

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

Christoph Benz Müller (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]

Formal Methods and Science in Philosophy III

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:

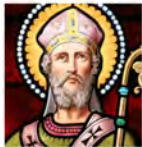
- ▶ Ed Zalta (& co) with PROVER9 at Stanford
- ▶ John Rushby with PVS at SRI

[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

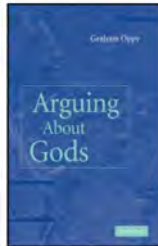
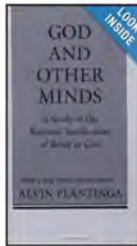
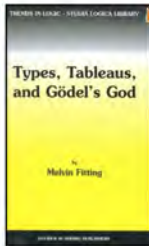


Leibniz



ÖSTERREICH €0.55

Gödel



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

Ontologische Beweise Feb 10, 1970

$P(\varphi)$ φ is positive (i.e. $\varphi \in P$)

At 1 $P(\varphi) \cdot P(\psi) \supset P(\varphi \wedge \psi)$ At 2 $P(\varphi) \supset P(\Box \varphi)$

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

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

$P \supset Nq \equiv N(p \supset q)$ Necessity

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

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

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

Ax 3 $P(E)$

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

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

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

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

M = possibility

any two instances of x are nec. equivalent

exclusive or * and for any number of Atom-words

$M(\exists x) G(x)$ means ^{the system} all possible This is

At 4: $P(\varphi) \cdot \varphi \supset N \psi$

~~hence~~ $\begin{cases} x=x & \text{is possible} \\ x \neq x & \text{is not} \end{cases}$

But if a system S of it would mean that a (positive) would be $x \neq x$

Positive means positive in the moral sense (independently of the accidental structure of the world). Only in the actual time. It means also moral "attribution" as opposed to "privation (or containing privation)". This is part of the proof

If φ is positive: $(x) N \sim \varphi(x) \cdot \text{hence } \varphi(x) \supset N \sim \varphi(x)$

hence $x \neq x$ positive or $x=x$ is necessary At 4

or the exp. of pos. At 4
 i.e. the normal form in terms of elem. prop. contains a member without negation.



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

Ontologische Beweise Feb. 10, 1970

$P(\varphi)$ φ is positive (i.e. $\varphi \in P$)

At. 1 $P(\varphi) \cdot P(\psi) \supset P(\varphi \cdot \psi)$ $\wedge \neg P(\varphi) \vee \neg P(\psi)$

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

[2] $\varphi \text{ Em } x = (\psi) [\varphi(x) \supset \psi(x)]$ (Essence of x)

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

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

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

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

Ax 3 $P(E)$

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

$M(x) G(x)$ means ^{the system of} all pos. props. is compatible
 This is true because of:

At. 4: $P(\varphi) \cdot \varphi \supset N \psi \supset P(\psi)$ which impl.

~~is~~ $\begin{cases} x=x & \text{is positive} \\ x \neq x & \text{is negative} \end{cases}$

But if a system S of pos. props. were inconsistent it would mean that the prop. S (which is positive) would be $x \neq x$

Positive means positive in the moral aesthetic sense (independently of the accidental structure of the world). Only in the ^{pure} at. sense. It also means "attribution" as opposed to "privation".

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)

Ontologische Beweise

Feb 10, 1970

$M(x) \supset E(x)$ means: all x that are possible are actual

In the end we prove

- **Necessarily (N)**, there exists God.

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

$\neg P(\varphi) \supset N \neg P(\varphi)$ from the nature of the property

Th. $G(x) \supset GEm.x$

Df. $E(x) \equiv \lambda \varphi [\varphi Em.x \supset N \exists x \varphi(x)]$ necessary E is the

$Ax \exists P(E)$

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

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

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

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

any two theorems are not contradictory
exclusive or * and for any number of theorems

(positive) would be $x \neq x$

Positive means positive in the moral sense (independently of the accidental structure of the world). Only in the actual time. It also means "attribution" as opposed to "privation" (or containing privation). This is a positive property.

If φ is a property: $(x) N \neg \varphi(x) \supset \varphi(x) \supset x \neq x$

hence $x \neq x$ positive or $x = x$ is necessary. At the end of the proof.

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

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

Ontologische Beweise Feb 10, 1970

$P(\varphi)$ φ is positive (i.e. $\varphi \in P$)

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

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

[2] $\varphi \text{ Ens } x \equiv (\psi) [\psi(x) \supset N(\exists y) [\varphi(y) \supset \psi(y)]]$ (Ensures x)

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

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

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

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

Ax 3 $P(E)$

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

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

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

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

M = possibility

any two instances of x are nec. equivalent

exclusive or * and for any number of Attributands

$M(\exists x) G(x)$ means "all possible" ^(the system) This is:

At 4: $P(\varphi) \cdot \varphi \supset N \psi$

~~hence~~ $\begin{cases} x=x & \text{is possible} \\ x \neq x & \text{is not} \end{cases}$

But if a system S of \mathcal{L} would mean that the same prop. A (which is positive) would be $x \neq x$

Positive means positive in the modal sense (independently of the accidental structure of the world). Only in the actual time. It means also essential attribution as opposed to "privation (or containing privation)". This is part of the proof.

If φ is positive: $(x) N \neg \varphi(x) \equiv \text{Necessary } \varphi(x) \supset x \neq x$

hence $x \neq x$ positive iff $x=x$ is necessary At 4

or the equiv. of pos. At 4: $\varphi(x) \supset N \neg \varphi(x)$ i.e. the normal form in terms of elem. prop. contains a member without negation.



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

Ontologische Beweise Feb. 10, 1970

$P(\varphi)$ φ is positive (i.e. $\varphi \in P$.)

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

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

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

$P \supset N \sim P$ (Necessity)

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

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

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

Ax 3 $P(E)$

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

$M(\exists x) G(x)$ means ^{the system of} all pos. props. is: consistent
 This is true because of:

At. 4: $P(\varphi) \cdot \varphi \supset N \psi \supset P(\psi)$ which impl.

~~is~~ $\begin{cases} x=x & \text{is positive} \\ x \neq x & \text{is negative} \end{cases}$

But if a system S of pos. props. were inconsistent it would mean that the prop. S (which is positive) would be $x \neq x$

Positive means positive in the modal sense (independently of the accidental structure of the world). ^{pure} Only in the actual world. It also means "attribution" as opposed to "privation" (or certain negation). This is the modal sense.

(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.

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

$\Box P$

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 $\Diamond P$

$$\Box P$$

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 $\Diamond P$

$$\Box P$$

P is necessary

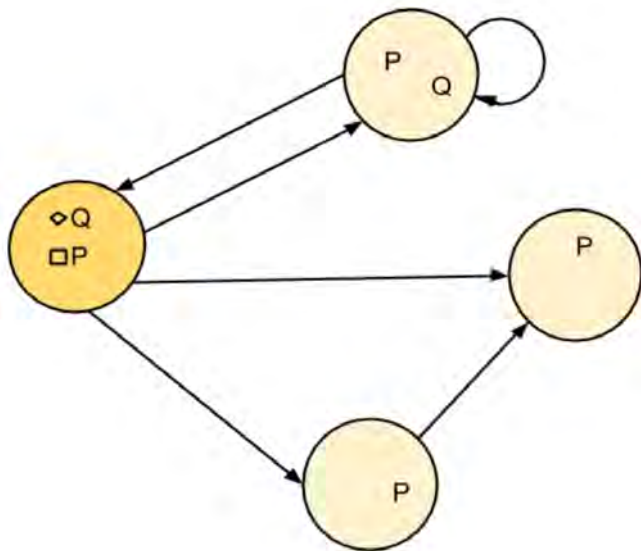
$$\Diamond P$$

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:

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

Standard translation extended to quantifiers:

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

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

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

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

Standard translation:

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

Standard translation extended to quantifiers:

- ▶ in HOL: $\forall x . \phi x$ shorthand for $\Pi(\lambda x . \phi x)$
- ▶ $\Box \forall x . P x$ is represented as $\Box \Pi'(\lambda x_\alpha . \lambda w_i . P x w)$ where $\Pi' := \lambda \Phi_{\alpha \rightarrow i \rightarrow o} . \lambda w_i . \Pi(\lambda x_\alpha . \Phi x w)$ and \Box is resolved as above

$$\begin{aligned}\Box \forall x . P x &\equiv \Box \Pi'(\lambda x . \lambda w . P x w) \\ &\equiv \Box((\lambda \Phi . \lambda w . \Pi(\lambda x . \Phi x w))(\lambda x . \lambda w . P x w)) \\ &\equiv \Box(\lambda w . \Pi(\lambda x . (\lambda w . P x w) x w)) \\ &\equiv \Box(\lambda w . \Pi(\lambda x . P x w)) \\ &\equiv (\lambda \varphi . \lambda w . \forall v . (R w v \rightarrow \varphi v))(\lambda w . \Pi(\lambda x . P x w)) \\ &\equiv (\lambda \varphi . \lambda w . \Pi(\lambda v . R w v \rightarrow \varphi v))(\lambda w . \Pi(\lambda x . P x w)) \\ &\equiv (\lambda w . \Pi(\lambda v . R w v \rightarrow (\lambda w . \Pi(\lambda x . P x w) v))) \\ &\equiv (\lambda w . \Pi(\lambda v . R w v \rightarrow \Pi(\lambda x . P x v))) \\ &\equiv (\lambda w . \forall v . (R w v \rightarrow \forall x . P x v))\end{aligned}$$

