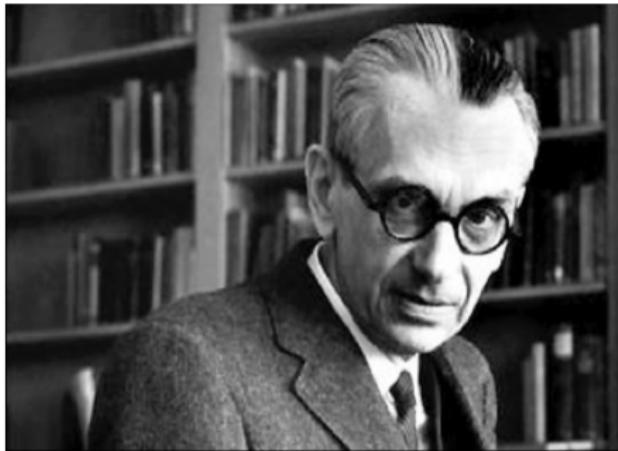


Computational Metaphysics: New Insights on Gödel's Ontological Argument and Modal Collapse

Christoph Benzmüller (jww D. Fuenmayor)

Freie Universität Berlin | University of Luxembourg



“There is a scientific (exact) philosophy and theology,
which deals with concepts of the highest abstractness;
and this is also most highly fruitful for science.”

- Kurt Gödel (Wang, 1996)[p. 316]

Presentation Outline

A Ontological Argument of Gödel & Scott on the Computer

- ▶ Recap of Methodology and Main Findings (jww B. Woltzenlogel-Paleo)

B Relevant Notions for this Talk:

- ▶ Intension vs. extension of properties
- ▶ Ultrafilter

C Comparative Analysis on the Computer:

- ▶ Gödel/Scott (1972) variant
- ▶ Anderson's (1990) variant
- ▶ Fitting's (2002) variant

D Discussion & Conclusion



Part A

— Computational Metaphysics (recap) —

Ontological Argument by Gödel & Scott on the Computer

Related work:

- ▶ Ed Zalta (& co) with PROVER9 at Stanford
- ▶ John Rushby with PVS at SRI

[AJP 2011, CADE 2015]
[CAV-WS 2013, JAL 2018]

Ontological Proofs of God's Existence

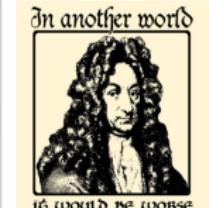
A Long and Continuing Tradition in Philosophy



St. Anselm



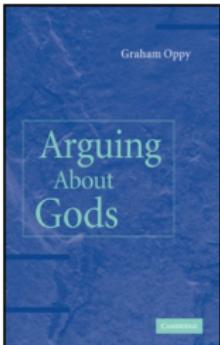
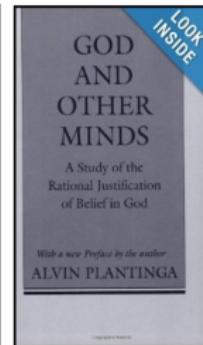
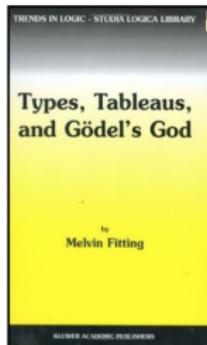
Descartes



Leibniz



Gödel



Computational Metaphysics: Kurt Gödel's Ontological Argument (1970)

Ontologischer Beweis Feb 10, 1970

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

At 1 $P(\varphi), P(\psi) \supset P(\varphi \wedge \psi)$ At 2 $P(\varphi) \supset P(\neg \varphi)$

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

P2 $\varphi \text{ Emx} \equiv (\psi)[\forall x(\varphi(x) \supset N(y)[\varphi(y) \supset \psi(y)])]$ (Existence)

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

At 2 $\begin{cases} P(\varphi) \supset N(P(\varphi)) \\ \neg P(\varphi) \supset N \neg P(\varphi) \end{cases}$ } because it follows from the nature of the property

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

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

At 3 $P(E)$

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

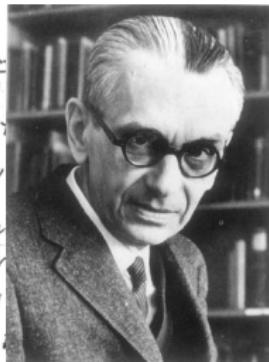
thus $(\exists x) G(x) \supset N(\exists y) G(y)$

" $M(\exists x) G(x) \supset M N(\exists y) G(y)$

" $\supset N(\exists y) G(y)$ M = permuting

any two instances of x are met. equivalent
exclusive or and for any number of numerants

$M(x) G(x)$ means "all
possible This is:
At 4: $P(\varphi), \varphi \supset \psi$
~~True~~ $\begin{cases} x=x & \text{is pr.} \\ x \neq x & \text{is not pr.} \end{cases}$
 But if a system S is
it would mean, that
(if positive) would be $x \neq x$



Positive means positive in the moral aesthetic sense (independently of the accidental structure of the world). Only ~~the~~ the at time. It may also mean "affirmation" as opposed to "privation" (or crushing privation). This supports the platonist

$\neg \varphi$ is negative $(\exists x) N \neg P(x)$ Otherwise $\varphi(x) \supset x \neq x$
hence $x \neq x$ (positive) $\neg x \neq x$ (negative) At the end of proof At

i.e. the formal form in terms of elem. prop. contains a member without negation.

Computational Metaphysics: Kurt Gödel's Ontological Argument (1970)

Ontologischer Beweis FEB 10, 1970

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

At. 1: $P(\varphi), P(\psi) \vdash P(\varphi \wedge \psi) \quad \vdash P(\varphi) \wedge P(\neg \varphi)$

P1 $G(x) \equiv (\varphi) [P(\varphi) \supset \varphi(x)] \quad (\text{God})$

P2 $\varphi \text{ FM } x = (\forall y) E_{\text{Ex. } x} y [P(y) \supset \varphi(y)] \quad (\text{Ex. of } x)$

$P \supset_N q = N(P \supset q) \quad \text{Necessity}$

At. 2 $\begin{array}{l} P(\varphi) \supset N P(\varphi) \\ \sim P(\varphi) \supset N \sim P(\varphi) \end{array} \quad \left. \begin{array}{l} \text{because it follows} \\ \text{from the nature of the} \\ \text{property} \end{array} \right.$

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

Df. $E(x) \equiv (\forall y) [G(x) \supset N \exists x \varphi(x)] \quad \text{necessary Exist.}$

At. 3 $P(E)$

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

$M(x) F(x)$: means all pos. prop. w.r.t. com-
patible ^{the system of}
This is true because of:

At 4: $P(\varphi), \varphi \supset \psi \vdash P(\psi)$ which impl.
~~True~~ { $x=x$ is positive
~~False~~ { $x \neq x$ is negative

But if a system S of pos. prop. were incon-
sistent it would mean that the non-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 ~~in the~~ ^{at the} at. time. It re-
fers also to "Attribution" as opposed to "privatism".

Notion of "Godlike":

- Being Godlike is equivalent to having all positive properties.

Note: this definition is "second-order".

Computational Metaphysics: Kurt Gödel's Ontological Argument (1970)

Ontologischer Beweis

FEB 10, 1970

$M(\exists x)F(x)$ means all non-empty sets in the system of

In the end we prove

- Necessarily (N), there exists God.

Note: we need to formalize "necessity" and "possibility".

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

Df. $E(x) \equiv \exists p(p \in x \wedge \forall y \exists q(q \in y \rightarrow q \in x))$ necessary Existence

Ax 3 $P(E)$

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

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

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

" $\supset N(\exists y)G(y)$ M-penitent

any two elements of M are not equivalent,
exclusive or * and for any number of members in M

a positive would be $x \neq x$

Positive means positive in the moral aesthetic sense (independently of the accidental structure of the world). Only ~~the~~ ^{the} at time. It may also mean "affirmation" as opposed to "privation (or crushing of privation)." This supports the plausibility of

$\exists y \supset p(y) \wedge (\forall x)(N(\neg p(x)) \rightarrow \neg G(x) \supset x \neq x)$

hence $x \neq x$ (positive) $\wedge x = x$ (negative) At the epiphany of privation

i.e. the formal formulation of elem. prop. contains a member without negation.

Computational Metaphysics: Gödel's (1970) and Scott's Variants (1972)

Onkelogischer Bereich Feb 10, 1970

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

At 1 $P(\varphi), P(\psi) \supset P(\varphi \wedge \psi)$ At 2 $P(\varphi) \supset P(\neg \varphi)$

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

P2 $\varphi \text{ Emx} \equiv (\psi)[\psi(x) \supset N(y)[\varphi(y) \supset \psi(y)]]$ (Emx $\neq x$)

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

At 2 $\begin{array}{l} P(\varphi) \supset N P(\varphi) \\ \sim P(\varphi) \supset N \sim P(\varphi) \end{array} \left. \begin{array}{l} \text{because it follows} \\ \text{from the nature of the} \\ \text{property} \end{array} \right\}$

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

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

At 3 $P(E)$

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

 how $(\exists x) G(x) \supset N(\exists y) G(y)$

 " $M(\exists x) G(x) \supset M N(\exists y) G(y)$

 " $\supset N(\exists y) G(y)$ M = permitting

any two elements of X are nec. equivalent
exclusive or and for any number of nonmembers

$M(\exists x) G(x)$ means all
possible This is:
possible $\left\{ \begin{array}{l} x=x \text{ is pr.} \\ x \neq x \text{ is pr.} \end{array} \right.$
But if a system is of
it would mean, that the num.prop. is (which
is positive) would be $x \neq x$



Positive means positive in the moral aest.
sense. (independently of the accidental structure of
the world). Only ~~the~~ the at. time. It may
also mean "affirmation" as opposed to "privatization"
(or containing privation). This supports the pl. part

$\neg \varphi$ is negative $(\exists x) N \sim P(x)$ Otherwise $\varphi(x) \supset x \neq x$
hence $x \neq x$ (positive) $\neg x \neq x$ (negative) At
or the absence of $\neg x \neq x$

dog
i.e. the formal form in terms of elem. prop. contains a
Member without negation.

Computational Metaphysics: Gödel's (1970) and Scott's Variants (1972)

Onkologischer Beweis FEB 10, 1970

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

At 1 $P(\varphi), P(\psi) \vdash P(\varphi \wedge \psi)$ At 2 $P(\varphi) \nvdash P(\neg \varphi)$

Pl $G(x) \equiv (\exists y)[P(y) \Rightarrow \varphi(x)]$ (Good)

P2 $\varphi \text{ Emx} \equiv (\forall y)[\forall z(y \neq z) \neg P(y) \Rightarrow \neg \varphi(z)]$ (Emx $\neq x$)

$\neg \varphi = N(\varphi, \neg \varphi)$ Necessity

At 2 $\begin{cases} P(\varphi) \supset N(P(\varphi)) \\ \neg \varphi \supset N \neg P(\varphi) \end{cases}$ because it follows from the nature of the property

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

Df. $E(x) \equiv (\forall y)[\varphi \text{ Emx} \supset N \exists z \varphi(z)]$ necessary Existence

At 3 $P(E)$

Th. $G(x) \supset N(\forall y)G(y)$

$M(x)G(x)$: means all pos. propo. in: compatible
This is true because of:

At 4: $P(\varphi), \varphi \supset \psi \vdash P(\psi)$ which impl.

- { $x=x$ is positive
- { $x \neq x$ is negative

But if a system S of pos. propo. were incongruous it would mean, that the num.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 ~~the~~ ^{at} the at. time. It may also mean "Attribution" as opposed to "privation" (or contains negation). This is an aside and

(Main) Difference between Gödel and Scott: Def. of "Essence (Ess.)"

- **Gödel:** Property E is Ess. of x iff all of x's properties are nec. entailed by E.
- ex **Scott:** Property E is Ess. of x iff x has E and all of x's properties are nec. entailed by E.

(Higher-Order) Modal Logic

$\Box P$

P is necessary

$\Diamond P$

P is possible

\Box and \Diamond are not truth-functional

Higher-Order Logic can be extended by $\Box P$ and $\Diamond P$

(Higher-Order) Modal Logic

$\Box P$

P is necessary

$\Diamond P$

P is possible

\Box and \Diamond are not truth-functional

Higher-Order Logic can be extended by $\Box P$ and $\Diamond P$

(Higher-Order) Modal Logic

$\Box P$

P is necessary

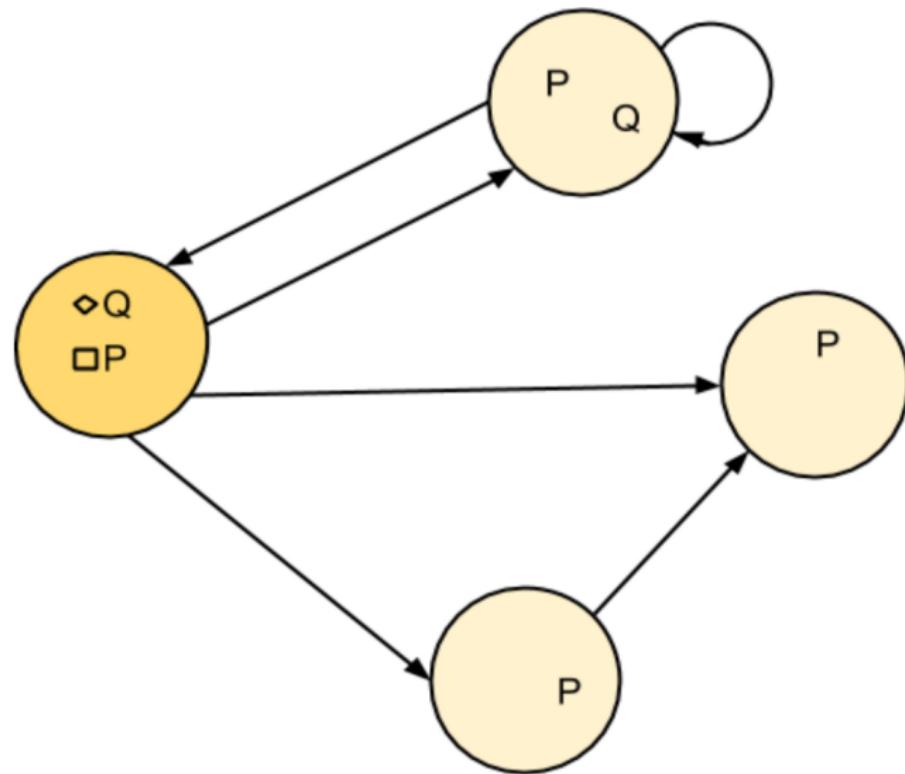
$\Diamond P$

P is possible

\Box and \Diamond are not truth-functional

Higher-Order Logic can be extended by $\Box P$ and $\Diamond P$

(Higher-Order) Modal Logics: Kripke-style Semantics - Possible Worlds



Semantical Embedding of Higher-Order Model Logic in Higher-Order Logic (HOL)

Standard translation:

- ▶ $\vee_{(i \rightarrow o) \rightarrow (i \rightarrow o) \rightarrow (i \rightarrow o)} := \lambda\varphi_{i \rightarrow o}.\lambda\psi_{i \rightarrow o}.(\lambda w_i.\varphi w \vee \psi w)$
- ▶ $\square_{(i \rightarrow o) \rightarrow (i \rightarrow o)} := \lambda\varphi_{i \rightarrow o}.(\lambda w_i.\forall v_i.Rwv \rightarrow \varphi v)$ ($R_{i \rightarrow i \rightarrow o}$: accessibility relation)

Standard translation extended to quantifiers:

- ▶ in HOL: $\forall x.\phi x$ shorthand for $\Pi(\lambda x.\phi x)$
- ▶ $\square\forall x.Px$ is represented as $\square\Pi'(\lambda x_a.\lambda w_i.\Pi(x_a,\Phi_{xw}))$ where $\Pi' := \lambda\Phi_{\alpha \rightarrow i \rightarrow o}.\lambda w_i.\Pi(\lambda x_a.\Phi_{xw})$ and \square is resolved as above
 - $\square\forall x.Px \equiv \square\Pi'(\lambda x.\lambda w.Pxw)$
 - $\equiv \square((\lambda\Phi.\lambda w.\Pi(\lambda x.\Phi_{xw}))(\lambda x.\lambda w.Pxw))$
 - $\equiv \square(\lambda w.\Pi(\lambda x.(\lambda x.\lambda w.Pxw)xw))$
 - $\equiv \square(\lambda w.\Pi(\lambda x.Pxw))$
 - $\equiv (\lambda\varphi.\lambda w.\forall v.(Rwv \rightarrow \varphi v))(\lambda w.\Pi(\lambda x.Pxw))$
 - $\equiv (\lambda\varphi.\lambda w.\Pi(\lambda v.Rwv \rightarrow \varphi v))(\lambda w.\Pi(\lambda x.Pxw))$
 - $\equiv (\lambda w.\Pi(\lambda v.Rwv \rightarrow (\lambda w.\Pi(\lambda x.Pxw))v))$
 - $\equiv (\lambda w.\Pi(\lambda v.Rwv \rightarrow \Pi(\lambda x.Pxv)))$
 - $\equiv (\lambda w.\forall v.(Rwv \rightarrow \forall x.Pxv))$

- ▶ 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

Semantical Embedding of Higher-Order Model Logic in Higher-Order Logic (HOL)

Standard translation:

- ▶ $\vee_{(i \rightarrow o) \rightarrow (i \rightarrow o) \rightarrow (i \rightarrow o)} := \lambda\varphi_{i \rightarrow o}.\lambda\psi_{i \rightarrow o}.(\lambda w_i.\varphi w \vee \psi w)$
- ▶ $\square_{(i \rightarrow o) \rightarrow (i \rightarrow o)} := \lambda\varphi_{i \rightarrow o}.(\lambda w_i.\forall v_i.Rwv \rightarrow \varphi v)$ ($R_{i \rightarrow i \rightarrow o}$: accessibility relation)

Standard translation extended to quantifiers:

- ▶ in HOL: $\forall x.\phi x$ shorthand for $\Pi(\lambda x.\phi x)$
- ▶ $\square\forall x.Px$ is represented as $\square\Pi'(\lambda x_a.\lambda w_i.Pxw)$ where
 $\Pi' := \lambda\Phi_{\alpha \rightarrow i \rightarrow o}.\lambda w_i.\Pi(\lambda x_a.\Phi xw)$ and \square is resolved as above
 - $\square\forall x.Px \equiv \square\Pi'(\lambda x.\lambda w.Pxw)$
 - $\equiv \square((\lambda\Phi.\lambda w.\Pi(\lambda x.\Phi xw))(\lambda x.\lambda w.Pxw))$
 - $\equiv \square(\lambda w.\Pi(\lambda x.(\lambda x.\lambda w.Pxw)xw))$
 - $\equiv \square(\lambda w.\Pi(\lambda x.Pxw))$
 - $\equiv (\lambda\varphi.\lambda w.\forall v.(Rwv \rightarrow \varphi v))(\lambda w.\Pi(\lambda x.Pxw))$
 - $\equiv (\lambda\varphi.\lambda w.\Pi(\lambda v.Rwv \rightarrow \varphi v))(\lambda w.\Pi(\lambda x.Pxw))$
 - $\equiv (\lambda w.\Pi(\lambda v.Rwv \rightarrow (\lambda w.\Pi(\lambda x.Pxw))v))$
 - $\equiv (\lambda w.\Pi(\lambda v.Rwv \rightarrow \Pi(\lambda x.Pxv)))$
 - $\equiv (\lambda w.\forall v.(Rwv \rightarrow \forall x.Pxv))$

- ▶ 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

Semantical Embedding of Higher-Order Model Logic in Higher-Order Logic (HOL)

Standard translation:

- ▶ $\vee_{(i \rightarrow o) \rightarrow (i \rightarrow o) \rightarrow (i \rightarrow o)} := \lambda\varphi_{i \rightarrow o}.\lambda\psi_{i \rightarrow o}.(\lambda w_i.\varphi w \vee \psi w)$
- ▶ $\square_{(i \rightarrow o) \rightarrow (i \rightarrow o)} := \lambda\varphi_{i \rightarrow o}.(\lambda w_i.\forall v_i.Rwv \rightarrow \varphi v)$ ($R_{i \rightarrow i \rightarrow o}$: accessibility relation)

Standard translation extended to quantifiers:

- ▶ in HOL: $\forall x.\phi x$ shorthand for $\Pi(\lambda x.\phi x)$
- ▶ $\square\forall x.Px$ is represented as $\square\Pi'(\lambda x_a.\lambda w_i.Pxw)$ where
 $\Pi' := \lambda\Phi_{\alpha \rightarrow i \rightarrow o}.\lambda w_i.\Pi(\lambda x_a.\Phi xw)$ and \square is resolved as above

$$\begin{aligned}\square\forall x.Px &\equiv \square\Pi'(\lambda x.\lambda w.Pxw) \\ &\equiv \square((\lambda\Phi.\lambda w.\Pi(\lambda x.\Phi xw))(\lambda x.\lambda w.Pxw)) \\ &\equiv \square(\lambda w.\Pi(\lambda x.(\lambda x.\lambda w.Pxw)xw)) \\ &\equiv \square(\lambda w.\Pi(\lambda x.Pxw)) \\ &\equiv (\lambda\varphi.\lambda w.\forall v.(Rwv \rightarrow \varphi v))(\lambda w.\Pi(\lambda x.Pxw)) \\ &\equiv (\lambda\varphi.\lambda w.\Pi(\lambda v.Rwv \rightarrow \varphi v))(\lambda w.\Pi(\lambda x.Pxw)) \\ &\equiv (\lambda w.\Pi(\lambda v.Rwv \rightarrow (\lambda w.\Pi(\lambda x.Pxw))v)) \\ &\equiv (\lambda w.\Pi(\lambda v.Rwv \rightarrow \Pi(\lambda x.Pxv))) \\ &\equiv (\lambda w.\forall v.(Rwv \rightarrow \forall x.Pxv))\end{aligned}$$

- ▶ 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

Semantical Embedding of Higher-Order Model Logic in Higher-Order Logic (HOL)

Standard translation:

- ▶ $\vee_{(i \rightarrow o) \rightarrow (i \rightarrow o) \rightarrow (i \rightarrow o)} := \lambda\varphi_{i \rightarrow o}.\lambda\psi_{i \rightarrow o}.(\lambda w_i.\varphi w \vee \psi w)$
- ▶ $\square_{(i \rightarrow o) \rightarrow (i \rightarrow o)} := \lambda\varphi_{i \rightarrow o}.(\lambda w_i.\forall v_i.Rwv \rightarrow \varphi v)$ ($R_{i \rightarrow i \rightarrow o}$: accessibility relation)

Standard translation extended to quantifiers:

