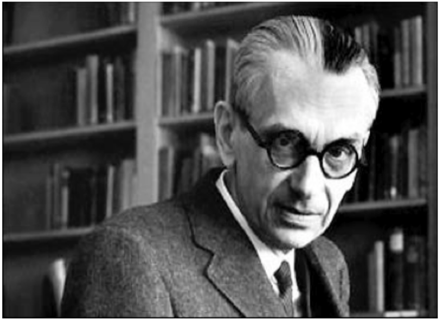
Computational Metaphysics: New Insights on Gödel's Ontological Argument and Modal Collapse

Christoph Benzmüller (jww D. Fuenmayor)

Freie Universität Berlin | University of Luxemburg



"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

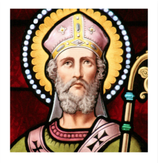
Part A
— Computational Metaphysics (recap) —
Ontological Argument by Gödel & Scott on the Computer

Related work:

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

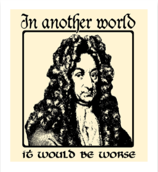
Ontological Proofs of God's Existence

A Long and Continuing Tradition in Philosophy



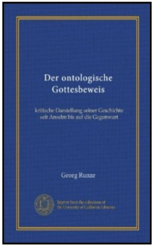
St. Anselm

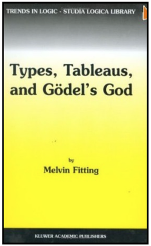
Descartes

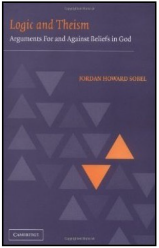


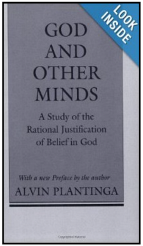
Leibniz

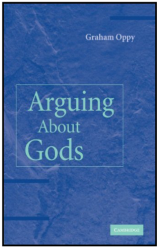
Gödel



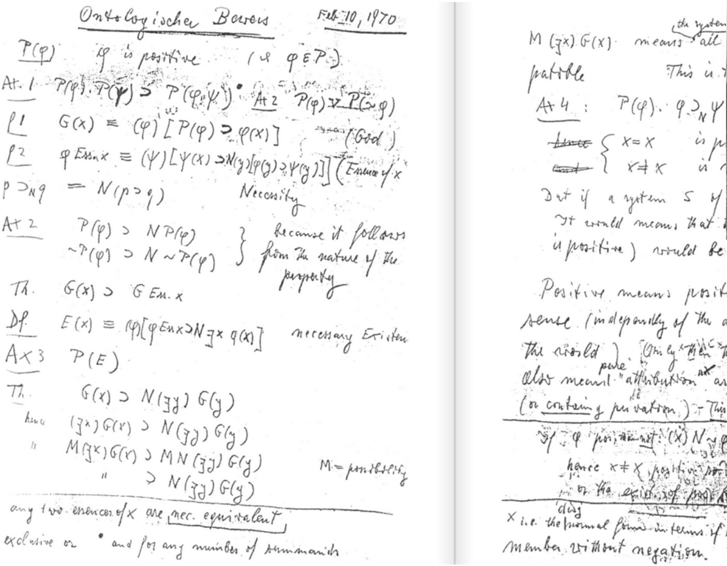








Computational Metaphysics: Kurt Gödel's Ontological Argument (1970)



Ontologischer Beweis Feb 10, 1970

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

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

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

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

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

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

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

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

Ax 3 $P(E)$

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

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

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

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

any two essences of x are nec. equivalent

exclusive or • and for any number of summands

$M(\exists x) G(x)$ means all the system ... patible This is ...

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

{ $x = x$ is p... ; $x \neq x$ is ...

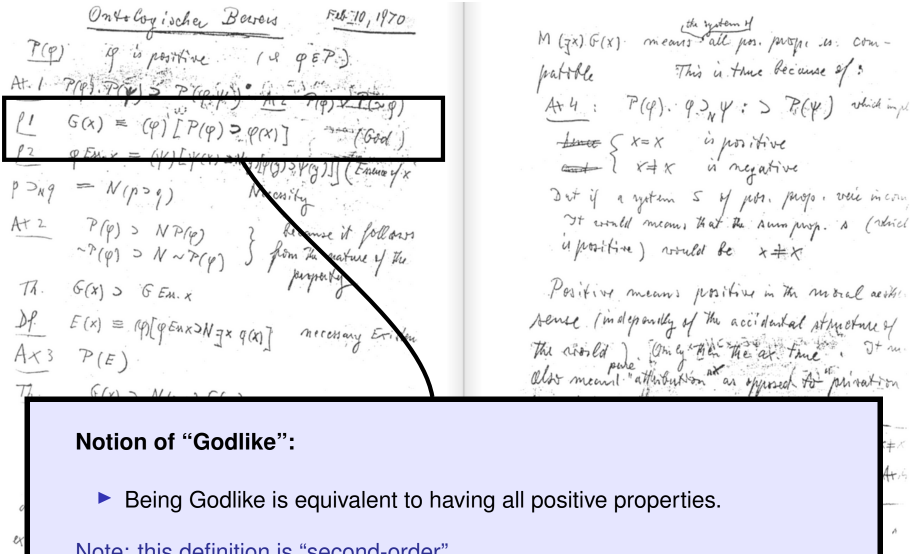
But if a system S of ... It would mean, that ... is positive) would be $x \neq x$

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

If φ positive: $(x) N \sim \varphi(x)$. Otherwise: $\varphi(x) \supset_N x \neq x$ hence $x \neq x$ positive so $x = x$ ... contrary At. ... or the exist. of pos. prop.

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

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



Ontologischer Beweis Feb 10, 1970

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

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

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

P2 $\varphi\, Em.x \equiv (\psi)[\psi(x) \supset N(y)[\varphi(y) \supset \psi(y)]]$ (Essence of x)

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

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

Th. $G(x) \supset G\, Em.x$

Df. $E(x) \equiv \varphi[\varphi\, Em\, x \supset N \exists x\, \varphi(x)]$ necessary Existence

Ax 3 $P(E)$

Th.

$M(\exists x)G(x)$ means the system of all pos. prop. is compatible. This is true because of:

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

$x = x$ is positive, $x \neq x$ is negative

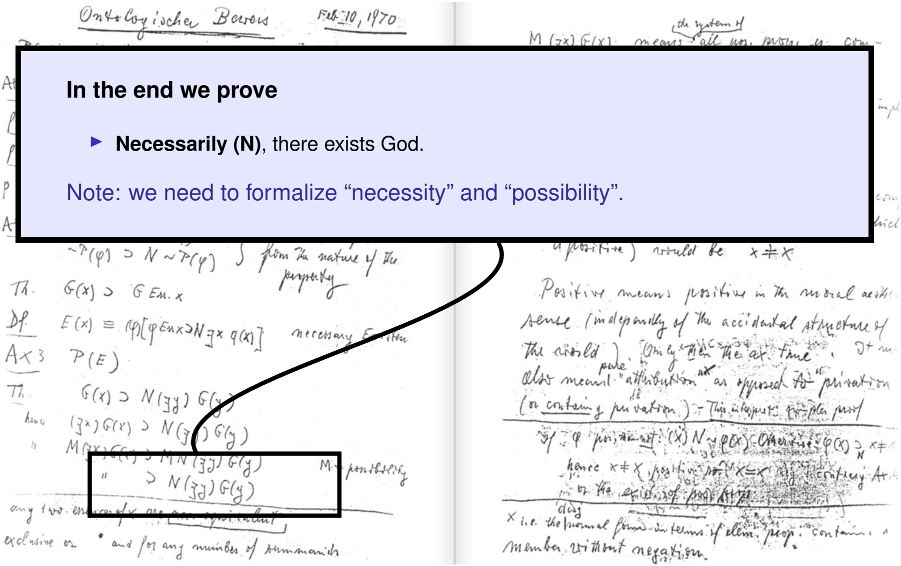
But if a system S of pos. prop. were incom it would mean, that the sum prop. s (which is positive) would be $x \neq x$

Positive means positive in the moral aesth sense (independently of the accidental structure of the world). Only then the ax. true. It m also means pure "attribution" as opposed to "privation"

Notion of "Godlike":

- Being Godlike is equivalent to having all positive properties.

Note: this definition is "second-order".



In the end we prove

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

Note: we need to formalize “necessity” and “possibility”.

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

Ontologischer Beweis Feb 10, 1970

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

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

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

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

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

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

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

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

Ax 3 $P(E)$

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

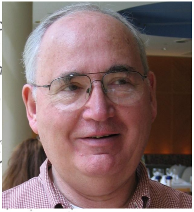
hence $(\exists x)G(x) \supset N(\exists y)\, G(y)$

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

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

any two essences of x are nec. equivalent

exclusive or * and for any number of summands



$M(\exists x)G(x)$ means all [the system] ...

patible This is ...

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

~~true~~ $x = x$ is p...

~~false~~ $x \neq x$ is ...

But if a system S of ...

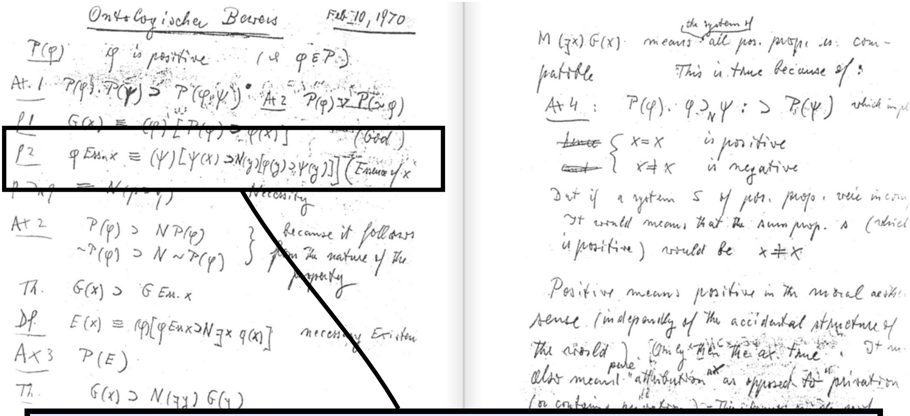
It would mean that the sum prop. s (which is positive) would be $x \neq x$

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

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

* 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)



Ontologischer Beweis   Feb 10, 1970

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

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

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

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

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

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

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

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

Ax 3 $P(E)$

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

$M(\exists x)\, G(x)$ means the system of all pos. prop. is compatible. This is true because of:

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

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

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

Positive means positive in the moral aesth… sense (independently of the accidental structure of the world). Only then the ax. true. It m… also mean pure "attribution" as opposed to "privation" (or containing privation) …

(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.