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

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

Standard translation:

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

Standard translation extended to quantifiers:

- ▶ in HOL: $\forall x . \phi x$ shorthand for $\Pi(\lambda x . \phi x)$
- ▶ $\Box \forall x . P x$ is represented as $\Box \Pi'(\lambda x_\alpha . \lambda w_i . P x w)$ where $\Pi' := \lambda \Phi_{\alpha \rightarrow i \rightarrow o} . \lambda w_i . \Pi(\lambda x_\alpha . \Phi x w)$ and \Box is resolved as above

$$\begin{aligned}\Box \forall x . P x &\equiv \Box \Pi'(\lambda x . \lambda w . P x w) \\ &\equiv \Box((\lambda \Phi . \lambda w . \Pi(\lambda x . \Phi x w))(\lambda x . \lambda w . P x w)) \\ &\equiv \Box(\lambda w . \Pi(\lambda x . (\lambda w . P x w) x w)) \\ &\equiv \Box(\lambda w . \Pi(\lambda x . P x w)) \\ &\equiv (\lambda \varphi . \lambda w . \forall v . (R w v \rightarrow \varphi v))(\lambda w . \Pi(\lambda x . P x w)) \\ &\equiv (\lambda \varphi . \lambda w . \Pi(\lambda v . R w v \rightarrow \varphi v))(\lambda w . \Pi(\lambda x . P x w)) \\ &\equiv (\lambda w . \Pi(\lambda v . R w v \rightarrow (\lambda w . \Pi(\lambda x . P x w)) v)) \\ &\equiv (\lambda w . \Pi(\lambda v . R w v \rightarrow \Pi(\lambda x . P x v))) \\ &\equiv (\lambda w . \forall v . (R w v \rightarrow \forall x . P x v))\end{aligned}$$

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

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

Standard translation:

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

Standard translation extended to quantifiers:

- ▶ in HOL: $\forall x . \phi x$ shorthand for $\Pi(\lambda x . \phi x)$
- ▶ $\Box \forall x . P x$ is represented as $\Box \Pi'(\lambda x_\alpha . \lambda w_i . P x w)$ where
 $\Pi' := \lambda \Phi_{\alpha \rightarrow i \rightarrow o} . \lambda w_i . \Pi(\lambda x_\alpha . \Phi x w)$ and \Box is resolved as above

$$\begin{aligned}\Box \forall x . P x &\equiv \Box \Pi'(\lambda x . \lambda w . P x w) \\ &\equiv \Box((\lambda \Phi . \lambda w . \Pi(\lambda x . \Phi x w))(\lambda x . \lambda w . P x w)) \\ &\equiv \Box(\lambda w . \Pi(\lambda x . (\lambda x . \lambda w . P x w) x w)) \\ &\equiv \Box(\lambda w . \Pi(\lambda x . P x w)) \\ &\equiv (\lambda \varphi . \lambda w . \forall v . (R w v \rightarrow \varphi v))(\lambda w . \Pi(\lambda x . P x w)) \\ &\equiv (\lambda \varphi . \lambda w . \Pi(\lambda v . R w v \rightarrow \varphi v))(\lambda w . \Pi(\lambda x . P x w)) \\ &\equiv (\lambda w . \Pi(\lambda v . R w v \rightarrow (\lambda w . \Pi(\lambda x . P x w) v))) \\ &\equiv (\lambda w . \Pi(\lambda v . R w v \rightarrow \Pi(\lambda x . P x v))) \\ &\equiv (\lambda w . \forall v . (R w v \rightarrow \forall x . P x v))\end{aligned}$$

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

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

Standard translation:

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

Standard translation extended to quantifiers:

- ▶ in HOL: $\forall x . \phi x$ shorthand for $\Pi(\lambda x . \phi x)$
- ▶ $\Box \forall x . P x$ is represented as $\Box \Pi'(\lambda x_\alpha . \lambda w_i . P x w)$ where
 $\Pi' := \lambda \Phi_{\alpha \rightarrow i \rightarrow o} . \lambda w_i . \Pi(\lambda x_\alpha . \Phi x w)$ and \Box is resolved as above

$$\begin{aligned}\Box \forall x . P x &\equiv \Box \Pi'(\lambda x . \lambda w . P x w) \\ &\equiv \Box((\lambda \Phi . \lambda w . \Pi(\lambda x . \Phi x w))(\lambda x . \lambda w . P x w)) \\ &\equiv \Box(\lambda w . \Pi(\lambda x . (\lambda x . \lambda w . P x w) x w)) \\ &\equiv \Box(\lambda w . \Pi(\lambda x . P x w)) \\ &\equiv (\lambda \varphi . \lambda w . \forall v . (R w v \rightarrow \varphi v))(\lambda w . \Pi(\lambda x . P x w)) \\ &\equiv (\lambda \varphi . \lambda w . \Pi(\lambda v . R w v \rightarrow \varphi v))(\lambda w . \Pi(\lambda x . P x w)) \\ &\equiv (\lambda w . \Pi(\lambda v . R w v \rightarrow (\lambda w . \Pi(\lambda x . P x w) v))) \\ &\equiv (\lambda w . \Pi(\lambda v . R w v \rightarrow \Pi(\lambda x . P x v))) \\ &\equiv (\lambda w . \forall v . (R w v \rightarrow \forall x . P x v))\end{aligned}$$

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

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

```
Isabelle2018/HOL - IHOML.nix
IHOML.thy (~/.GITHUBS/chrisgitlab/talks/2019-Dubrovnik/)

1 theory IHOML imports Main
2 begin
3   typedecl i (* possible worlds *) typedecl e (* individuals *)
4   (*Logical Operators as Truth-Sets *)
5   abbreviation mnnot ("¬" [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"∧"51) where "φ ∧ ψ ≡ λw. (φ w) ∧ (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
```

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.

$$\Diamond \exists x G(x)$$

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.

$$\Box \exists x G(x)$$

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.

$$\Box \exists x G(x)$$

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

GoedelProof.thy (~chrisGIT/tex/talks/2018-AISSQ/)

```
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 ↔ (∀Y.(Z Y) → {Y x})))"
8 definition h3 (infix "⇒" 60) where "X ⇒ Y ≡ □(∀z. 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 A1a: "[∀X. P (→X) → ¬(P X)]" and A1b: "[∀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 → ◇3 X]" by (metis A1a A2 h3_def)
16 (*T2*) theorem T2: "[P G]" proof -
17   {have 1: "∀v. ∃Z X. (P G v 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: "[◇3 G]" using T1 T2 by simp
21
22 (**Part II**)
23 (*T4*) theorem T4: "[◇3 G]" using T1 T2 by simp
```

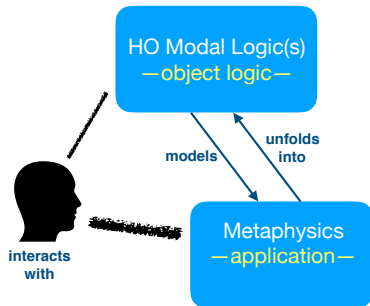
Documentation Sidekick State Theories

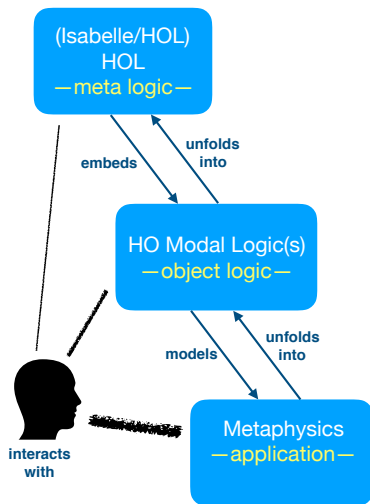
theorem U3: $P \subseteq P \wedge P \subseteq$
Undefined fact: "T6"

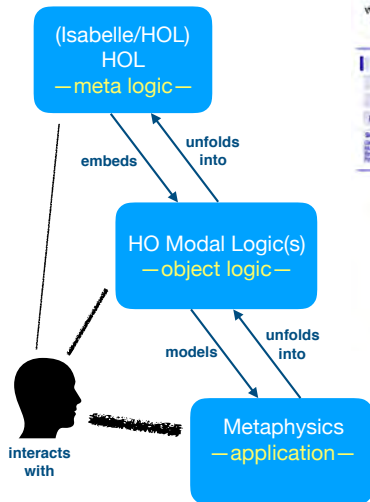
00:00 | ————— -02:26

Output Query Sledgehammer Symbols









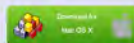
Isabelle



What is Isabelle?