- ▶ in HOL: $\forall x.\phi x$ shorthand for $\Pi(\lambda x.\phi x)$
- ▶ $\square\forall x.Px$ is represented as $\square\Pi'(\lambda x_a.\lambda w_i.Pxw)$ where
 $\Pi' := \lambda\Phi_{\alpha \rightarrow i \rightarrow o}.\lambda w_i.\Pi(\lambda x_a.\Phi xw)$ and \square is resolved as above

$$\begin{aligned}\square\forall x.Px &\equiv \square\Pi'(\lambda x.\lambda w.Pxw) \\ &\equiv \square((\lambda\Phi.\lambda w.\Pi(\lambda x.\Phi xw))(\lambda x.\lambda w.Pxw)) \\ &\equiv \square(\lambda w.\Pi(\lambda x.(\lambda x.\lambda w.Pxw)xw)) \\ &\equiv \square(\lambda w.\Pi(\lambda x.Pxw)) \\ &\equiv (\lambda\varphi.\lambda w.\forall v.(Rwv \rightarrow \varphi v))(\lambda w.\Pi(\lambda x.Pxw)) \\ &\equiv (\lambda\varphi.\lambda w.\Pi(\lambda v.Rwv \rightarrow \varphi v))(\lambda w.\Pi(\lambda x.Pxw)) \\ &\equiv (\lambda w.\Pi(\lambda v.Rwv \rightarrow (\lambda w.\Pi(\lambda x.Pxw))v)) \\ &\equiv (\lambda w.\Pi(\lambda v.Rwv \rightarrow \Pi(\lambda x.Pxv))) \\ &\equiv (\lambda w.\forall v.(Rwv \rightarrow \forall x.Pxv))\end{aligned}$$

- ▶ 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

Semantical Embedding of Higher-Order Model Logic in Higher-Order Logic (HOL)

Standard translation:

- ▶ $\vee_{(i \rightarrow o) \rightarrow (i \rightarrow o) \rightarrow (i \rightarrow o)} := \lambda\varphi_{i \rightarrow o}.\lambda\psi_{i \rightarrow o}.(\lambda w_i.\varphi w \vee \psi w)$
- ▶ $\square_{(i \rightarrow o) \rightarrow (i \rightarrow o)} := \lambda\varphi_{i \rightarrow o}.(\lambda w_i.\forall v_i.Rwv \rightarrow \varphi v)$ ($R_{i \rightarrow i \rightarrow o}$: accessibility relation)

Standard translation extended to quantifiers:

- ▶ in HOL: $\forall x.\phi x$ shorthand for $\Pi(\lambda x.\phi x)$
- ▶ $\square\forall x.Px$ is represented as $\square\Pi'(\lambda x_a.\lambda w_i.Pxw)$ where
 $\Pi' := \lambda\Phi_{\alpha \rightarrow i \rightarrow o}.\lambda w_i.\Pi(\lambda x_a.\Phi xw)$ and \square is resolved as above

$$\begin{aligned}\square\forall x.Px &\equiv \square\Pi'(\lambda x.\lambda w.Pxw) \\ &\equiv \square((\lambda\Phi.\lambda w.\Pi(\lambda x.\Phi xw))(\lambda x.\lambda w.Pxw)) \\ &\equiv \square(\lambda w.\Pi(\lambda x.(\lambda x.\lambda w.Pxw)xw)) \\ &\equiv \square(\lambda w.\Pi(\lambda x.Pxw)) \\ &\equiv (\lambda\varphi.\lambda w.\forall v.(Rwv \rightarrow \varphi v))(\lambda w.\Pi(\lambda x.Pxw)) \\ &\equiv (\lambda\varphi.\lambda w.\Pi(\lambda v.Rwv \rightarrow \varphi v))(\lambda w.\Pi(\lambda x.Pxw)) \\ &\equiv (\lambda w.\Pi(\lambda v.Rwv \rightarrow (\lambda w.\Pi(\lambda x.Pxw))v)) \\ &\equiv (\lambda w.\Pi(\lambda v.Rwv \rightarrow \Pi(\lambda x.Pxv))) \\ &\equiv (\lambda w.\forall v.(Rwv \rightarrow \forall x.Pxv))\end{aligned}$$

- ▶ 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

Semantical Embedding of Higher-Order Model Logic in Higher-Order Logic (HOL)

Isabelle2018/HOL - IHOML.thy

```
□ IHOML.thy (~/GITHUBS/chrisgitlab/talks/2019-Dubrovnik/)

1 theory IHOML imports Main
2 begin
3   typedcl i (* possible worlds *) typedcl e (* individuals *)
4 (*Logical Operators as Truth-Sets*)
5   abbreviation mnot ("¬_"[52]53) where "¬φ ≡ λw. ¬(φ w)"
6   abbreviation negpred ("¬_"[52]53) where "¬Φ ≡ λx. ¬(Φ x)"
7   abbreviation mnegpred ("¬_"[52]53) where "¬Φ ≡ λx.λw. ¬(Φ x w)"
8   abbreviation mand (infixr "Λ"51) where "φΛψ ≡ λw. (φ w)Λ(ψ w)"
9   abbreviation mor (infixr "∨"50) where "φ∨ψ ≡ λw. (φ w)∨(ψ w)"
10  abbreviation mimp (infixr "→"49) where "φ→ψ ≡ λw. (φ w)→(ψ w)"
11  abbreviation mequ (infixr "↔"48) where "φ↔ψ ≡ λw. (φ w)↔(ψ w)"
12 (*Possibilist Quantification*)
13  abbreviation mforall ("∀") where "∀Φ ≡ λw.∀x. (Φ x w)"
14  abbreviation mforallB (binder "∀" [8]9) where "∀x. φ(x) ≡ ∀φ"
15  abbreviation mexists ("∃") where "∃Φ ≡ λw.∃x. (Φ x w)"
16  abbreviation mexistsB (binder "∃" [8]9) where "∃x. φ(x) ≡ ∃φ"
17 (*Actualist Quantification*)
18  consts Exists:::"(e⇒i⇒bool)" ("existsAt")
19  abbreviation mforallAct ("∀ᴱ") where "∀ᴱΦ ≡ λw.∀x. (existsAt x w)→(Φ x w)"
20  abbreviation mexistsAct ("∃ᴱ") where "∃ᴱΦ ≡ λw.∃x. (existsAt x w) ∧ (Φ x w)"
21  abbreviation mforallActB (binder "∀ᴱ" [8]9) where "∀ᴱx. φ(x) ≡ ∀ᴱφ"
22  abbreviation mexistsActB (binder "∃ᴱ" [8]9) where "∃ᴱx. φ(x) ≡ ∃ᴱφ"
23 (*Modal Operators*)
24  consts aRel:::"i⇒i⇒bool" (infixr "r" 70) (*accessibility relation r*)
25  abbreviation mbox ("◻_"[52]53) where "◻φ ≡ λw.∀v. (w r v)→(φ v)"
26  abbreviation mdia ("◊_"[52]53) where "◊φ ≡ λw.∃v. (w r v)∧(φ v)"
27 (*Meta-logical Predicates*)
28  abbreviation valid ("◻["[8]) where "◻[ψ] ≡ ∀w. (ψ w)"
29 (*Consistency and some useful definitions on (accessibility) relations*)
30  lemma True nitpick[satisfy] oops (*Model found by model finder Nitpick*)
31 end
32
```

Computational Metaphysics: Scott's and Gödel's Variants — Demo

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

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

Def. A Godlike being possesses all positive properties: $G(x) \leftrightarrow \forall\phi[P(\phi) \rightarrow \phi(x)]$

Axiom The property of being Godlike is positive: $P(G)$

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

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

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

Thm. Being Godlike is an essence of any Godlike being: $\forall x[G(x) \rightarrow G \text{ ess. } x]$

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

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

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

Computational Metaphysics: Scott's and Gödel's Variants — Demo

Axiom	$\forall\phi[P(\neg\phi) \leftrightarrow \neg P(\phi)]$
Axiom	$\forall\phi\forall\psi[(P(\phi) \wedge \Box\forall x[\phi(x) \rightarrow \psi(x)]) \rightarrow P(\psi)]$
Thm.	$\forall\phi[P(\phi) \rightarrow \Diamond\exists x\phi(x)]$
Def.	$G(x) \leftrightarrow \forall\phi[P(\phi) \rightarrow \phi(x)]$
Axiom	$P(G)$
Cor.	$\Diamond\exists xG(x)$
Axiom	$\forall\phi[P(\phi) \rightarrow \Box P(\phi)]$
Def.	$\phi \text{ ess. } x \leftrightarrow \phi(x) \wedge \forall\psi(\psi(x) \rightarrow \Box\forall y(\phi(y) \rightarrow \psi(y)))$
Thm.	$\forall x[G(x) \rightarrow G \text{ ess. } x]$
Def.	$NE(x) \leftrightarrow \forall\phi[\phi \text{ ess. } x \rightarrow \Box\exists y\phi(y)]$
Axiom	$P(NE)$
Thm.	$\Box\exists xG(x)$

Computational Metaphysics: Scott's and Gödel's Variants — Demo

Axiom

$$\forall\phi[P(\neg\phi) \leftrightarrow \neg P(\phi)]$$

Axiom

$$\forall\phi\forall\psi[(P(\phi) \wedge \Box\forall x[\phi(x) \rightarrow \psi(x)]) \rightarrow P(\psi)]$$

Def.

$$G(x) \leftrightarrow \forall\phi[P(\phi) \rightarrow \phi(x)]$$

Axiom

$$P(G)$$

Axiom

$$\forall\phi[P(\phi) \rightarrow \Box P(\phi)]$$

Def.

$$\phi \text{ ess. } x \leftrightarrow \phi(x) \wedge \forall\psi(\psi(x) \rightarrow \Box\forall y(\phi(y) \rightarrow \psi(y)))$$

Def.

$$NE(x) \leftrightarrow \forall\phi[\phi \text{ ess. } x \rightarrow \Box\exists y\phi(y)]$$

Axiom

$$P(NE)$$

Thm.

$$\Box\exists xG(x)$$

Computational Metaphysics: Scott's and Gödel's Variants — Demo

The screenshot shows a computer interface for formalizing mathematics. At the top, a window titled "UltrafilterMovie1" displays a proof script in Isabelle/HOL. The script is titled "GoedelProof.thy" and follows Fitting's textbook. It includes definitions for "pos", "positiveProperty", and "A3", and axioms A1, A2, and A3. Theorems T1 and T2 are proved using metis, and T3 is proved using simp. Part II of the proof is indicated. Below the script, a video player shows a video titled "UltrafilterMovie1" with a duration of 02:26. The video player has controls for back, forward, and play/pause, and a progress bar. At the bottom, there are tabs for Output, Query, Sledgehammer, and Symbols.

```
theory GoedelProof imports IHOML      (* This formalization follows Fitting's textbook *)
begin
(*Positiveness/perfection: uninterpreted constant symbol*)
consts positiveProperty::"(e⇒i⇒bool)⇒i⇒bool" ("P")
(*Some auxiliary definitions needed to formalise A3*)
definition h1 ("pos")    where "pos Z ≡ ∀X. Z X → P X"
definition h2 (infix "∩" 60) where "X ∩ Z ≡ □(∀x.(X x ↔ (¬(Y z) → (Y x))))"
definition h3 (infix "⇒" 60) where "X ⇒ Y ≡ □(∀z. X z → Y z)"

(**Part I**)
(*D1*) definition G ("G") where "G ≡ (λx. ∀Y. P Y → Y x)"
(*A1*) axiomatization where Ala: "[| ∀X. P (¬X) → ¬(P X)|]" and Alb:"[| ∀X. ¬(P X) → P (¬X)|]"
(*A2*) axiomatization where A2: "[| ∀X Y. (P X ∧ (X ⇒ Y)) → P Y|]"
(*A3*) axiomatization where A3: "[| ∀Z X. (pos Z ∧ X ∩ Z) → P X|]"
(*T1*) theorem T1: "[| ∀X. P X → ◇∃ X|]" by (metis Ala A2 h3_def)
(*T2*) theorem T2: "[| P G |]" proof -
  {have 1: "¬∀w. ∃Z X. (P G ∨ pos Z ∧ X ∩ Z ∧ ¬P X) w" by (metis(full_types) G_def h1_def h2_def)
  have 2: "[| ∃Z X. (pos Z ∧ X ∩ Z) → P X |] → [| P G |]" using 1 by auto}
  thus ?thesis using A3 by blast qed
(*T3*) theorem T3: "[| ◇∃ G |]" using T1 T2 by simp
(**Part II**)
(*Logic VDK*) axiomatization where summa: "Symmetric Axiom"
```

theorem U3: $P' \subseteq P \wedge P \subseteq$
Undefined fact: "T6"Δ

00:00 - 02:26

100%

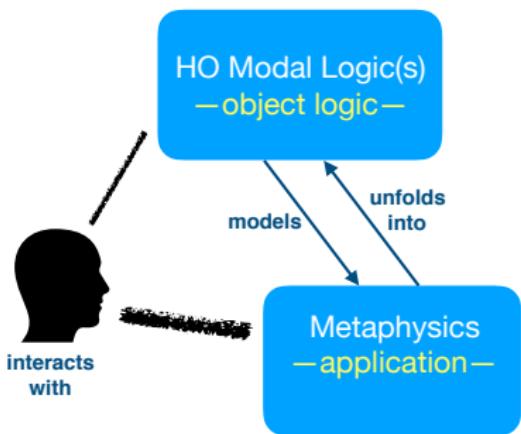
Output Query Sledgehammer Symbols

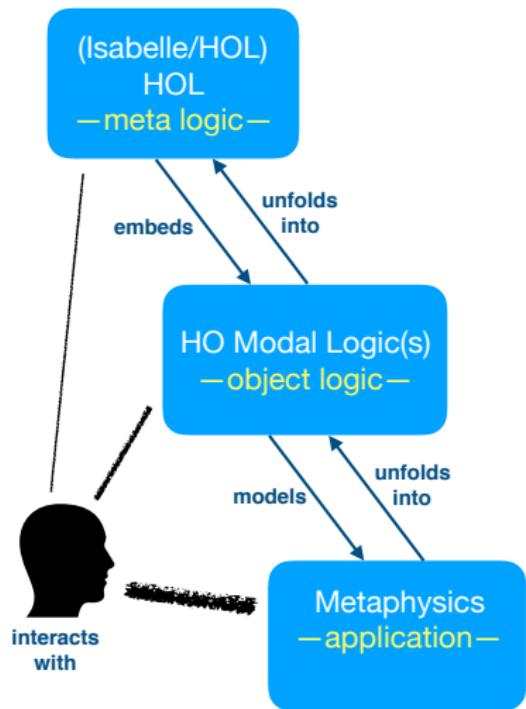


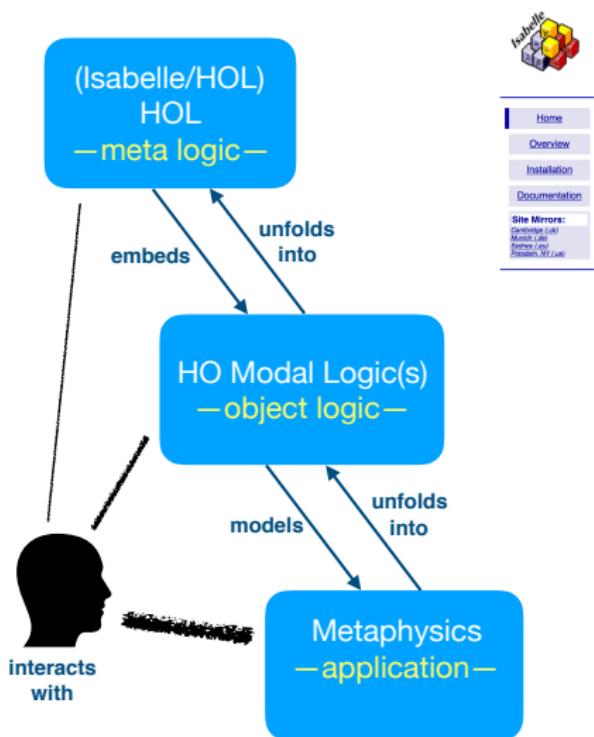
interacts
with



Metaphysics
—application—







Isabelle



What is Isabelle?

Isabelle is a generic proof assistant. It allows mathematical formulae to be expressed in a formal language and provides tools for proving those formulae 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](#)

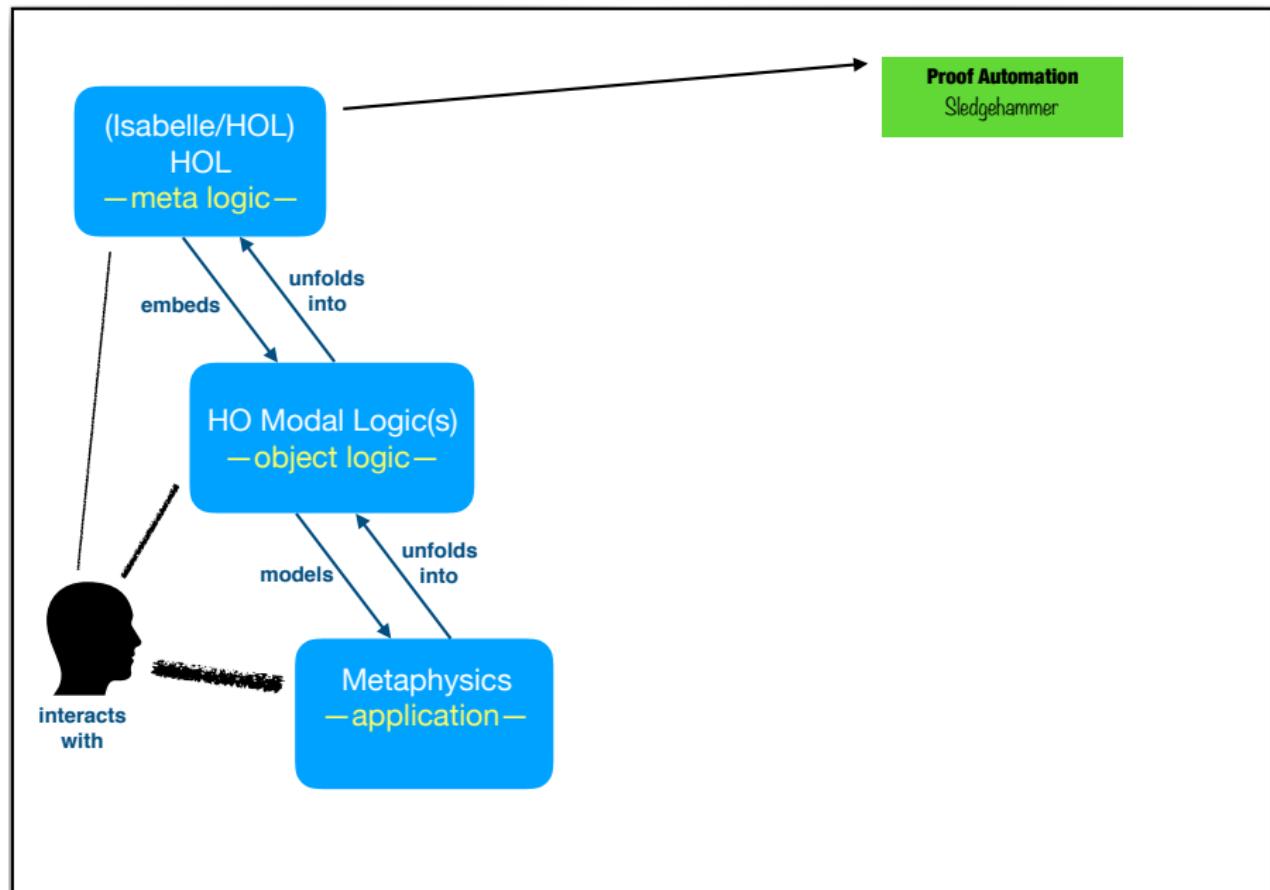
Some notable changes:

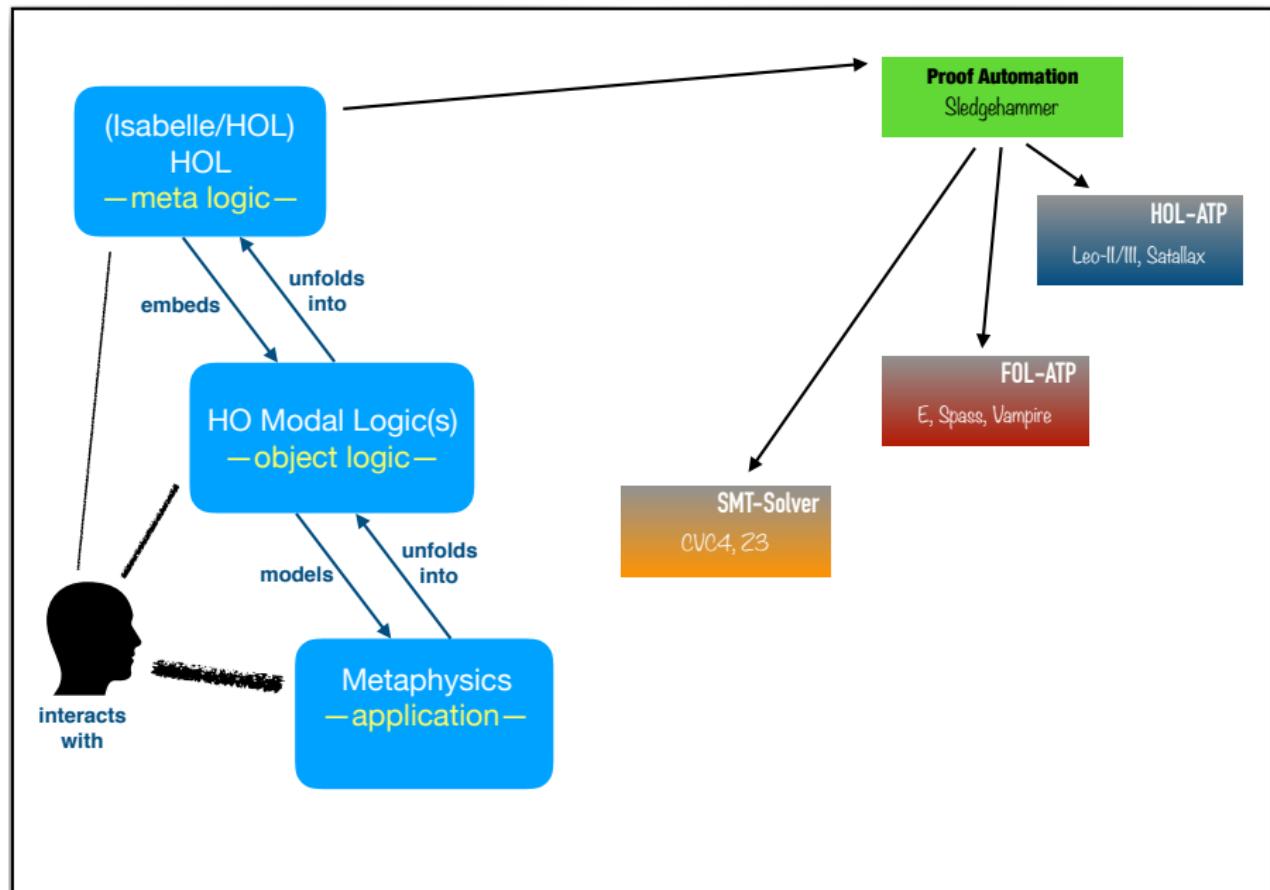
- Experimental support for Visual Studio Code as alternative PIDE front-end.
- Improved Isabelle/Edit Prover IDE: management of session sources independently of editor buffers, removal of unused theories, explicit indication of theory status, more careful auto-indentation.
- Session-qualified theory imports.
- Code completion 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.
- SML database support in Isabelle/Scala.