(Higher-Order) Modal Logic

$\Box P$

P is necessary

$\Diamond P$

P is possible

$\Box$ and $\Diamond$ are not truth-functional

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

(Higher-Order) Modal Logic

$\Box P$

P is necessary

$\Diamond P$

P is possible

$\Box$ and $\Diamond$ are not truth-functional

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

(Higher-Order) Modal Logic

$\Box P$

P is necessary

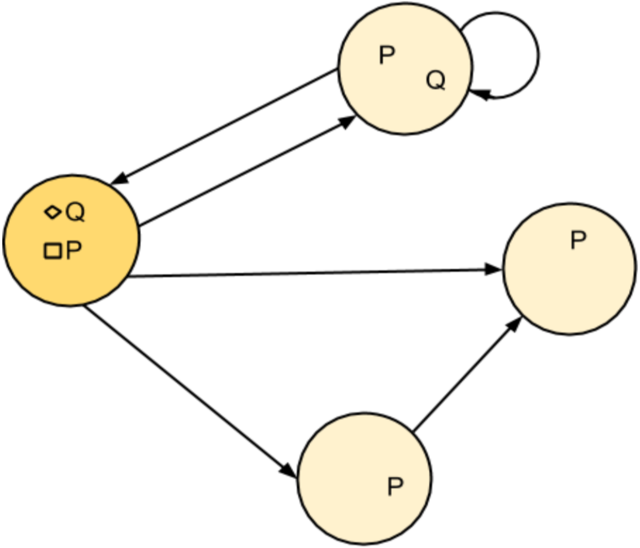
$\Diamond P$

P is possible

$\Box$ and $\Diamond$ are not truth-functional

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

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



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

Standard translation:

- $\vee_{(i\to o)\to(i\to o)\to(i\to o)} := \lambda\varphi_{i\to o}.\lambda\psi_{i\to o}.(\lambda w_i.\varphi w \vee \psi w)$
- $\Box_{(i\to o)\to(i\to o)} := \lambda\varphi_{i\to o}.(\lambda w_i.\forall v_i.Rwv \to \varphi v)$ ($R_{i\to i\to o}$: 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 $\Pi' := \lambda\Phi_{\alpha\to i\to o}.\lambda w_i.\Pi(\lambda x_\alpha.\Phi xw)$ and $\Box$ is resolved as above

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

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

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

Standard translation:

- $\vee_{(i\to o)\to(i\to o)\to(i\to o)} := \lambda\varphi_{i\to o}.\lambda\psi_{i\to o}.(\lambda w_i.\varphi w \vee \psi w)$
- $\Box_{(i\to o)\to(i\to o)} := \lambda\varphi_{i\to o}.(\lambda w_i.\forall v_i.Rwv \to \varphi v)$ ($R_{i\to i\to o}$: 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 $\Pi' := \lambda\Phi_{\alpha\to i\to o}.\lambda w_i.\Pi(\lambda x_\alpha.\Phi xw)$ and $\Box$ is resolved as above

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

- above: possibilist quantification
- actualist quantification: $\Pi' := \lambda\Phi.\lambda w.\Pi(\lambda x.$**existsAt x w** $\to \Phi xw)$
- also supported: local and global validity and consequence

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

Standard translation:

- $\vee_{(i\to o)\to(i\to o)\to(i\to o)} := \lambda\varphi_{i\to o}.\lambda\psi_{i\to o}.(\lambda w_i.\varphi w \vee \psi w)$
- $\Box_{(i\to o)\to(i\to o)} := \lambda\varphi_{i\to o}.(\lambda w_i.\forall v_i.Rwv \to \varphi v)$ ($R_{i\to i\to o}$: 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 $\Pi' := \lambda\Phi_{\alpha\to i\to o}.\lambda w_i.\Pi(\lambda x_\alpha.\Phi xw)$ and $\Box$ is resolved as above

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

- above: possibilist quantification
- actualist quantification: $\Pi' := \lambda\Phi.\lambda w.\Pi(\lambda x.$**existsAt x w** $\to \Phi xw)$
- also supported: local and global validity and consequence

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

Standard translation:

- $\vee_{(i\to o)\to(i\to o)\to(i\to o)} := \lambda\varphi_{i\to o}.\lambda\psi_{i\to o}.(\lambda w_i.\varphi w \vee \psi w)$
- $\Box_{(i\to o)\to(i\to o)} := \lambda\varphi_{i\to o}.(\lambda w_i.\forall v_i.Rwv \to \varphi v)$ ($R_{i\to i\to o}$: 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
  $\Pi' := \lambda\Phi_{\alpha\to i\to o}.\lambda w_i.\Pi(\lambda x_\alpha.\Phi xw)$ and $\Box$ is resolved as above

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

- above: possibilist quantification
- actualist quantification: $\Pi' := \lambda\Phi.\lambda w.\Pi(\lambda x.$**existsAt x w** $\to \Phi xw)$
- also supported: local and global validity and consequence

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

Standard translation:

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

Standard translation extended to quantifiers:

- in HOL: $\forall x.\phi x$ shorthand for $\Pi(\lambda x.\phi x)$
- $\Box\forall x.Px$ 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