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

Now available: Isabelle2017 (October 2017)



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

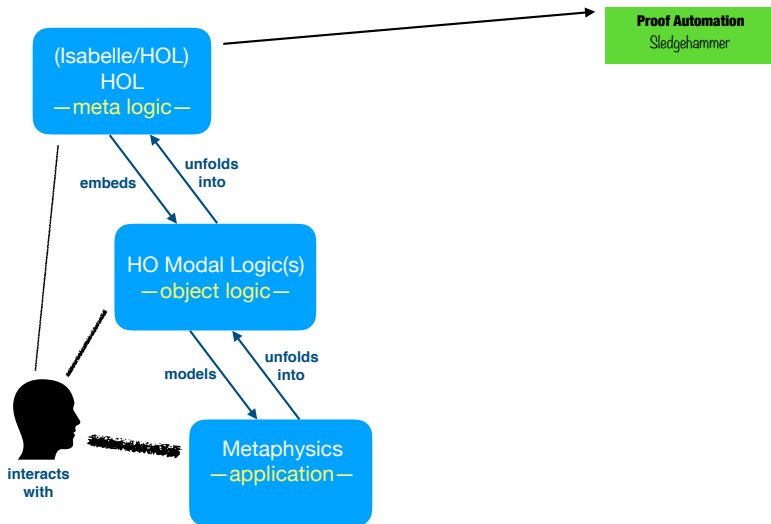
Some notable changes:

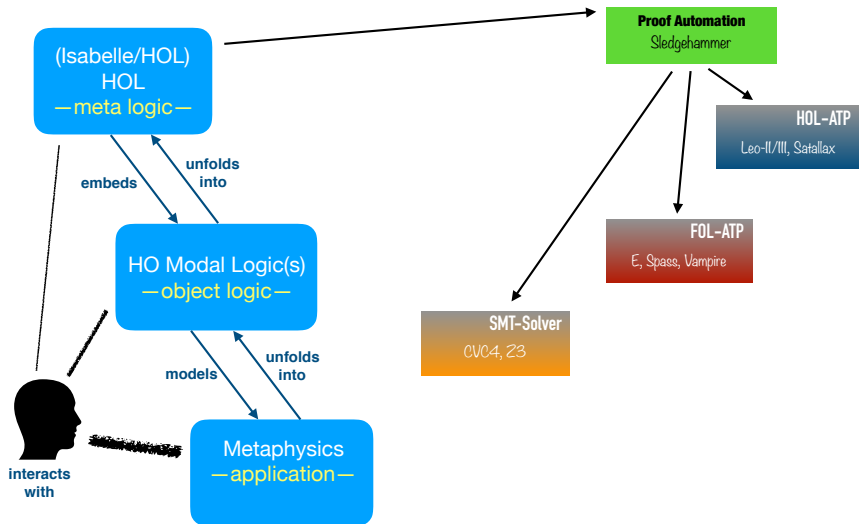
- Experimental support for Visual Studio Code as alternative IDE front-end.
- Improved Isabelle/Proof Power (IDE) management of session sources independently of editor buffers, removal of unused theorems, explicit indication of theory status, more careful auto-indentation.
- Session-qualified theory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

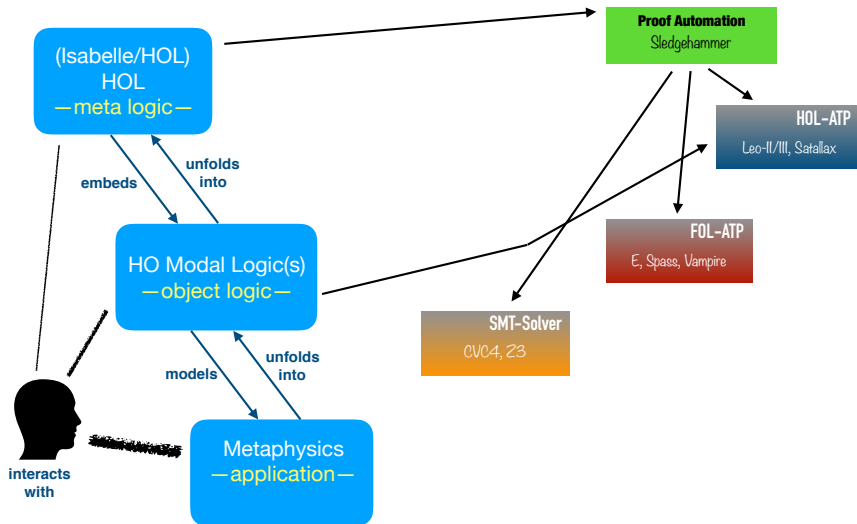
See also the cumulative [NEWS](#).

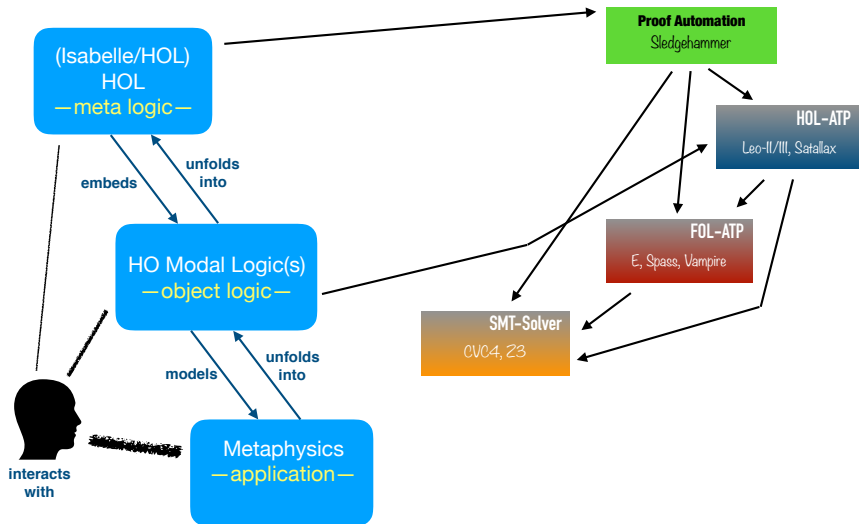
Distribution & Support

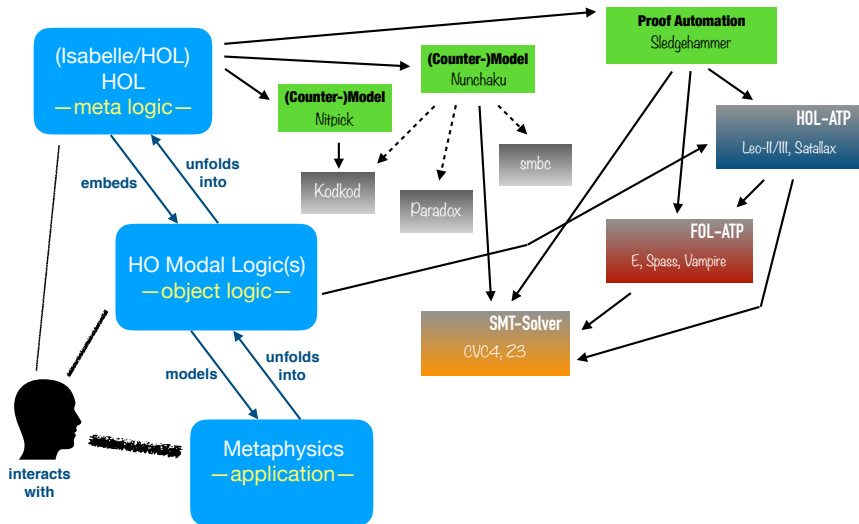
Isabelle is distributed for free under a conglomerate of open-source licenses, but the main code-base is subject to BSD-style regulations. The application bundles include source and binary packages and documentation, see the detailed [Installation Instructions](#). A vast collection of Isabelle examples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