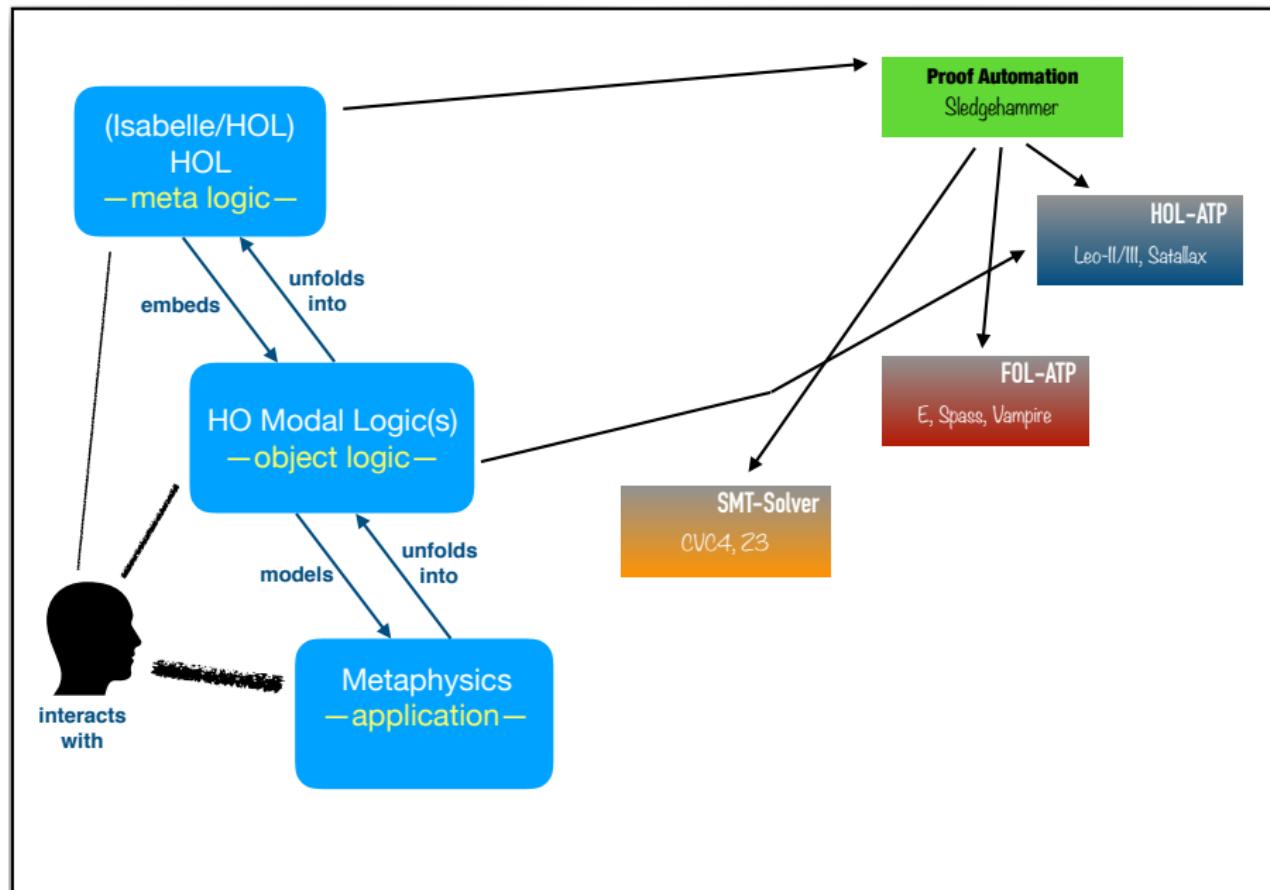
See also the cumulative [NEWS](#).

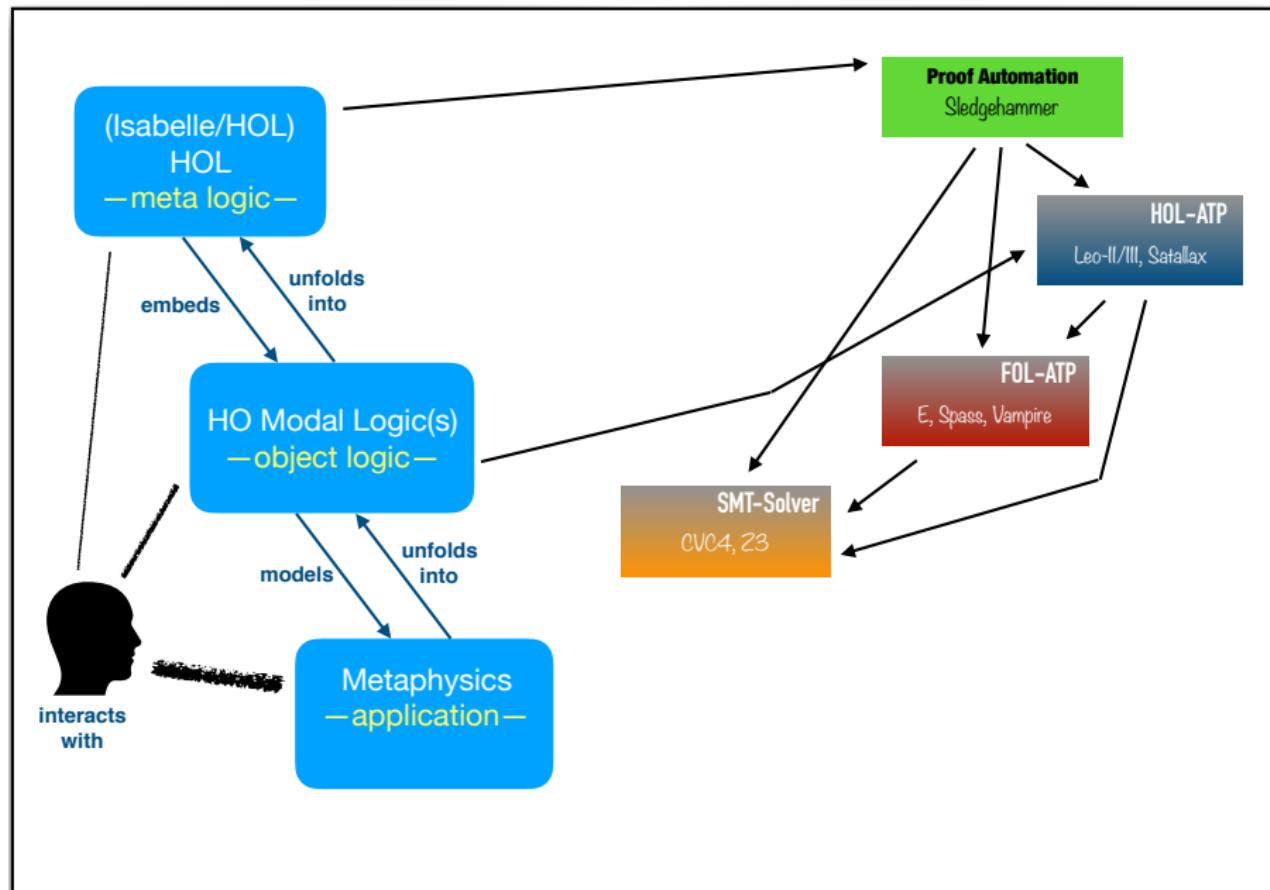
Distribution & Support

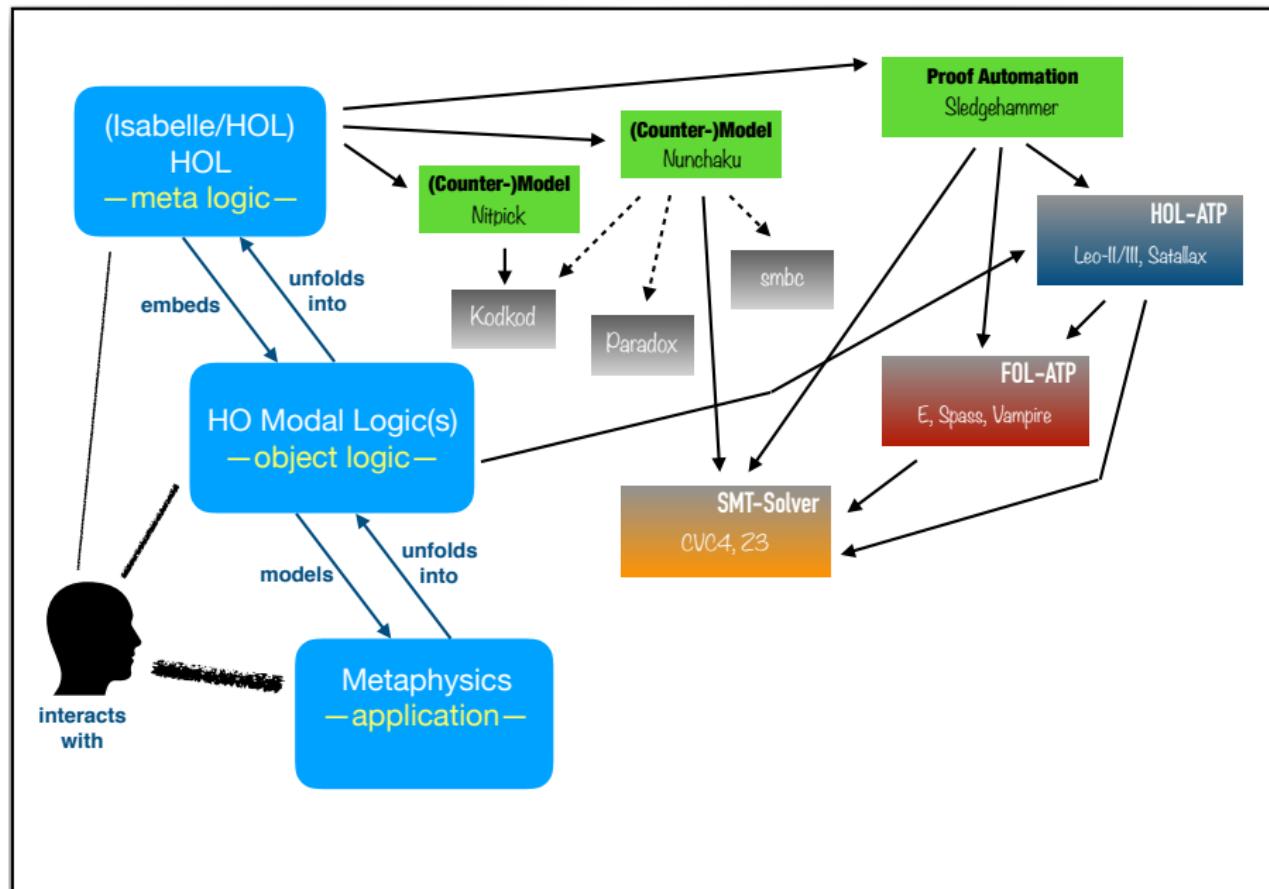
Isabelle is distributed for free under a conglomeration of open-source licenses, but the main code-base is subject to BSD-style regulations. The application bundles include source and binary packages and documentation, see the detailed [installation instructions](#). A vast collection of Isabelle examples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

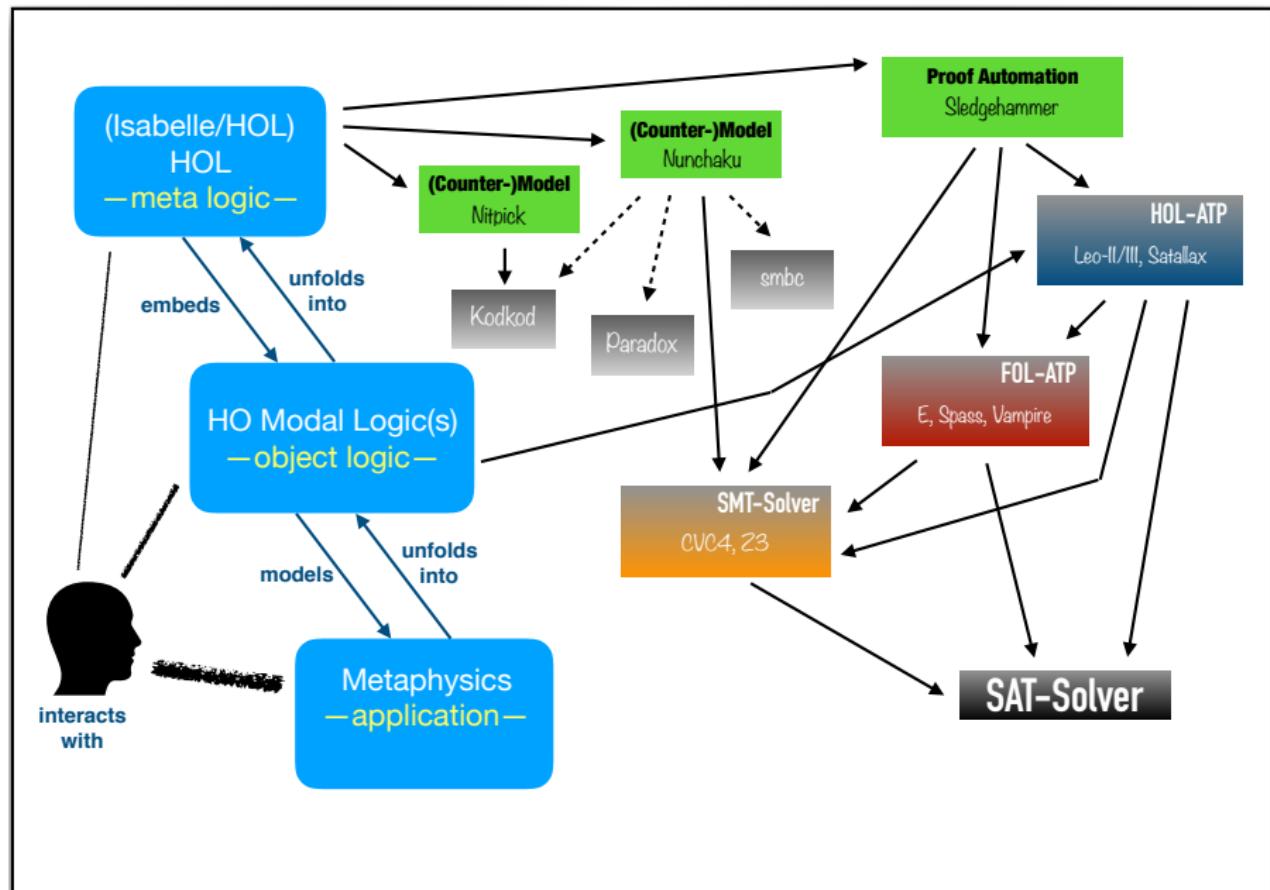


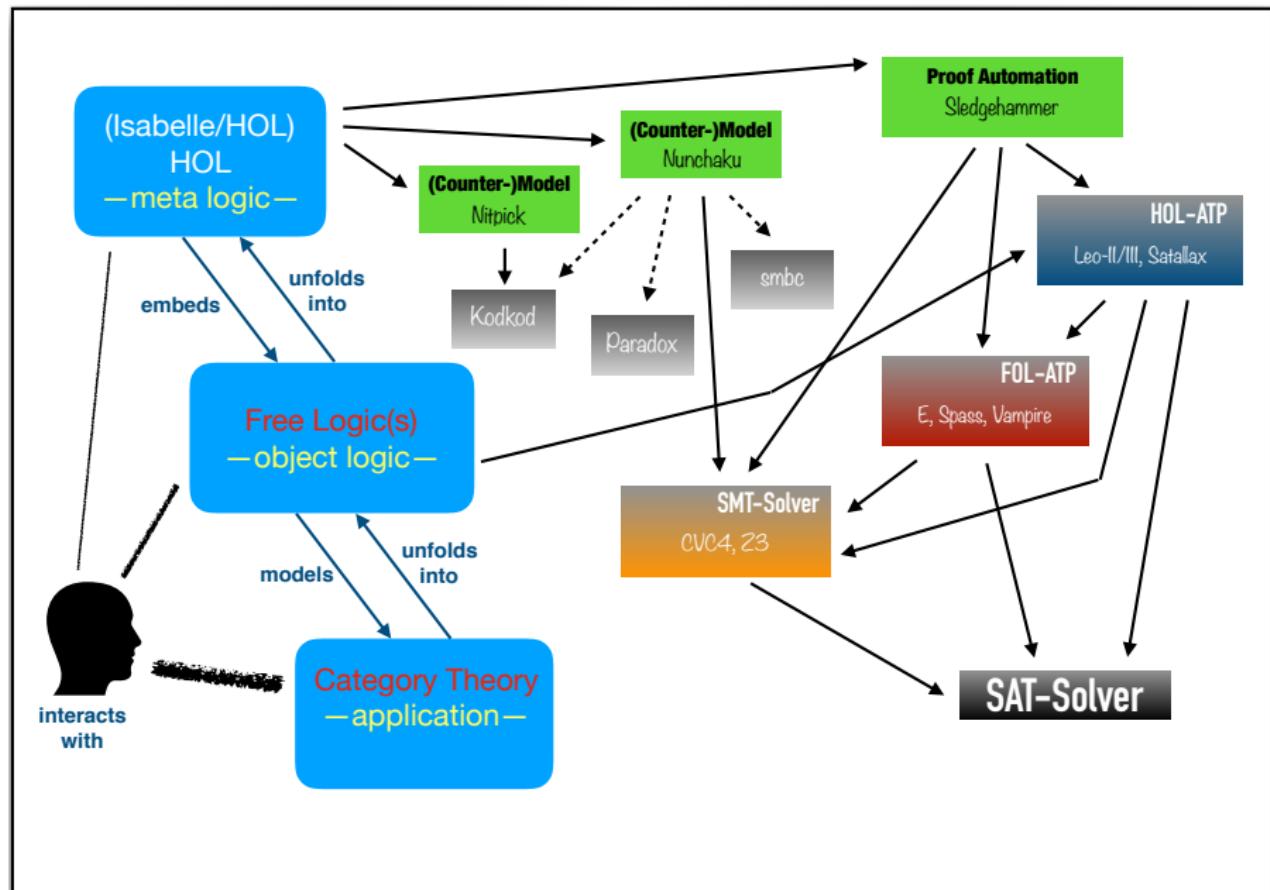


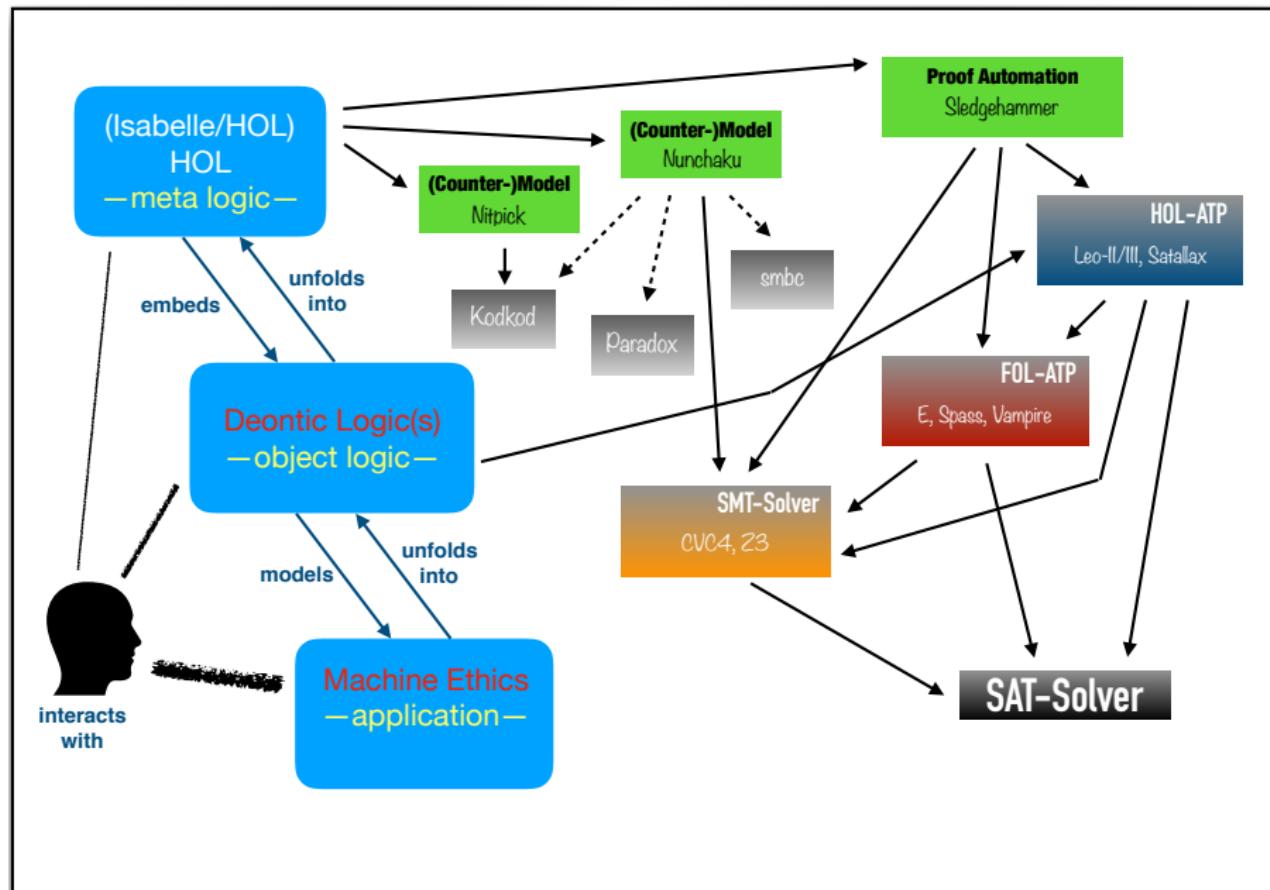


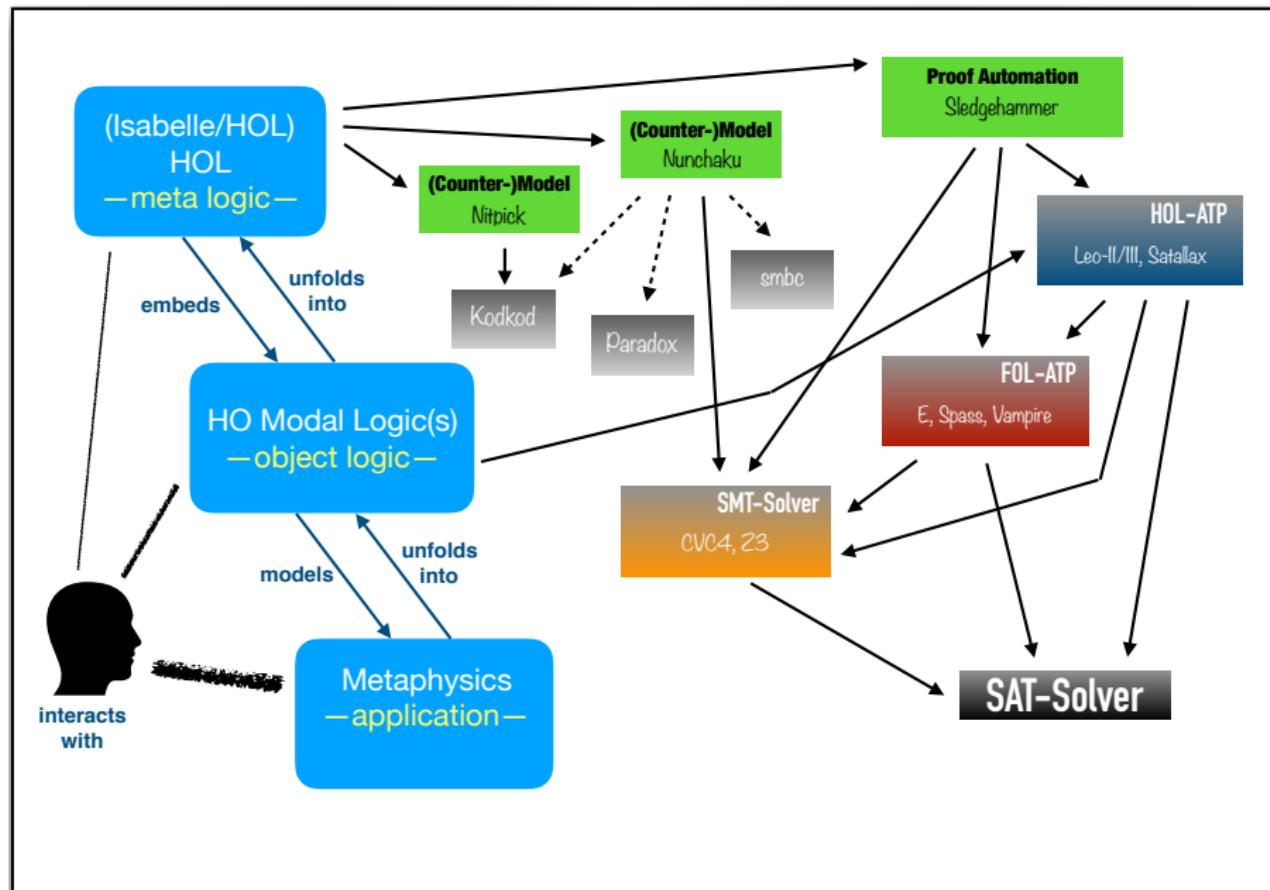














Results of our Experiments (jww B. Woltzenlogel-Paleo)
(see also [Savijnanam 2017], [IJCAI 2016], [ECAI 2014])

Results of our Experiments

Variant of Dana Scott (1972)

- ▶ the premises are **consistent**
- ▶ all argument steps are **logically correct** in (higher-order, extensional) modal logic
 - correct in logic **S5**
 - weaker logic **KB** is already sufficient
 - critique about use of S5 not justified



With our technology it is possible ...
... to verify (selected) masterpiece arguments in philosophy.

