. *)
12  A2: "[~process_data_lawfully → 0(erase_data)]"
13  (* Given a situation where data is processed unlawfully. *) and
14  A3: "[~process_data_lawfully]_cw"
15
16 (*
17 l
18 l
19 l
20 l
21 lemma "[0(~erase_data)]" sledgehammer nitpick oops (* Should the data be kept? *)
22 lemma "[0(kill_boss)]" sledgehammer nitpick oops (* Should the boss be killed? *)
23 end
```

A large orange rectangular box with a black border is overlaid on the code editor, containing the text:

Danger Zone:
Paradoxes and Inconsistencies!

At the bottom of the interface, there are several buttons: 'Proof state' (checked), 'Auto update' (checked), 'Update', 'Search:' (with a dropdown menu), and a zoom slider set to 100%. Below these buttons, a message says "Sledgehammering...". Underneath, it says "Proof found..." followed by a detailed log of the proof search:

"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)

At the very bottom, there are tabs for 'Output', 'Query', 'Sledgehammer', and 'Symbols'.

"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, 1978)

- ▶ Gewirth's PGC has
 - ▶ stirred much controversy in moral philosophy
 - ▶ been discussed as means to bound the impact of artificial general intellig. (AGI)
- ▶ Idea (in a nutshell):
 - ▶ emendation of the Golden Rule (treat others as you would wish to be treated)
 - ▶ adopting an agent perspective, the PGC expresses, and deductively justifies, a related upper moral principle, according to which any intelligent agent, by virtue of its self-understanding as an agent, is rationally committed to asserting that:
 - (i) it has rights to freedom and well-being, and
 - (ii) all other agents have those same rights.
- ▶ References
 - ▶ A. Gewirth. Reason and morality. U of Chicago Press, 1978.
 - ▶ D. Beyleveld. The dialectical necessity of morality: An analysis and defense of Alan Gewirth's argument to the principle of generic consistency. UCP 1991.
 - ▶ A. Kornai. Bounding the impact of AGI. J. Experimental & Theoretical AI, 2014.

"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**, 1978)

► **Gewirth's PGC has**

- ▶ stirred much controversy in moral philosophy
- ▶ been discussed as means to bound the impact of artificial general intellig. (AGI)

► **Idea (in a nutshell):**

- ▶ emendation of the Golden Rule (treat others as you would wish to be treated)
- ▶ adopting an agent perspective, the PGC expresses, and deductively justifies, a related upper moral principle, according to which any intelligent agent, by virtue of its self-understanding as an agent, is rationally committed to asserting that:
 - (i) it has rights to freedom and well-being, and
 - (ii) all other agents have those same rights.

► **References**

- ▶ A. Gewirth. *Reason and morality*. U of Chicago Press, 1978.
- ▶ D. Beyleveld. The dialectical necessity of morality: An analysis and defense of Alan Gewirth's argument to the principle of generic consistency. UCP 1991.
- ▶ A. Kornai. Bounding the impact of AGI. *J. Experimental & Theoretical AI*, 2014.

"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**, 1978)

- ▶ **Gewirth's PGC has**
 - ▶ stirred much controversy in moral philosophy
 - ▶ been discussed as means to bound the impact of artificial general intellig. (AGI)
- ▶ **Idea (in a nutshell):**
 - ▶ emendation of the Golden Rule (treat others as you would wish to be treated)
 - ▶ adopting an agent perspective, the PGC expresses, and deductively justifies, a related upper moral principle, according to which any intelligent agent, by virtue of its self-understanding as an agent, is rationally committed to asserting that:
 - (i) it has rights to freedom and well-being, and
 - (ii) all other agents have those same rights.
- ▶ **References**
 - ▶ A. Gewirth. Reason and morality. U of Chicago Press, 1978.
 - ▶ D. Beyleveld. The dialectical necessity of morality: An analysis and defense of Alan Gewirth's argument to the principle of generic consistency. UCP 1991.
 - ▶ A. Kornai. Bounding the impact of AGI. J. Experimental & Theoretical AI, 2014.