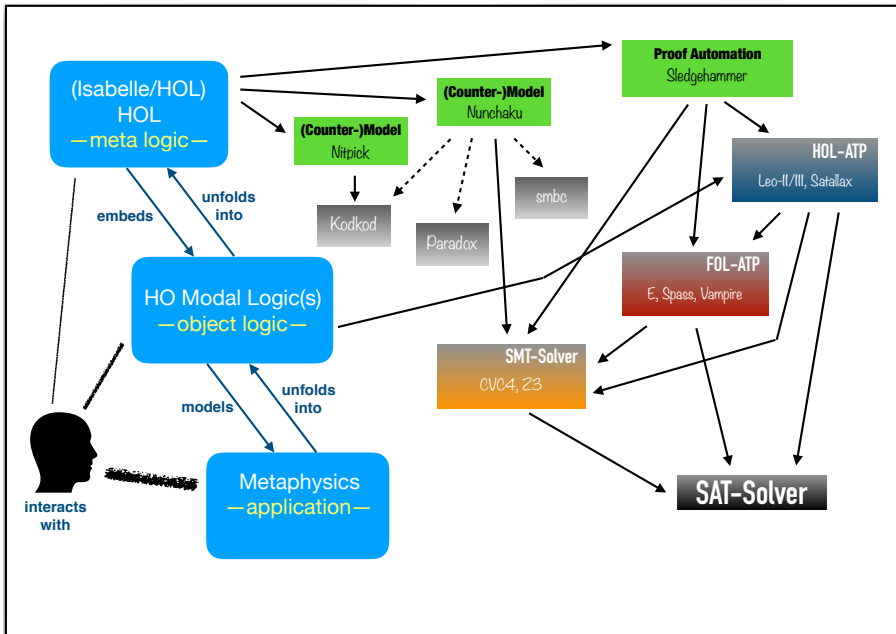


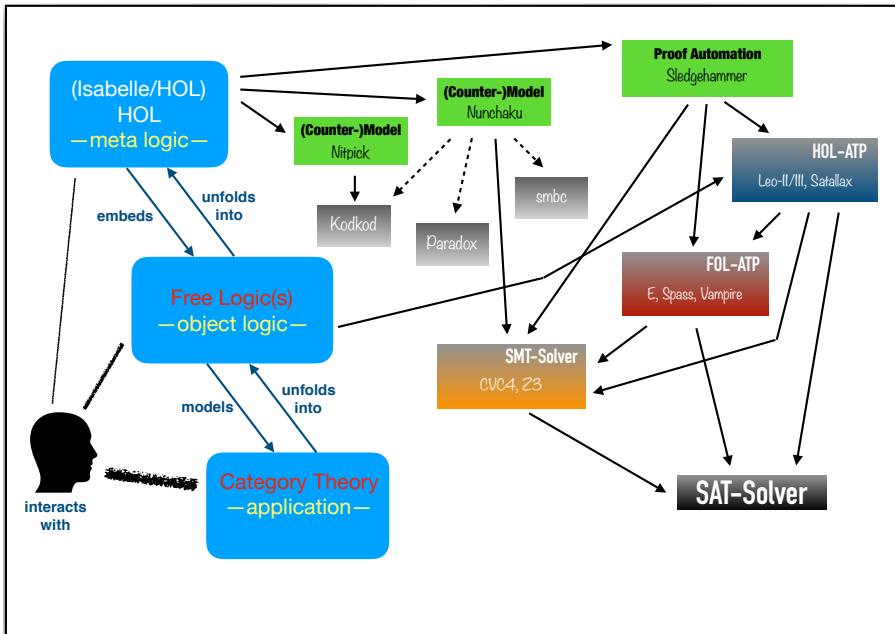


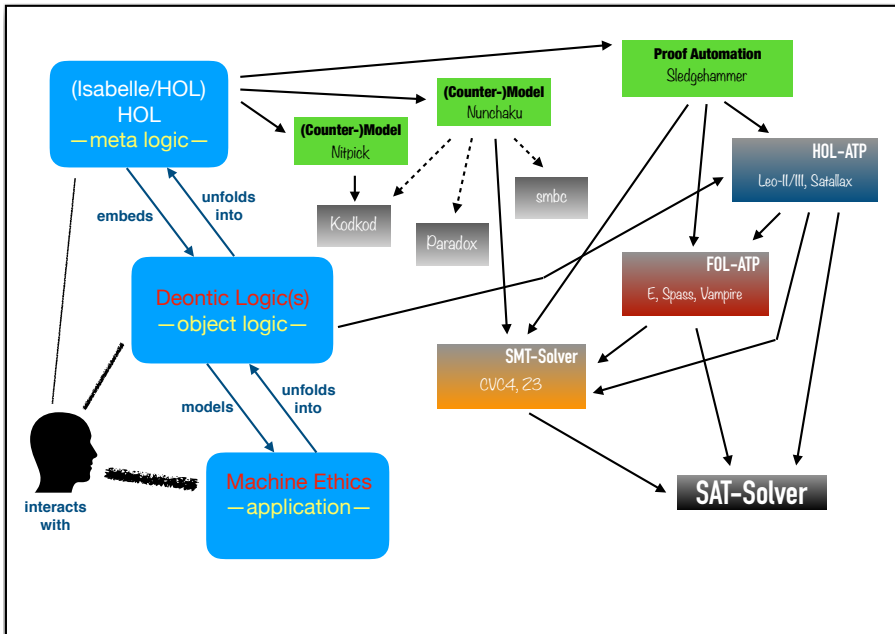


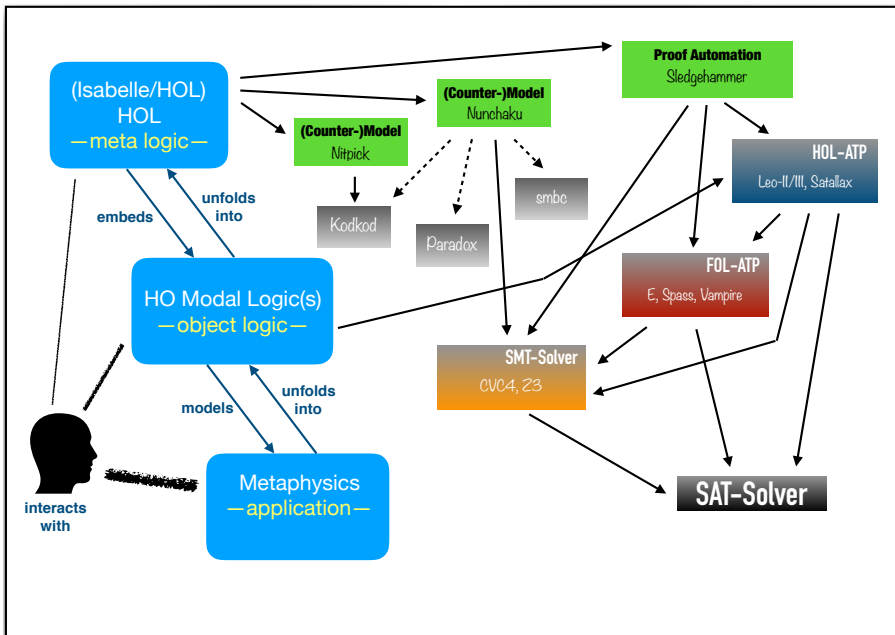














Results of our Experiments (jww B. Woltzenlogel-Paleo)
(see also [Savijnanam 2017], [IJCAI 2016], [ECAI 2014])



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.



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.



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.



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.

Variant of Kurt Gödel (1970)

- ▶ the premises are **inconsistent/contradictory**
(since they imply $\Diamond\Box\perp$)
- ▶ **everything follows (ex false quod libet)!**
- ▶ 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.



Variant of Kurt Gödel (1970)

- ▶ the premises are **inconsistent/contradictory**
(since they imply $\Diamond\Box\perp$)
- ▶ **everything follows (ex false quod libet)!**
- ▶ 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.



Variant of Kurt Gödel (1970)

- ▶ the premises are **inconsistent/contradictory**
(since they imply $\Diamond\Box\perp$)
- ▶ **everything follows (ex false quod libet)!**
- ▶ 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.



Variant of Kurt Gödel (1970)

- ▶ the premises are **inconsistent/contradictory**
(since they imply $\Diamond\Box\perp$)
- ▶ **everything follows (ex falso quod libet)!**
- ▶ 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**

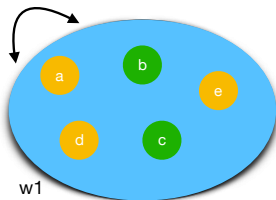


Part B

Some Relevant Pillar Stones for this Talk

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

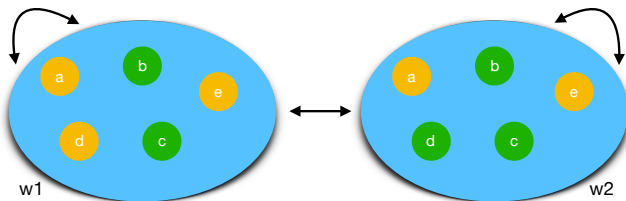
Example predicate: **IsChessGrandmaster**



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

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

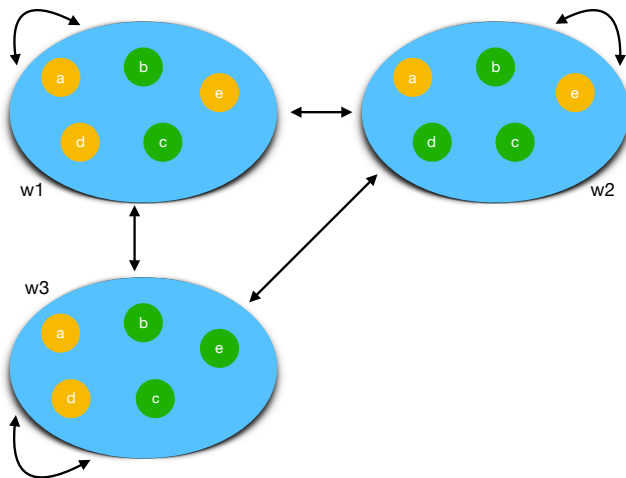
Example predicate: **IsChessGrandmaster**



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

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

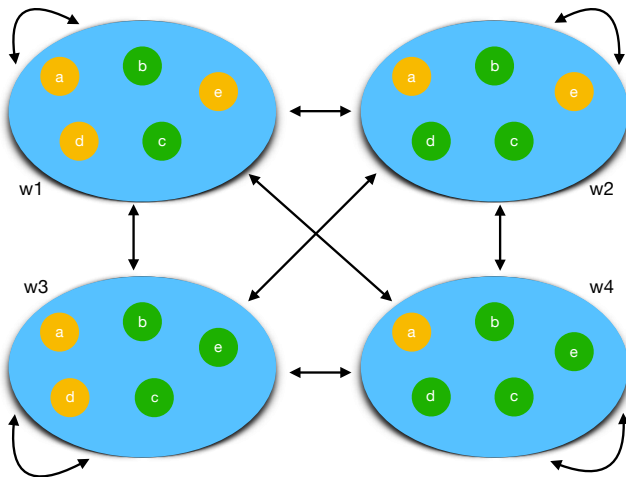
Example predicate: **IsChessGrandmaster**



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

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

Example predicate: **IsChessGrandmaster**



— Intensional Predicate **IsChessGrandmaster** (ICG)

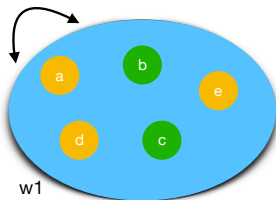
— Extensions of **ICG** in possible worlds w1-w4:

ICG w1 = {b,c} **ICG** w2 = {b,c,d} **ICG** w3 = {b,c,e} **ICG** w4 = {b,c,d,e}

“Rigidly Intensionalised Extension” of a Predicate

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

Example predicate: **IsChessGrandmaster**

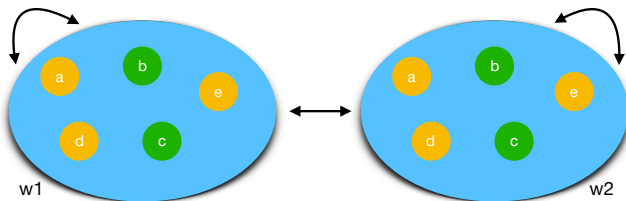


- Intensional Predicate **IsChessGrandmaster** (**ICG**)
- Rigidified extension of **ICG** in world w_1 :
ICG $w_1 = \{b, c\}$