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

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

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

Isabelle2018/HOL - IHOML.thy

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

```
theory IHOML imports Main
begin
  typedecl i (* possible worlds *) typedecl e (* individuals *)
(*Logical Operators as Truth-Sets *)
  abbreviation mnot ("¬_"[52]53) where "¬φ ≡ λw. ¬(φ w)"
  abbreviation negpred ("⇁_"[52]53) where "⇁Φ ≡ λx. ¬(Φ x)"
  abbreviation mnegpred ("⇁_"[52]53) where "⇁Φ ≡ λx.λw. ¬(Φ x w)"
  abbreviation mand (infixr"∧"51) where "φ∧ψ ≡ λw. (φ w)∧(ψ w)"
  abbreviation mor (infixr"∨"50) where "φ∨ψ ≡ λw. (φ w)∨(ψ w)"
  abbreviation mimp (infixr"→"49) where "φ→ψ ≡ λw. (φ w)⟶(ψ w)"
  abbreviation mequ (infixr"↔"48) where "φ↔ψ ≡ λw. (φ w)⟷(ψ w)"
(*Possibilist Quantification*)
  abbreviation mforall ("∀") where "∀Φ ≡ λw.∀x. (Φ x w)"
  abbreviation mforallB (binder"∀"[8]9) where "∀x. φ(x) ≡ ∀φ"
  abbreviation mexists ("∃") where "∃Φ ≡ λw.∃x. (Φ x w)"
  abbreviation mexistsB (binder"∃"[8]9) where "∃x. φ(x) ≡ ∃φ"
(*Actualist Quantification*)
  consts Exists::"(e⇒i⇒bool)" ("existsAt")
  abbreviation mforallAct ("∀^E") where "∀^EΦ ≡ λw.∀x. (existsAt x w)⟶(Φ x w)"
  abbreviation mexistsAct ("∃^E") where "∃^EΦ ≡ λw.∃x. (existsAt x w) ∧ (Φ x w)"
  abbreviation mforallActB (binder"∀^E"[8]9) where "∀^Ex. φ(x) ≡ ∀^Eφ"
  abbreviation mexistsActB (binder"∃^E"[8]9) where "∃^Ex. φ(x) ≡ ∃^Eφ"
(*Modal Operators*)
  consts aRel::"i⇒i⇒bool" (infixr "r" 70)  (*accessibility relation r*)
  abbreviation mbox ("□_"[52]53) where "□φ ≡ λw.∀v. (w r v)⟶(φ v)"
  abbreviation mdia ("◇_"[52]53) where "◇φ ≡ λw.∃v. (w r v)∧(φ v)"
(*Meta-logical Predicates*)
  abbreviation valid ("⌊_⌋"[8]) where "⌊ψ⌋ ≡  ∀w.(ψ w)"
(*Consistency and some useful definitions on (accessibility) relations*)
  lemma True nitpick[satisfy] oops  (*Model found by model finder Nitpick*)
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: $\phi\ ess.\ x \leftrightarrow \phi(x) \wedge \forall\psi(\psi(x) \rightarrow \Box\forall y(\phi(y) \rightarrow \psi(y)))$

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

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

Axiom 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

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

Thm. $\forall\phi[P(\phi) \to \Diamond\exists x\phi(x)]$

Def. $G(x) \leftrightarrow \forall\phi[P(\phi) \to \phi(x)]$

Axiom $P(G)$

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

Axiom $\forall\phi[P(\phi) \to \Box P(\phi)]$

Def.

$$\phi\ ess.\ x \leftrightarrow \phi(x)\ \wedge\ \forall\psi(\psi(x) \to \Box\forall y(\phi(y) \to \psi(y)))$$

Thm. $\forall x[G(x) \to G\ ess.\ x]$

Def.

$$NE(x) \leftrightarrow \forall\phi[\phi\ ess.\ x \to \Box\exists y\phi(y)]$$

Axiom $P(NE)$

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

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

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

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

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

Axiom $P(G)$

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

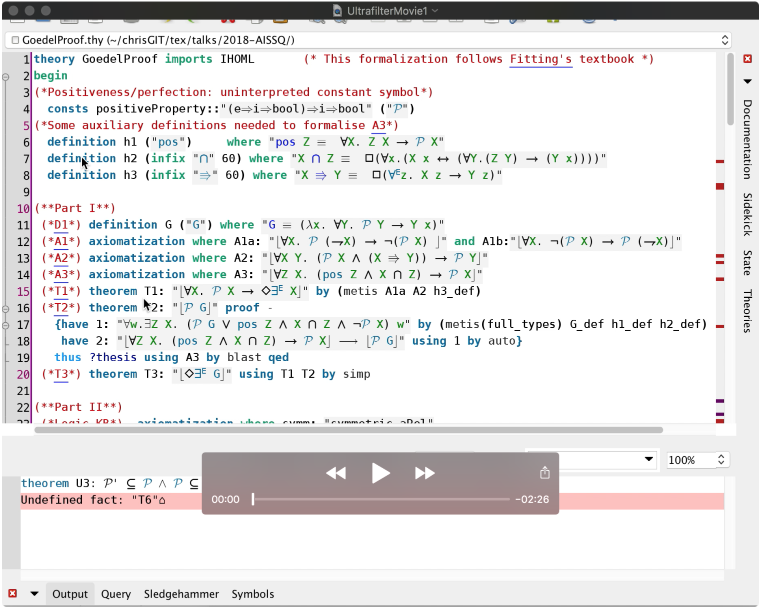
Def. $\phi \; ess. \; x \leftrightarrow \phi(x) \; \wedge \; \forall \psi (\psi(x) \rightarrow \Box \forall y (\phi(y) \rightarrow \psi(y)))$

Def. $NE(x) \leftrightarrow \forall \phi [\phi \; ess. \; x \rightarrow \Box \exists y \phi(y)]$

Axiom $P(NE)$

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

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

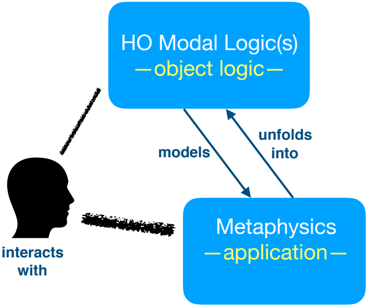


Universal (Meta-)Logical Reasoning Framework

[SCP, 2019]

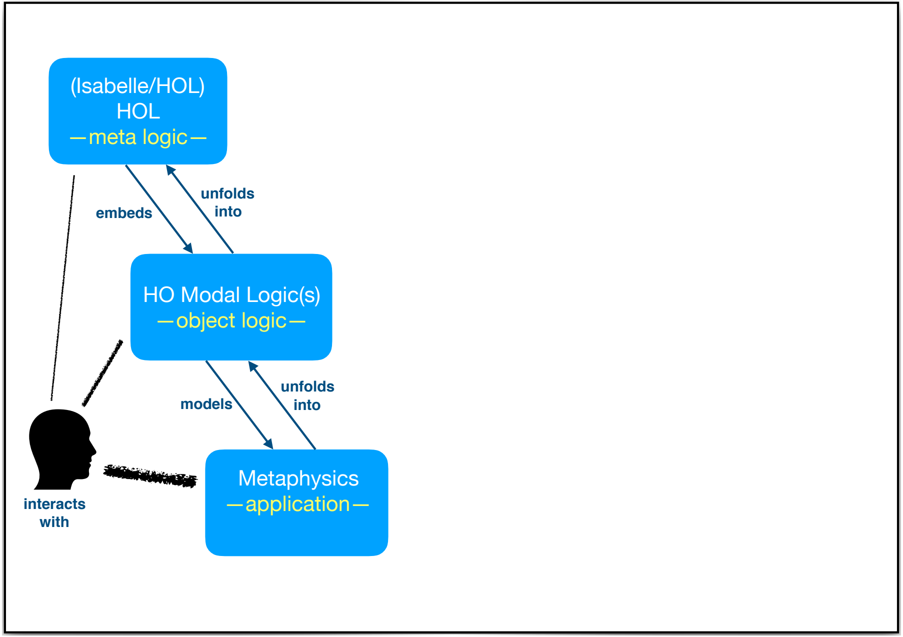
Universal (Meta-)Logical Reasoning Framework

[SCP, 2019]



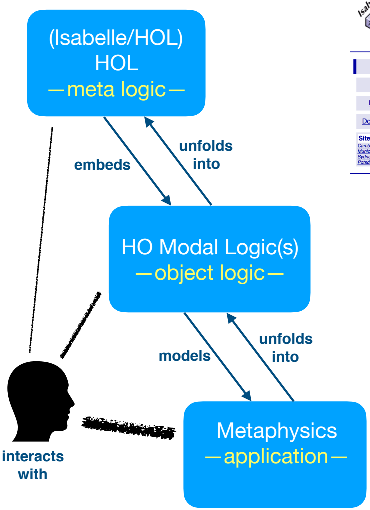
Universal (Meta-)Logical Reasoning Framework

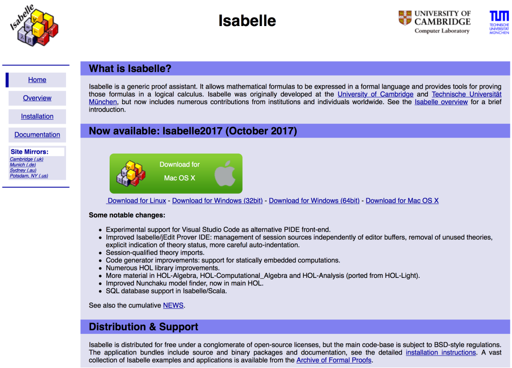
[SCP, 2019]



Universal (Meta-)Logical Reasoning Framework

[SCP, 2019]





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 Mac OS X

Download for Linux - Download for Windows (32bit) - Download for Windows (64bit) - Download for Mac OS X

Some notable changes:

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

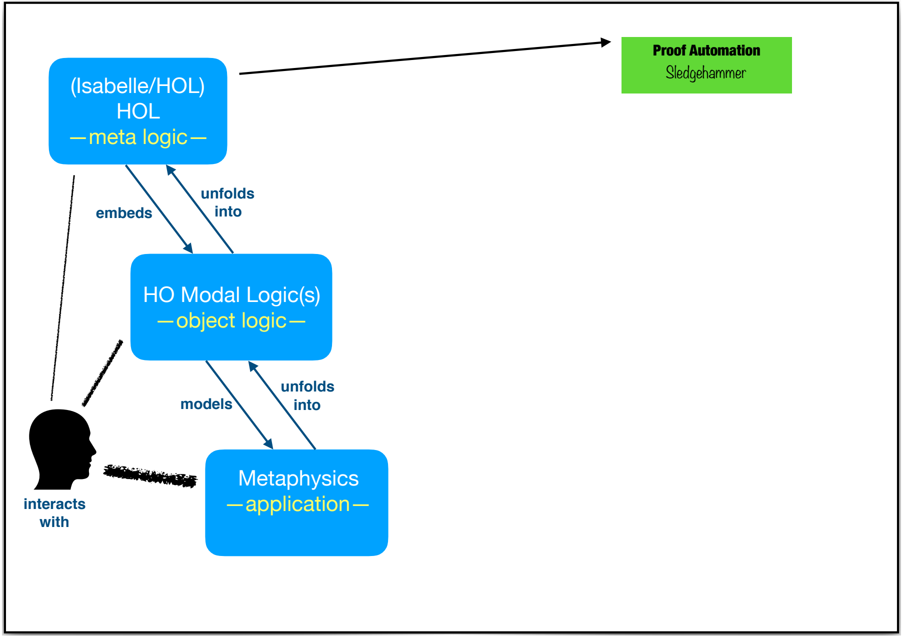
See also the cumulative NEWS.

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.

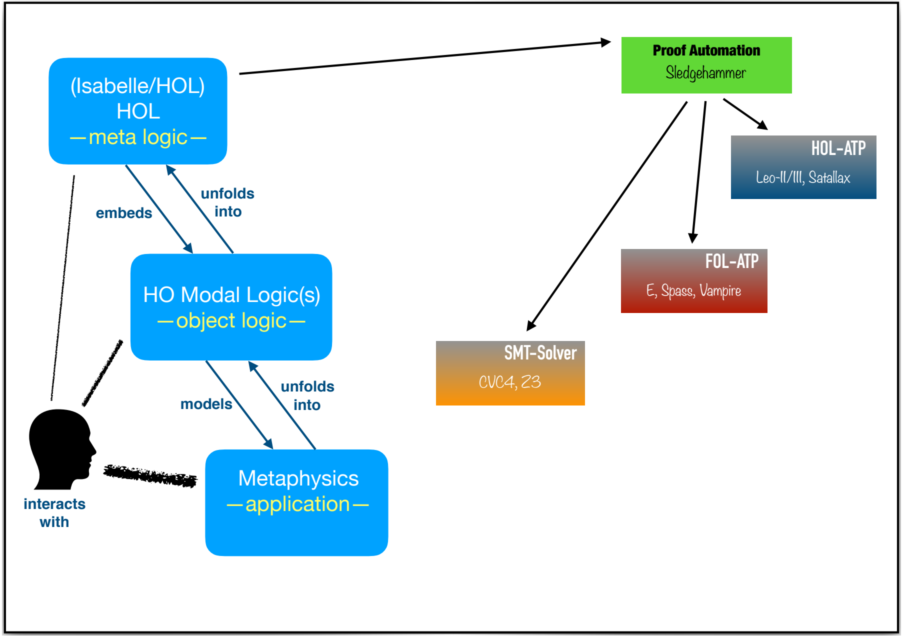
Universal (Meta-)Logical Reasoning Framework

[SCP, 2019]



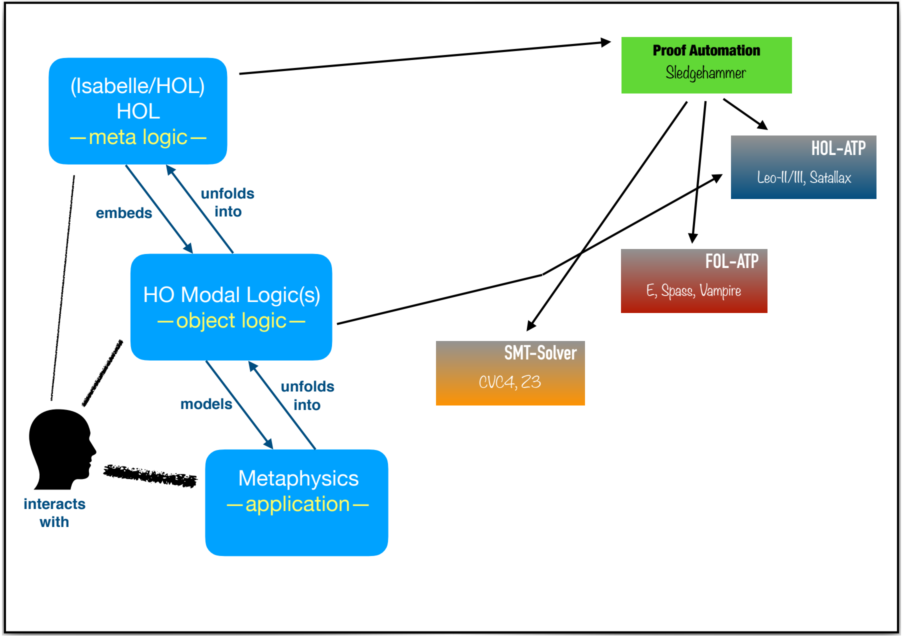
Universal (Meta-)Logical Reasoning Framework

[SCP, 2019]



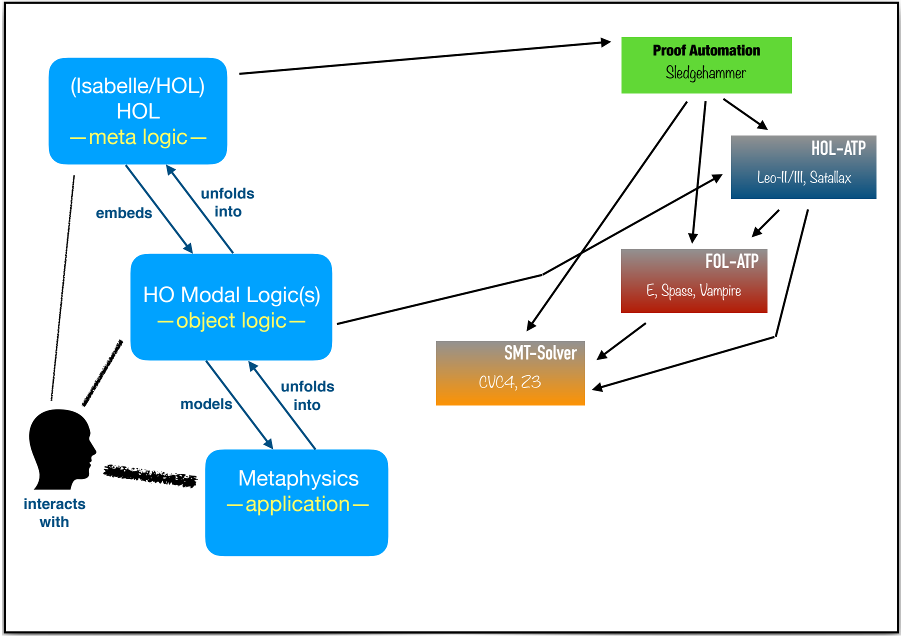
Universal (Meta-)Logical Reasoning Framework

[SCP, 2019]



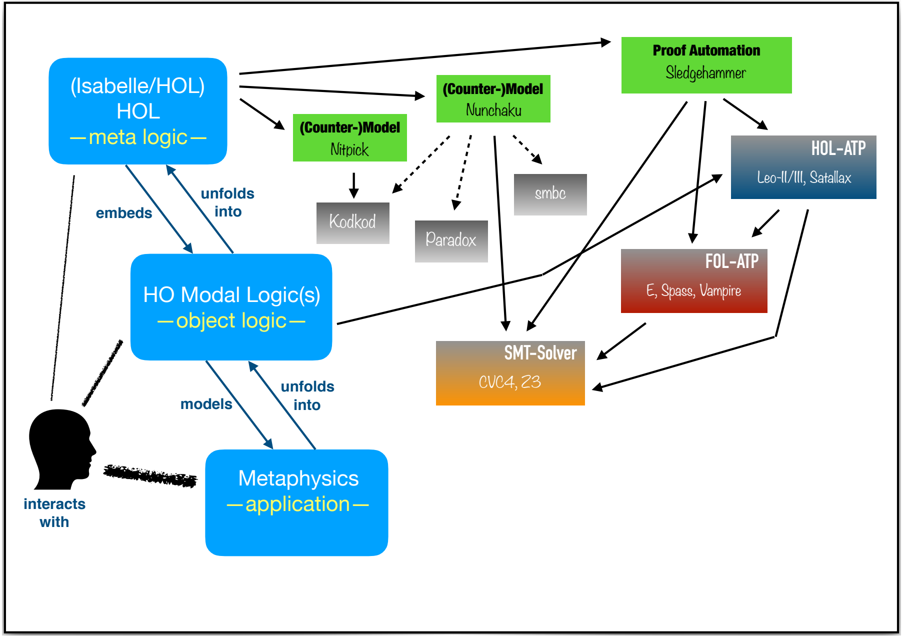
Universal (Meta-)Logical Reasoning Framework

[SCP, 2019]



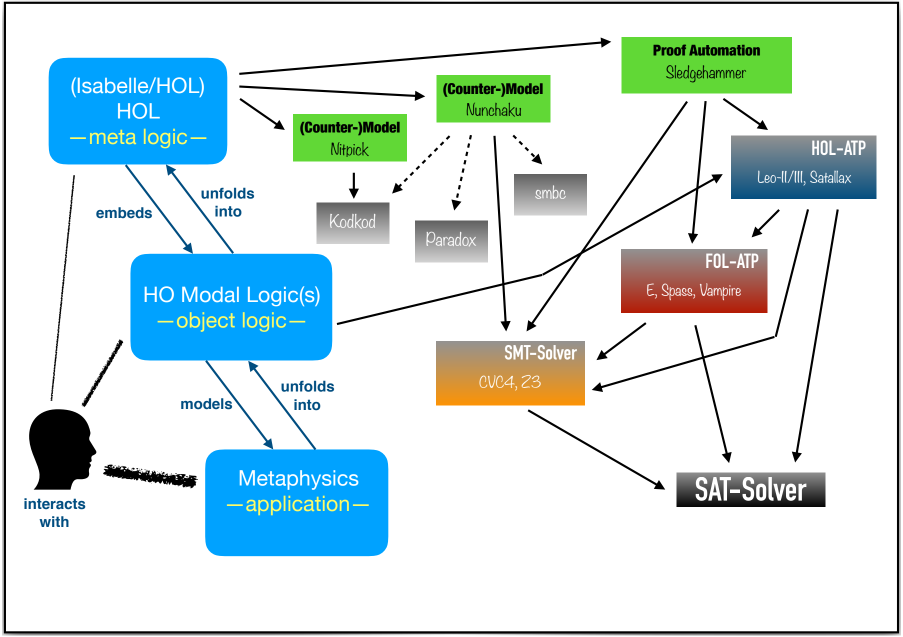
Universal (Meta-)Logical Reasoning Framework

[SCP, 2019]



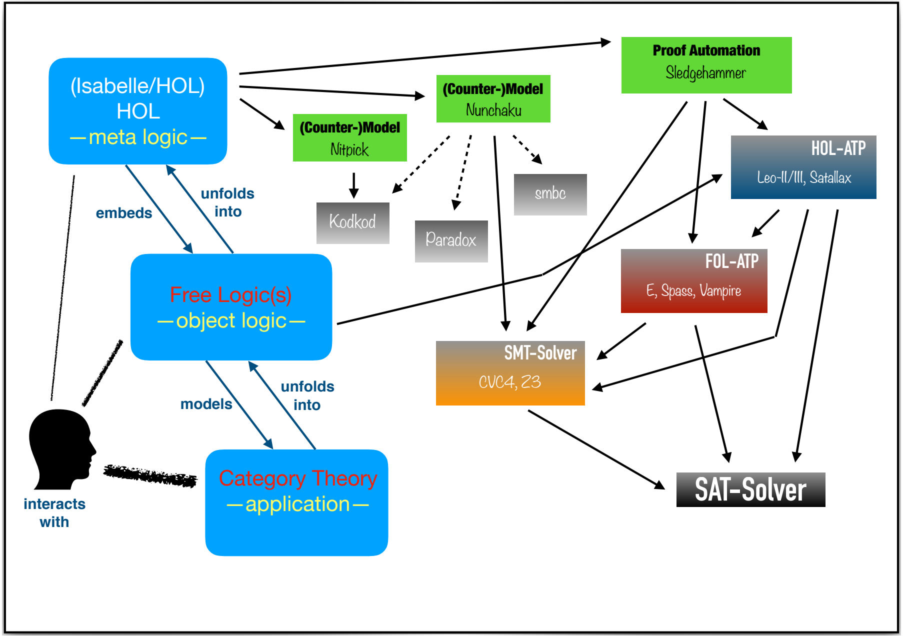
Universal (Meta-)Logical Reasoning Framework

[SCP, 2019]

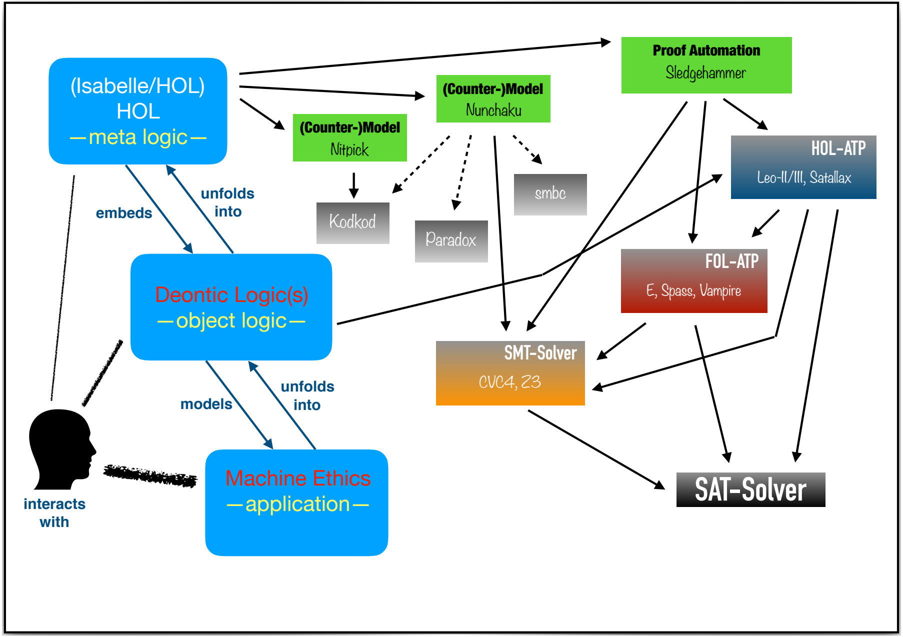


Universal (Meta-)Logical Reasoning Framework

[SCP, 2019]

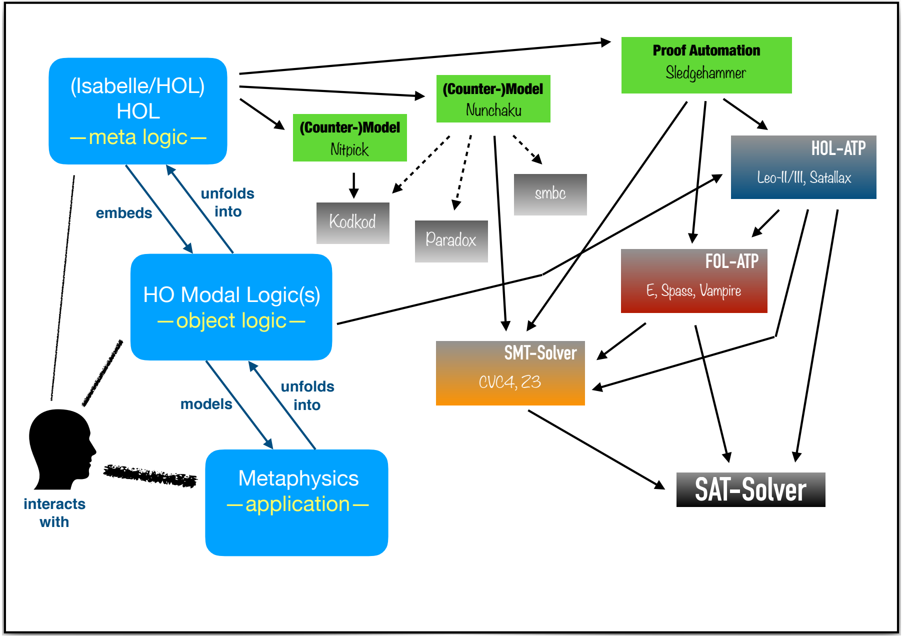


Universal (Meta-)Logical Reasoning Framework [SCP, 2019]



Universal (Meta-)Logical Reasoning Framework

[SCP, 2019]



Results of our Experiments (jww B. Woltzenlogel-Paleo)

(see also [Savijnanam 2017], [IJCAI 2016], [ECAI 2014])

Results of our Experiments

Variant of Dana Scott (1972)

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

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

Results of our Experiments

Variant of Dana Scott (1972)

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

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

Results of our Experiments

Variant of Dana Scott (1972)

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

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

Results of our Experiments

Variant of Dana Scott (1972)

- the premises are **consistent**
- all argument steps are **logically correct** in (higher-order, extensional) modal logic

  - correct in logic **S5**
  - weaker logic **KB** is already sufficient
  - critique about use of S5 not justfied

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

Results of our Experiments

Variant of Kurt Gödel (1970)

- the premises are inconsistent/contradictory (since they imply $\Diamond\Box\bot$)
- 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.

Results of our Experiments

Variant of Kurt Gödel (1970)

- the premises are inconsistent/contradictory (since they imply $\Diamond\Box\bot$)
- 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.

Results of our Experiments

Variant of Kurt Gödel (1970)

- the premises are inconsistent/contradictory (since they imply $\Diamond\Box\bot$)
- 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.

Results of our Experiments

Variant of Kurt Gödel (1970)

- the premises are inconsistent/contradictory (since they imply $\Diamond\Box\bot$)
- 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.

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: $\varphi \longrightarrow \Box \varphi$

  - 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: $\varphi \longrightarrow \Box \varphi$
  - 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: $\varphi \longrightarrow \Box\,\varphi$

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

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

— Can the modal collapse be avoided? —

Remainder of this Talk

We will have a closer look at

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

Questions:

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

To answer this questions we will apply some notions from

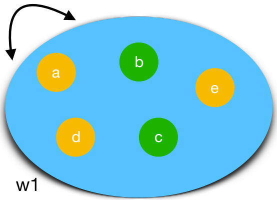
- mathematics: **ultrafilters**
- philosophy of language: **extension and intension of predicates**

Part B

Some Relevant Pillar Stones for this Talk

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

Example predicate: IsChessGrandmaster

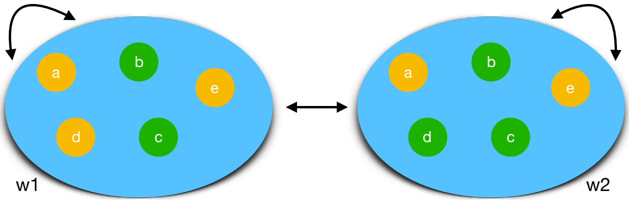


— Intensional Predicate **IsChessGrandmaster** (**ICG**)
— Extensions of **ICG** in possible worlds w1-w4:

ICG w1 = {**b**,**c**}

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

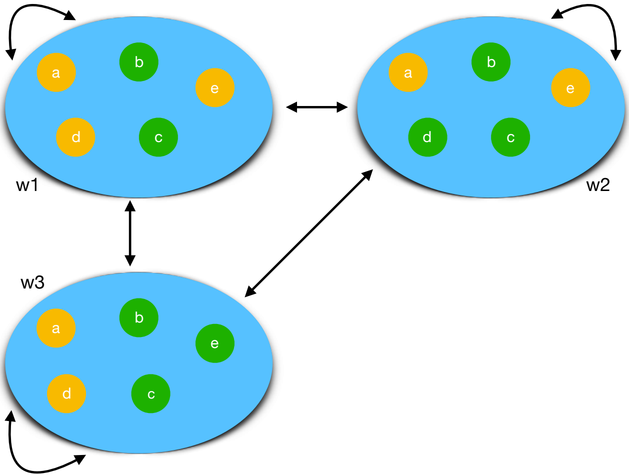
Example predicate: IsChessGrandmaster



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

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

Example predicate: IsChessGrandmaster