"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**, 1978)

- ▶ **Gewirth's PGC has**
 - ▶ stirred much controversy in moral philosophy
 - ▶ been discussed as means to bound the impact of artificial general intellig. (AGI)
- ▶ **Idea (in a nutshell):**
 - ▶ emendation of the Golden Rule (treat others as you would wish to be treated)
 - ▶ adopting an agent perspective, the PGC expresses, and deductively justifies, a related upper moral principle, according to which any intelligent agent, by virtue of its self-understanding as an agent, is rationally committed to asserting that:
 - (i) it has rights to freedom and well-being, and
 - (ii) all other agents have those same rights.
- ▶ **References**
 - ▶ A. Gewirth. Reason and morality. U of Chicago Press, 1978.
 - ▶ D. Beyleveld. The dialectical necessity of morality: An analysis and defense of Alan Gewirth's argument to the principle of generic consistency. UCP 1991.
 - ▶ A. Kornai. Bounding the impact of AGI. J. Experimental & Theoretical AI, 2014.

The idea is to constrain potential AGI's to reason in the following way

- ▶ It is necessary for me (as an AGI) to accept that:

- (P1) I act voluntarily on purpose E (equivalent by definition 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

Gewirth3.thy

```

15 (** C9: "Every PPA has a necessary right to their freedom and well-being")
16 theorem C9: "[ $\forall a. \text{PPA } a \rightarrow \Box_{\text{FWB}} \text{RightTo } a \text{ FWB}]^a"
17 proof -
18 {
19   fix I {
20     fix E {
21       (** Stage I *)
22       assume P1: "[ActsOnPurpose I E]^a" (*I act voluntarily on purpose E*)
23       from P1 have P1_var: "[PPA I]^a" by auto (*definition of PPA*)
24       from P1 have C2: "[Good I E]^a" using explicationGoodness1 by blast (*E is good for me (I)*)
25       hence C4: "[\Box_{\text{FWB}} \text{Good I (FWB I)}]^a" using explicationGoodness2 P3 by blast (*My F_WB are necessary goods*)
26       (** Stage II *)
27       hence C4a: "[\Box_{\text{FWB}} I \mid \Box_{\text{FWB}} \text{Good I (FWB I)}]^a" using explicationGoodness3 explicationFWB1 by blast
28       hence C4b: "[\Box_{\text{FWB}} I]^a" using explicationFWB1 explicationFWB2 C4 CJ_14p by blast
29       hence C4c: "[\Box_{\text{FWB}} (\Box_{\text{FWB}} I)]^a" using OIOAC by auto
30       hence C5a: "[\Box_{\text{FWB}} (\forall a. \neg \text{InterferesWith } a \text{ (FWB I)})]^a" using explicationInterference2 by auto
31       hence C5: "[\Box_{\text{FWB}} \text{RightTo I FWB}]^a" by simp (*I have a claim right to my freedom and well-being*)
32       hence C5_var: "[\Box_{\text{FWB}} \text{RightTo I FWB}]^a" by simp
33     }
34     (** Stage IIIa *)
35     hence C6: "[\Box_{\text{FWB}} \text{RightTo I FWB}]^a" by (rule impI)
36   }
37   hence C7: "[\forall P. \Box_{\text{FWB}} \text{RightTo I FWB}]^a" by (rule allI)
38 }
39 hence C8: "[\forall a. \forall P. \Box_{\text{FWB}} \text{RightTo a P} \rightarrow \Box_{\text{FWB}} \text{RightTo a FWB}]^a" by (rule allI)
40 hence C9_var: "[\forall a. \text{PPA } a \rightarrow \Box_{\text{FWB}} \text{RightTo a FWB}]^a"
41   by simp (*Every PPA has a necessary right to their freedom and well-being*)
42 thus ?thesis by simp
43 qed$ 
```

Proof state Auto update Update Search: 100%

proof (prove)
goal (1 subgoal):
1. ($\lambda x. [\text{PPA } x]^a \sqsubseteq (\lambda x. \text{pv aw} \sqsubseteq \text{O}_1(\lambda w. \forall x. (\neg \text{InterferesWith } xa \text{ (FWB } x)) w))$)

Output Query Sledgehammer Symbols

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

- ▶ challenging: **alethic & deontic modalities, quantification and indexicals**
- ▶ uninterpreted predicate/relation symbols: FWB, Good (1-ary);
ActsOnPurpose, InterferesWith (both 2-ary); NeedsForPurpose (3-ary)
- ▶ PPA $a := \exists E. \text{ActsOnPurpose } a E$ (prospective purposive agents)
an additional axiom postulates that being a PPA is identity-constitutive for
any individual: $[\forall a. \text{PPA}a \rightarrow \square^D(\text{PPA}a)]^D$.