“Rigidly Intensionalised Extension” of a Predicate

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

Example predicate: **IsChessGrandmaster**

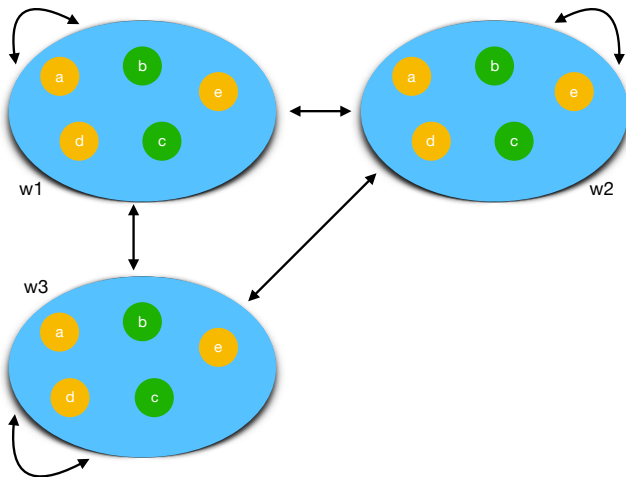


- Intensional Predicate **IsChessGrandmaster** (ICG)
- Rigidified extension of **ICG** in world w_1 :
ICG $w_1 = \{b, c\}$ **ICG** $w_2 = \{b, c\}$

“Rigidly Intensionalised Extension” of a Predicate

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

Example predicate: **IsChessGrandmaster**

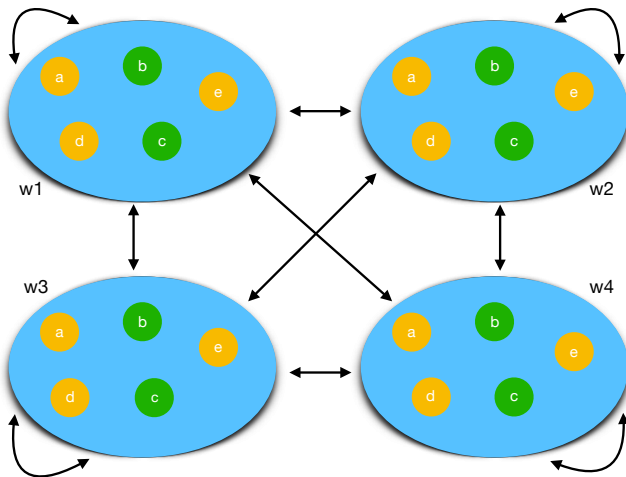


- Intensional Predicate **IsChessGrandmaster** (**ICG**)
- Rigidified extension of **ICG** in world w_1 :
ICG $w_1 = \{b, c\}$ **ICG** $w_2 = \{b, c\}$ **ICG** $w_3 = \{b, c\}$

“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 w_1 :

ICG $w_1 = \{b, c\}$ **ICG** $w_2 = \{b, c\}$ **ICG** $w_3 = \{b, c\}$ **ICG** $w_4 = \{b, c\}$

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:

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

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

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:

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

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

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

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:

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

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

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

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:

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

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

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

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:

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

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

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

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

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:

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

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

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

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

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:

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

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

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

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

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:

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

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

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

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

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

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:

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

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

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

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

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

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:

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

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

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

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

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

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:

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

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

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

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

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

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

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:

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

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

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

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

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

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

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:

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

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

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

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

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

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

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:

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

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

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

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

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

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

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:

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

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

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

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

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

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

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:

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

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

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

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

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

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

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:

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

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

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

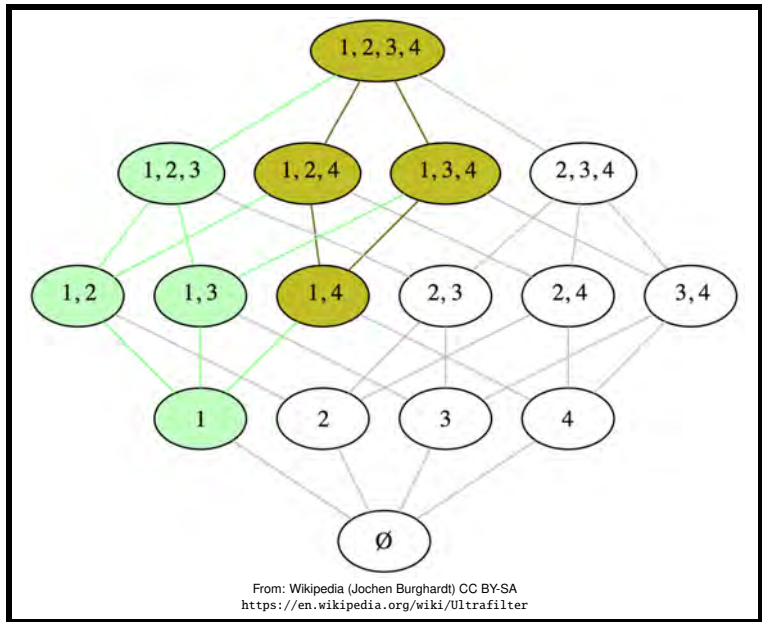
$$U^2 = \{ \qquad \qquad \qquad \{1, 4\}, \qquad \qquad \{1, 2, 4\}, \{1, 3, 4\}, \{1, 2, 3, 4\} \}$$

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

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

$\mathbf{1}$ is element of all sets in U ($\mathbf{1}$ 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

A2 **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 D3 Necessary existence of an individual is the necessary instantiation of all its essences.

From A5 Necessary existence is a positive property.

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

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

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

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

“Modal Collapse” is implied by these axioms: $\varphi \supset \Box \varphi$

► determinism

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

► \mathcal{P} is an ultrafilter

Let \mathcal{P}' be the set of “rigidly intensionalised extensions” of positive properties. ($\mathcal{P}'\varphi := \mathcal{P}(\downarrow \varphi)$)

We can prove:

► \mathcal{P}' is an ultrafilter

► $\mathcal{P} = \mathcal{P}'$

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 ↔ (∀Y.(Z Y) → (Y x))))"
8 definition h3 (infix "⇒" 60) where "X ⇒ Y ≡ □(∀Z. 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 A1a: "[∀X. P (→X) → ¬(P X)]" and A1b: "[∀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 A1a 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
22 (**Part II**)
23 (*Logic KB*) axiomatization where symm: "symmetric aRel"
24 (*A4*) axiomatization where A4: " $\forall X. P\ X \rightarrow \Box(P\ X)$ "
25 (*D2*) definition ess ("E") where " $E\ Y\ x \equiv (Y\ x) \wedge (\forall Z. Z\ x \rightarrow Y \Rightarrow Z)$ "
26 (*T4*) theorem T4: " $\forall X. G\ x \rightarrow (E\ G\ x)$ " by (metis Alb A4 G_def h3_def ess_def)
27 (*D3*) definition NE ("NE") where " $NE\ x \equiv (\lambda w. (\forall Y. E\ Y\ x \rightarrow \Box^E Y)\ w)$ "
28 (*A5*) axiomatization where A5: " $\Box NE$ "
29 (*T5*) theorem T5: " $\Box(\Box^E G) \rightarrow \Box^E G$ " by (metis A5 G_def NE_def T4 symm)
30 (*T6*) theorem T6: " $\Box\Box^E G$ " using T3 T5 by blast
31
32 (**Consistency**)
33 lemma True nitpick[satisfy] oops (*Model found by Nitpick: the axioms are consistent*)
34
35 (**Modal Collapse**)
36 lemma ModalCollapse: " $\forall \Phi. (\Phi \rightarrow (\Box\ \Phi))$ " proof -
37   {fix w fix Q
38     have " $\forall x. G\ x\ w \rightarrow (\forall Z. Z\ x \rightarrow \Box(\forall^E z. G\ z \rightarrow Z\ z))\ w$ " by (metis Alb A4 G_def)
39     hence 1: " $(\exists x. G\ x\ w) \rightarrow ((Q \rightarrow \Box(\forall^E z. G\ z \rightarrow Q))\ w)$ " by force
40     have " $\exists x. G\ x\ w$ " using T3 T6 symm by blast
41     hence " $(Q \rightarrow \Box Q)\ w$ " 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: " $\forall X. \neg P\ X \rightarrow \Box(\neg P\ X)$ " using A4 symm by blast
47   definition h4 ("N") where " $N\ X \equiv \neg P\ X$ "
48 (*C3*) theorem C3: " $\forall X. N\ X \rightarrow \Box(N\ X)$ " by (simp add: C2 h4_def)
```

Ontological Argument: Variant by Gödel/Scott