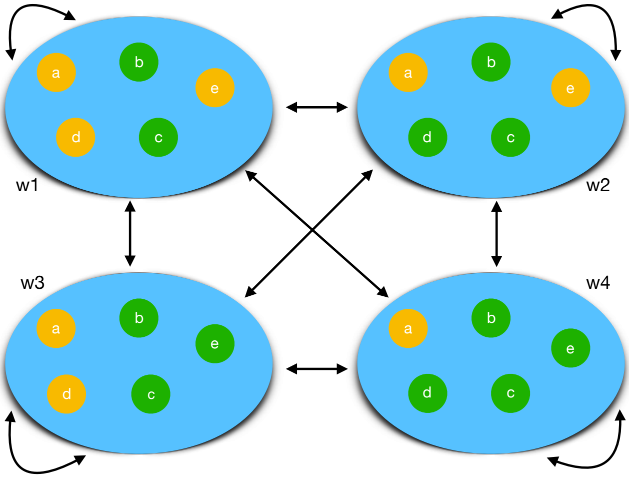


— Intensional Predicate **IsChessGrandmaster** (**ICG**)
— Extensions of **ICG** in possible worlds w1-w4:

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

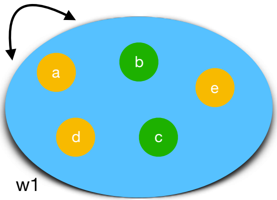


— Intensional Predicate **IsChessGrandmaster** (**ICG**)
— Extensions of **ICG** in possible worlds w1-w4:

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)

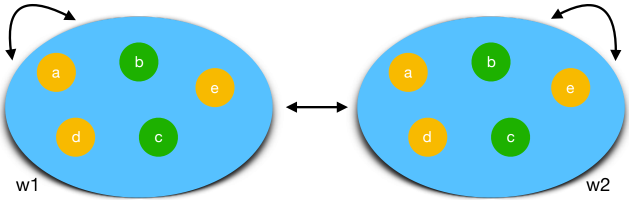
Example predicate: IsChessGrandmaster



— Intensional Predicate **IsChessGrandmaster** (**ICG**)
— Rigidified extension of **ICG** in world w1:
ICG w1 = {**b**,**c**}

"Rigidly Intensionalised Extension" of a Predicate (cf. $@_i\varphi$ from Patrick's talk)

Example predicate: IsChessGrandmaster



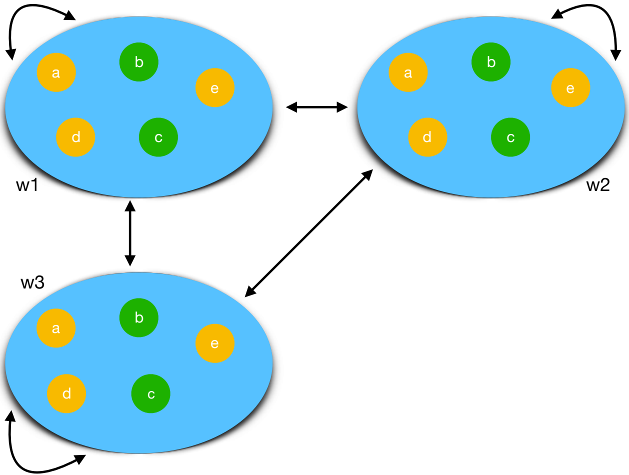
— Intensional Predicate **IsChessGrandmaster** (**ICG**)

— Rigidified extension of **ICG** in world w1:

ICG w1 = {**b**,**c**}   **ICG** w2 = {**b**,**c**}

"Rigidly Intensionalised Extension" of a Predicate (cf. $@_i\varphi$ from Patrick's talk)

Example predicate: IsChessGrandmaster



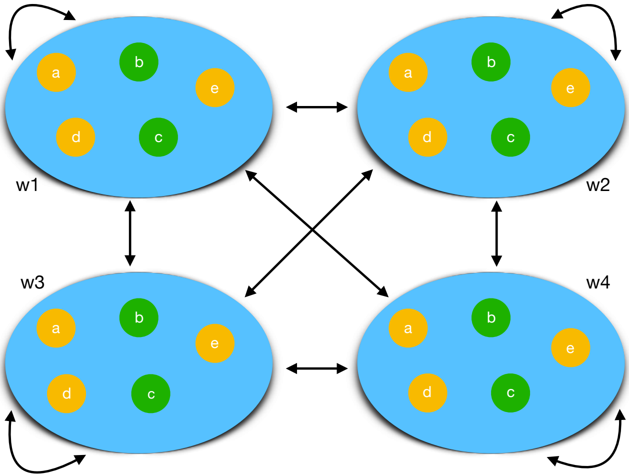
— Intensional Predicate **IsChessGrandmaster** (**ICG**)

— Rigidified extension of **ICG** in world w1:

ICG w1 = {**b**,**c**} **ICG** w2 = {**b**,**c**} **ICG** w3 = {**b**,**c**}

"Rigidly Intensionalised Extension" of a Predicate (cf. $@_i\varphi$ from Patrick's talk)

Example predicate: IsChessGrandmaster



— Intensional Predicate **IsChessGrandmaster** (**ICG**)
— Rigidified extension of **ICG** in world w1:

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

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

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

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

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:

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

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

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

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:

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

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

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

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:

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

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

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

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:

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

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

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

Ultrafilter

Definition of Ultrafilter:

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

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

Example:

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

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

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

Ultrafilter

Definition of Ultrafilter:

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

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

Example:

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

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

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

Ultrafilter

Definition of Ultrafilter:

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

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

Example:

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

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

$$\begin{array}{lllll}
U^1 = \{ & & \{1,4\}, & & \} \\
U^2 = \{ & & \{1,4\}, & \{1,2,4\}, \{1,3,4\}, \{1,2,3,4\}\} & \\
U^3 = \{ & \{1\}, & \{1,4\}, & \{1,2,4\}, \{1,3,4\}, \{1,2,3,4\}\} &
\end{array}$$

Ultrafilter

Definition of Ultrafilter:

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

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

Example:

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

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

Ultrafilter

Definition of Ultrafilter:

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

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

Example:

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

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

Ultrafilter

Definition of Ultrafilter:

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

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

Example:

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

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

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

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

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

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

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:

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

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

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

Ultrafilter

Definition of Ultrafilter:

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

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

Example:

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

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

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

Ultrafilter

Definition of Ultrafilter:

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

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

Example:

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

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

$$\begin{array}{llllll}
U^1 = \{ & & \{1,4\}, & & & \} \\
U^2 = \{ & & \{1,4\}, & & \{1,2,4\}, \{1,3,4\}, \{1,2,3,4\}\} & \\
U^3 = \{ & \{1\}, & \{1,4\}, & & \{1,2,4\}, \{1,3,4\}, \{1,2,3,4\}\} & \\
U^4 = \{ & \{1\}, \{1,2\}, \{1,3\}, & \{1,4\}, & \{1,2,3\}, & \{1,2,4\}, \{1,3,4\}, \{1,2,3,4\}\} &
\end{array}$$

Ultrafilter

Definition of Ultrafilter:

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

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

Example:

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

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

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

Ultrafilter

Definition of Ultrafilter:

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

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

Example:

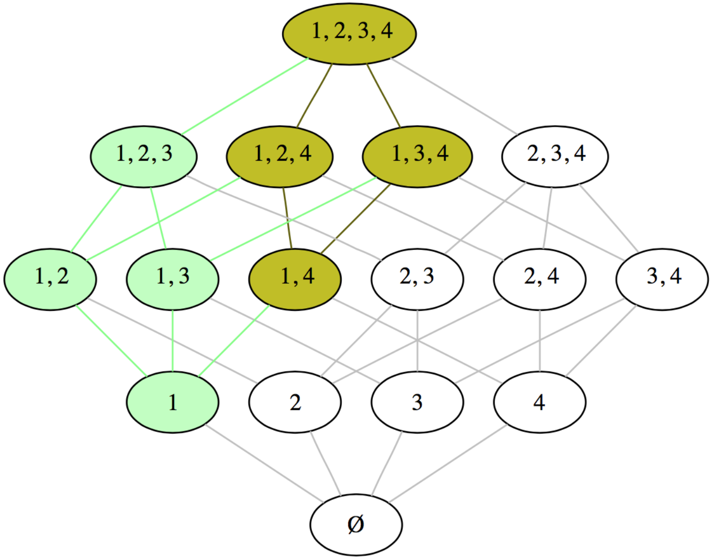
$$X = \{1, 2, 3, 4\}$$

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

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

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

Ultrafilter



From: Wikipedia (Jochen Burghardt) CC BY-SA
https://en.wikipedia.org/wiki/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

A1

A2

A3

Fro

T1

Fro

T2

Fro

T3

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 properties are nec. entailed by E.[a]

A4 Positive properties are necessarily positive.

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

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

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

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.

---

[a] The underlined part in definition D2 has been added by Scott. Gödel originally omitted this part.

Part I - Proving that God's existence is possible

D1
A1
A2
A3
Fro
T1
Fro
T2
Fro
T3

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 p
A4 F
From
T4 E
D3 N ... all i
A5 N
From
T5 E
And ... **ms,**
e.g. 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.

[a] The underlined part in definition D2 has been added by Scott. Gödel originally omitted this part.

Ontological Argument: Variant by Gödel/Scott

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

(**Part I**)
 (*D1*) definition G ("G") where "G ≡ (λx. ∀Y. 𝒫 Y ⟶ Y x)"
 (*A1*) axiomatization where A1a: "⌊∀X. 𝒫 (⇁X) ⟶ ¬(𝒫 X) ⌋" and A1b:"⌊∀X. ¬(𝒫 X) ⟶ 𝒫 (⇁X)⌋"
 (*A2*) axiomatization where A2: "⌊∀X Y. (𝒫 X ∧ (X ⇛ Y)) ⟶ 𝒫 Y⌋"
 (*A3*) axiomatization where A3: "⌊∀Z X. (pos Z ∧ X ⊓ Z) ⟶ 𝒫 X⌋"
 (*T1*) theorem T1: "⌊∀X. 𝒫 X ⟶ ◇∃ᴱ X⌋" by (metis A1a A2 h3_def)
 (*T2*) theorem T2: "⌊𝒫 G⌋" proof -
   {have 1: "∀w.∃Z X. (𝒫 G ∨ pos Z ∧ X ⊓ Z ∧ ¬𝒫 X) w" by (metis(full_types) G_def h1_def h2_def)
    have 2: "⌊∀Z X. (pos Z ∧ X ⊓ Z) ⟶ 𝒫 X⌋ ⟶ ⌊𝒫 G⌋" using 1 by auto}
   thus ?thesis using A3 by blast qed
 (*T3*) theorem T3: "⌊◇∃ᴱ G⌋" sledgehammer using T1 T2 by simp
```