Results of our Experiments

Variant of Dana Scott (1972)

- ▶ the premises are **consistent**
- ▶ all argument steps are **logically correct** in (higher-order, extensional) modal logic
 - correct in logic **S5**
 - weaker logic **KB** is already sufficient
 - critique about use of S5 not justified



With our technology it is possible ...
... to verify (selected) masterpiece arguments in philosophy.

Results of our Experiments

Variant of Dana Scott (1972)

- ▶ the premises are **consistent**
- ▶ all argument steps are **logically correct**
in (higher-order, extensional) modal logic
 - correct in logic **S5**
 - weaker logic **KB** is already sufficient
 - critique about use of S5 not justified



With our technology it is possible ...
... to verify (selected) masterpiece arguments in philosophy.

Results of our Experiments

Variant of Dana Scott (1972)

- ▶ the premises are **consistent**
- ▶ all argument steps are **logically correct**
in (higher-order, extensional) modal logic
 - correct in logic **S5**
 - weaker logic **KB** is already sufficient
 - critique about use of S5 not justified



With our technology it is possible ...
... to verify (selected) masterpiece arguments in philosophy.

Results of our Experiments

Variant of Kurt Gödel (1970)

- ▶ the premises are inconsistent/contradictory
(since they imply $\Diamond\Box\perp$)
- ▶ everything follows (ex false quodlibet)!
- ▶ humans had not seen this before
- ▶ ... but my theorem prover LEO-II did



Our technology ...
... can reveal flawed arguments and can even contribute new knowledge.

Results of our Experiments

Variant of Kurt Gödel (1970)

- ▶ the premises are inconsistent/contradictory
(since they imply $\Diamond \Box \perp$)
- ▶ everything follows (ex false quodlibet)!
- ▶ humans had not seen this before
- ▶ ... but my theorem prover LEO-II did



Our technology ...
... can reveal flawed arguments and can even contribute new knowledge.

Results of our Experiments

Variant of Kurt Gödel (1970)

- ▶ the premises are inconsistent/contradictory
(since they imply $\Diamond \Box \perp$)
- ▶ everything follows (ex false quodlibet)!
- ▶ humans had not seen this before
- ▶ ... but my theorem prover LEO-II did



Our technology ...
... can reveal flawed arguments and can even contribute new knowledge.

Results of our Experiments

Variant of Kurt Gödel (1970)

- ▶ the premises are inconsistent/contradictory
(since they imply $\Diamond\Box\perp$)
- ▶ everything follows (ex false quodlibet)!
- ▶ humans had not seen this before
- ▶ ... but my theorem prover LEO-II did



Our technology ...
... can reveal flawed arguments and can even contribute new knowledge.

Results of our Analysis

... we continue with Scott's version

Further corollaries we can prove

- ▶ Monotheism
- ▶ Gott is flawless (has only positive properties)
- ▶ ...
- ▶ Modal Collapse: $\varphi \rightarrow \Box \varphi$

- ▶ there are no contingent truths
- ▶ no alternative worlds
- ▶ everything is determined
- ▶ no free will



Challenge:

Can the Modal Collapse be avoided (with minimal changes)?

Results of our Analysis

... we continue with Scott's version

Further corollaries we can prove

- ▶ Monotheism
- ▶ Gott is flawless (has only positive properties)
- ▶ ...
- ▶ Modal Collapse: $\varphi \rightarrow \Box \varphi$

- ▶ there are no contingent truths
- ▶ no alternative worlds
- ▶ everything is determined
- ▶ no free will



Challenge:

Can the Modal Collapse be avoided (with minimal changes)?

Results of our Analysis

... we continue with Scott's version

Further corollaries we can prove

- ▶ Monotheism
- ▶ Gott is flawless (has only positive properties)
- ▶ ...
- ▶ Modal Collapse: $\varphi \rightarrow \Box \varphi$



- ▶ there are no contingent truths
- ▶ no alternative worlds
- ▶ everything is determined
- ▶ no free will



Challenge: Can the Modal Collapse be avoided (with minimal changes)?



— Can the modal collapse be avoided? —

Remainder of this Talk

We will have a closer look at

- ▶ Gödel/Scott (1972) modal collapse
- ▶ C. Anthony Anderson (1990) avoids modal collapse
- ▶ Melvin Fitting (2002) avoids modal collapse

Questions:

- ▶ How do Anderson and Fitting the avoid modal collapse?
- ▶ Are their solutions related?

To answer this questions we will apply some notions from

- ▶ mathematics: **ultrafilters**
- ▶ philosophy of language: **extension and intension of predicates**

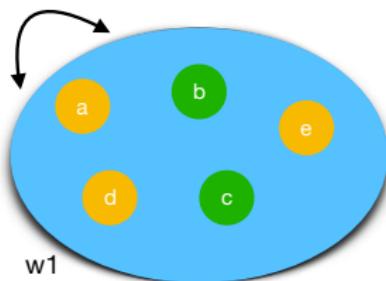


Part B

Some Relevant Pillar Stones for this Talk

Intension vs. Extension of a Predicate (following [Fitting, 2002]))

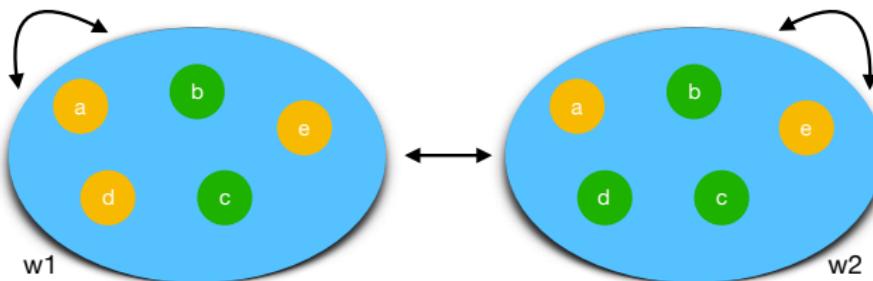
Example predicate: **IsChessGrandmaster**



- Intensional Predicate **IsChessGrandmaster (ICG)**
- Extensions of **ICG** in possible worlds w1-w4:
ICG w1 = {**b,c**}

Intension vs. Extension of a Predicate (following [Fitting, 2002]))

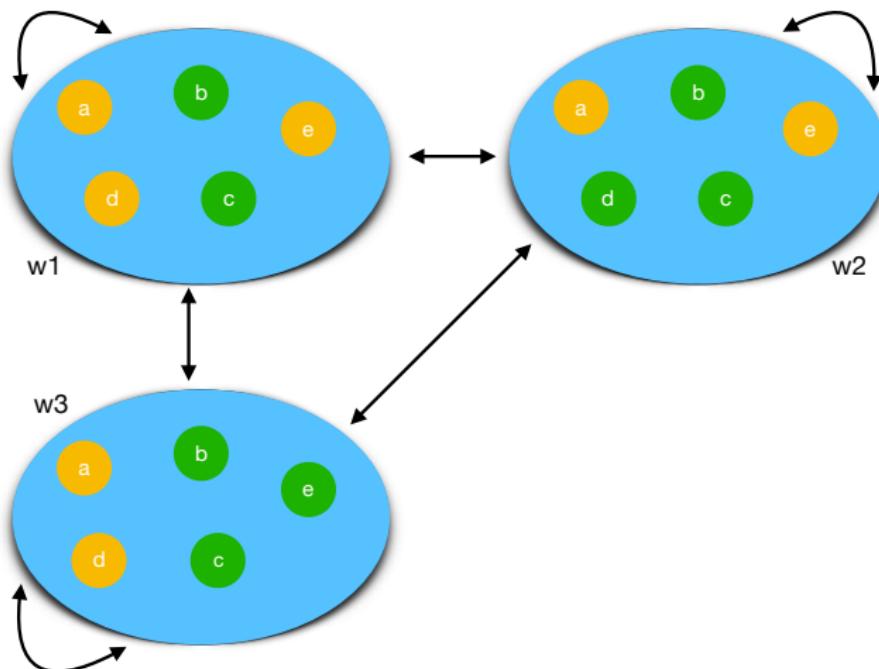
Example predicate: IsChessGrandmaster



- Intensional Predicate **IsChessGrandmaster (ICG)**
- Extensions of **ICG** in possible worlds w_1-w_4 :
 $\text{ICG } w_1 = \{\text{b,c}\}$ $\text{ICG } w_2 = \{\text{b,c,d}\}$

Intension vs. Extension of a Predicate (following [Fitting, 2002]))

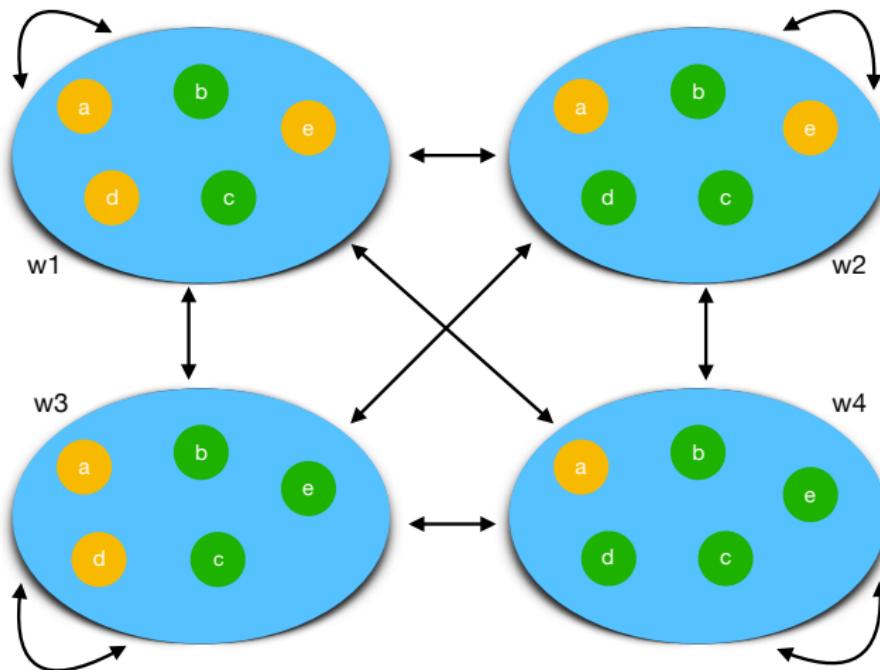
Example predicate: IsChessGrandmaster



- Intensional Predicate **IsChessGrandmaster (ICG)**
- Extensions of **ICG** in possible worlds w1-w4:
 $\text{ICG w1} = \{\text{b,c}\}$ $\text{ICG w2} = \{\text{b,c,d}\}$ $\text{ICG w3} = \{\text{b,c,e}\}$

Intension vs. Extension of a Predicate (following [Fitting, 2002]))

Example predicate: IsChessGrandmaster

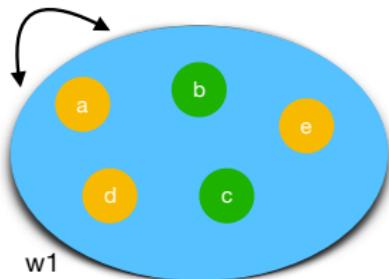


- Intensional Predicate **IsChessGrandmaster (ICG)**
- Extensions of **ICG** in possible worlds w1-w4:
 $\text{ICG } w1 = \{\text{b,c}\}$ $\text{ICG } w2 = \{\text{b,c,d}\}$ $\text{ICG } w3 = \{\text{b,c,e}\}$ $\text{ICG } w4 = \{\text{b,c,d,e}\}$

“Rigidly Intensionalised Extension” of a Predicate

(cf. $@_i\varphi$ from Patrick’s talk)

Example predicate: **IsChessGrandmaster**

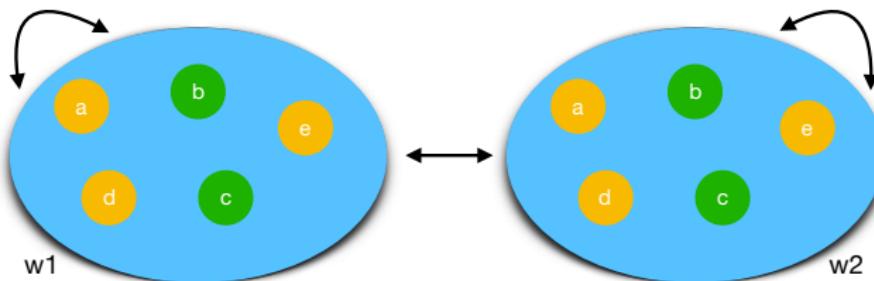


- Intensional Predicate **IsChessGrandmaster (ICG)**
- Rigidified extension of **ICG** in world w_1 :
 $\text{ICG } w_1 = \{b, c\}$

“Rigidly Intensionalised Extension” of a Predicate

(cf. $@_i\varphi$ from Patrick’s talk)

Example predicate: IsChessGrandmaster

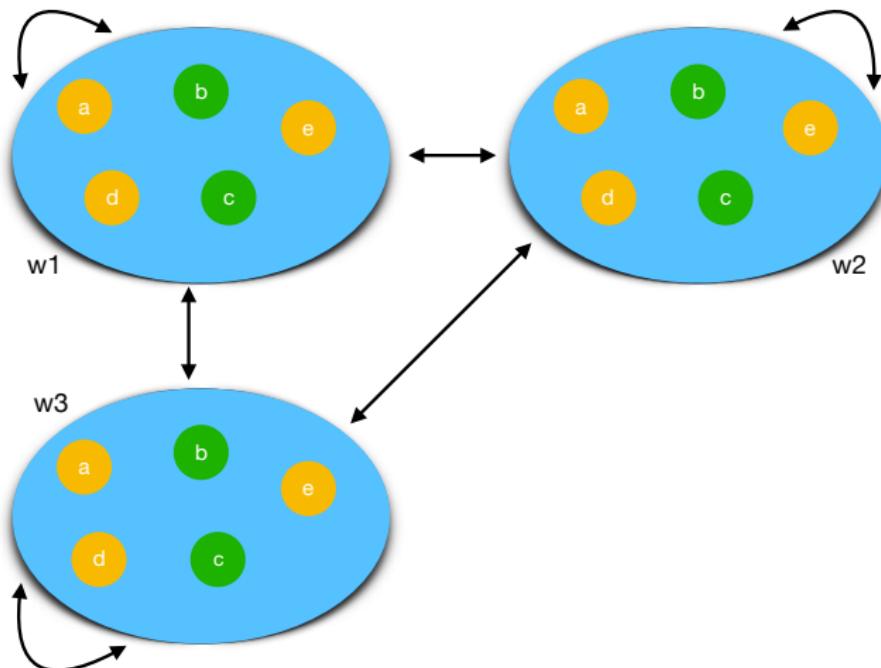


- Intensional Predicate **IsChessGrandmaster (ICG)**
- Rigidified extension of **ICG** in world $w1$:
 $\text{ICG } w1 = \{\mathbf{b}, \mathbf{c}\}$ $\text{ICG } w2 = \{\mathbf{b}, \mathbf{c}\}$

“Rigidly Intensionalised Extension” of a Predicate

(cf. $@_i\varphi$ from Patrick’s talk)

Example predicate: IsChessGrandmaster

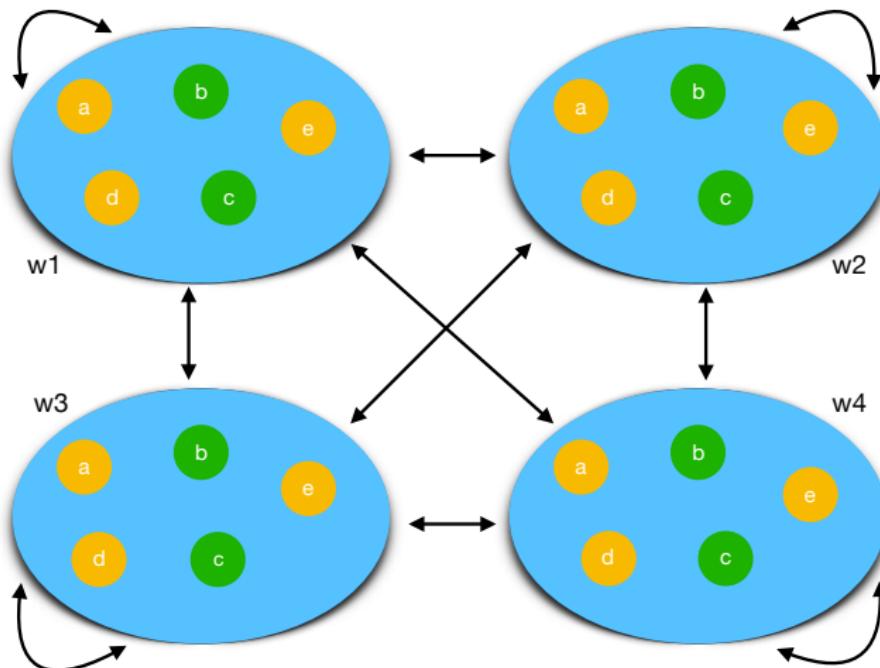


- Intensional Predicate **IsChessGrandmaster (ICG)**
- Rigidified extension of **ICG** in world w1:
 $\text{ICG w1} = \{\mathbf{b}, \mathbf{c}\}$ $\text{ICG w2} = \{\mathbf{b}, \mathbf{c}\}$ $\text{ICG w3} = \{\mathbf{b}, \mathbf{c}\}$

“Rigidly Intensionalised Extension” of a Predicate

(cf. $@_i\varphi$ from Patrick’s talk)

Example predicate: IsChessGrandmaster



- Intensional Predicate **IsChessGrandmaster (ICG)**
- Rigidified extension of **ICG** in world w1:
 $\text{ICG w1} = \{\text{b,c}\}$ $\text{ICG w2} = \{\text{b,c}\}$ $\text{ICG w3} = \{\text{b,c}\}$ $\text{ICG w4} = \{\text{b,c}\}$

Ultrafilter

Definition of Ultrafilter:

Given an arbitrary set X . An ultrafilter U on the powerset $\mathcal{P}(X)$ is a subset of $\mathcal{P}(X)$ such that (where $A, B \in \mathcal{P}(X)$):

1. \emptyset is not an element of U .
2. If A is subset of B and A is element of U , then B is also element of U .
3. If A and B are elements of U , then so is their intersection.
4. Either A or its relative complement $X \setminus A$ is an element of U .

Ultrafilter

Definition of Ultrafilter:

Given an arbitrary set X . An ultrafilter U on the powerset $\mathcal{P}(X)$ is a subset of $\mathcal{P}(X)$ such that (where $A, B \in \mathcal{P}(X)$):

1. \emptyset is not an element of U .
2. If A is subset of B and A is element of U , then B is also element of U .
3. If A and B are elements of U , then so is their intersection.
4. Either A or its relative complement $X \setminus A$ is an element of U .

Example:

$$X = \{1, 2, 3, 4\}$$

Ultrafilter

Definition of Ultrafilter:

Given an arbitrary set X . **An ultrafilter U on the powerset $\mathcal{P}(X)$ is** a subset of $\mathcal{P}(X)$ such that (where $A, B \in \mathcal{P}(X)$):

1. \emptyset is not an element of U .
2. If A is subset of B and A is element of U , then B is also element of U .
3. If A and B are elements of U , then so is their intersection.
4. Either A or its relative complement $X \setminus A$ is an element of U .

Example:

$$X = \{1, 2, 3, 4\}$$

$$\begin{aligned}\mathcal{P}(X) = & \{\emptyset, \{1\}, \{2\}, \{3\}, \{4\}, \{1, 2\}, \{1, 3\}, \{1, 4\}, \{2, 3\}, \{2, 4\}, \{3, 4\}, \\ & \{1, 2, 3\}, \{1, 2, 4\}, \{1, 3, 4\}, \{2, 3, 4\}, \{1, 2, 3, 4\}\}\end{aligned}$$

Ultrafilter

Definition of Ultrafilter:

Given an arbitrary set X . An ultrafilter U on the powerset $\mathcal{P}(X)$ is **a subset of $\mathcal{P}(X)$ such that** (where $A, B \in \mathcal{P}(X)$):

1. \emptyset is not an element of U .
2. If A is subset of B and A is element of U , then B is also element of U .
3. If A and B are elements of U , then so is their intersection.
4. Either A or its relative complement $X \setminus A$ is an element of U .