```

49
50 (**Positive Properties and Ultrafilters**)
51 abbreviation emptySet ("0") where "0 ≡ λx w. False"
52 abbreviation entails (infixr "⊆" 51) where "φ ⊆ ψ ≡ ∀x w. φ x w → ψ x w"
53 abbreviation andPred (infixr "∩" 51) where "φ ∩ ψ ≡ λx w. φ x w ∧ ψ x w"
54 abbreviation negpred ("¬" [52] 53) where "¬φ ≡ λx w. ¬φ x w"
55 abbreviation "ultrafilter Φ cw ≡
56   ¬(Φ 0 cw)
57   ∧ (∀φ. ∀ψ. (Φ φ cw ∧ Φ ψ cw) → (Φ (φ ∩ ψ) cw))
58   ∧ (∀φ::e⇒i⇒bool. ∀ψ::e⇒i⇒bool. (Φ φ cw ∨ Φ (¬φ) cw) ∧ ¬(Φ φ cw ∧ Φ (¬φ) cw))
59   ∧ (∀φ::e⇒i⇒bool. ∀ψ::e⇒i⇒bool. (Φ φ cw ∧ φ ⊆ ψ) → Φ ψ cw)"
60 lemma helpA: "∀w. ¬(P 0 w)" using T1 by auto
61 lemma helpB: "∀φ ψ w. (P φ w ∧ P ψ w) → (P (φ ∩ ψ) w)" by (smt Alb G_def T3 T6 symm)
62 lemma helpC: "∀φ ψ w. (P φ w ∨ P (¬φ) w) ∧ ¬(P φ w ∧ P (¬φ) w)" using Ala Alb by blast
63 lemma helpD: "∀φ ψ w. (P φ w ∧ (φ ⊆ ψ)) → P ψ w" by (metis Alb A4 G_def T1 T6)
64
65 (*U1*) theorem U1: "∀w. ultrafilter P w" using helpA helpB helpC helpD by simp
66
67 (*⌊φ⌋ converts an extensional object φ into 'rigid' intensional one*)
68 abbreviation trivialConversion ("⌊_⌋") where "⌊φ⌋ ≡ (λw. φ)"
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 nextPredArg (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 ⌊0⌋) w)" using T1 by blast
75 lemma helpF: "∀φ ψ w. ((P ⌊φ⌋) w ∧ (P ⌊ψ⌋) w) → ((P ⌊(φ ∩ 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*)

```

Ontological Argument: Variant by Gödel/Scott

GoedelProof.thy (~chrisGIT/tex/talks/2018-AISSQ/)

```
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 ↔ (∀Y. (Z Y) → (Y x))))"
8 definition h3 (infix "⇒" 60) where "X ⇒ Y ≡ □(∀z. 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 A1a: "[∀X. P (→X) → ¬(P X)]" and A1b: "[∀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 → ◇∃ X]" by (metis A1a A2 h3_def)
16 (*T2*) theorem T2: "[P G]" proof -
17   {have 1: "∀w. ∃Z X. (P G V 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: "[◇∃ G]" using T1 T2 by simp
21
22 (**Part II**)
23 (*Axiomization where sum: "sumetric ∼B1")
```

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

00:00 | ————— -02:26

Output Query Sledgehammer Symbols

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

T3 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.

From A5 Necessary existence is a positive property.

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

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

T3 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 "Modal Collapse" is **not** implied by these axioms

A1b $\varphi \supset \Box \varphi$ (has countermodel)

A4 From $\Box \varphi$ no determinism

A2 T4 "positive properties (\mathcal{P})" are applied here to intensional properties.

From D3 We have:

T1 \mathcal{P} is **not** an ultrafilter (has countermodel)

From A5 Let \mathcal{P}' be the set of all "rigidly intensionalised extensions" of positive properties. We can prove:

T2 From T5 \mathcal{P}' is an ultrafilter

From T3 And $\mathcal{P} \neq \mathcal{P}'$

e.g. S 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 (*A1*) axiomatization where A1a: "[∀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 A1a 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

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

Ontological Argument: Variant by Anderson

```
1 theory AndersonProof imports IHOML
```

```
38 (**Positive Properties and Ultrafilters**)
```

```
39 abbreviation emptySet ("0") where "0 ≡ λx w. False"
```

```
40 abbreviation entails (infixr "⊆" 51) where "φ ⊆ ψ ≡ ∀x w. φ x w ⟶ ψ x w"
```

```
41 abbreviation andPred (infixr "∩" 51) where "φ ∩ ψ ≡ λx w. φ x w ∧ ψ x w"
```

```
42 abbreviation negpred ("¬" [52] 53) where "¬φ ≡ λx w. ¬φ x w"
```

```
43 abbreviation "ultrafilter" φ CW ≡
```

```
44   ¬(φ 0 CW)
```

```
45   ∧ (∀φ. ∀ψ. (φ ⊆ ψ ∧ φ 0 CW) ⟶ (ψ 0 CW))
```

```
46   ∧ (∀φ::e⇒i⇒bool. ∀ψ::e⇒i⇒bool. (φ ⊆ ψ ∨ φ (¬ψ) CW) ∧ ¬(φ ⊆ ψ ∧ φ (¬ψ) CW))
```

```
47   ∧ (∀φ::e⇒i⇒bool. ∀ψ::e⇒i⇒bool. (φ ⊆ ψ ∧ φ ⊆ ψ) ⟶ φ 0 CW)"
```

```
49 (*U1*) theorem U1: "∀w. ultrafilter P w" nitpick[user_axioms,format=2,show_all] oops (*counterterm.*)
```

```
50 lemma helpC: "∀φ ψ w. (P φ w ∨ P (¬φ) w) ∧ ¬(P φ w ∧ P (¬φ) w)" nitpick oops (*countermodel*)
```

```
52 (*⌊φ⌋ converts an extensional object φ into 'rigid' intensional one*)
```

```
53 abbreviation trivialConversion ("⌊⌋") where "⌊φ⌋ ≡ (λw. φ)"
```

```
54 (*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"*)
```

```
56 abbreviation nextPredArg (infix "⌊" 60) where "Q ⌊φ⌋ ≡ λw. Q (λx. ⌊φ x w⌋) w"
```

```
57 lemma "∀Q φ. Q φ = Q ⌊φ⌋" nitpick oops (*countermodel: the two notions are not the same*)
```

```
59 lemma helpE: "∀w. ¬((P ⌊0⌋) w)" using T1 by blast
```

```
60 lemma helpF: "∀φ ψ w. ((P ⌊φ⌋) w ∧ (P ⌊ψ⌋) w) ⟶ ((P ⌊φ ∩ ψ⌋) w)" by (smt GA_def T3 T5 symm)
```

```
61 lemma helpG: "∀w. ((P ⌊φ⌋) w ∨ (P ⌊¬φ⌋) w) ∧ ¬((P ⌊φ⌋) w ∧ (P ⌊¬φ⌋) w)" by (smt GA_def T3 T5 symm)
```

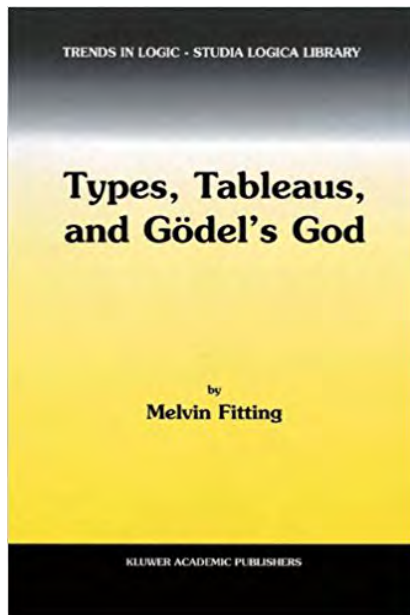
```
62 lemma helpH: "∀w. ((P ⌊φ⌋) w ∧ φ ⊆ ψ) ⟶ (P ⌊ψ⌋) w" by (metis (no_types, lifting) A4 C2 GA_def T3)
```

```
64 abbreviation "Pi φ ≡ (P ⌊φ⌋)" (*Pi: the set of all rigidly intensionalised positive properties*)
```

```
66 (*U2*) theorem U2: "∀w. ultrafilter Pi w" using helpE helpF helpG helpH by simp
```

```
67 (*U3*) theorem U3: "(Pi ⊆ P) ∧ (P ⊆ Pi)" nitpick oops (*countermodel: Pi 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 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 D3 D3 Necessary existence of an individual is the necessary instantiation of all its essences.

T2 A5 Necessary existence is a positive property.

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.

^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 A4 Positive properties are necessarily positive.

From T1 T4 **"Modal Collapse" is *not* implied by these axioms**

From T2 D3 $\varphi \supset \Box \varphi$ (has countermodel)

T3 A5 We can prove that these "positive property extensions" (which corresponds to \mathcal{P}' from before) form an ultrafilter.