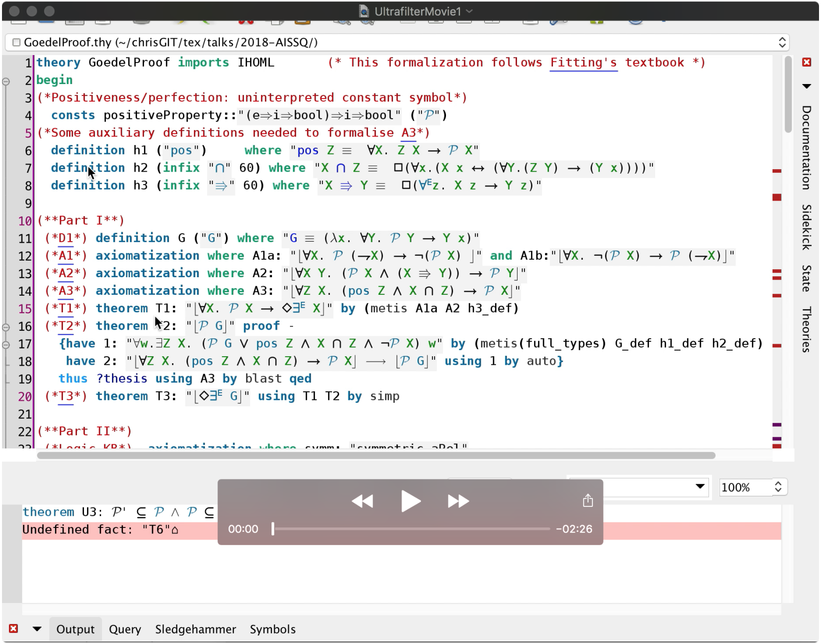
Ontological Argument: Variant by Gödel/Scott

```
(**Part II**)
 (*Logic KB*)  axiomatization where symm: "symmetric aRel"
 (*A4*) axiomatization where A4: "⌊∀X. 𝒫 X → □(𝒫 X)⌋"
 (*D2*) definition ess ("ℰ") where  "ℰ Y x ≡ (Y x) ∧ (∀Z. Z x → Y ⇛ Z)"
 (*T4*) theorem T4: "⌊∀x. G x → (ℰ G x)⌋" by (metis A1b A4 G_def h3_def ess_def)
 (*D3*) definition NE ("NE") where "NE x  ≡ (λw. (∀Y.  ℰ Y x → □∃ᴱ Y) w)"
 (*A5*) axiomatization where A5: "⌊𝒫 NE⌋"
 (*T5*) theorem T5: "⌊(◇∃ᴱ G) → □∃ᴱ G⌋" by (metis A5 G_def NE_def T4 symm)
 (*T6*) theorem T6: "⌊□∃ᴱ G⌋" using T3 T5 by blast

(**Consistency**)
  lemma True nitpick[satisfy] oops  (*Model found by Nitpick: the axioms are consistent*)

(**Modal Collapse**)
  lemma ModalCollapse: "⌊∀Φ.(Φ → (□ Φ))⌋" proof -
    {fix w fix Q
      have "∀x. G x w ⟶ (∀Z. Z x → □(∀ᴱz. G z → Z z)) w" by (metis A1b A4 G_def)
      hence 1: "(∃x. G x w) ⟶ ((Q → □(∀ᴱz. G z → Q)) w)" by force
      have "∃x. G x w" using T3 T6 symm by blast
      hence "(Q → □Q) w" using 1 T6 by blast
    } thus ?thesis by auto qed

(**Some Corollaries**)
 (*C1*) theorem C1: "⌊∀E P x. ((ℰ E x) ∧ (P x)) → (E ⇛ P)⌋" by (metis ess_def)
 (*C2*) theorem C2: "⌊∀X. ¬𝒫 X → □(¬𝒫 X)⌋" using A4 symm by blast
  definition h4 ("𝒩") where "𝒩 X ≡  ¬𝒫 X"
 (*C3*) theorem C3: "⌊∀X. 𝒩 X → □(𝒩 X)⌋" by (simp add: C2 h4_def)
```