- ▶ $[\varphi]^D$ is indexical validity of φ : true in all contexts (following Kaplan's LD)
- ▶ RightTo $a \varphi := \mathbf{O}_i(\forall b. \neg \text{InterferesWith } b (\varphi a))$
 a has (claim) right to property φ iff it is obligatory that every b does not
interfere with state of affairs (φa)
- ▶ \mathbf{O}_i is *ideal* obligation operator from Carmo & Jones' DDL (\mathbf{O}_a also fine)
- ▶ further axioms interrelate the concepts of goodness and agency
- ▶ axiom $[\forall P. \forall a. \text{NeedsForPurpose } a \text{ FWB } P]^D$ expresses that FWB is always
required in order to act on any purpose
- ▶ FWB is postulated a contingent property: $[\forall a. \diamond_p \text{ FWB } a \wedge \diamond_p \neg \text{FWB } a]^D$
- ▶ Note that both **first-order** and **higher-order quantifiers** are required

- ▶ challenging: **alethic & deontic modalities, quantification and indexicals**
- ▶ uninterpreted predicate/relation symbols: FWB, Good (1-ary);
`ActsOnPurpose`, `InterferesWith` (both 2-ary); `NeedsForPurpose` (3-ary)
- ▶ $\text{PPA } a := \exists E. \text{ActsOnPurpose } a E$ (prospective purposive agents)
an additional axiom postulates that being a PPA is identity-constitutive for any individual: $[\forall a. \text{PPA } a \rightarrow \square^D(\text{PPA } a)]^D$.
- ▶ $[\varphi]^D$ is indexical validity of φ : true in all contexts (following Kaplan's LD)
- ▶ $\text{RightTo } a \varphi := \mathbf{O}_i(\forall b. \neg \text{InterferesWith } b (\varphi a))$
 a has (claim) right to property φ iff it is obligatory that every b does not interfere with state of affairs (φa)
- ▶ \mathbf{O}_i is *ideal* obligation operator from Carmo & Jones' DDL (\mathbf{O}_a also fine)
- ▶ further axioms interrelate the concepts of goodness and agency
- ▶ axiom $[\forall P. \forall a. \text{NeedsForPurpose } a \text{ FWB } P]^D$ expresses that FWB is always required in order to act on any purpose
- ▶ FWB is postulated a contingent property: $[\forall a. \diamond_p \text{ FWB } a \wedge \diamond_p \neg \text{FWB } a]^D$
- ▶ Note that both **first-order** and **higher-order quantifiers** are required

- ▶ challenging: **alethic & deontic modalities, quantification and indexicals**
- ▶ uninterpreted predicate/relation symbols: FWB, Good (1-ary);
ActsOnPurpose, InterferesWith (both 2-ary); NeedsForPurpose (3-ary)
- ▶ PPA $a := \exists E. \text{ActsOnPurpose } a E$ (prospective purposive agents)
an additional axiom postulates that being a PPA is identity-constitutive for
any individual: $[\forall a. \text{PPA}a \rightarrow \square^D(\text{PPA}a)]^D$.