From 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 (*⌊⌋ converts an extensional object φ into 'rigid' intensional one*)
7 abbreviation trivialConversion ("⌊_⌋") where "⌊φ⌋ ≡ (λw. φ)"
8 abbreviation Entails (infix "⇒" 60) where "X⇒Y ≡ □(∀z. (X z)→(Y z))"
9 (*φ's argument is a relativized term (of extensional type) derived from an intensional predicate.*)
10 abbreviation extPredArg (infix "⌊" 60) where "φ ⌊P ≡ λw. φ (λx. P x w) w"
11 (*A variant of the latter where φ takes intensional terms as argument.*)
12 abbreviation nextPredArg (infix "⌊" 60) where "φ ⌊P ≡ λw. φ (λx. (P x w)) w"
13 (*Another variant where φ has two arguments (the first one being relativized).*)
14 abbreviation extPredArg1 (infix "⌊₁" 60) where "φ ⌊₁P ≡ λz. λw. φ (λx. P x w) z w"
15
16 (**Part I**)
17 (*D1*) abbreviation God ("G") where "G ≡ (λx. ∀Y. P Y → (Y x))"
18 (*A1*) axiomatization where A1a:"[∀X. P (→X) → ¬(P X)]" and A1b:"[∀X. ¬(P X) → P (→X)]"
19 (*A2*) axiomatization where A2: "[∀X Y. (P X ∧ (X ⇒ Y)) → P Y]"
20 (*T1*) theorem T1: "[∀X. P X → ◇(∃z. (X z))]" using A1a A2 by blast
21 (*T2*) axiomatization where T2: "[P ⌊G]"
22 (*T3*) theorem T3deRe: "[ (λX. ◇∃x X) ⌊G]" using T1 T2 by simp
23 theorem T3deDicto: "[◇∃x ⌊G]" nitpickoops (*countermodel*)
24

```

Ontological Argument: Variant by Fitting (2002)

```
1 theory FittingProof imports IHOML
2 begin
3 (*Positiveness/perfection: uninterpreted constant symbol*)
4 consts Positiveness: "(p=beel) = i=beel" (*?*)
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25 (**Part II*)
26 (*Logic KB*) axiomatization where symm: "symmetric aRel"
27 (*A4*) axiomatization where A4: "[ $\forall x. P\ x \rightarrow \Box(P\ x)$ ]"
28 (*D2*) abbreviation Essence ("E") where " $E\ Y\ x \equiv (Y\ x) \wedge (\forall z. (Z\ x) \rightarrow (Y \Rightarrow Z))$ "
29 (*T4*) theorem T4: "[ $\forall x. G\ x \rightarrow (\{E\ \_1\ G\}\ x)$ ]" using Alb by metis
30 (*D3*) definition NE ("NE") where "NE x  $\equiv \lambda w. (\forall y. E\ Y\ x \rightarrow \Box(\exists z. (Y\ z)))\ w$ "
31 (*A5*) axiomatization where A5: "[ $P\ \_1\ NE$ ]"
32 lemma help1: "[ $\exists\ \_1\ G \rightarrow \Box\ \_1\ G$ ]" sorry (*longer interactive proof, omitted here*)
33 lemma help2: "[ $\exists\ \_1\ G \rightarrow (\{\lambda x. \Box\ \_1\ G\}\ \_1\ G)$ ]" by (metis A4 help1)
34 (*T5*) theorem T5deDicto: "[ $\Box\ \_1\ G \rightarrow \Box\ \_1\ G$ ]" using T3deRe help1 by blast
35 theorem T5deRe: "[ $\{\lambda x. \Box\ \_1\ G\}\ \_1\ G \rightarrow \{\lambda x. \Box\ \_1\ G\}\ \_1\ G$ ]" by (metis A4 help2)
36 (*T6*) theorem T6deDicto: "[ $\Box\ \_1\ G$ ]" using T3deRe help1 by blast
37 theorem T6deRe: "[ $\{\lambda x. \Box\ \_1\ G\}\ \_1\ G$ ]" using T3deRe help2 by blast
38
39 (**Consistency**)
40 lemma True nitpick[satisfy] oops (*Model found by Nitpick: the axioms are consistent*)
41
42 (**Modal Collapse**)
43 lemma ModalCollapse: "[ $\forall \phi. (\phi \rightarrow (\Box\ \phi))$ ]" nitpick oops (*countermodel*)
44
45 (**Some Corollaries**)
46 (* TODO (*C1*) theorem C1: "[ $\forall E\ P\ x. (\{E\ E\ x\} \wedge (P\ x)) \rightarrow (E \Rightarrow P)$ ]" by (metis ess_def) *)
47 (*C2*) theorem C2: "[ $\forall X. \neg P\ X \rightarrow \Box(\neg P\ X)$ ]" using A4 symm by blast
48 definition h4 ("N") where "N X  $\equiv \neg P\ X$ "
49 (*C3*) theorem C3: "[ $\forall X. N\ X \rightarrow \Box(N\ X)$ ]" by (simp add: C2 h4_def)
50 definition rigid  $\varphi \equiv \forall x. \varphi\ x \rightarrow \Box(\varphi\ x)$ 
51 (*C4*) theorem "[ $\forall \varphi. P\ \varphi \rightarrow \text{rigid } (\lambda x. (\varphi\ x))$ ]" by (simp add: rigid_def)
52 (*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 consts Positiveness: "(e⇒bool) ⇒ (e⇒bool)" (*?*)
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25 (**Part II*)
26 (*Logic KB*) axiomatization where symm: "symmetric aRel"
27 (*A4*) axiomatization where A4: "[ $\forall x. P\ x \rightarrow \Box(P\ x)$ ]"
28 (*D2*) abbreviation Essence ("E") where " $E\ Y\ x \equiv (Y\ x) \wedge (\forall z. (Z\ x) \rightarrow (Y \Rightarrow Z))$ "
29 (*T4*) theorem T4: "[ $\forall x. G\ x \rightarrow \{(\mathcal{E}\ x)\}$ ]" using Alb by metis
30 (*D3*) definition NE ("NE") where "NE x  $\equiv \lambda w. (\forall Y. E\ Y\ x \rightarrow \Box(\exists z. (Y\ z)))\ w$ "
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54 (**Positive Properties and Ultrafilters**)
55 abbreviation empty ("∅") where "∅  $\equiv \lambda x. False$ "
56 abbreviation intersect (infix "∩" 52) where " $\varphi \cap \psi \equiv (\lambda x. \varphi\ x \wedge \psi\ x)$ "
57 abbreviation nnegpred ("¬" [52] 53) where " $\neg \varphi \equiv \lambda x. \neg \varphi(x)$ "
58 abbreviation entail (infixr "⊆" 51) where " $\varphi \subseteq \psi \equiv \forall x. \varphi\ x \rightarrow \psi\ x$ "
59 abbreviation "ultrafilter  $\Phi\ cw \equiv$ "
60   " $\neg(\Phi\ \emptyset\ cw) \wedge (\forall \varphi::(e \Rightarrow bool). \forall \psi::(e \Rightarrow bool). (\Phi\ \varphi\ cw \wedge \Phi\ \psi\ cw) \rightarrow (\Phi\ (\varphi \cap \psi)\ cw))$ "
61   " $\wedge (\forall \varphi::(e \Rightarrow bool). \forall \psi::(e \Rightarrow bool). (\Phi\ \varphi\ cw \vee \Phi\ (\neg \varphi)\ cw) \wedge \neg(\Phi\ \varphi\ cw \wedge \Phi\ (\neg \varphi)\ cw))$ "
62   " $\wedge (\forall \varphi::(e \Rightarrow bool). \forall \psi::(e \Rightarrow bool). (\Phi\ \varphi\ cw \wedge \varphi \subseteq \psi) \rightarrow \Phi\ \psi\ cw)"$ "
63
64 lemma lemmaA: " $\forall w. \neg(P\ \emptyset\ w)$ " using T1 by blast
65 lemma lemmaB: " $\forall w. (P\ \varphi\ w \wedge P\ \psi\ w) \rightarrow (P\ (\varphi \cap \psi)\ w)$ " by (metis Alb T3deRe)
66 lemma lemmaC: " $\forall w. (P\ \varphi\ w \vee P\ (\neg \varphi)\ w) \wedge \neg(P\ \varphi\ w \wedge P\ (\neg \varphi)\ w)$ " using Ala Alb by blast
67 lemma lemmaD: " $\forall w. (P\ \varphi\ w \wedge \varphi \subseteq \psi) \rightarrow P\ \psi\ w$ " by (smt A2)
68
69 (*U1*) theorem " $\forall w. \text{ultrafilter } P\ w$ " by (smt lemmaA lemmaB lemmaC lemmaD)
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151 (*C4*) theorem "[ $\forall \varphi. P\ \varphi \rightarrow \text{rigid } (\lambda x. (\varphi\ x))$ ]" by (simp add: rigid_def)
152 (*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

```
AndersonProof.thy (~/.GITHUBS/chrisgitlab/talks/2019-Dubrovnik/)
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 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 GA]" (*here we postulate T2 instead of proving it*)
13 (*T3*) theorem T3: "[◇∃E GA]" 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 ("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 ≡ (λw. [∀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
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) 587MB 6:25 PM

DEMO: Anderson's Variant

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

29
30 (**Positive Properties and Ultrafilters**)
31 abbreviation emptySet ("∅") where "∅ ≡ λx w. False"
32 abbreviation entails (infixr "⊆") where "⊆ ≡ λφ ψ. ⊆ ≡ ∀x w. φ x w ⟶ ψ x w"
33 abbreviation andPred (infixr "∩") where "∩ ≡ λx w. φ x w ∧ ψ x w"
34 abbreviation negpred ("¬") [52] [53] where "¬ ≡ λx w. ¬ φ x w"
35 abbreviation "ultrafilter" φ cW ≡
36   ¬(φ ∅ cW)
37   ∧ (∀φ. ∀ψ. (φ ⊆ cW ∧ φ ⊆ ψ cW) ⟶ (φ ∩ ψ) cW)
38   ∧ (∀φ::e⇒i⇒bool. ∀ψ::e⇒i⇒bool. (φ ⊆ cW ∨ φ (¬ψ) cW) ∧ ¬(φ ⊆ cW ∧ φ (¬ψ) cW))
39   ∧ (∀φ::e⇒i⇒bool. ∀ψ::e⇒i⇒bool. (φ ⊆ cW ∧ ψ ⊆ cW) ⟶ φ ⊆ cW)
40
41 (*U1*) theorem U1: "∀w. ultrafilter P w" nitpick[user_axioms,format=2,show_all] oops (*counterterm.*)

Proof state  Auto update  Update  Search:  100%

ψ = (λx. _)((e1, i1) := False, (e1, i2) := False)
w = i1
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)
existsSat = (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

41,46 (2225/5923)  (isabelle,isabelle,UTF-8-Isabelle)  0.00 s  502MB  6:38 PM
```