Example:

$$X = \{1, 2, 3, 4\}$$

$$\begin{aligned}\mathcal{P}(X) = & \{\emptyset, \{1\}, \{2\}, \{3\}, \{4\}, \{1, 2\}, \{1, 3\}, \{1, 4\}, \{2, 3\}, \{2, 4\}, \{3, 4\}, \\ & \{1, 2, 3\}, \{1, 2, 4\}, \{1, 3, 4\}, \{2, 3, 4\}, \{1, 2, 3, 4\}\}\end{aligned}$$

$$U^1 = \{ \quad \{1, 4\}, \quad \}$$

Ultrafilter

Definition of Ultrafilter:

Given an arbitrary set X . An ultrafilter U on the powerset $\mathcal{P}(X)$ is a subset of $\mathcal{P}(X)$ such that (where $A, B \in \mathcal{P}(X)$):

- 1. \emptyset is not an element of U .**
- 2. If A is subset of B and A is element of U , then B is also element of U .**
- 3. If A and B are elements of U , then so is their intersection.**
- 4. Either A or its relative complement $X \setminus A$ is an element of U .**

Example:

$$X = \{1, 2, 3, 4\}$$

$$\begin{aligned}\mathcal{P}(X) = & \{\emptyset, \{1\}, \{2\}, \{3\}, \{4\}, \{1, 2\}, \{1, 3\}, \{1, 4\}, \{2, 3\}, \{2, 4\}, \{3, 4\}, \\ & \{1, 2, 3\}, \{1, 2, 4\}, \{1, 3, 4\}, \{2, 3, 4\}, \{1, 2, 3, 4\}\}\end{aligned}$$

$$U^1 = \{ \quad \{1, 4\}, \quad \}$$

Ultrafilter

Definition of Ultrafilter:

Given an arbitrary set X . An ultrafilter U on the powerset $\mathcal{P}(X)$ is a subset of $\mathcal{P}(X)$ such that (where $A, B \in \mathcal{P}(X)$):

1. \emptyset is not an element of U .
2. **If A is subset of B and A is element of U , then B is also element of U .**
3. If A and B are elements of U , then so is their intersection.
4. Either A or its relative complement $X \setminus A$ is an element of U .

Example:

$$X = \{1, 2, 3, 4\}$$

$$\begin{aligned}\mathcal{P}(X) = & \{\emptyset, \{1\}, \{2\}, \{3\}, \{4\}, \{1, 2\}, \{1, 3\}, \{1, 4\}, \{2, 3\}, \{2, 4\}, \{3, 4\}, \\ & \{1, 2, 3\}, \{1, 2, 4\}, \{1, 3, 4\}, \{2, 3, 4\}, \{1, 2, 3, 4\}\}\end{aligned}$$

$$U^1 = \{ \quad \{1, 4\}, \quad \}$$

Ultrafilter

Definition of Ultrafilter:

Given an arbitrary set X . An ultrafilter U on the powerset $\mathcal{P}(X)$ is a subset of $\mathcal{P}(X)$ such that (where $A, B \in \mathcal{P}(X)$):

1. \emptyset is not an element of U .
2. **If A is subset of B and A is element of U , then B is also element of U .**
3. If A and B are elements of U , then so is their intersection.
4. Either A or its relative complement $X \setminus A$ is an element of U .

Example:

$$X = \{1, 2, 3, 4\}$$

$$\begin{aligned}\mathcal{P}(X) = & \{\emptyset, \{1\}, \{2\}, \{3\}, \{4\}, \{1, 2\}, \{1, 3\}, \{1, 4\}, \{2, 3\}, \{2, 4\}, \{3, 4\}, \\ & \{1, 2, 3\}, \{1, 2, 4\}, \{1, 3, 4\}, \{2, 3, 4\}, \{1, 2, 3, 4\}\}\end{aligned}$$

$$\begin{aligned}U^1 = & \{ \quad \{1, 4\}, \quad \quad \quad \} \\ U^2 = & \{ \quad \{1, 4\}, \quad \{1, 2, 4\}, \{1, 3, 4\}, \{1, 2, 3, 4\} \}\end{aligned}$$

Ultrafilter

Definition of Ultrafilter:

Given an arbitrary set X . An ultrafilter U on the powerset $\mathcal{P}(X)$ is a subset of $\mathcal{P}(X)$ such that (where $A, B \in \mathcal{P}(X)$):

1. \emptyset is not an element of U .
2. If A is subset of B and A is element of U , then B is also element of U .
3. **If A and B are elements of U , then so is their intersection.**
4. Either A or its relative complement $X \setminus A$ is an element of U .

Example:

$$X = \{1, 2, 3, 4\}$$

$$\begin{aligned}\mathcal{P}(X) = & \{\emptyset, \{1\}, \{2\}, \{3\}, \{4\}, \{1, 2\}, \{1, 3\}, \{1, 4\}, \{2, 3\}, \{2, 4\}, \{3, 4\}, \\ & \{1, 2, 3\}, \{1, 2, 4\}, \{1, 3, 4\}, \{2, 3, 4\}, \{1, 2, 3, 4\}\}\end{aligned}$$

$$\begin{aligned}U^1 = & \{ \quad \{1, 4\}, \quad \quad \quad \} \\ U^2 = & \{ \quad \{1, 4\}, \quad \{1, 2, 4\}, \{1, 3, 4\}, \{1, 2, 3, 4\} \}\end{aligned}$$

Ultrafilter

Definition of Ultrafilter:

Given an arbitrary set X . An ultrafilter U on the powerset $\mathcal{P}(X)$ is a subset of $\mathcal{P}(X)$ such that (where $A, B \in \mathcal{P}(X)$):

1. \emptyset is not an element of U .
2. If A is subset of B and A is element of U , then B is also element of U .
3. If A and B are elements of U , then so is their intersection.
4. **Either A or its relative complement $X \setminus A$ is an element of U .**

Example:

$$X = \{1, 2, 3, 4\}$$

$$\begin{aligned}\mathcal{P}(X) = & \{\emptyset, \{1\}, \{2\}, \{3\}, \{4\}, \{1, 2\}, \{1, 3\}, \{1, 4\}, \{2, 3\}, \{2, 4\}, \{3, 4\}, \\ & \{1, 2, 3\}, \{1, 2, 4\}, \{1, 3, 4\}, \{2, 3, 4\}, \{1, 2, 3, 4\}\}\end{aligned}$$

$$\begin{aligned}U^1 = & \{ \quad \{1, 4\}, \quad \quad \quad \} \\ U^2 = & \{ \quad \{1, 4\}, \quad \{1, 2, 4\}, \{1, 3, 4\}, \{1, 2, 3, 4\} \}\end{aligned}$$

Ultrafilter

Definition of Ultrafilter:

Given an arbitrary set X . An ultrafilter U on the powerset $\mathcal{P}(X)$ is a subset of $\mathcal{P}(X)$ such that (where $A, B \in \mathcal{P}(X)$):

1. \emptyset is not an element of U .
2. If A is subset of B and A is element of U , then B is also element of U .
3. If A and B are elements of U , then so is their intersection.
4. **Either A or its relative complement $X \setminus A$ is an element of U .**

Example:

$$X = \{1, 2, 3, 4\}$$

$$\begin{aligned}\mathcal{P}(X) = & \{\emptyset, \{1\}, \{2\}, \{3\}, \{4\}, \{1, 2\}, \{1, 3\}, \{1, 4\}, \{2, 3\}, \{2, 4\}, \{3, 4\}, \\ & \{1, 2, 3\}, \{1, 2, 4\}, \{1, 3, 4\}, \{2, 3, 4\}, \{1, 2, 3, 4\}\}\end{aligned}$$

$$U^1 = \{ \quad \quad \quad \{1, 4\}, \quad \quad \quad \}$$

$$U^2 = \{ \quad \quad \quad \{1, 4\}, \quad \quad \quad \{1, 2, 4\}, \{1, 3, 4\}, \{1, 2, 3, 4\} \}$$

$$U^3 = \{\{1\}, \quad \quad \quad \{1, 4\}, \quad \quad \quad \{1, 2, 4\}, \{1, 3, 4\}, \{1, 2, 3, 4\}\}$$

Ultrafilter

Definition of Ultrafilter:

Given an arbitrary set X . An ultrafilter U on the powerset $\mathcal{P}(X)$ is a subset of $\mathcal{P}(X)$ such that (where $A, B \in \mathcal{P}(X)$):

- 1. \emptyset is not an element of U .**
- 2. If A is subset of B and A is element of U , then B is also element of U .**
- 3. If A and B are elements of U , then so is their intersection.**
- 4. Either A or its relative complement $X \setminus A$ is an element of U .**

Example:

$$X = \{1, 2, 3, 4\}$$

$$\begin{aligned}\mathcal{P}(X) = & \{\emptyset, \{1\}, \{2\}, \{3\}, \{4\}, \{1, 2\}, \{1, 3\}, \{1, 4\}, \{2, 3\}, \{2, 4\}, \{3, 4\}, \\ & \{1, 2, 3\}, \{1, 2, 4\}, \{1, 3, 4\}, \{2, 3, 4\}, \{1, 2, 3, 4\}\}\end{aligned}$$

$$U^1 = \{ \quad \{1, 4\}, \quad \quad \quad \}$$

$$U^2 = \{ \quad \{1, 4\}, \quad \{1, 2, 4\}, \{1, 3, 4\}, \{1, 2, 3, 4\} \}$$

$$U^3 = \{\{1\}, \quad \{1, 4\}, \quad \{1, 2, 4\}, \{1, 3, 4\}, \{1, 2, 3, 4\}\}$$

Ultrafilter

Definition of Ultrafilter:

Given an arbitrary set X . An ultrafilter U on the powerset $\mathcal{P}(X)$ is a subset of $\mathcal{P}(X)$ such that (where $A, B \in \mathcal{P}(X)$):

1. \emptyset is not an element of U .
2. If A is subset of B and A is element of U , then B is also element of U .
3. If A and B are elements of U , then so is their intersection.
4. Either A or its relative complement $X \setminus A$ is an element of U .

Example:

$$X = \{1, 2, 3, 4\}$$

$$\begin{aligned}\mathcal{P}(X) = & \{\emptyset, \{1\}, \{2\}, \{3\}, \{4\}, \{1, 2\}, \{1, 3\}, \{1, 4\}, \{2, 3\}, \{2, 4\}, \{3, 4\}, \\ & \{1, 2, 3\}, \{1, 2, 4\}, \{1, 3, 4\}, \{2, 3, 4\}, \{1, 2, 3, 4\}\}\end{aligned}$$

$$U^1 = \{ \quad \quad \quad \{1, 4\}, \quad \quad \quad \}$$

$$U^2 = \{ \quad \quad \quad \{1, 4\}, \quad \quad \quad \{1, 2, 4\}, \{1, 3, 4\}, \{1, 2, 3, 4\} \}$$

$$U^3 = \{\{1\}, \quad \quad \quad \{1, 4\}, \quad \quad \quad \{1, 2, 4\}, \{1, 3, 4\}, \{1, 2, 3, 4\}\}$$

Ultrafilter

Definition of Ultrafilter:

Given an arbitrary set X . An ultrafilter U on the powerset $\mathcal{P}(X)$ is a subset of $\mathcal{P}(X)$ such that (where $A, B \in \mathcal{P}(X)$):

1. \emptyset is not an element of U .
2. If A is subset of B and A is element of U , then B is also element of U .
3. If A and B are elements of U , then so is their intersection.
4. Either A or its relative complement $X \setminus A$ is an element of U .

Example:

$$X = \{1, 2, 3, 4\}$$

$$\begin{aligned}\mathcal{P}(X) = & \{\emptyset, \{1\}, \{2\}, \{3\}, \{4\}, \{1, 2\}, \{1, 3\}, \{1, 4\}, \{2, 3\}, \{2, 4\}, \{3, 4\}, \\ & \{1, 2, 3\}, \{1, 2, 4\}, \{1, 3, 4\}, \{2, 3, 4\}, \{1, 2, 3, 4\}\}\end{aligned}$$

$$U^1 = \{ \quad \{1, 4\}, \quad \quad \quad \}$$

$$U^2 = \{ \quad \{1, 4\}, \quad \{1, 2, 4\}, \{1, 3, 4\}, \{1, 2, 3, 4\} \}$$

$$U^3 = \{\{1\}, \quad \{1, 4\}, \quad \{1, 2, 4\}, \{1, 3, 4\}, \{1, 2, 3, 4\}\}$$

$$U^4 = \{\{1\}, \{1, 2\}, \{1, 3\}, \{1, 4\}, \{1, 2, 3\}, \{1, 2, 4\}, \{1, 3, 4\}, \{1, 2, 3, 4\}\}$$

Ultrafilter

Definition of Ultrafilter:

Given an arbitrary set X . An ultrafilter U on the powerset $\mathcal{P}(X)$ is a subset of $\mathcal{P}(X)$ such that (where $A, B \in \mathcal{P}(X)$):

1. \emptyset is not an element of U .
2. If A is subset of B and A is element of U , then B is also element of U .
3. **If A and B are elements of U , then so is their intersection.**
4. Either A or its relative complement $X \setminus A$ is an element of U .

Example:

$$X = \{1, 2, 3, 4\}$$

$$\begin{aligned}\mathcal{P}(X) = & \{\emptyset, \{1\}, \{2\}, \{3\}, \{4\}, \{1, 2\}, \{1, 3\}, \{1, 4\}, \{2, 3\}, \{2, 4\}, \{3, 4\}, \\ & \{1, 2, 3\}, \{1, 2, 4\}, \{1, 3, 4\}, \{2, 3, 4\}, \{1, 2, 3, 4\}\}\end{aligned}$$

$$U^1 = \{ \quad \{1, 4\}, \quad \quad \quad \}$$

$$U^2 = \{ \quad \{1, 4\}, \quad \{1, 2, 4\}, \{1, 3, 4\}, \{1, 2, 3, 4\} \}$$

$$U^3 = \{\{1\}, \quad \{1, 4\}, \quad \{1, 2, 4\}, \{1, 3, 4\}, \{1, 2, 3, 4\}\}$$

$$U^4 = \{\{1\}, \{1, 2\}, \{1, 3\}, \{1, 4\}, \{1, 2, 3\}, \{1, 2, 4\}, \{1, 3, 4\}, \{1, 2, 3, 4\}\}$$

Ultrafilter

Definition of Ultrafilter:

Given an arbitrary set X . An ultrafilter U on the powerset $\mathcal{P}(X)$ is a subset of $\mathcal{P}(X)$ such that (where $A, B \in \mathcal{P}(X)$):

1. \emptyset is not an element of U .
2. If A is subset of B and A is element of U , then B is also element of U .
3. **If A and B are elements of U , then so is their intersection.**
4. Either A or its relative complement $X \setminus A$ is an element of U .

Example:

$$X = \{1, 2, 3, 4\}$$

$$\begin{aligned}\mathcal{P}(X) = & \{\emptyset, \{1\}, \{2\}, \{3\}, \{4\}, \{1, 2\}, \{1, 3\}, \{1, 4\}, \{2, 3\}, \{2, 4\}, \{3, 4\}, \\ & \{1, 2, 3\}, \{1, 2, 4\}, \{1, 3, 4\}, \{2, 3, 4\}, \{1, 2, 3, 4\}\}\end{aligned}$$

$$U^1 = \{ \quad \quad \quad \{1, 4\}, \quad \quad \quad \}$$

$$U^2 = \{ \quad \quad \quad \{1, 4\}, \quad \quad \quad \{1, 2, 4\}, \{1, 3, 4\}, \{1, 2, 3, 4\} \}$$

$$U^3 = \{ \{1\}, \quad \quad \quad \{1, 4\}, \quad \quad \quad \{1, 2, 4\}, \{1, 3, 4\}, \{1, 2, 3, 4\} \}$$

$$U^4 = \{ \{1\}, \{1, 2\}, \{1, 3\}, \{1, 4\}, \{1, 2, 3\}, \{1, 2, 4\}, \{1, 3, 4\}, \{1, 2, 3, 4\} \}$$

Ultrafilter

Definition of Ultrafilter:

Given an arbitrary set X . An ultrafilter U on the powerset $\mathcal{P}(X)$ is a subset of $\mathcal{P}(X)$ such that (where $A, B \in \mathcal{P}(X)$):

1. \emptyset is not an element of U .
2. If A is subset of B and A is element of U , then B is also element of U .
3. If A and B are elements of U , then so is their intersection.
4. **Either A or its relative complement $X \setminus A$ is an element of U .**

Example:

$$X = \{1, 2, 3, 4\}$$

$$\begin{aligned}\mathcal{P}(X) = & \{\emptyset, \{1\}, \{2\}, \{3\}, \{4\}, \{1, 2\}, \{1, 3\}, \{1, 4\}, \{2, 3\}, \{2, 4\}, \{3, 4\}, \\ & \{1, 2, 3\}, \{1, 2, 4\}, \{1, 3, 4\}, \{2, 3, 4\}, \{1, 2, 3, 4\}\}\end{aligned}$$

$$U^1 = \{ \quad \{1, 4\}, \quad \quad \quad \}$$

$$U^2 = \{ \quad \{1, 4\}, \quad \{1, 2, 4\}, \{1, 3, 4\}, \{1, 2, 3, 4\} \}$$

$$U^3 = \{\{1\}, \quad \{1, 4\}, \quad \{1, 2, 4\}, \{1, 3, 4\}, \{1, 2, 3, 4\}\}$$

$$U^4 = \{\{1\}, \{1, 2\}, \{1, 3\}, \{1, 4\}, \{1, 2, 3\}, \{1, 2, 4\}, \{1, 3, 4\}, \{1, 2, 3, 4\}\}$$

Ultrafilter

Definition of Ultrafilter:

Given an arbitrary set X . An ultrafilter U on the powerset $\mathcal{P}(X)$ is a subset of $\mathcal{P}(X)$ such that (where $A, B \in \mathcal{P}(X)$):

- 1. \emptyset is not an element of U .**
- 2. If A is subset of B and A is element of U , then B is also element of U .**
- 3. If A and B are elements of U , then so is their intersection.**
- 4. Either A or its relative complement $X \setminus A$ is an element of U .**

Example:

$$X = \{1, 2, 3, 4\}$$

$$\begin{aligned}\mathcal{P}(X) = & \{\emptyset, \{1\}, \{2\}, \{3\}, \{4\}, \{1, 2\}, \{1, 3\}, \{1, 4\}, \{2, 3\}, \{2, 4\}, \{3, 4\}, \\ & \{1, 2, 3\}, \{1, 2, 4\}, \{1, 3, 4\}, \{2, 3, 4\}, \{1, 2, 3, 4\}\}\end{aligned}$$

$$U^1 = \{ \quad \{1, 4\}, \quad \quad \quad \}$$

$$U^2 = \{ \quad \{1, 4\}, \quad \{1, 2, 4\}, \{1, 3, 4\}, \{1, 2, 3, 4\} \}$$

$$U^3 = \{ \{1\}, \quad \{1, 4\}, \quad \{1, 2, 4\}, \{1, 3, 4\}, \{1, 2, 3, 4\} \}$$

$$U^4 = \{ \{1\}, \{1, 2\}, \{1, 3\}, \{1, 4\}, \{1, 2, 3\}, \{1, 2, 4\}, \{1, 3, 4\}, \{1, 2, 3, 4\} \}$$

Ultrafilter

Definition of Ultrafilter:

Given an arbitrary set X . An ultrafilter U on the powerset $\mathcal{P}(X)$ is a subset of $\mathcal{P}(X)$ such that (where $A, B \in \mathcal{P}(X)$):

1. **\emptyset is not an element of U .**
2. **If A is subset of B and A is element of U , then B is also element of U .**
3. **If A and B are elements of U , then so is their intersection.**
4. **Either A or its relative complement $X \setminus A$ is an element of U .**

Example:

$$X = \{1, 2, 3, 4\}$$

$$\begin{aligned}\mathcal{P}(X) = & \{\emptyset, \{1\}, \{2\}, \{3\}, \{4\}, \{1, 2\}, \{1, 3\}, \{1, 4\}, \{2, 3\}, \{2, 4\}, \{3, 4\}, \\ & \{1, 2, 3\}, \{1, 2, 4\}, \{1, 3, 4\}, \{2, 3, 4\}, \{1, 2, 3, 4\}\}\end{aligned}$$

$$U^1 = \{ \quad \{1, 4\}, \quad \quad \quad \}$$

$$U^2 = \{ \quad \{1, 4\}, \quad \{1, 2, 4\}, \{1, 3, 4\}, \{1, 2, 3, 4\} \}$$

$$U^3 = \{\{1\}, \quad \{1, 4\}, \quad \{1, 2, 4\}, \{1, 3, 4\}, \{1, 2, 3, 4\}\}$$

$$U^4 = \{\{1\}, \{1, 2\}, \{1, 3\}, \{1, 4\}, \{1, 2, 3\}, \{1, 2, 4\}, \{1, 3, 4\}, \{1, 2, 3, 4\}\} = \mathbf{U}$$

Ultrafilter

Definition of Ultrafilter:

Given an arbitrary set X . An ultrafilter U on the powerset $\mathcal{P}(X)$ is a subset of $\mathcal{P}(X)$ such that (where $A, B \in \mathcal{P}(X)$):

1. \emptyset is not an element of U .
2. If A is subset of B and A is element of U , then B is also element of U .
3. If A and B are elements of U , then so is their intersection.
4. Either A or its relative complement $X \setminus A$ is an element of U .