- ▶ $[\varphi]^D$ is indexical validity of φ : true in all contexts (following Kaplan's LD)
- ▶ RightTo $a \varphi := \mathbf{O}_i(\forall b. \neg \text{InterferesWith } b (\varphi a))$
 a has (claim) right to property φ iff it is obligatory that every b does not
interfere with state of affairs (φa)
- ▶ \mathbf{O}_i is *ideal* obligation operator from Carmo & Jones' DDL (\mathbf{O}_a also fine)
- ▶ further axioms interrelate the concepts of goodness and agency
- ▶ axiom $[\forall P. \forall a. \text{NeedsForPurpose } a \text{ FWB } P]^D$ expresses that FWB is always
required in order to act on any purpose
- ▶ FWB is postulated a contingent property: $[\forall a. \diamond_p \text{ FWB } a \wedge \diamond_p \neg \text{FWB } a]^D$
- ▶ Note that both **first-order** and **higher-order quantifiers** are required

- ▶ challenging: **alethic & deontic modalities, quantification and indexicals**
- ▶ uninterpreted predicate/relation symbols: FWB, Good (1-ary);
ActsOnPurpose, InterferesWith (both 2-ary); NeedsForPurpose (3-ary)
- ▶ $\text{PPA } a := \exists E. \text{ActsOnPurpose } a E$ (prospective purposive agents)
an additional axiom postulates that being a PPA is identity-constitutive for
any individual: $[\forall a. \text{PPA } a \rightarrow \square^D(\text{PPA } a)]^D$.
- ▶ $[\varphi]^D$ is indexical validity of φ : true in all contexts (following Kaplan's LD)
- ▶ $\text{RightTo } a \varphi := \mathbf{O}_i(\forall b. \neg \text{InterferesWith } b (\varphi a))$
 a has (claim) right to property φ iff it is obligatory that every b does not
interfere with state of affairs (φa)
- ▶ \mathbf{O}_i is *ideal* obligation operator from Carmo & Jones' DDL (\mathbf{O}_a also fine)
- ▶ further axioms interrelate the concepts of goodness and agency
- ▶ axiom $[\forall P. \forall a. \text{NeedsForPurpose } a \text{ FWB } P]^D$ expresses that FWB is always
required in order to act on any purpose
- ▶ FWB is postulated a contingent property: $[\forall a. \diamond_p \text{ FWB } a \wedge \diamond_p \neg \text{FWB } a]^D$
- ▶ Note that both **first-order** and **higher-order quantifiers** are required

- ▶ challenging: **alethic & deontic modalities, quantification and indexicals**
- ▶ uninterpreted predicate/relation symbols: FWB, Good (1-ary);
ActsOnPurpose, InterferesWith (both 2-ary); NeedsForPurpose (3-ary)
- ▶ $\text{PPA } a := \exists E. \text{ActsOnPurpose } a E$ (prospective purposive agents)
an additional axiom postulates that being a PPA is identity-constitutive for
any individual: $[\forall a. \text{PPA } a \rightarrow \square^D(\text{PPA } a)]^D$.
- ▶ $[\varphi]^D$ is indexical validity of φ : true in all contexts (following Kaplan's LD)
- ▶ $\text{RightTo } a \varphi := \mathbf{O}_i (\forall b. \neg \text{InterferesWith } b (\varphi a))$
 a has (claim) right to property φ iff it is obligatory that every b does not
interfere with state of affairs (φa)
- ▶ \mathbf{O}_i is *ideal* obligation operator from Carmo & Jones' DDL (\mathbf{O}_a also fine)
- ▶ further axioms interrelate the concepts of goodness and agency
- ▶ axiom $[\forall P. \forall a. \text{NeedsForPurpose } a \text{ FWB } P]^D$ expresses that FWB is always
required in order to act on any purpose
- ▶ FWB is postulated a contingent property: $[\forall a. \diamond_p \text{ FWB } a \wedge \diamond_p \neg \text{FWB } a]^D$
- ▶ Note that both **first-order** and **higher-order quantifiers** are required

- ▶ challenging: **alethic & deontic modalities, quantification and indexicals**
- ▶ uninterpreted predicate/relation symbols: FWB, Good (1-ary);
ActsOnPurpose, InterferesWith (both 2-ary); NeedsForPurpose (3-ary)
- ▶ $\text{PPA } a := \exists E. \text{ActsOnPurpose } a E$ (prospective purposive agents)
an additional axiom postulates that being a PPA is identity-constitutive for
any individual: $[\forall a. \text{PPA } a \rightarrow \square^D(\text{PPA } a)]^D$.