Ontological Argument: Variant by Gödel/Scott

```
(**Positive Properties and Ultrafilters**)
  abbreviation emptySet ("∅") where "∅ ≡ λx w. False"
  abbreviation entails (infixr"⊆"51) where "φ⊆ψ  ≡ ∀x w. φ x w ⟶ ψ x w"
  abbreviation andPred (infixr"⊓"51) where "φ⊓ψ  ≡ λx w. φ x w ∧ ψ x w"
  abbreviation negpred ("¯_"[52]53) where "¯ψ   ≡ λx w. ¬ψ x w"
  abbreviation "ultrafilter Φ cw ≡
        ¬(Φ ∅ cw)
    ∧  (∀φ. ∀ψ. (Φ φ cw ∧ Φ ψ cw) ⟶ (Φ (φ ⊓ ψ) cw))
    ∧  (∀φ::e⇒i⇒bool. ∀ψ::e⇒i⇒bool. (Φ φ cw ∨ Φ (¯φ) cw) ∧ ¬(Φ φ cw ∧ Φ (¯φ) cw))
    ∧  (∀φ::e⇒i⇒bool. ∀ψ::e⇒i⇒bool.  (Φ φ cw ∧ φ ⊆ ψ) ⟶ Φ ψ cw)"
  lemma helpA: "∀w. ¬(𝒫 ∅ w)" using T1 by auto
  lemma helpB: "∀φ ψ w. (𝒫 φ w ∧ 𝒫 ψ w) ⟶ (𝒫 (φ ⊓ ψ) w)" by (smt A1b G_def T3 T6 symm)
  lemma helpC: "∀φ ψ w. (𝒫 φ w ∨ 𝒫 (¯φ) w) ∧ ¬(𝒫 φ w ∧ 𝒫 (¯φ) w)" using A1a A1b by blast
  lemma helpD: "∀φ ψ w. (𝒫 φ w  ∧ (φ ⊆ ψ))  ⟶ 𝒫 ψ w" by (metis A1b A4 G_def T1 T6)

(*U1*) theorem U1: "∀w. ultrafilter 𝒫 w" using helpA helpB helpC helpD by simp

(*⦇φ⦈ converts an extensional object φ into `rigid' intensional one*)
  abbreviation trivialConversion ("⦇_⦈") where "⦇φ⦈ ≡ (λw. φ)"
(*Q ↓φ: the extension of a (possibly) non-rigid predicate φ is turned into a rigid intensional one,
  then Q is applied to the latter; ↓φ can be read as "the rigidly intensionalised predicate φ"*)
  abbreviation mextPredArg (infix "↓" 60) where "Q ↓φ ≡ λw. Q (λx. ⦇φ x w⦈) w"
  lemma "∀Q φ. Q φ = Q ↓φ" nitpick oops (*countermodel: the two notions are not the same*)

  lemma helpE: "∀w.¬((𝒫 ↓∅) w)" using T1 by blast
  lemma helpF: "∀φ ψ w.((𝒫 ↓φ) w ∧ (𝒫 ↓ψ) w) ⟶ ((𝒫 ↓(φ⊓ψ)) w)" by (smt A1b C2 G_def T3 symm)
  lemma helpG: "∀w.((𝒫 ↓φ) w ∨ (𝒫 ↓(¯φ)) w) ∧ ¬((𝒫 ↓φ) w ∧ (𝒫 ↓(¯φ)) w)" using A1a A1b by blast
  lemma helpH: "∀w.((𝒫 ↓φ) w ∧ φ⊆ψ) ⟶ (𝒫 ↓ψ) w" by (metis A1b A5 G_def NE_def T3 T4 symm)

  abbreviation "𝒫' φ ≡ (𝒫 ↓φ)" (*𝒫': the set of all rigidly intensionalised positive properties*)

(*U2*) theorem U2: "∀w. ultrafilter 𝒫' w"  using helpE helpF helpG helpH by simp
(*U3*) theorem U3: "(𝒫' ⊆ 𝒫) ∧ (𝒫 ⊆ 𝒫')" by (smt A1b G_def T1 T6 symm) (*𝒫' and 𝒫 are equal*)
```

Ontological Argument: Variant by Gödel/Scott



Ontological Argument: Variant by Anderson (1990)

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'

A1a

~~A1b~~

A2

A3

Fro

T1

Fro

T2

Fro

T3

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 properties are nec. entailed by E.

A4 Positive properties are necessarily positive.

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

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

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

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.

Ontological Argument: Variant by Anderson

Part I - Proving that God's existence is possible

D1'

A1a

~~A1b~~

A2

A3

Fro

T1

Fro

T2

Fro

T3

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

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.**

A4 Positive properties are necessarily positive.

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

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

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

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.

Ontological Argument: Variant by Anderson

Part I - Proving that God's existence is possible

D1'

A1a

~~A1b~~

A2

A3

Fro

T1

Fro

T2

Fro

T3

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

D2' A ... l of x ... all p

A4 F

From

T4 E

D3 N ... all i

A5 N

From

T5 E

And ... ms, e.g. S

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

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

- no determinism

"positive properties ($\mathcal{P}$)" are applied here to intensional properties. We have:

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

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

- $\mathcal{P}'$ is an ultrafilter
- $\mathcal{P} \neq \mathcal{P}'$

T6 Being Godlike is necessarily instantiated.

Ontological Argument: Variant by Anderson

```
theory AndersonProof imports IHOML
begin
(*Positiveness/perfection: uninterpreted constant symbol*)
  consts positiveProperty::"(e⇒i⇒bool)⇒i⇒bool" ("𝒫")
(*Some auxiliary definitions*)
  definition h3 (infix "⇛" 60) where "X ⇛ Y ≡  □(∀ᴱz. X z ⟶ Y z)"

(**Part I**)
 (*D1'*)  definition GA ("Gᴬ") where "Gᴬ ≡ λx. ∀Y. (𝒫 Y) ↔ □(Y x)"
 (*A1a*) axiomatization where A1a:"⌊∀X. 𝒫 (⇁X) ⟶ ¬(𝒫 X) ⌋"
 (*A2*)  axiomatization where A2: "⌊∀X Y. (𝒫 X ∧ (X ⇛ Y)) ⟶ 𝒫 Y⌋"
 (*T1*)  theorem T1: "⌊∀X. 𝒫 X ⟶ ◇∃ᴱ X⌋" using A1a A2 h3_def by metis
 (*T2*)  axiomatization where T2: "⌊𝒫 Gᴬ⌋"  (*here we postulate T2 instead of proving it*)
 (*T3*)  theorem T3: "⌊◇∃ᴱ Gᴬ⌋" using T1 T2 h3_def by blast
```

Ontological Argument: Variant by Anderson

```
theory AndersonProof imports IHOML
```

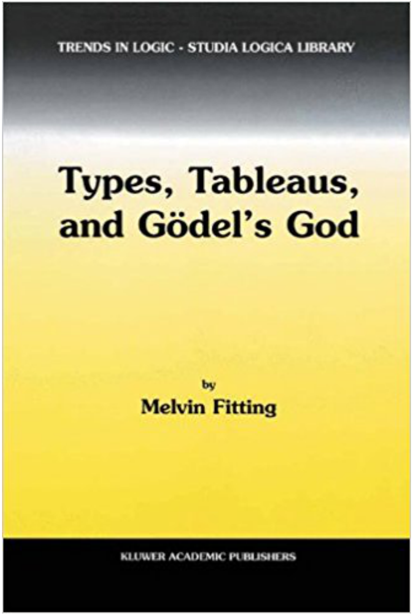
```
(**Part II**)
 (*Logic KB*) axiomatization where symm: "symmetric aRel"
 (*A4*)  axiomatization where A4: "⌊∀X. 𝒫 X → □(𝒫 X)⌋"
 (*D2'*) abbreviation essA ("ℰᴬ") where "ℰᴬ Y x ≡ (∀Z. □(Z x) ↔ Y ⇛ Z)"
 (*T4*)  theorem T4: "⌊∀x. Gᴬ x → (ℰᴬ Gᴬ x)⌋" by (metis A2 GA_def T2 symm h3_def)
 (*D3*)  abbreviation NEA ("NEᴬ") where "NEᴬ x  ≡ (λw. (∀Y. ℰᴬ Y x → □∃ᴱ Y) w)"
 (*A5*)  axiomatization where A5: "⌊𝒫 NEᴬ⌋"
 (*T5*)  theorem T5: "⌊◇∃ᴱ Gᴬ⌋ ⟶ ⌊□∃ᴱ Gᴬ⌋" by (metis A2 GA_def T2 symm h3_def)
 (*T6*)  theorem T6: "⌊□∃ᴱ Gᴬ⌋" using T3 T5 by blast

(**Modal collapse is countersatisfiable**)
lemma "⌊∀Φ.(Φ → (□ Φ))⌋" nitpick oops (*Countermodel found by Nitpick*)

(**Consistency**)
  lemma True nitpick[satisfy] oops  (*model found by Nitpick: the axioms are consistent*)

(**Some Corollaries**)
 (*C1*) theorem C1: "⌊∀E P x. ((ℰᴬ E x) ∧ (P x)) → (E ⇛ P)⌋" nitpick oops (*countermodel*)
 (*C2*) theorem C2: "⌊∀X. ¬𝒫 X → □(¬𝒫 X)⌋" using A4 symm by blast
   definition h4 ("𝒩") where "𝒩 X ≡  ¬𝒫 X"
 (*C3*) theorem C3: "⌊∀X. 𝒩 X → □(𝒩 X)⌋" by (simp add: C2 h4_def)
```