Example:

$$X = \{1, 2, 3, 4\}$$

$$\mathcal{P}(X) = \{\emptyset, \{1\}, \{2\}, \{3\}, \{4\}, \{1, 2\}, \{1, 3\}, \{1, 4\}, \{2, 3\}, \{2, 4\}, \{3, 4\}, \\ \{1, 2, 3\}, \{1, 2, 4\}, \{1, 3, 4\}, \{2, 3, 4\}, \{1, 2, 3, 4\}\}$$

$$U^1 = \{ \quad \quad \quad \{1, 4\}, \quad \quad \quad \}$$

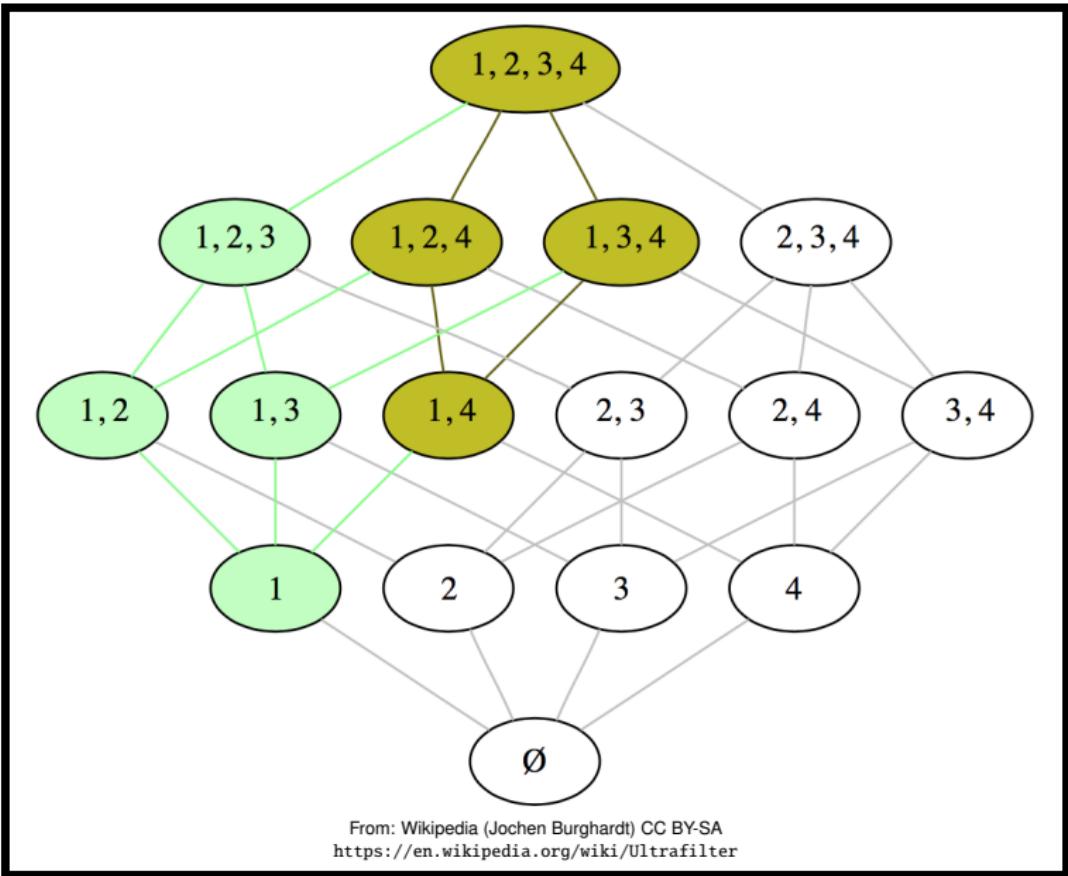
$$U^2 = \{ \quad \quad \quad \{1, 4\}, \quad \quad \quad \{1, 2, 4\}, \{1, 3, 4\}, \{1, 2, 3, 4\} \}$$

$$U^3 = \{\{1\}, \quad \quad \quad \{1, 4\}, \quad \quad \quad \{1, 2, 4\}, \{1, 3, 4\}, \{1, 2, 3, 4\}\}$$

$$U^4 = \{\{1\}, \{1, 2\}, \{1, 3\}, \{1, 4\}, \{1, 2, 3\}, \{1, 2, 4\}, \{1, 3, 4\}, \{1, 2, 3, 4\}\} = \mathbf{U}$$

1 is element of all sets in U (**1** has all properties of U)

Ultrafilter





Part C
— Comparative Analysis —
Variants of Gödel/Scott, Anderson and Fitting

Ontological Argument: Variant by Gödel/Scott

Part I - Proving that God's existence is possible

D1 Being Godlike is equivalent to having all positive properties.

A1 Exactly one of a property or its negation is positive.

A2 Any property entailed by a positive property is positive.

A3 The combination of any collection of positive properties is itself positive.

From A1 and A2 follows theorem T1:

T1 Every positive property is possibly instantiated.

From D1 and A3 follows:

T2 Being Godlike is a positive property.

From T1 and T2 follows:

T3 Being Godlike is possibly instantiated.

Ontological Argument: Variant by Gödel/Scott

Part I - Proving that God's existence is possible

D1 Part II - Proving that God's existence is necessary, if possible

A1 D2 A property E is the essence of an individual x iff x has E and all of x's properties are nec. entailed by E.^a

A3 A4 Positive properties are necessarily positive.

From A1 and A4 (using definitions D1 and D2) follows:

T1 T4 Being Godlike is an essential property of any Godlike individual.

From T2 and D3 (using D1, D2) follows:

D3 Necessary existence of an individual is the necessary instantiation of all its essences.

From T3 and A5 (using D1, D2, D3) follows:

A5 Necessary existence is a positive property.

T5 Being Godlike, if instantiated, is necessarily instantiated.

And finally from T3, T5 (together with some implicit modal axioms, e.g. S5) the existence of (at least a) God follows:

T6 Being Godlike is necessarily instantiated.

^aThe underlined part in definition D2 has been added by Scott. Gödel originally omitted this part.

Ontological Argument: Variant by Gödel/Scott

Part I - Proving that God's existence is possible

D1 Part II - Proving that God's existence is necessary, if possible

A1 D2 A property F is the essence of an individual x iff x has F and all of x 's

A2 p "Modal Collapse" is implied by these axioms: $\varphi \supset \Box\varphi$

A3 A4 P ▶ determinism

From T1 From "positive properties (\mathcal{P})" applied here to (non-rigid) intensions of properties. We can prove:

T2 T4 E From D3 M ▶ \mathcal{P} is an ultrafilter all

T2 T4 E From A5 M Let \mathcal{P}' be the set of "rigidly intensionalised extensions" of positive properties. $(\mathcal{P}'\varphi := \mathcal{P}(\downarrow\varphi))$

T3 T5 E From And e.g. S We can prove: ms,

And ▶ \mathcal{P}' is an ultrafilter

e.g. S ▶ $\mathcal{P} = \mathcal{P}'$

T6 Being Godlike is necessarily instantiated.

^aThe underlined part in definition D2 has been added by Scott. Gödel originally omitted this part.

Ontological Argument: Variant by Gödel/Scott

```
1 theory GoedelProof imports IHOML      (* This formalization follows Fitting's textbook *)
2 begin
3 (*Positiveness/perfection: uninterpreted constant symbol*)
4 consts positiveProperty::"(e⇒i⇒bool)⇒i⇒bool" ("P")
5 (*Some auxiliary definitions needed to formalise A3*)
6 definition h1 ("pos")    where "pos Z ≡ ∀X. Z X → P X"
7 definition h2 (infix "⊓" 60) where "X ⊓ Z ≡ □(∀x.(X x ↔ (YY.(Z Y) → (Y x))))"
8 definition h3 (infix "⇒" 60) where "X ⇒ Y ≡ □(∀EZ. X z → Y z)"
9
10 (**Part I**)
11 (*D1*) definition G ("G") where "G ≡ (λx. ∀Y. P Y → Y x)"
12 (*A1*) axiomatization where Ala: "[∀X. P (¬X) → ¬(P X)]" and Alb:"[∀X. ¬(P X) → P (¬X)]"
13 (*A2*) axiomatization where A2: "[∀X Y. (P X ∧ (X ⇒ Y)) → P Y]"
14 (*A3*) axiomatization where A3: "[∀Z X. (pos Z ∧ X ⊓ Z) → P X]"
15 (*T1*) theorem T1: "[∀X. P X → ◊∃E X]" by (metis Ala A2 h3_def)
16 (*T2*) theorem T2: "[P G]" proof -
17   {have 1: "∀w. ∃Z X. (P G ∨ pos Z ∧ X ⊓ Z ∧ ¬P X) w" by (metis(full_types) G_def h1_def h2_def)
18   have 2: "[∀Z X. (pos Z ∧ X ⊓ Z) → P X] → [P G]" using 1 by auto}
19   thus ?thesis using A3 by blast qed
20 (*T3*) theorem T3: "[◊∃E G]" sledgehammer using T1 T2 by simp
21
```

Ontological Argument: Variant by Gödel/Scott

```
21 (**Part II**)
22 (*Logic KB*) axiomatization where symm: "symmetric aRel"
23 (*A4*) axiomatization where A4: "[ $\forall x. P x \rightarrow \square(P x)$ ]"
24 (*D2*) definition ess ("E") where " $E Y x \equiv (Y x) \wedge (\forall z. z x \rightarrow Y \Rightarrow z)$ "
25 (*T4*) theorem T4: "[ $\forall x. G x \rightarrow (E G x)$ ]" by (metis Alb A4 G_def h3_def ess_def)
26 (*D3*) definition NE ("NE") where " $NE x \equiv (\lambda w. (\forall y. E y x \rightarrow \square^E y) w)$ "
27 (*A5*) axiomatization where A5: "[ $P NE$ ]"
28 (*T5*) theorem T5: "[ $(\Diamond \exists^E G) \rightarrow \square \exists^E G$ ]" by (metis A5 G_def NE_def T4 symm)
29 (*T6*) theorem T6: "[ $\square \exists^E G$ ]" using T3 T5 by blast
30
31
32 (**Consistency**)
33 lemma True nitpick[satisfy] oops (*Model found by Nitpick: the axioms are consistent*)
34
35 (**Modal Collapse**)
36 lemma ModalCollapse: "[ $\forall \Phi. (\Phi \rightarrow (\square \Phi))$ ]" proof -
37   {fix w fix Q
38     have " $\forall x. G x w \longrightarrow (\forall z. z x \rightarrow \square(\forall e. G z \rightarrow z z)) w$ " by (metis Alb A4 G_def)
39     hence 1: " $(\exists x. G x w) \longrightarrow ((Q \rightarrow \square(\forall e. G z \rightarrow Q)) w)$ " by force
40     have " $\exists x. G x w$ " using T3 T6 symm by blast
41     hence " $(Q \rightarrow \square Q) w$ " using 1 T6 by blast
42   } thus ?thesis by auto qed
43
44 (**Some Corollaries**)
45 (*C1*) theorem C1: "[ $\forall E P x. ((E E x) \wedge (P x)) \rightarrow (E \Rightarrow P)$ ]" by (metis ess_def)
46 (*C2*) theorem C2: "[ $\forall X. \neg P X \rightarrow \square(\neg P X)$ ]" using A4 symm by blast
47   definition h4 ("N") where " $N X \equiv \neg P X$ "
48 (*C3*) theorem C3: "[ $\forall X. N X \rightarrow \square(N X)$ ]" by (simp add: C2 h4_def)
```

Ontological Argument: Variant by Gödel/Scott

```
49 (**Positive Properties and Ultrafilters**)
50 abbreviation emptySet ("∅") where "∅ ≡ λx w. False"
51 abbreviation entails (infixr "C" 51) where "φ C ψ ≡ ∀x w. φ x w → ψ x w"
52 abbreviation andPred (infixr "Π" 51) where "φ Π ψ ≡ λx w. φ x w ∧ ψ x w"
53 abbreviation negpred ("¬" [52]53) where "¬ψ ≡ λx w. ¬ψ x w"
54 abbreviation "ultrafilter Φ cw ≡
55   ¬(Φ ∅ cw)
56   ∧ (∀φ. ∀ψ. (Φ φ cw ∧ Φ ψ cw) → (Φ (φ Π ψ) cw))
57   ∧ (∀φ::e⇒i=bool. ∀ψ::e⇒i=bool. (Φ φ cw ∨ Φ (¬φ) cw) ∧ ¬(Φ φ cw ∧ Φ (¬φ) cw))
58   ∧ (∀φ::e⇒i=bool. ∀ψ::e⇒i=bool. (Φ φ cw ∧ φ ⊆ ψ) → Φ ψ cw)"
59 lemma helpA: "∀w. ¬(P ∅ w)" using T1 by auto
60 lemma helpB: "∀φ ψ w. (P φ w ∧ P ψ w) → (P (φ Π ψ) w)" by (smt Alb G_def T3 T6 symm)
61 lemma helpC: "∀φ ψ w. (P φ w ∨ P (¬φ) w) ∧ ¬(P φ w ∧ P (¬φ) w)" using Ala Alb by blast
62 lemma helpD: "∀φ ψ w. (P φ w ∧ (φ ⊆ ψ)) → P ψ w" by (metis Alb A4 G_def T1 T6)
63
64 (*U1*) theorem U1: "∀w. ultrafilter P w" using helpA helpB helpC helpD by simp
65
66 (*(φ) converts an extensional object φ into 'rigid' intensional one*)
67 abbreviation trivialConversion ("(L)") where "(φ) ≡ (λw. φ)"
68
69 (*Q ↓φ: the extension of a (possibly) non-rigid predicate φ is turned into a rigid intensional one,
70 then Q is applied to the latter; ↓φ can be read as "the rigidly intensionalised predicate φ"*)
71 abbreviation mextPredArg (infix "↓" 60) where "Q ↓φ ≡ λw. Q (λx. (φ x w)) w"
72 lemma "∀Q φ. Q φ = Q ↓φ" nitpick oops (*countermodel: the two notions are not the same*)
73
74 lemma helpE: "∀w. ¬((P ↓∅) w)" using T1 by blast
75 lemma helpF: "∀φ ψ w. ((P ↓φ) w ∧ (P ↓ψ) w) → ((P ↓(φ Π ψ)) w)" by (smt Alb C2 G_def T3 symm)
76 lemma helpG: "∀w. ((P ↓φ) w ∨ (P ↓(¬φ)) w) ∧ ¬((P ↓φ) w ∧ (P ↓(¬φ)) w)" using Ala Alb by blast
77 lemma helpH: "∀w. ((P ↓φ) w ∧ φ ⊆ ψ) → (P ↓ψ) w" by (metis Alb A5 G_def NE_def T3 T4 symm)
78
79 abbreviation "P' φ ≡ (P ↓φ)" (*P': the set of all rigidly intensionalised positive properties*)
80
81 (*U2*) theorem U2: "∀w. ultrafilter P' w" using helpE helpF helpG helpH by simp
82 (*U3*) theorem U3: "(P' ⊆ P) ∧ (P ⊆ P')" by (smt Alb G_def T1 T6 symm) (*P' and P are equal*)
83
```

Ontological Argument: Variant by Gödel/Scott

The screenshot shows the UltrafilterMovie1 interface running a formalization in GoedelProof.thy. The interface has a top bar with tabs for "Documentation", "Sidekick", "State", and "Theories". The main area displays the Isabelle/HOL code for the argument.

```
theory GoedelProof imports IHOML      (* This formalization follows Fitting's textbook *)
begin

(*Positiveness/perfection: uninterpreted constant symbol*)
consts positiveProperty::"(e⇒i⇒bool)⇒i⇒bool" ("P")
(*Some auxiliary definitions needed to formalise A3*)
definition h1 ("pos")    where "pos Z ≡ ∀X. Z X → P X"
definition h2 (infix "∩" 60) where "X ∩ Z ≡ □(∀x. (X x ↔ (¬Y. (Z Y → (Y x)))))"
definition h3 (infix "⇒" 60) where "X ⇒ Y ≡ □(∀z. X z → Y z)"

(**Part I**)
(*D1*) definition G ("G") where "G ≡ (λx. ∀Y. P Y → Y x)"
(*A1*) axiomatization where Ala: "[∀X. P (→X) → ¬(P X)]" and Alb: "[∀X. ¬(P X) → P (→X)]"
(*A2*) axiomatization where A2: "[∀X Y. (P X ∧ (X ⇒ Y)) → P Y]"
(*A3*) axiomatization where A3: "[∀Z X. (pos Z ∧ X ∩ Z) → P X]"
(*T1*) theorem T1: "[∀X. P X → ∃E X]" by (metis Ala A2 h3_def)
(*T2*) theorem T2: "[P G]" proof -
  {have 1: "¬∀w. ∃Z X. (P G ∨ pos Z ∧ X ∩ Z ∧ ¬P X) w" by (metis(full_types) G_def h1_def h2_def)
   have 2: "[∀Z X. (pos Z ∧ X ∩ Z) → P X] → [P G]" using 1 by auto}
  thus ?thesis using A3 by blast qed
(*T3*) theorem T3: "[¬∃E G]" using T1 T2 by simp

(**Part II**)
(*Logic KB*) axiomatization where symm: "symmetric apol"
```

The bottom part of the interface shows a timeline with a play button and a progress bar from 00:00 to -02:26, indicating the video is playing backwards. The status bar at the bottom shows "theorem U3: P' ⊆ P ∧ P ⊆" and "Undefined fact: "T6"△".

SOME EMENDATIONS OF GÖDEL'S ONTOLOGICAL PROOF

C. Anthony Anderson

Kurt Gödel's version of the ontological argument was shown by J. Howard Sobel to be defective, but some plausible modifications in the argument result in a version which is immune to Sobel's objection. A definition is suggested which permits the proof of some of Gödel's axioms.

[Faith and Philosophy 1990]

Ontological Argument: Variant by Anderson

Part I - Proving that God's existence is possible

D1 Being Godlike is equivalent to having all positive properties.

A1 Exactly one of a property or its negation is positive.

A2 Any property entailed by a positive property is positive.

A3 The combination of any collection of positive properties is itself positive.

From A1 and A2 follows theorem T1:

T1 Every positive property is possibly instantiated.

From D1 and A3 follows:

T2 Being Godlike is a positive property.

From T1 and T2 follows:

T3 Being Godlike is possibly instantiated.

Ontological Argument: Variant by Anderson

Part I - Proving that God's existence is possible

D1 Being Godlike is equivalent to having all positive properties.

A1a If a property is positive, then its negation is not positive.

A1b If the negation of a property is not positive, then the property is positive.

A2 Any property entailed by a positive property is positive.

A3 The combination of any collection of positive properties is itself positive.

From A1 and A2 follows theorem T1:

T1 Every positive property is possibly instantiated.

From D1 and A3 follows:

T2 Being Godlike is a positive property.

From T1 and T2 follows:

T3 Being Godlike is possibly instantiated.

Ontological Argument: Variant by Anderson

Part I - Proving that God's existence is possible

D1 Being Godlike is equivalent to having all positive properties.

A1a If a property is positive, then its negation is not positive.

A1b ~~If the negation of a property is not positive, then the property is positive.~~

A2 Any property entailed by a positive property is positive.

A3 The combination of any collection of positive properties is itself positive.

From A1 and A2 follows theorem T1:

T1 Every positive property is possibly instantiated.

From D1 and A3 follows:

T2 Being Godlike is a positive property.

From T1 and T2 follows:

T3 Being Godlike is possibly instantiated.

Ontological Argument: Variant by Anderson

Part I - Proving that God's existence is possible

D1' Being Godlike is equivalent to having all and only the positive properties as necessary properties.

A1a If a property is positive, then its negation is not positive.

A1b ~~If the negation of a property is not positive, then the property is positive.~~

A2 Any property entailed by a positive property is positive.

A3 The combination of any collection of positive properties is itself positive.

From A1 and A2 follows theorem T1:

T1 Every positive property is possibly instantiated.

From D1 and A3 follows:

T2 Being Godlike is a positive property.

From T1 and T2 follows:

T3 Being Godlike is possibly instantiated.

Ontological Argument: Variant by Anderson

Part I - Proving that God's existence is possible

D1' Part II - Proving that God's existence is necessary, if possible

A1a D2 A property E is the essence of an individual x iff x has E and all of x's properties are nec. entailed by E.

A1b A4 Positive properties are necessarily positive.

From A1 and A4 (using definitions D1 and D2) follows:

A2 T4 Being Godlike is an essential property of any Godlike individual.

A3 D3 Necessary existence of an individual is the necessary instantiation of all its essences.

T1 A5 Necessary existence is a positive property.

From T4 and A5 (using D1, D2, D3) follows:

T2 T5 Being Godlike, if instantiated, is necessarily instantiated.

From T5 and A2 (using D1, D2, D3) follows:
And finally from T3, T5 (together with some implicit modal axioms, e.g. S5) the existence of (at least a) God follows:

T6 Being Godlike is necessarily instantiated.

Ontological Argument: Variant by Anderson

Part I - Proving that God's existence is possible

D1' Part II - Proving that God's existence is necessary, if possible

A1a D2' A property E is an essence (\mathcal{E}^A) of an individual x if and only if all of x's necessary properties are nec. entailed by E and (conversely) all properties nec. entailed by E are necessary properties of x.

A1b A4 Positive properties are necessarily positive.

A2 From A1 and A4 (using definitions D1 and D2) follows:

A3 T4 Being Godlike is an essential property of any Godlike individual.

From D3 Necessary existence of an individual is the necessary instantiation of all its essences.

T1 From A5 Necessary existence is a positive property.

T2 From T4 and A5 (using D1, D2, D3) follows:

T3 T5 Being Godlike, if instantiated, is necessarily instantiated.

And finally from T3, T5 (together with some implicit modal axioms, e.g. S5) the existence of (at least a) God follows:

T6 Being Godlike is necessarily instantiated.

Ontological Argument: Variant by Anderson

Part I - Proving that God's existence is possible

D1'

Part II - Proving that God's existence is necessary, if possible

D2'

✗ “Modal Collapse” is *not* implied by these axioms

A1a

A1b

A2

A4 From ▶ no determinism

A3

T4 From E “positive properties (\mathcal{P})” are applied here to intensional properties.

Fro

D3 M We have:

T1

i. ▶ \mathcal{P} is *not* an ultrafilter (has countermodel)

Fro

A5 M Let \mathcal{P}' be the set of all “rigidly intensionalised extensions” of posi-

T2

From E tive properties. We can prove:

Fro

T5 E ▶ \mathcal{P}' is an ultrafilter

T3

And e.g. S ▶ $\mathcal{P} \neq \mathcal{P}'$

l of
all

all

ns,

T6 Being Godlike is necessarily instantiated.