- ▶ $[\varphi]^D$ is indexical validity of φ : true in all contexts (following Kaplan's LD)
- ▶ $\text{RightTo } a \varphi := \mathbf{O}_i (\forall b. \neg \text{InterferesWith } b (\varphi a))$
 a has (claim) right to property φ iff it is obligatory that every b does not
interfere with state of affairs (φa)
- ▶ \mathbf{O}_i is *ideal* obligation operator from Carmo & Jones' DDL (\mathbf{O}_a also fine)
- ▶ further axioms interrelate the concepts of goodness and agency
- ▶ axiom $[\forall P. \forall a. \text{NeedsForPurpose } a \text{ FWB } P]^D$ expresses that FWB is always
required in order to act on any purpose
- ▶ FWB is postulated a contingent property: $[\forall a. \diamond_p \text{ FWB } a \wedge \diamond_p \neg \text{FWB } a]^D$
- ▶ Note that both **first-order** and **higher-order quantifiers** are required

- ▶ challenging: **alethic & deontic modalities, quantification and indexicals**
- ▶ uninterpreted predicate/relation symbols: FWB, Good (1-ary);
ActsOnPurpose, InterferesWith (both 2-ary); NeedsForPurpose (3-ary)
- ▶ $\text{PPA } a := \exists E. \text{ActsOnPurpose } a E$ (prospective purposive agents)
an additional axiom postulates that being a PPA is identity-constitutive for
any individual: $[\forall a. \text{PPA } a \rightarrow \square^D(\text{PPA } a)]^D$.
- ▶ $[\varphi]^D$ is indexical validity of φ : true in all contexts (following Kaplan's LD)
- ▶ $\text{RightTo } a \varphi := \mathbf{O}_i (\forall b. \neg \text{InterferesWith } b (\varphi a))$
 a has (claim) right to property φ iff it is obligatory that every b does not
interfere with state of affairs (φa)
- ▶ \mathbf{O}_i is *ideal* obligation operator from Carmo & Jones' DDL (\mathbf{O}_a also fine)
- ▶ further axioms interrelate the concepts of goodness and agency
- ▶ axiom $[\forall P. \forall a. \text{NeedsForPurpose } a \text{ FWB } P]^D$ expresses that FWB is always
required in order to act on any purpose
- ▶ FWB is postulated a contingent property: $[\forall a. \diamond_p \text{ FWB } a \wedge \diamond_p \neg \text{FWB } a]^D$
- ▶ Note that both **first-order** and **higher-order quantifiers** are required

- ▶ challenging: **alethic & deontic modalities, quantification and indexicals**
- ▶ uninterpreted predicate/relation symbols: FWB, Good (1-ary);
ActsOnPurpose, InterferesWith (both 2-ary); NeedsForPurpose (3-ary)
- ▶ $\text{PPA } a := \exists E. \text{ActsOnPurpose } a E$ (prospective purposive agents)
an additional axiom postulates that being a PPA is identity-constitutive for
any individual: $[\forall a. \text{PPA } a \rightarrow \square^D(\text{PPA } a)]^D$.
- ▶ $[\varphi]^D$ is indexical validity of φ : true in all contexts (following Kaplan's LD)
- ▶ $\text{RightTo } a \varphi := \mathbf{O}_i (\forall b. \neg \text{InterferesWith } b (\varphi a))$
 a has (claim) right to property φ iff it is obligatory that every b does not
interfere with state of affairs (φa)
- ▶ \mathbf{O}_i is *ideal* obligation operator from Carmo & Jones' DDL (\mathbf{O}_a also fine)
- ▶ further axioms interrelate the concepts of goodness and agency
- ▶ axiom $[\forall P. \forall a. \text{NeedsForPurpose } a \text{ FWB } P]^D$ expresses that FWB is always
required in order to act on any purpose
- ▶ FWB is postulated a contingent property: $[\forall a. \diamond_p \text{ FWB } a \wedge \diamond_p \neg \text{FWB } a]^D$
- ▶ Note that both **first-order** and **higher-order quantifiers** are required

- ▶ challenging: **alethic & deontic modalities, quantification and indexicals**
- ▶ uninterpreted predicate/relation symbols: FWB, Good (1-ary);
ActsOnPurpose, InterferesWith (both 2-ary); NeedsForPurpose (3-ary)
- ▶ $\text{PPA } a := \exists E. \text{ActsOnPurpose } a E$ (prospective purposive agents)
an additional axiom postulates that being a PPA is identity-constitutive for
any individual: $[\forall a. \text{PPA } a \rightarrow \square^D(\text{PPA } a)]^D$.