Ontological Argument: Variant by Anderson

```
theory AndersonProof imports IHOML
```

```
(**Positive Properties and Ultrafilters**)
  abbreviation emptySet ("∅") where "∅ ≡ λx w. False"
  abbreviation entails (infixr"⊆"51) where "φ⊆ψ  ≡ ∀x w. φ x w ⟶ ψ x w"
  abbreviation andPred (infixr"⊓"51) where "φ⊓ψ  ≡ λx w. φ x w ∧ ψ x w"
  abbreviation negpred ("¯_"[52]53) where "¯ψ   ≡ λx w. ¬ψ x w"
  abbreviation "ultrafilter Φ cw ≡
      ¬(Φ ∅ cw)
    ∧  (∀φ. ∀ψ. (Φ φ cw ∧ Φ ψ cw) ⟶ (Φ (φ ⊓ ψ) cw))
    ∧  (∀φ::e⇒i⇒bool. ∀ψ::e⇒i⇒bool. (Φ φ cw ∨ Φ (¯φ) cw) ∧ ¬(Φ φ cw ∧ Φ (¯φ) cw))
    ∧  (∀φ::e⇒i⇒bool. ∀ψ::e⇒i⇒bool.  (Φ φ cw ∧ φ ⊆ ψ) ⟶ Φ ψ cw)"

(*U1*) theorem U1: "∀w. ultrafilter 𝒫 w" nitpick[user_axioms,format=2,show_all] oops (*counterm.*)
  lemma helpC: "∀φ ψ w. (𝒫 φ w ∨ 𝒫 (¯φ) w) ∧ ¬(𝒫 φ w ∧ 𝒫 (¯φ) w)" nitpick oops (*countermodel*)

(*⦇φ⦈ converts an extensional object φ into `rigid' intensional one*)
  abbreviation trivialConversion ("⦇_⦈") where "⦇φ⦈ ≡ (λw. φ)"
(*Q ↓φ: the extension of a (possibly) non-rigid predicate φ is turned into a rigid intensional one,
  then Q is applied to the latter; ↓φ can be read as "the rigidly intensionalised predicate φ"*)
  abbreviation mextPredArg (infix "↓" 60) where "Q ↓φ ≡ λw. Q (λx. ⦇φ x w⦈) w"
  lemma "∀Q φ. Q φ = Q ↓φ" nitpick oops (*countermodel: the two notions are not the same*)

  lemma helpE: "∀w.¬((𝒫 ↓∅) w)" using T1 by blast
  lemma helpF: "∀φ ψ w.((𝒫 ↓φ) w ∧ (𝒫 ↓ψ) w) ⟶ ((𝒫 ↓(φ⊓ψ)) w)" by (smt GA_def T3 T5 symm)
  lemma helpG: "∀w.((𝒫 ↓φ)w ∨ (𝒫 ↓(¯φ))w) ∧ ¬((𝒫 ↓φ)w ∧ (𝒫 ↓(¯φ))w)" by (smt GA_def T3 T5 symm)
  lemma helpH: "∀w.((𝒫 ↓φ) w ∧ φ⊆ψ) ⟶ (𝒫 ↓ψ) w" by (metis (no_types, lifting) A4 C2 GA_def T3)

  abbreviation "𝒫' φ ≡ (𝒫 ↓φ)" (*𝒫': the set of all rigidly intensionalised positive properties*)

(*U2*) theorem U2: "∀w. ultrafilter 𝒫' w"  using helpE helpF helpG helpH by simp
(*U3*) theorem U3: "(𝒫' ⊆ 𝒫) ∧ (𝒫 ⊆ 𝒫')" nitpick oops (*countermodel: 𝒫' and 𝒫 are not equal*)
```

Ontological Argument: Variant by Fitting (2002)



Ontological Argument: Variant by Fitting (2002)

Part I - Proving that God's existence is possible

D1 Being Godlike is equivalent to having all positive properties.

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

A2 Any property entailed by a positive property is positive.

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

From A1 and A2 follows theorem T1:

T1 Every positive property is possibly instantiated.

From D1 and A3 follows:

T2 Being Godlike is a positive property.

From T1 and T2 follows:

T3 Being Godlike is possibly instantiated.

Fully analogous to Gödel/Scott.

But: "positive properties" applied to extensions of properties only!

Ontological Argument: Variant by Fitting (2002)

Part I - Proving that God's existence is possible

D1

A1

A2

A3

Fro

T1

Fro

T2

Fro

T3

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 properties are nec. entailed by E.[a]

A4 Positive properties are necessarily positive.

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

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

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

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.

[a] The 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

A1

A2

A3

Fro

T1

Fro

T2

Fro

T3

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

D2 A property E is the essence of an individual x iff <u>x has E and</u> all of x's properties are nec. entailed by E.[a]

A4 Positive properties are necessarily positive.

From

T4 E

D3 N
i

A5 N

From

T5 E

And

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

T6 Being Godlike is necessarily instantiated.

[a] The underlined part in definition D2 has been added by Scott. Gödel originally omitted this part.

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

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

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

Ontological Argument: Variant by Fitting (2002)

```
theory FittingProof imports IHOML
begin
(*Positiveness/perfection: uninterpreted constant symbol*)
  consts Positiveness::"(e⇒bool)⇒i⇒bool" ("𝒫")
(*Some auxiliary definitions*)
 (*⦇φ⦈ converts an extensional object φ into `rigid' intensional one*)
  abbreviation trivialConversion ("⦇_⦈") where "⦇φ⦈ ≡ (λw. φ)"
  abbreviation Entails (infix"⇛" 60) where "X⇛Y ≡ □(∀^E z. ⦇X z⦈→⦇Y z⦈)"
 (*φ's argument is a relativized term (of extensional type) derived from an intensional predicate.*)
  abbreviation extPredArg (infix "↓" 60) where "φ ↓P ≡ λw. φ (λx. P x w) w"
 (*A variant of the latter where φ takes intensional terms as argument.*)
  abbreviation mextPredArg (infix "↓" 60) where "φ ↓P ≡ λw. φ (λx. ⦇P x w⦈) w"
 (*Another variant where φ has two arguments (the first one being relativized).*)
  abbreviation extPredArg1 (infix "↓_1" 60) where "φ ↓_1P ≡ λz. λw. φ (λx. P x w) z w"

(**Part I**)
 (*D1*) abbreviation God ("G") where "G ≡ (λx. ∀Y. 𝒫 Y → ⦇Y x⦈)"
 (*A1*) axiomatization where A1a:"⌊∀X. 𝒫 (⇁X) → ¬(𝒫 X) ⌋" and A1b:"⌊∀X. ¬(𝒫 X) → 𝒫 (⇁X)⌋"
 (*A2*) axiomatization where A2: "⌊∀X Y. (𝒫 X ∧ (X ⇛ Y)) → 𝒫 Y⌋"
 (*T1*) theorem T1: "⌊∀X. 𝒫 X → ◇(∃^E z. ⦇X z⦈)⌋" using A1a A2 by blast
 (*T2*) axiomatization where T2: "⌊𝒫 ↓G⌋"
 (*T3*) theorem T3deRe: "⌊(λX. ◇∃^E X) ↓G⌋" using T1 T2 by simp
        theorem T3deDicto: "⌊◇∃^E ↓G⌋" nitpick oops (*countermodel*)
```

Ontological Argument: Variant by Fitting (2002)

```
theory FittingProof imports IHOML
begin
(*Positiveness/perfection: uninterpreted constant symbol*)
  consts Positiveness::"(e⇒bool)⇒i⇒bool" ("𝒫")
```

```
(**Part II*)
 (*Logic KB*)  axiomatization where symm: "symmetric aRel"
 (*A4*) axiomatization where A4: "⌊∀X. 𝒫 X → □(𝒫 X)⌋"
 (*D2*) abbreviation Essence ("ℰ") where "ℰ Y x ≡ ⦇Y x⦈ ∧ (∀Z.⦇Z x⦈→(Y⇛Z))"
 (*T4*) theorem T4: "⌊∀x. G x → ((ℰ ↓₁G) x)⌋" using A1b by metis
 (*D3*) definition NE ("NE") where "NE x  ≡ λw. (∀Y.  ℰ Y x → □(∃ᴱz. ⦇Y z⦈)) w"
 (*A5*) axiomatization where A5: "⌊𝒫 ↓NE⌋"
   lemma help1: "⌊∃ ↓G → □∃ᴱ ↓G⌋" sorry (*longer interactive proof, omitted here*)
   lemma help2: "⌊∃ ↓G → ((λX. □∃ᴱ X) ↓G)⌋" by (metis A4 help1)
 (*T5*) theorem T5deDicto:"⌊◇∃ᴱ ↓G⌋⟶⌊□∃ᴱ ↓G⌋" using T3deRe help1 by blast
        theorem T5deRe:"⌊(λX. ◇∃ᴱ X) ↓G⌋ ⟶ ⌊(λX. □∃ᴱ X) ↓G⌋" by (metis A4 help2)
 (*T6*) theorem T6deDicto: "⌊□∃ᴱ ↓G⌋" using T3deRe help1 by blast
        theorem T6deRe: "⌊(λX. □∃ᴱ X) ↓G⌋" using T3deRe help2 by blast

(**Consistency**)
lemma True nitpick[satisfy] oops  (*Model found by Nitpick: the axioms are consistent*)

(**Modal Collapse**)
  lemma ModalCollapse: "⌊∀Φ.(Φ → (□ Φ))⌋" nitpick oops (*countermodel*)

(**Some Corollaries**)
(* Todo (*C1*) theorem C1: "⌊∀E P x. ((ℰ E x) ∧ (P x)) → (E ⇛ P)⌋" by (metis ess_def) *)
 (*C2*) theorem C2: "⌊∀X. ¬𝒫 X → □(¬𝒫 X)⌋" using A4 symm by blast
   definition h4 ("𝒩") where "𝒩 X ≡  ¬𝒫 X"
 (*C3*) theorem C3: "⌊∀X. 𝒩 X → □(𝒩 X)⌋" by (simp add: C2 h4_def)
   definition "rigid φ ≡ ∀x. φ x → □(φ x)"
 (*C4*) theorem "⌊∀φ. 𝒫 φ  → rigid (λx. ⦇φ x⦈)⌋" by (simp add: rigid_def)
 (*C5*) theorem "⌊rigid 𝒫⌋" by (simp add: A4 rigid_def)
```