Ontological Argument: Variant by Anderson

```
1 theory AndersonProof imports IHOML
2 begin
3 (*Positiveness/perfection: uninterpreted constant symbol*)
4 consts positiveProperty::"(e⇒i⇒bool)⇒i⇒bool" ("P")
5 (*Some auxiliary definitions*)
6 definition h3 (infix "⇒" 60) where "X ⇒ Y ≡ □(∀z. X z → Y z)"
7
8 (**Part I**)
9 (*D1*) definition GA ("GA") where "GA ≡ λx. ∀Y. (P Y) ↔ □(Y x)"
10 (*Ala*) axiomatization where Ala:"[∀X. P (→X) → ¬(P X)]"
11 (*A2*) axiomatization where A2: "[∀X Y. (P X ∧ (X ⇒ Y)) → P Y]"
12 (*T1*) theorem T1: "[∀X. P X → ◊∃E X]" using Ala A2 h3_def by metis
13 (*T2*) axiomatization where T2: "[P GA]" (*here we postulate T2 instead of proving it*)
14 (*T3*) theorem T3: "[◊∃E GA]" using T1 T2 h3_def by blast
15
```

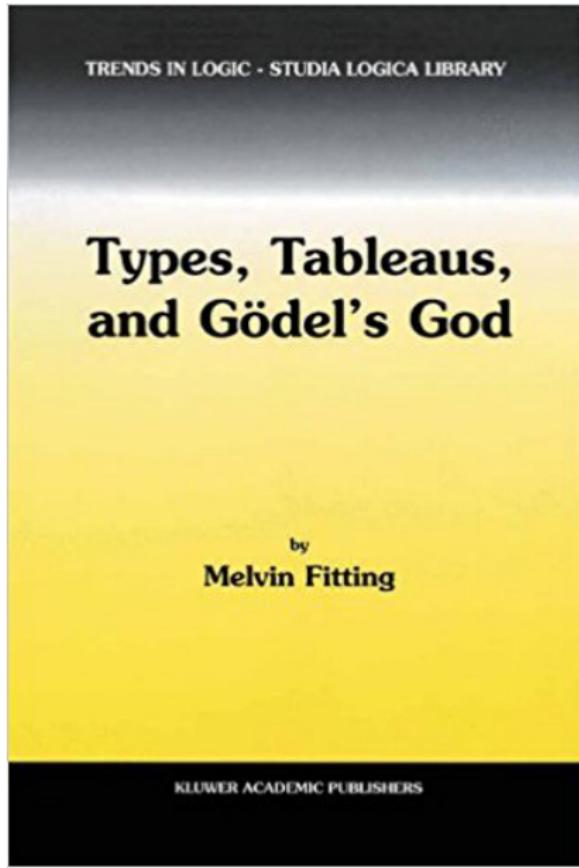
Ontological Argument: Variant by Anderson

```
1 theory AndersonProof imports IHOML
2
3 16 (**Part II**)
4 17 (*Logic KB*) axiomatization where symm: "symmetric aRel"
5 18 (*A4*) axiomatization where A4: "[ $\forall X. P X \rightarrow \square(P X)$ ]"
6 19 (*D2'*) abbreviation essA (" $E^A$ ") where " $E^A Y x \equiv (\forall Z. \square(Z x) \leftrightarrow Y \Rightarrow Z)$ "
7 20 (*T4*) theorem T4: "[ $\forall x. G^A x \rightarrow (E^A G^A x)$ ]" by (metis A2 GA_def T2 symm h3_def)
8 21 (*D3*) abbreviation NEA (" $NE^A$ ") where " $NE^A x \equiv (\lambda w. (\forall Y. E^A Y x \rightarrow \square \exists^E Y) w)$ "
9 22 (*A5*) axiomatization where A5: "[ $P NE^A$ ]"
10 23 (*T5*) theorem T5: "[ $\square \exists^E G^A \longrightarrow \square \exists^E G^A$ ]" by (metis A2 GA_def T2 symm h3_def)
11 24 (*T6*) theorem T6: "[ $\square \exists^E G^A$ ]" using T3 T5 by blast
12 25
13 26 (**Modal collapse is countersatisfiable**)
14 27 lemma "[ $\forall \Phi. (\Phi \rightarrow (\square \Phi))$ ]" nitpick oops (*Countermodel found by Nitpick*)
15 28
16 29 (**Consistency**)
17 30 lemma True nitpick[satisfy] oops (*model found by Nitpick: the axioms are consistent*)
18 31
19 32 (**Some Corollaries**)
20 33 (*C1*) theorem C1: "[ $\forall E x. ((E^A E x) \wedge (P x)) \rightarrow (E \Rightarrow P)$ ]" nitpick oops (*countermodel*)
21 34 (*C2*) theorem C2: "[ $\forall X. \neg P X \rightarrow \square(\neg P X)$ ]" using A4 symm by blast
22 35 definition h4 (" $\mathcal{N}$ ") where " $\mathcal{N} X \equiv \neg P X$ "
23 36 (*C3*) theorem C3: "[ $\forall X. \mathcal{N} X \rightarrow \square(\mathcal{N} X)$ ]" by (simp add: C2 h4_def)
```

Ontological Argument: Variant by Anderson

```
1 theory AndersonProof imports IHOML
2
3 (*Positive Properties and Ultrafilters*)
4 abbreviation emptySet ("∅") where "∅ ≡ λx w. False"
5 abbreviation entails (infixr "⊆" 51) where "φ ⊆ ψ ≡ ∀x w. φ x w → ψ x w"
6 abbreviation andPred (infixr "Π" 51) where "φ Π ψ ≡ λx w. φ x w ∧ ψ x w"
7 abbreviation negpred ("¬" [52] 53) where "¬ψ ≡ λx w. ¬ψ x w"
8 abbreviation "ultrafilter" Φ cw ≡
9   ¬(Φ ∅ cw)
10  ∧ ( ∀φ. ∀ψ. (Φ φ cw ∧ Φ ψ cw) → (Φ (φ Π ψ) cw))
11  ∧ ( ∀ψ::e⇒i⇒bool. ∀ψ::e⇒i⇒bool. (Φ φ cw ∨ Φ (¬φ) cw) ∧ ¬(Φ φ cw ∧ Φ (¬φ) cw))
12  ∧ ( ∀ψ::e⇒i⇒bool. ∀ψ::e⇒i⇒bool. (Φ φ cw ∧ φ ⊆ ψ) → Φ ψ cw)
13
14 (*U1*) theorem U1: "∀w. ultrafilter P w" nitpick[user_axioms,format=2,show_all] oops (*countermodel*)
15 lemma helpC: "∀φ ψ w. (P φ w ∨ P (¬φ) w) ∧ ¬(P φ w ∧ P (¬φ) w)" nitpick oops (*countermodel*)
16
17 (*(φ) converts an extensional object φ into 'rigid' intensional one*)
18 abbreviation trivialConversion ("(φ)") where "(φ) ≡ (λw. φ)"
19 (*Q ↓φ: the extension of a (possibly) non-rigid predicate φ is turned into a rigid intensional one,
20 then Q is applied to the latter; ↓φ can be read as "the rigidly intensionalised predicate φ"*)
21 abbreviation mextPredArg (infix "↓" 60) where "Q ↓φ ≡ λw. Q (λx. (φ x w)) w"
22 lemma "∀Q φ. Q φ = Q ↓φ" nitpick oops (*countermodel*: the two notions are not the same*)
23
24 lemma helpE: "∀w. ¬((P ↓∅) w)" using T1 by blast
25 lemma helpF: "∀φ ψ w. ((P ↓φ) w ∧ (P ↓ψ) w) → ((P ↓(φ Π ψ)) w)" by (smt GA_def T3 T5 symm)
26 lemma helpG: "∀w. ((P ↓φ) w ∨ (P ↓(¬φ)) w) ∧ ¬((P ↓φ) w ∧ (P ↓(¬φ)) w)" by (smt GA_def T3 T5 symm)
27 lemma helpH: "∀w. ((P ↓φ) w ∧ φ ⊆ ψ) → (P ↓ψ) w" by (metis (no_types, lifting) A4 C2 GA_def T3)
28
29 abbreviation "P' φ ≡ (P ↓φ)" (*P': the set of all rigidly intensionalised positive properties*)
30
31 (*U2*) theorem U2: "∀w. ultrafilter P' w" using helpE helpF helpG helpH by simp
32 (*U3*) theorem U3: "(P' ⊆ P) ∧ (P ⊆ P')" nitpick oops (*countermodel*: P' and P are not equal*)
```

Ontological Argument: Variant by Fitting (2002)



Ontological Argument: Variant by Fitting (2002)

Part I - Proving that God's existence is possible

D1 Being Godlike is equivalent to having all positive properties.

A1 Exactly one of a property or its negation is positive.

A2 Any property entailed by a positive property is positive.

A3 The combination of any collection of positive properties is itself positive.

From A1 and A2 follows theorem T1:

T1 Every positive property is possibly instantiated.

From D1 and A3 follows:

T2 Being Godlike is a positive property.

From T1 and T2 follows:

T3 Being Godlike is possibly instantiated.

Fully analogous to Gödel/Scott.

But: “positive properties” applied to extensions of properties only!

Ontological Argument: Variant by Fitting (2002)

Part I - Proving that God's existence is possible

D1 Part II - Proving that God's existence is necessary, if possible

A1 D2 A property E is the essence of an individual x iff x has E and all of x's properties are nec. entailed by E.^a

A2 A3 Positive properties are necessarily positive.

From A1 and A4 (using definitions D1 and D2) follows:

T1 T4 Being Godlike is an essential property of any Godlike individual.

From T1 and D3 (using definition D2) follows:

T2 D3 Necessary existence of an individual is the necessary instantiation of all its essences.

From T2 and A5 (using D1, D2, D3) follows:

T3 A5 Necessary existence is a positive property.

From T3 and T4 (using D1, D2, D3) follows:

T5 T5 Being Godlike, if instantiated, is necessarily instantiated.

And finally from T3, T5 (together with some implicit modal axioms, e.g. S5) the existence of (at least a) God follows:

T6 T6 Being Godlike is necessarily instantiated.

^aThe underlined part in definition D2 has been added by Scott. Gödel originally omitted this part.

Ontological Argument: Variant by Fitting (2002)

Part I - Proving that God's existence is possible

D1 Part II - Proving that God's existence is necessary, if possible

A1 D2 A property E is the essence of an individual x iff x has E and all of x's properties are nec. entailed by E.^a

A2 A3 Positive properties are necessarily positive.

From T1

“Modal Collapse” is *not* implied by these axioms

T1 D3

$$\varphi \supset \Box\varphi \quad (\text{has countermodel})$$

From T2

A5

We can prove that these “positive property extensions” (which corresponds to \mathcal{P}' from before) form an ultrafilter.

From T3

T5

And

e.g. S5) the existence of (at least a) God follows:

T6 Being Godlike is necessarily instantiated.

^aThe underlined part in definition D2 has been added by Scott. Gödel originally omitted this part.

Ontological Argument: Variant by Fitting (2002)

```
1 theory FittingProof imports IHOML
2 begin
3 (*Positiveness/perfection: uninterpreted constant symbol*)
4 consts Positiveness::"(e⇒bool)⇒i⇒bool" ("P")
5 (*Some auxiliary definitions*)
6 (* $\varphi$  converts an extensional object  $\varphi$  into 'rigid' intensional one*)
7 abbreviation trivialConversion ("(_)"") where " $\langle\varphi\rangle \equiv (\lambda w. \varphi)$ ""
8 abbreviation Entails (infix " $\Rightarrow$ " 60) where " $X \Rightarrow Y \equiv \Box(\forall z. (X z) \rightarrow (Y z))$ ""
9 (* $\varphi$ 's argument is a relativized term (of extensional type) derived from an intensional predicate.*)
10 abbreviation extPredArg (infix " $\Downarrow$ " 60) where " $\varphi \Downarrow P \equiv \lambda w. \varphi (\lambda x. P x w) w$ ""
11 (*A variant of the latter where  $\varphi$  takes intensional terms as argument.*)
12 abbreviation mextPredArg (infix " $\Downarrow$ " 60) where " $\varphi \Downarrow P \equiv \lambda w. \varphi (\lambda x. (P x w)) w$ ""
13 (*Another variant where  $\varphi$  has two arguments (the first one being relativized).*)
14 abbreviation extPredArg1 (infix " $\Downarrow_1$ " 60) where " $\varphi \Downarrow_1 P \equiv \lambda z. \lambda w. \varphi (\lambda x. P x w) z w$ ""
15
16 (**Part I**)
17 (*D1*) abbreviation God ("G") where " $G \equiv (\lambda x. \forall y. P Y \rightarrow (Y x))$ ""
18 (*A1*) axiomatization where Ala:"[ $\forall X. P (\neg X) \rightarrow \neg(P X)$ ]'' and Alb:"[ $\forall X. \neg(P X) \rightarrow P (\neg X)$ ]''"
19 (*A2*) axiomatization where A2: "[ $\forall X Y. (P X \wedge (X \Rightarrow Y)) \rightarrow P Y$ ]''"
20 (*T1*) theorem T1: "[ $\forall X. P X \rightarrow \Diamond(\exists z. (X z))$ ]'' using Ala A2 by blast
21 (*T2*) axiomatization where T2: "[ $P \Downarrow G$ ]''"
22 (*T3*) theorem T3deRe: "[ $(\lambda X. \Diamond \exists^E X) \Downarrow G$ ]'' using T1 T2 by simp
23 theorem T3deDicto: "[ $\Diamond \exists^E \Downarrow G$ ]'' nitpick oops (*countermodel*)"
24
```

Ontological Argument: Variant by Fitting (2002)

```
1 theory FittingProof imports IHOML
2 begin
3 (*Positiveness/perfection: uninterpreted constant symbol*)
4 const Positiveness :: "(o::bool) -> i::bool" ("P")
5
6 (**Part II*)
7 (*Logic KB*) axiomatization where symm: "symmetric aRel"
8 (*A4*) axiomatization where A4: "[|X. P X → □(P X)|]"
9 (*D2*) abbreviation Essence ("E") where "E Y x ≡ (Y x) ∧ (VZ.(Z x)→(Y⇒Z))"
10 (*T4*) theorem T4: "[|X. G x → ((E ↓_1 G) x)|]" using Alb by metis
11 (*D3*) definition NE ("NE") where "NE x ≡ λw. (VY. E Y x → □(EZ. (Y z))) w"
12 (*A5*) axiomatization where A5: "[|P |NE|"
13   lemma help1: "|[|E ↓ G → □EZ ↓ G|]| sorry (*longer interactive proof, omitted here*)
14   lemma help2: "|[|E ↓ G → ((λX. □EZ X) ↓ G)|]| by (metis A4 help1)
15 (*T5*) theorem T5deDicto: "|[|◇EZ ↓ G|] → [|EZ ↓ G|]|" using T3deRe help1 by blast
16   theorem T5deRe: "|[|(|λX. ◇EZ X) ↓ G|] → |[|(|λX. □EZ X) ↓ G|]|" by (metis A4 help2)
17 (*T6*) theorem T6deDicto: "|[|EZ ↓ G|]|" using T3deRe help1 by blast
18   theorem T6deRe: "|[|(|λX. □EZ X) ↓ G|]|" using T3deRe help2 by blast
19
20 (**Consistency**)
21 40 lemma True nitpick[satisfy] oops (*Model found by Nitpick: the axioms are consistent*)
22
23 (**Modal Collapse**)
24 43 lemma ModalCollapse: "|[|Φ. (Φ → (□ Φ))|]| nitpick oops (*countermodel*)"
25
26 (**Some Corollaries**)
27 (* Todo (*C1*) theorem C1: "|[|VE P x. ((E E x) ∧ (P x)) → (E ⇒ P)|]|" by (metis ess_def) *)
28 (*C2*) theorem C2: "|[|X. ¬P X → □(¬P X)|]|" using A4 symm by blast
29   definition h4 ("N") where "N X ≡ ¬P X"
30 (*C3*) theorem C3: "|[|X. N X → □(N X)|]|" by (simp add: C2 h4_def)
31   definition "rigid φ ≡ ∀x. φ x → □(φ x)"
32 (*C4*) theorem "|[|φ. P φ → rigid (λx. (φ x))|]|" by (simp add: rigid_def)
33 (*C5*) theorem "|[|rigid P|]|" by (simp add: A4 rigid_def)
```

Ontological Argument: Variant by Fitting (2002)

```
1 theory FittingProof imports IHOML
2 begin
3 (*Positiveness/perfection: uninterpreted constant symbol*)
4 const Positiveness :: "(o::bool) -> i::bool" ("P")
5
6 (**Part II*)
7 (*Logic KB*) axiomatization where symm: "symmetric aRel"
8 (*A4*) axiomatization where A4: "[|X. P X → □(P X)|]"
9 (*D2*) abbreviation Essence ("E") where "E Y x ≡ (Y x) ∧ (VZ. (Z x) → (Y ⇒ Z))"
10 (*T4*) theorem T4: "[|X. G x → ((E ↓G) x)|]" using Alb by metis
11 (*D3*) definition NE ("NE") where "NE x ≡ λw. (VY. E Y x → □(∃z. (Y z))) w"
12 (*A5*) theorem A5: "[|X. G x → ((E ↓G) x)|]" using T4, NE by metis
13
14 (**Positive Properties and Ultrafilters**)
15 abbreviation empty ("∅") where "∅ ≡ λx. False"
16 abbreviation intersect (infix "∩" 52) where "φ ∩ ψ ≡ (λx. φ x ∧ ψ x)"
17 abbreviation nnegpred ("¬_" "[52]53) where "¬ψ ≡ λx. ¬ψ(x)"
18 abbreviation entail (infixr "⊆" 51) where "φ ⊆ ψ ≡ ∀x. φ x → ψ x"
19 abbreviation "ultrafilter" Φ cw ≡
20   ¬(Φ ∅ cw) (* The empty set is not an element of U *)
21   ∧ (∀ψ::(e⇒bool). ∀ψ::(e⇒bool). (Φ φ cw ∧ Φ ψ cw) → (Φ (φ∩ψ) cw))
22   ∧ (∀ψ::(e⇒bool). ∀ψ::(e⇒bool). (Φ φ cw ∨ Φ (¬φ) cw) ∧ ¬(Φ φ cw ∧ Φ (¬φ) cw))
23   ∧ (∀ψ::(e⇒bool). ∀ψ::(e⇒bool). (Φ φ cw ∧ φ ⊆ ψ) → Φ ψ cw)
24 lemma lemmaA: "∀w. ¬(P ∅ w)" using T1 by blast
25 lemma lemmaB: "∀w. (P φ w ∧ P ψ w) → (P (φ∩ψ) w)" by (metis Alb T3deRe)
26 lemma lemmaC: "∀w. (P φ w ∨ P (¬φ) w) ∧ ¬(P φ w ∧ P (¬φ) w)" using Ala Alb by blast
27 lemma lemmaD: "∀w. (P φ w ∧ φ ⊆ ψ) → P ψ w" by (smt A2)
28
29 (*U1*) theorem "∀w. ultrafilter P w" by (smt lemmaA lemmaB lemmaC lemmaD)
30
31 (*C4*) theorem "[|φ. P φ → rigid (λx. (φ x))|]" by (simp add: rigid_def)
32 (*C5*) theorem "[|rigid P|]" by (simp add: A4 rigid_def)
```

Summary of Results

- ▶ “Godlike” has been defined in terms of “positive properties”
- ▶ “positive properties” has been linked with the notion of “ultrafilter”.
- ▶ In our experiments we then distinguished between
 - \mathcal{P} : positive intensional properties
 - \mathcal{P}' : positive (“rigidly intensionalised”) extensions of properties
- ▶ Gödel/Scott variant axiomatises \mathcal{P} : $\mathcal{P} = \mathcal{P}'$ is an ultrafilter
- ▶ Anderson’s variant axiomatises \mathcal{P} : $\mathcal{P} \neq \mathcal{P}'$; only \mathcal{P}' is an ultrafilter
- ▶ Fitting’s variant axiomatises only \mathcal{P}' : \mathcal{P}' is an ultrafilter

Modal collapse holds for Gödel/Scott variant, but not for Anderson’s & Fitting’s!

They achieve this in seemingly different ways.

Mathematically, however, their solutions appear closely related.

Summary of Results

- ▶ “Godlike” has been defined in terms of “positive properties”
- ▶ “positive properties” has been linked with the notion of “ultrafilter”.
- ▶ In our experiments we then distinguished between
 - \mathcal{P} : positive intensional properties
 - \mathcal{P}' : positive (“rigidly intensionalised”) extensions of properties
- ▶ Gödel/Scott variant axiomatises \mathcal{P} : $\mathcal{P} = \mathcal{P}'$ is an ultrafilter
- ▶ Anderson’s variant axiomatises \mathcal{P} : $\mathcal{P} \neq \mathcal{P}'$; only \mathcal{P}' is an ultrafilter
- ▶ Fitting’s variant axiomatises only \mathcal{P}' : \mathcal{P}' is an ultrafilter

Modal collapse holds for Gödel/Scott variant, but not for Anderson’s & Fitting’s!

They achieve this in seemingly different ways.

Mathematically, however, their solutions appear closely related.

Summary of Results

- ▶ “Godlike” has been defined in terms of “positive properties”
- ▶ “positive properties” has been linked with the notion of “ultrafilter”.
- ▶ In our experiments we then distinguished between
 - \mathcal{P} : positive intensional properties
 - \mathcal{P}' : positive (“rigidly intensionalised”) extensions of properties
- ▶ Gödel/Scott variant axiomatises \mathcal{P} : $\mathcal{P} = \mathcal{P}'$ is an ultrafilter
- ▶ Anderson’s variant axiomatises \mathcal{P} : $\mathcal{P} \neq \mathcal{P}'$; only \mathcal{P}' is an ultrafilter
- ▶ Fitting’s variant axiomatises only \mathcal{P}' : \mathcal{P}' is an ultrafilter

Modal collapse holds for Gödel/Scott variant, but not for Anderson’s & Fitting’s!

They achieve this in seemingly different ways.

Mathematically, however, their solutions appear closely related.

Summary of Results

- ▶ “Godlike” has been defined in terms of “positive properties”
- ▶ “positive properties” has been linked with the notion of “ultrafilter”.
- ▶ In our experiments we then distinguished between
 - \mathcal{P} : positive intensional properties
 - \mathcal{P}' : positive (“rigidly intensionalised”) extensions of properties
- ▶ Gödel/Scott variant axiomatises \mathcal{P} : $\mathcal{P} = \mathcal{P}'$ is an ultrafilter
- ▶ Anderson’s variant axiomatises \mathcal{P} : $\mathcal{P} \neq \mathcal{P}'$; only \mathcal{P}' is an ultrafilter
- ▶ Fitting’s variant axiomatises only \mathcal{P}' : \mathcal{P}' is an ultrafilter

Modal collapse holds for Gödel/Scott variant, but not for Anderson’s & Fitting’s!

They achieve this in seemingly different ways.

Mathematically, however, their solutions appear closely related.

Summary of Results

- ▶ “Godlike” has been defined in terms of “positive properties”
- ▶ “positive properties” has been linked with the notion of “ultrafilter”.
- ▶ In our experiments we then distinguished between
 - \mathcal{P} : positive intensional properties
 - \mathcal{P}' : positive (“rigidly intensionalised”) extensions of properties
- ▶ Gödel/Scott variant axiomatises \mathcal{P} : $\mathcal{P} = \mathcal{P}'$ is an ultrafilter
- ▶ Anderson’s variant axiomatises \mathcal{P} : $\mathcal{P} \neq \mathcal{P}'$; only \mathcal{P}' is an ultrafilter
- ▶ Fitting’s variant axiomatises only \mathcal{P}' : \mathcal{P}' is an ultrafilter

Modal collapse holds for Gödel/Scott variant, but not for Anderson’s & Fitting’s!

They achieve this in seemingly different ways.

Mathematically, however, their solutions appear closely related.

Summary of Results

- ▶ “Godlike” has been defined in terms of “positive properties”
- ▶ “positive properties” has been linked with the notion of “ultrafilter”.
- ▶ In our experiments we then distinguished between
 - \mathcal{P} : positive intensional properties
 - \mathcal{P}' : positive (“rigidly intensionalised”) extensions of properties
- ▶ Gödel/Scott variant axiomatises \mathcal{P} : $\mathcal{P} = \mathcal{P}'$ is an ultrafilter
- ▶ Anderson’s variant axiomatises \mathcal{P} : $\mathcal{P} \neq \mathcal{P}'$; only \mathcal{P}' is an ultrafilter
- ▶ Fitting’s variant axiomatises only \mathcal{P}' : \mathcal{P}' is an ultrafilter

Modal collapse holds for Gödel/Scott variant, but not for Anderson’s & Fitting’s!