- ▶ $[\varphi]^D$ is indexical validity of φ : true in all contexts (following Kaplan's LD)
- ▶ $\text{RightTo } a \varphi := \mathbf{O}_i (\forall b. \neg \text{InterferesWith } b (\varphi a))$
 a has (claim) right to property φ iff it is obligatory that every b does not
interfere with state of affairs (φa)
- ▶ \mathbf{O}_i is *ideal* obligation operator from Carmo & Jones' DDL (\mathbf{O}_a also fine)
- ▶ further axioms interrelate the concepts of goodness and agency
- ▶ axiom $[\forall P. \forall a. \text{NeedsForPurpose } a \text{ FWB } P]^D$ expresses that FWB is always
required in order to act on any purpose
- ▶ FWB is postulated a contingent property: $[\forall a. \diamond_p \text{ FWB } a \wedge \diamond_p \neg \text{FWB } a]^D$
- ▶ Note that both **first-order** and **higher-order quantifiers** are required

- ▶ challenging: **alethic & deontic modalities, quantification and indexicals**
- ▶ uninterpreted predicate/relation symbols: FWB, Good (1-ary);
ActsOnPurpose, InterferesWith (both 2-ary); NeedsForPurpose (3-ary)
- ▶ $\text{PPA } a := \exists E. \text{ActsOnPurpose } a E$ (prospective purposive agents)
an additional axiom postulates that being a PPA is identity-constitutive for
any individual: $[\forall a. \text{PPA } a \rightarrow \square^D(\text{PPA } a)]^D$.
- ▶ $[\varphi]^D$ is indexical validity of φ : true in all contexts (following Kaplan's LD)
- ▶ $\text{RightTo } a \varphi := \mathbf{O}_i (\forall b. \neg \text{InterferesWith } b (\varphi a))$
 a has (claim) right to property φ iff it is obligatory that every b does not
interfere with state of affairs (φa)
- ▶ \mathbf{O}_i is *ideal* obligation operator from Carmo & Jones' DDL (\mathbf{O}_a also fine)
- ▶ further axioms interrelate the concepts of goodness and agency
- ▶ axiom $[\forall P. \forall a. \text{NeedsForPurpose } a \text{ FWB } P]^D$ expresses that FWB is always
required in order to act on any purpose
- ▶ FWB is postulated a contingent property: $[\forall a. \diamond_p \text{ FWB } a \wedge \diamond_p \neg \text{FWB } a]^D$
- ▶ Note that both **first-order** and **higher-order quantifiers** are required

Ethics

Argued for explicit ethical reasoning competencies in IASs

- ▶ development of normative reasoning experimentation platform
- ▶ utilising HOL as universal meta-logic
- ▶ practical evidence (metaphysics, category theory, etc.)
- ▶ suitable also for teaching

Ongoing and further work

- ▶ workbench of (expressive) deontic logics and logic combinations
- ▶ formalisation and mechanisation of ethico-legal theories
- ▶ experimentation and deployment in concrete applications

Ethics

Argued for explicit ethical reasoning competencies in IASs

- ▶ development of normative reasoning experimentation platform
- ▶ utilising HOL as universal meta-logic
- ▶ practical evidence (metaphysics, category theory, etc.)
- ▶ suitable also for teaching

Ongoing and further work

- ▶ workbench of (expressive) deontic logics and logic combinations
- ▶ formalisation and mechanisation of ethico-legal theories
- ▶ experimentation and deployment in concrete applications