DEMO: Anderson's Variant

isabelle2018-4-01 - AndersonProof.thy (modified)

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

```
44 abbreviation trivialConversion ("( $\downarrow$ )") where " $\downarrow \equiv (\lambda w. \varphi)$ "
45 (*Q  $\downarrow \varphi$ : the extension of a (possibly) non-rigid predicate  $\varphi$  is turned into a rigid intensional one,
46 then Q is applied to the latter;  $\downarrow \varphi$  can be read as "the rigidly intensionalised predicate  $\varphi$ "*)
47 abbreviation nextPredArg (infix " $\downarrow$ " 60) where " $Q \downarrow \varphi \equiv \lambda w. Q (\lambda x. \downarrow \varphi x w)$ "
48
49 lemma " $\forall Q \varphi. Q \varphi = Q \downarrow \varphi$ " nitpick oops (*countermodel: the two notions are not the same*)
50
51 lemma helpE: " $\forall w. \neg ((P \downarrow \emptyset) w)$ " using T1 by blast
52 lemma helpF: " $\forall \varphi \psi w. ((P \downarrow \varphi) w \wedge (P \downarrow \psi) w) \longrightarrow ((P \downarrow (\varphi \Pi \psi)) w)$ " by (smt GA_def T3 T5 symm)
53 lemma helpG: " $\forall w. ((P \downarrow \varphi) w \vee (P \downarrow (\neg \varphi)) w) \wedge \neg ((P \downarrow \varphi) w \wedge (P \downarrow (\neg \varphi)) w)$ " by (smt GA_def T3 T5 symm)
54 lemma helpH: " $\forall w. ((P \downarrow \varphi) w \wedge \varphi \subseteq \psi) \longrightarrow (P \downarrow \psi) w$ " by (metis GA_def T3 T5 symm)
55
56 abbreviation " $\mathcal{P}' \varphi \equiv (P \downarrow \varphi)$ " (* $\mathcal{P}'$ : the set of all rigidly intensionalised positive properties*)
57
58 (*U2*) theorem U2: " $\forall w. \text{ultrafilter } \mathcal{P}' w$ " using helpE helpF helpG helpH by simp
59 (*U3*) theorem U3: " $(\mathcal{P}' \subseteq \mathcal{P}) \wedge (\mathcal{P} \subseteq \mathcal{P}')$ " nitpick[user_axioms] oops (*countermodel:  $\mathcal{P}', \mathcal{P}$  not equal*)
60
```

Proof state Auto update Update Search: 100%

theorem

U2: $[(\lambda w. (\neg \mathcal{P}') (\{False\}) w \wedge$
 $(\lambda \varphi \psi. (\mathcal{P} (\lambda x. \downarrow \varphi x w) \wedge \mathcal{P} (\lambda x. \downarrow \psi x w)) w) \subseteq (\lambda \varphi \psi. \mathcal{P}' (\varphi \Pi \psi) w) \wedge$
 $[(\lambda \varphi. [((\mathcal{P} (\lambda x. \downarrow \varphi x w) \vee \mathcal{P} (\lambda x. \downarrow (\neg \varphi) x w)) w \wedge (\neg (\wedge)) (\mathcal{P}' \varphi w) (\mathcal{P}' (\neg \varphi) w))]] \wedge$
 $(\lambda \varphi \psi. \mathcal{P}' \varphi w \wedge \varphi \subseteq \psi) \subseteq (\lambda \psi. \mathcal{P}' \psi w)]]$

Output Query Sledgehammer Symbols

58,79 (3239/5927) (isabelle,isabelle,UTF-8-Isabelle) 1 nm 20 UG 1/500MB 6:41 PM

DEMO: Anderson's Variant

Isabelle2018+HOL - AndersonProof.thy

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

```

44 abbreviation trivialConversion ("( $\_$ )") where " $(\_)$   $\equiv$  ( $\lambda z::t. \_$ )"
45 *Q  $\downarrow \varphi$ : the extension of a (possibly) non-rigid predicate  $\varphi$  is turned into a rigid intensional one,
46 then Q is applied to the latter;  $\downarrow \varphi$  can be read as "the rigidly intensionalised predicate  $\varphi$ "
47 abbreviation nextPredArg (infix "↓" 60) where "Q  $\downarrow \varphi \equiv \lambda w. Q (\lambda x. (\varphi x w)) w$ "
48
49 lemma " $\forall Q \varphi. Q \varphi = Q \downarrow \varphi$ " nitpick oops (*countermodel: the two notions are not the same*)
50
51 lemma helpE: " $\forall w. \neg ((P \downarrow \emptyset) w)$ " using T1 by blast
52 lemma helpF: " $\forall \varphi \psi w. ((P \downarrow \varphi) w \wedge (P \downarrow \psi) w) \longrightarrow ((P \downarrow (\varphi \sqcap \psi)) w)$ " by (smt GA_def T3 T5 symm)
53 lemma helpG: " $\forall w. ((P \downarrow \varphi) w \vee (P \downarrow (\neg \varphi)) w) \wedge \neg ((P \downarrow \varphi) w \wedge (P \downarrow (\neg \varphi)) w)$ " by (smt GA_def T3 T5 symm)
54 lemma helpH: " $\forall w. ((P \downarrow \varphi) w \wedge \varphi \subseteq \psi) \longrightarrow (P \downarrow \psi) w$ " by (metis GA_def T3 T5 symm)
55
56 abbreviation " $P' \varphi \equiv (P \downarrow \varphi)$ " (*P': the set of all rigidly intensionalised positive properties*)
57
58 *U2*) theorem U2: " $\forall w. \text{ultrafilter } P' w$ " using helpE helpF helpG helpH by simp
59 *U3*) theorem U3: " $(P' \subseteq P) \wedge (P \subseteq P')$ " nitpick[user_axioms] oops (*countermodel: P', P not equal*)
60

```

Proof state Auto update Update Search: 100%

theorem

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

Output Query Sledgehammer Symbols

58,80 (3243/5930) (isabelle,isabelle,UTF-8-Isabelle) 1 n m 2 0 UG 114 / 1 MB 6:44 PM

DEMO: Anderson's Variant

ModelChecker/ModelChecker - AndersonProof.thy

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

```
61 (**Modal Logic S5: Consistency and Modal Collapse**)
62 axiomatization where refl: "∀x. x r x" and trans: "∀x y z. x r y ∧ y r z → x r z"
63 lemma True nitpick[satisfy] oops (*Model found by Nitpick: the axioms are consistent*)
64 lemma ModalCollapse: "[∀φ. (φ → (□ φ))]" nitpick[user_axioms,show_all,format=2] oops [*countermodel.*]
65
66 (**Barcan and Converse Barcan Formula for Individuals (type e)**)
67 lemma BarcanInd1: "[{∀x::e. (□ (φ(x))) → (□ (∀x::e. φ(x)))}" nitpick oops (*countermodel*)
68 lemma ConvBarcanInd1: "[{(□ (∀x::e. φ(x))) → (∀x::e. (□ (φ(x))))}" nitpick oops (*countermodel*)
69 (**Barcan and Converse Barcan Formula for Properties (type e⇒i⇒bool)**)
70 lemma BarcanPred1: "[{∀x::e⇒i⇒bool. (□ (φ(x))) → (□ (∀x::e⇒i⇒bool. φ(x)))}" by simp
71 lemma ConvBarcanPred1: "[{(□ (∀x::e⇒i⇒bool. φ(x))) → (∀x::e⇒i⇒bool. (□ (φ(x))))}" by simp
```

✓ Proof state ✓ Auto update Update Search: 100%

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) UG 71MB 6:54 PM

DEMO: Anderson's Variant

```

73 (* Some tests *)
74 consts aw::i peter::e mary::e supreme_being::e loves::"e=e-i-bool"
75 axiomatization where t1: "[ $\lambda x. \text{loves } i \text{ mary}$ ]]" and
76 t2: " $\neg(\text{peter} = \text{mary})$ " and t3: " $\neg(\text{peter} = \text{supreme\_being})$ " and t4: " $\neg(\text{mary} = \text{supreme\_being})$ " and
77 t5: " $\neg(\exists^i \text{peter } aw)$ " and t6: " $\neg(\exists^i \text{mary } aw)$ " and t7: " $\neg(\text{loves } \text{peter } \text{mary } aw)$ "
78 consts P_prime::"(e=e-i-bool)⇒i-bool"
79 axiomatization where t8: "P_prime = P"
80
81 lemma "(ultrafilter P aw) ∧ ¬(ultrafilter P aw)"
82   nitpick[
83     user_axioms, show_all, format=3] (*counterex.*)
84   nitpick[satisfy, user_axioms, show_all, format=3, timeout=100]oops
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
10
```

DEMO: Anderson's Variant

Isabelle2018/HOL - AndersonProof.thy (modified)

```
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 (*Ala*) 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 G^A" (*here we postulate T2 instead of proving it*)
13 (*T3*) theorem T3: "◇∃E G^A" 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 ("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
```

✓ Proof state ✓ Auto update Update Search: 100%

theorem T6: "◇∃E G^A"

Output Query Sledgehammer Symbols

22,55 (1320/5444) (isabelle,isabelle,UTF-8-Isabelle) 7:50 PM