They achieve this in seemingly different ways.

Mathematically, however, their solutions appear closely related.



Part D

— Discussion —

Discussion

- ▶ There are consistent theistic theories which
 - ▶ imply the necessary existence of a supreme being
 - ▶ support different philosophical positions: determinism / non-determinism
- ▶ Theistic belief (at least in an abstract sense) not necessarily irrational
- ▶ By adopting the notion of “ultrafilter” these
theistic theories were mapped here to mathematical theories

Question

- ▶ Relevance of such existence results for the real world?
- ▶ Existence results in metaphysics vs. mathematics — difference?

Discussion

- ▶ There are consistent theistic theories which
 - ▶ imply the necessary existence of a supreme being
 - ▶ support different philosophical positions: determinism / non-determinism
- ▶ Theistic belief (at least in an abstract sense) not necessarily irrational
- ▶ By adopting the notion of “ultrafilter” these
theistic theories were mapped here to mathematical theories

Question

- ▶ Relevance of such existence results for the real world?
- ▶ Existence results in metaphysics vs. mathematics — difference?

Discussion

- ▶ There are consistent theistic theories which
 - ▶ imply the necessary existence of a supreme being
 - ▶ support different philosophical positions: determinism / non-determinism
- ▶ Theistic belief (at least in an abstract sense) not necessarily irrational
- ▶ By adopting the notion of “ultrafilter” these
theistic theories were mapped here to mathematical theories

Question

- ▶ Relevance of such existence results for the real world?
- ▶ Existence results in metaphysics vs. mathematics — difference?

Conclusion

- ▶ Experiments in Computational Metaphysics: Ontological Argument
- ▶ Universal Logical Reasoning Approach
- ▶ Interesting new results
- ▶ Approach applicable e.g. also to Ed Zalta's work
- ▶ Many other relevant and pressing applications (e.g., machine ethics)
- ▶ Should scale for Higher Partial Type Theory

Evidence provided for central claim of this talk

- ▶ Computers may help to sharpen our understanding of arguments
- ▶ Universal (meta-)logical reasoning approach particularly well suited

Related work

- ▶ Ed Zalta (& co) with PROVER9 at Stanford [AJP 2011, CADE 2015]
- ▶ John Rushby with PVS at SRI [CAV-WS 2013, JAL 2018]

Conclusion

- ▶ Experiments in Computational Metaphysics: Ontological Argument
- ▶ Universal Logical Reasoning Approach
- ▶ Interesting new results
- ▶ Approach applicable e.g. also to Ed Zalta's work
- ▶ Many other relevant and pressing applications (e.g., machine ethics)
- ▶ Should scale for Higher Partial Type Theory

Evidence provided for central claim of this talk

- ▶ Computers may help to sharpen our understanding of arguments
- ▶ Universal (meta-)logical reasoning approach particularly well suited

Related work

- ▶ Ed Zalta (& co) with PROVER9 at Stanford [AJP 2011, CADE 2015]
- ▶ John Rushby with PVS at SRI [CAV-WS 2013, JAL 2018]

Conclusion

- ▶ Experiments in Computational Metaphysics: Ontological Argument
- ▶ Universal Logical Reasoning Approach
- ▶ Interesting new results
- ▶ Approach applicable e.g. also to Ed Zalta's work
- ▶ Many other relevant and pressing applications (e.g., machine ethics)
- ▶ Should scale for Higher Partial Type Theory

Evidence provided for central claim of this talk

- ▶ Computers may help to sharpen our understanding of arguments
- ▶ Universal (meta-)logical reasoning approach particularly well suited

Related work

- ▶ Ed Zalta (& co) with PROVER9 at Stanford [AJP 2011, CADE 2015]
- ▶ John Rushby with PVS at SRI [CAV-WS 2013, JAL 2018]

DEMO: Anderson's Variant

isabelle2018/HOL - AndersonProof.thy (modified)

The screenshot shows the Isabelle2018 interface with the file "AndersonProof.thy" open. The code defines various logical constructs and proves theorems related to Anderson's variant. The interface includes a toolbar at the top, a vertical scroll bar on the right, and tabs for Documentation, Sidekick, State, and Theories on the far right.

```
1 theory AndersonProof imports IHOML
2 begin
3 (*Positiveness/perfection: uninterpreted constant symbol*)
4 consts positiveProperty::"(e⇒i=bool)⇒i⇒bool" ("P")
5 (*Some auxiliary definitions*)
6 definition h3 (infix "⇒" 60) where "X ⇒ Y ≡ □(∀z. X z → Y z)"
7 (**Part I**)
8 (*D1*) definition GA ("G^A") where "G^A ≡ λx. ∀Y. (P Y) ↔ □(Y x)"
9 (*A1a*) axiomatization where A1a:"[∀X. P (→X) → ¬(P X)]"
10 (*A2*) axiomatization where A2: "[∀X Y. (P X ∧ (X ⇒ Y)) → P Y]"
11 (*T1*) theorem T1: "[∀X. P X → □∃E X]" using A1a A2 h3_def by metis
12 (*T2*) axiomatization where T2: "[P G^A]" (*here we postulate T2 instead of proving it*)
13 (*T3*) theorem T3: "[□∃E G^A]" using T1 T2 h3_def by blast
14 (**Part II**)
15 (*Logic KB*) axiomatization where symm: "∀x y. x r y → y r x"
16 (*A4*) axiomatization where A4: "[∀X. P X → □(P X)]"
17 (*D2*) abbreviation essA ("E^A") where "E^A Y x ≡ (∀Z. □(Z x) ↔ Y ⇒ Z)"
18 (*T4*) theorem T4: "[∀x. G^A x → (E^A G^A x)]" by (metis A2 GA_def T2 symm h3_def)
19 (*D3*) abbreviation NEA ("NE^A") where "NE^A x ≡ (λw. (∀Y. E^A Y x → □∃E Y) w)"
20 (*A5*) axiomatization where A5: "[P NE^A]"
21 (*T5*) theorem T5: "[□∃E G^A] → [□∃E G^A]" by (metis A2 GA_def T2 symm h3_def)
22 (*T6*) theorem T6: "[□∃E G^A]" using T3 T5 by blast
23
24 (**Modal collapse is countersatisfiable**)
25 lemma "[∀Φ. (Φ → (□ Φ))]" nitpick[user_axioms] oops (*Countermodel found by Nitpick*)
26
```

Proof state Auto update Update Search: 100% Output Query Sledgehammer Symbols
28,53 (1529/6107) (isabelle,isabelle,UTF-8-Isabelle) | n m r o UG 195/587MB 6:25 PM

DEMO: Anderson's Variant

Isabelle2018/HOL - AndersonProof.thy

AndersonProof.thy (~/GITHUBS/chrisgitlab/talks/2019-Dubrovnik/)

29 (**Positive Properties and Ultrafilters**)
30 abbreviation emptySet (" \emptyset ") where " $\emptyset \equiv \lambda x. w. \text{False}$ "
31 abbreviation entails (infixr " \subseteq " 51) where " $\varphi \subseteq \psi \equiv \forall x. w. \varphi x w \rightarrow \psi x w$ "
32 abbreviation andPred (infixr " \sqcap " 51) where " $\varphi \sqcap \psi \equiv \lambda x. w. \varphi x w \wedge \psi x w$ "
33 abbreviation negpred (" \neg " [52] 53) where " $\neg \psi \equiv \lambda x. w. \neg \psi x w$ "
34 abbreviation "ultrafilter Φ cw" ≡
35 $\neg(\Phi \emptyset cw)$
36 $\wedge (\forall \varphi. \forall \psi. (\Phi \varphi cw \wedge \Phi \psi cw) \rightarrow (\Phi (\varphi \sqcap \psi) cw))$
37 $\wedge (\forall \varphi::e\Rightarrow i\Rightarrow \text{bool}. \forall \psi::e\Rightarrow i\Rightarrow \text{bool}. (\Phi \varphi cw \vee \Phi (\neg \varphi) cw) \wedge \neg(\Phi \varphi cw \wedge \Phi (\neg \varphi) cw))$
38 $\wedge (\forall \varphi::e\Rightarrow i\Rightarrow \text{bool}. \forall \psi::e\Rightarrow i\Rightarrow \text{bool}. (\Phi \varphi cw \wedge \varphi \subseteq \psi) \rightarrow \Phi \psi cw)"$
39
40
41 (*U1*) theorem U1: " $\forall w. \text{ultrafilter } P w$ " nitpick[user_axioms,format=2,show_all] oops (*counterm.*)
42

✓ Proof state ✓ Auto update Update Search: 100% Theories

$\psi = (\lambda x. _)((e_1, i_1) := \text{False}, (e_1, i_2) := \text{False})$
 $w = i_1$

Constants:

$P = (\lambda x. _)$
 $((\lambda x. _)((e_1, i_1) := \text{True}, (e_1, i_2) := \text{True}), i_1) := \text{True},$
 $((\lambda x. _)((e_1, i_1) := \text{True}, (e_1, i_2) := \text{True}), i_2) := \text{True},$
 $((\lambda x. _)((e_1, i_1) := \text{True}, (e_1, i_2) := \text{False}), i_1) := \text{False},$
 $((\lambda x. _)((e_1, i_1) := \text{True}, (e_1, i_2) := \text{False}), i_2) := \text{False},$
 $((\lambda x. _)((e_1, i_1) := \text{False}, (e_1, i_2) := \text{True}), i_1) := \text{False},$
 $((\lambda x. _)((e_1, i_1) := \text{False}, (e_1, i_2) := \text{True}), i_2) := \text{False},$
 $((\lambda x. _)((e_1, i_1) := \text{False}, (e_1, i_2) := \text{False}), i_1) := \text{False},$
 $((\lambda x. _)((e_1, i_1) := \text{False}, (e_1, i_2) := \text{False}), i_2) := \text{False})$
existsAt = $(\lambda x. _)((e_1, i_1) := \text{True}, (e_1, i_2) := \text{True})$
 $r = (\lambda x. _)((i_1, i_1) := \text{True}, (i_1, i_2) := \text{True}, (i_2, i_1) := \text{True}, (i_2, i_2) := \text{True})$

Output Query Sledgehammer Symbols

41,46 (2225/5923) (isabelle,isabelle,UTF-8-Isabelle) n m r o UG 200/502MB 6:38 PM

DEMO: Anderson's Variant

Isabelle2018/HOL - AndersonProof.thy (modified)

The screenshot shows the Isabelle2018 interface with the "AndersonProof.thy" file open. The code editor displays various definitions and theorems related to Anderson's variant. A proof state is shown at the bottom, and a search bar is available.

```
abbreviation trivialConversion ("(L)") where " $(\varphi) \equiv (\lambda w. \varphi)$ "  
(*Q ↓φ: the extension of a (possibly) non-rigid predicate φ is turned into a rigid intensional one,  
then Q is applied to the latter; ↓φ can be read as "the rigidly intensionalised predicate φ"*)  
abbreviation mextPredArg (infix "↓" 60) where " $Q \downarrow \varphi \equiv \lambda w. Q(\lambda x. (\varphi x w)) w$ "  
  
lemma "∀Q φ. Q φ = Q ↓φ" nitpick oops (*countermodel: the two notions are not the same*)  
  
lemma helpE: "∀w. ¬((P ↓∅) w)" using T1 by blast  
lemma helpF: "∀φ ψ w. ((P ↓φ) w ∧ (P ↓ψ) w) → ((P ↓(φ ∩ ψ)) w)" by (smt GA_def T3 T5 symm)  
lemma helpG: "∀w. ((P ↓φ) w ∨ (P ↓(¬φ)) w) ∧ ¬((P ↓φ) w ∧ (P ↓(¬φ)) w)" by (smt GA_def T3 T5 symm)  
lemma helpH: "∀w. ((P ↓φ) w ∧ φ ⊆ ψ) → (P ↓ψ) w" by (metis GA_def T3 T5 symm)  
  
abbreviation "P' φ ≡ (P ↓φ)" (*P': the set of all rigidly intensionalised positive properties*)  
  
(*U2*) theorem U2: "∀w. ultrafilter P' w" using helpE helpF helpG helpH by simp  
(*U3*) theorem U3: "(P' ⊆ P) ∧ (P ⊆ P')" nitpick[user_axioms] oops (*countermodel: P', P not equal*)
```

theorem

U2: $\vdash (\lambda w. (\neg P \wedge (\neg \text{False})) w \wedge$
 $(\lambda \varphi \psi. (P(\lambda x. (\varphi x w)) \wedge P(\lambda x. (\psi x w))) w) \subseteq (\lambda \varphi \psi. P'(\varphi \cap \psi) w) \wedge$
 $\vdash (\lambda \varphi. (\vdash ((P(\lambda x. (\varphi x w)) \vee P(\lambda x. ((\neg \varphi) x w))) w \wedge (\neg (\lambda)) (P' \varphi w) (P' (\neg \varphi) w)))) \wedge$
 $(\lambda \varphi \psi. P' \varphi w \wedge \varphi \subseteq \psi) \subseteq (\lambda \psi. P' \psi w))]$

Output Query Sledgehammer Symbols

58,79 (3239/5927) (isabelle,isabelle,UTF-8-Isabelle) n m r o UG 193/500MB 6:41 PM

DEMO: Anderson's Variant

Isabelle2018/HOL - AndersonProof.thy

AndersonProof.thy (~/GITHUBS/chrisgitlab/talks/2019-Dubrovnik/)

```
abbreviation trivialConversion ("(λ_)") where "(λφ) ≡ (λz::i. φ)"  
(*Q ↓φ: the extension of a (possibly) non-rigid predicate φ is turned into a rigid intensional one,  
then Q is applied to the latter; ↓φ can be read as "the rigidly intensionalised predicate φ"*)  
abbreviation mextPredArg (infix "↓" 60) where "Q ↓φ ≡ λw. Q (λx. (φ x w)) w"  
lemma "∀Q φ. Q φ = Q ↓φ" nitpick oops (*countermodel: the two notions are not the same*)  
lemma helpE: "∀w.¬((P ↓∅) w)" using T1 by blast  
lemma helpF: "∀φ ψ w.((P ↓φ) w ∧ (P ↓ψ) w) — ((P ↓(φ ∏ ψ)) w)" by (smt GA_def T3 T5 symm)  
lemma helpG: "∀w.((P ↓φ) w ∨ (P ↓(¬φ)) w) ∧ ¬((P ↓φ) w ∧ (P ↓(¬φ)) w)" by (smt GA_def T3 T5 symm)  
lemma helpH: "∀w.((P ↓φ) w ∧ φ ⊆ ψ) — (P ↓ψ) w" by (metis GA_def T3 T5 symm)  
abbreviation "P' φ ≡ (P ↓φ)" (*P': the set of all rigidly intensionalised positive properties*)  
(*U2*) theorem U2: "∀w. ultrafilter P' w" using helpE helpF helpG helpH by simp  
(*U3*) theorem U3: "(P' ⊆ P) ∧ (P ⊆ P')" nitpick[user_axioms] oops (*countermodel: P',P not equal*)
```

theorem

U2: $\vdash (\lambda w. (\neg P) \emptyset w \wedge$
 $(\lambda \varphi \psi. (P (\lambda x. (\varphi x w)) \wedge P (\lambda x. (\psi x w))) w) \subseteq (\lambda \varphi \psi. P' (\varphi \sqcap \psi) w) \wedge$
 $\vdash (\lambda \varphi. [\lambda \psi. (P (\lambda x. (\varphi x w)) \vee P (\lambda x. ((\neg \varphi) x w))) w \wedge (\neg (\lambda)) (P' \varphi w) (P' (\neg \varphi) w)])] \wedge$
 $(\lambda \varphi \psi. P' \varphi w \wedge \varphi \subseteq \psi) \subseteq (\lambda \varphi \psi. P' \psi w))]$

Output Query Sledgehammer Symbols

58,80 (3243/5930) (isabelle,isabelle,UTF-8-Isabelle) 314/501MB 6:44 PM

DEMO: Anderson's Variant

The screenshot shows the Isabelle 2018/HOL interface with a proof script named `AndersonProof.thy`. The script contains several lemmas related to modal logic and Barcan formulas. The interface includes a toolbar with various icons, a search bar, and a status bar at the bottom.

```
1 (*Modal logic S5: Consistency and Modal Collapse*)
2 axiomatization where refl: " $\forall x. x \rightarrow x$ " and trans: " $\forall x y z. x \rightarrow y \wedge y \rightarrow z \rightarrow x \rightarrow z$ "
3 lemma True nitpick[satisfy] oops (*Model found by Nitpick: the axioms are consistent*)
4 lemma ModalCollapse: " $\exists \Phi. (\Phi \rightarrow (\Box \Phi))$ " nitpick[user_axioms,show_all,format=2] oops (*countermodel*)
5
6 (**Barcan and Converse Barcan Formula for Individuals (type e)**)
7 lemma BarcanIndl: " $\exists \varphi(x). \Box(\varphi(x)) \rightarrow (\Box(\forall x. \varphi(x)))$ " nitpick oops (*countermodel*)
8 lemma ConvBarcanIndl: " $\exists \varphi(x). \Box(\forall x. \varphi(x)) \rightarrow (\forall x. \Box(\varphi(x)))$ " nitpick oops (*countermodel*)
9 (**Barcan and Converse Barcan Formula for Properties (type e⇒i⇒bool)**)
10 lemma BarcanPredl: " $\exists \varphi(x). \Box(\varphi(x)) \rightarrow (\Box(\forall x. \varphi(x)))$ " by simp
11 lemma ConvBarcanPredl: " $\exists \varphi(x). \Box(\forall x. \varphi(x)) \rightarrow (\forall x. \Box(\varphi(x)))$ " by simp
```

Nitpicking formula...

Nitpick found a counterexample for card $e = 1$ and card $i = 2$:

Skolem constants:

```
v = i1
w = i2
x = (λx. _) (i1 := False, i2 := True)
```

Constants:

```
P = (λx. _)
((λx. _)((e1, i1) := True, (e1, i2) := True), i1) := True,
((λx. _)((e1, i1) := True, (e1, i2) := True), i2) := True,
((λx. _)((e1, i1) := True, (e1, i2) := False), i1) := False,
((λx. _)((e1, i1) := True, (e1, i2) := False), i2) := False,
((λx. _)((e1, i1) := False, (e1, i2) := True), i1) := False,
((λx. _)((e1, i1) := False, (e1, i2) := True), i2) := False,
((λx. _)((e1, i1) := False, (e1, i2) := False), i1) := False,
((λx. _)((e1, i1) := False, (e1, i2) := False), i2) := False)
existsAt = (λx. _)((e1, i1) := True, (e1, i2) := True)
(r) = (λx. _)((i1, i1) := True, (i1, i2) := True, (i2, i1) := True, (i2, i2) := True)
```

Output Query Sledgehammer Symbols

64,49 (3629/5788) (isabelle,isabelle,UTF-8–Isabelle) 1 n m r o UG 320/571MB 6:54 PM

DEMO: Anderson's Variant

The screenshot shows the Isabelle2018-HOL interface with a proof script named `AndersonProof.thy`. The script defines constants and axioms related to a filter `P` and elements `aw`, `peter`, and `mary`. It includes a nitpick counterexample search and a lemma about ultrafilters.

```
73 (* Some tests *)
74 consts aw::i peter::e mary::e supreme_being::e loves::"e⇒e⇒i⇒bool"
75 axiomatization where t1: "[P (λx. loves x mary)]" and
76   t2: "¬(peter = mary)" and t3: "¬(peter = supreme_being)" and t4: "¬(mary = supreme_being)" and
77   t5: "¬(Gλ peter aw)" and t6: "¬(Gλ mary aw)" and t7: "¬(loves peter mary aw)"
78 consts P_prime::"(e⇒i⇒bool)⇒i⇒bool"
79 axiomatization where t8: 'P_prime = P'
80
81 lemma "(ultrafilter P' aw) ∧ ¬(ultrafilter P aw)"
82   nitpick[ user_axioms,show_all,format=3 ] (*counterm.*)
83   nitpick[satisfy,user_axioms,show_all,format=3,timeout=100] oops
```

The proof state window displays a long list of generated constants, all starting with `(λx. _)` and involving various combinations of `i1`, `i2`, `e1`, `e2`, `e3`, and `e`.

At the bottom, the output pane shows the command `nitpick[satisfy,user_axioms,show_all,format=3,timeout=100]` followed by the word `oops`.

Navigation tabs at the bottom include Output, Query, Sledgehammer, and Symbols.

DEMO: Anderson's Variant

The screenshot shows the Isabelle2018/HOL interface with a modified proof script named `AndersonProof.thy`. The interface includes a toolbar with various icons, a menu bar, and a sidebar with tabs for Documentation, Sidekick, State, and Theories.

```
1 theory AndersonProof imports IHOML
2 begin
3 (*Positiveness/perfection: uninterpreted constant symbol*)
4 consts positiveProperty::"(e⇒i⇒bool)⇒i⇒bool" ("P")
5 (*Some auxiliary definitions*)
6 definition h3 (infix "⇒" 60) where "X ⇒ Y ≡ □(∀z. X z → Y z)"
7 (**Part I**)
8 (*D1*) definition GA ("GA") where "GA ≡ λx. ∀Y. (P Y) ↔ □(Y x)"
9 (*A1a*) axiomatization where Ala:"[∀X. P (→X) → ¬(P X)]"
10 (*A2*) axiomatization where A2: "[∀X Y. (P X ∧ (X ⇒ Y)) → P Y]"
11 (*T1*) theorem T1: "[∀X. P X → ◇∃E X]" using Ala A2 h3_def by metis
12 (*T2*) axiomatization where T2: "[P GA]" (*here we postulate T2 instead of proving it*)
13 (*T3*) theorem T3: "[◇∃E GA]" by (metis Ala A2 T2 h3_def)
14 (**Part II**)
15 (*Logic KB*) axiomatization where symm: "∀x y. x r y → y r x"
16 (*A4*) axiomatization where A4: "[∀X. P X → □(P X)]"
17 (*D2*) abbreviation essA ("EA") where "EA Y x ≡ (∀z. □(Z x) ↔ Y ⇒ Z)"
18 (*T4*) theorem T4: "[∀x. GA x → (EA GA x)]" by (metis A2 GA_def T2 symm h3_def)
19 (*D3*) abbreviation NEA ("NEA") where "NEA x ≡ (∀w. (∀Y. EA Y x → □∃E Y) w)"
20 (*A5*) axiomatization where A5: "[P NEA]"
21 (*T5*) theorem T5: "[◇∃E GA] → [□∃E GA]" by (metis A2 GA_def T2 symm h3_def)
22 (*T6*) theorem T6: "[□∃E GA]" using T3 T5 by blast
```

The proof state at the bottom shows the goal: `theorem T6: [□mexistsActB GA]`. The interface includes checkboxes for Proof state and Auto update, and a search bar. The status bar at the bottom shows the date and time: 22.55 (1320/5444) (isabelle,isabelle,UTF-8-Isabelle) | n m r o UG 526/651MB 7:50 PM.