Ontological Argument: Variant by Fitting (2002)

```
theory FittingProof imports IHOML
begin
(*Positiveness/perfection: uninterpreted constant symbol*)
```

```
(**Part II*)
 (*Logic KB*)  axiomatization where symm: "symmetric aRel"
 (*A4*) axiomatization where A4: "⌊∀X. 𝒫 X → □(𝒫 X)⌋"
 (*D2*) abbreviation Essence ("ℰ") where "ℰ Y x ≡ ⦇Y x⦈ ∧ (∀Z.⦇Z x⦈→(Y⇛Z))"
 (*T4*) theorem T4: "⌊∀x. G x → ((ℰ ↓₁G) x)⌋" using A1b by metis
 (*D3*) definition NE ("NE") where "NE x  ≡ λw. (∀Y.  ℰ Y x → □(∃ᴱz. ⦇Y z⦈)) w"
```

```
(**Positive Properties and Ultrafilters**)
  abbreviation empty ("∅")   where "∅ ≡ λx. False"
  abbreviation intersect (infix "⊓" 52) where "φ ⊓ ψ ≡ (λx. φ x ∧ ψ x)"
  abbreviation nnegpred ("¬_"[52]53) where "¬ψ    ≡ λx. ¬ψ(x)"
  abbreviation entail (infixr"⊆"51)   where "φ⊆ψ   ≡ ∀x. φ x ⟶ ψ x"
  abbreviation "ultrafilter Φ cw ≡
       ¬(Φ ∅ cw)   (* The empty set is not an element of U *)
   ∧  (∀φ::(e⇒bool). ∀ψ::(e⇒bool). (Φ φ cw ∧ Φ ψ cw) ⟶ (Φ (φ⊓ψ) cw))
   ∧  (∀φ::(e⇒bool). ∀ψ::(e⇒bool). (Φ φ cw ∨ Φ (¬φ) cw) ∧ ¬(Φ φ cw ∧ Φ (¬φ) cw))
   ∧  (∀φ::(e⇒bool). ∀ψ::(e⇒bool).  (Φ φ cw ∧ φ⊆ψ) ⟶ Φ ψ cw)"
  lemma lemmaA: "∀w. ¬(𝒫 ∅ w)"  using T1 by blast
  lemma lemmaB: "∀w. (𝒫 φ w ∧ 𝒫 ψ w) ⟶ (𝒫 (φ⊓ψ) w)" by (metis A1b T3deRe)
  lemma lemmaC: "∀w. (𝒫 φ w ∨ 𝒫 (¬φ) w) ∧ ¬(𝒫 φ w ∧ 𝒫 (¬φ) w)" using A1a A1b by blast
  lemma lemmaD: "∀w. (𝒫 φ w ∧ φ⊆ψ) ⟶ 𝒫 ψ w" by (smt A2)

(*U1*) theorem "∀w. ultrafilter 𝒫 w"  by (smt lemmaA lemmaB lemmaC lemmaD)
```

```
 (*C4*) theorem "⌊∀φ. 𝒫 φ  → rigid (λx. ⦇φ x⦈)⌋" by (simp add: rigid_def)
 (*C5*) theorem "⌊rigid 𝒫⌋" by (simp add: A4 rigid_def)
```

Summary of Results

- "Godlike" has been defined in terms of "positive properties"
- "positive properties" has been linked with the notion of "ultrafilter".
- In our experiments we then distinguished between

  $\mathcal{P}$: positive intensional properties

  $\mathcal{P}'$: positive ("rigidly intensionalised") extensions of properties
- Gödel/Scott variant axiomatises $\mathcal{P}$: $\mathcal{P} = \mathcal{P}'$ is an ultrafilter
- Anderson's variant axiomatises $\mathcal{P}$: $\mathcal{P} \neq \mathcal{P}'$; only $\mathcal{P}'$ is an ultrafilter
- Fitting's variant axiomatises only $\mathcal{P}'$: $\mathcal{P}'$ is an ultrafilter

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

They achieve this in seemingly different ways.

Mathematically, however, their solutions appear closely related.

Summary of Results

- "Godlike" has been defined in terms of "positive properties"
- "positive properties" has been linked with the notion of "ultrafilter".
- In our experiments we then distinguished between

  $\mathcal{P}$: positive intensional properties
  $\mathcal{P}'$: positive ("rigidly intensionalised") extensions of properties

- Gödel/Scott variant axiomatises $\mathcal{P}$: $\mathcal{P} = \mathcal{P}'$ is an ultrafilter
- Anderson's variant axiomatises $\mathcal{P}$: $\mathcal{P} \neq \mathcal{P}'$; only $\mathcal{P}'$ is an ultrafilter
- Fitting's variant axiomatises only $\mathcal{P}'$: $\mathcal{P}'$ is an ultrafilter

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

They achieve this in seemingly different ways.

Mathematically, however, their solutions appear closely related.

Summary of Results

- "Godlike" has been defined in terms of "positive properties"
- "positive properties" has been linked with the notion of "ultrafilter".
- In our experiments we then distinguished between

  $\mathcal{P}$: positive intensional properties
  $\mathcal{P}'$: positive ("rigidly intensionalised") extensions of properties

- Gödel/Scott variant axiomatises $\mathcal{P}$: $\mathcal{P} = \mathcal{P}'$ is an ultrafilter
- Anderson's variant axiomatises $\mathcal{P}$: $\mathcal{P} \neq \mathcal{P}'$; only $\mathcal{P}'$ is an ultrafilter
- Fitting's variant axiomatises only $\mathcal{P}'$: $\mathcal{P}'$ is an ultrafilter

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

They achieve this in seemingly different ways.

Mathematically, however, their solutions appear closely related.

Summary of Results

- "Godlike" has been defined in terms of "positive properties"
- "positive properties" has been linked with the notion of "ultrafilter".
- In our experiments we then distinguished between

  $\mathcal{P}$: positive intensional properties

  $\mathcal{P}'$: positive ("rigidly intensionalised") extensions of properties

- Gödel/Scott variant axiomatises $\mathcal{P}$: $\mathcal{P} = \mathcal{P}'$ is an ultrafilter
- Anderson's variant axiomatises $\mathcal{P}$: $\mathcal{P} \neq \mathcal{P}'$; only $\mathcal{P}'$ is an ultrafilter
- Fitting's variant axiomatises only $\mathcal{P}'$: $\mathcal{P}'$ is an ultrafilter

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

They achieve this in seemingly different ways.

Mathematically, however, their solutions appear closely related.

Summary of Results

- "Godlike" has been defined in terms of "positive properties"
- "positive properties" has been linked with the notion of "ultrafilter".
- In our experiments we then distinguished between

  $\mathcal{P}$: positive intensional properties
  $\mathcal{P}'$: positive ("rigidly intensionalised") extensions of properties

- Gödel/Scott variant axiomatises $\mathcal{P}$: $\mathcal{P} = \mathcal{P}'$ is an ultrafilter
- Anderson's variant axiomatises $\mathcal{P}$: $\mathcal{P} \neq \mathcal{P}'$; only $\mathcal{P}'$ is an ultrafilter
- Fitting's variant axiomatises only $\mathcal{P}'$: $\mathcal{P}'$ is an ultrafilter

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

They achieve this in seemingly different ways.

Mathematically, however, their solutions appear closely related.

Summary of Results

- "Godlike" has been defined in terms of "positive properties"
- "positive properties" has been linked with the notion of "ultrafilter".
- In our experiments we then distinguished between

  $\mathcal{P}$: positive intensional properties

  $\mathcal{P}'$: positive ("rigidly intensionalised") extensions of properties
- Gödel/Scott variant axiomatises $\mathcal{P}$: $\mathcal{P} = \mathcal{P}'$ is an ultrafilter
- Anderson's variant axiomatises $\mathcal{P}$: $\mathcal{P} \neq \mathcal{P}'$; only $\mathcal{P}'$ is an ultrafilter
- Fitting's variant axiomatises only $\mathcal{P}'$: $\mathcal{P}'$ is an ultrafilter

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

They achieve this in seemingly different ways.

Mathematically, however, their solutions appear closely related.

Part D

— Discussion —

Discussion

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

Question

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

Discussion

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

Question

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

Discussion

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

Question

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

Conclusion

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

Evidence provided for central claim of this talk

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

Related work

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

Conclusion

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

Evidence provided for central claim of this talk

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

Related work

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

Conclusion

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

Evidence provided for central claim of this talk

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

Related work

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

DEMO: Anderson's Variant

Isabelle2018/HOL - AndersonProof.thy (modified)

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

```
theory AndersonProof imports IHOML
begin
(*Positiveness/perfection: uninterpreted constant symbol*)
  consts positiveProperty::"(e⇒i⇒bool)⇒i⇒bool" ("𝒫")
(*Some auxiliary definitions*)
  definition h3 (infix "⇛" 60) where "X ⇛ Y ≡  □(∀ᴱz. X z → Y z)"
(**Part I**)
 (*D1'*)  definition GA ("Gᴬ") where "Gᴬ ≡ λx. ∀Y. (𝒫 Y) ↔ □(Y x)"
 (*A1a*) axiomatization where A1a:"⌊∀X. 𝒫 (→X) → ¬(𝒫 X) ⌋"
 (*A2*)  axiomatization where A2: "⌊∀X Y. (𝒫 X ∧ (X ⇛ Y)) → 𝒫 Y⌋"
 (*T1*)  theorem T1: "⌊∀X. 𝒫 X → ◇∃ᴱ X⌋" using A1a A2 h3_def by metis
 (*T2*)  axiomatization where T2: "⌊𝒫 Gᴬ⌋"  (*here we postulate T2 instead of proving it*)
 (*T3*)  theorem T3: "⌊◇∃ᴱ Gᴬ⌋" using T1 T2 h3_def by blast
(**Part II**)
 (*Logic KB*) axiomatization where symm: "∀x y. x r y ⟶ y r x"
 (*A4*)  axiomatization where A4: "⌊∀X. 𝒫 X → □(𝒫 X)⌋"
 (*D2'*) abbreviation essA ("ℰᴬ") where "ℰᴬ Y x ≡ (∀Z. □(Z x) ↔ Y ⇛ Z)"
 (*T4*)  theorem T4: "⌊∀x. Gᴬ x → (ℰᴬ Gᴬ x)⌋" by (metis A2 GA_def T2 symm h3_def)
 (*D3*)  abbreviation NEA ("NEᴬ") where "NEᴬ x  ≡ (λw. (∀Y. ℰᴬ Y x → □∃ᴱ Y) w)"
 (*A5*)  axiomatization where A5: "⌊𝒫 NEᴬ⌋"
 (*T5*)  theorem T5: "⌊◇∃ᴱ Gᴬ⌋ ⟶ ⌊□∃ᴱ Gᴬ⌋" by (metis A2 GA_def T2 symm h3_def)
 (*T6*)  theorem T6: "⌊□∃ᴱ Gᴬ⌋" using T3 T5 by blast

(**Modal collapse is countersatisfiable**)
lemma "⌊∀Φ.(Φ → (□ Φ))⌋" nitpick[user_axioms] oops (*Countermodel found by Nitpick*)
```

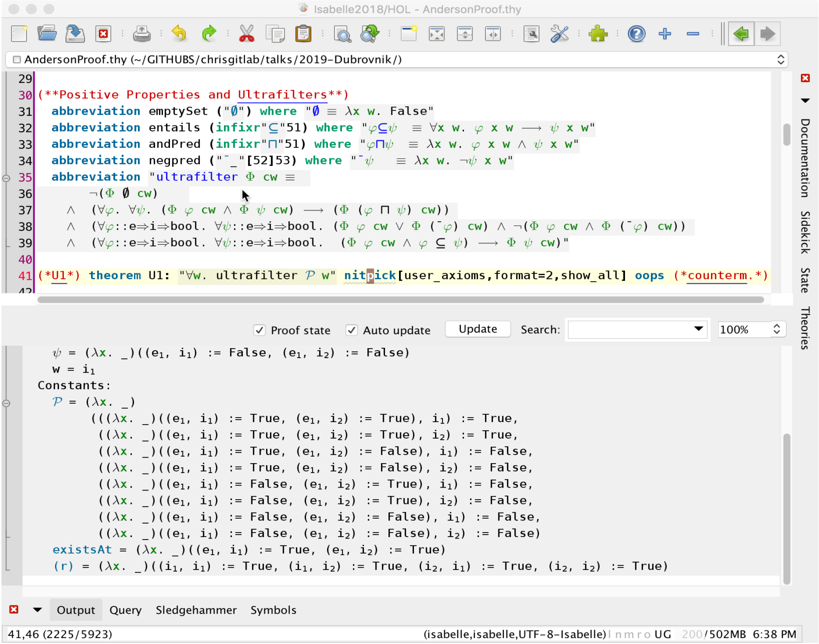
Proof state | Auto update | Update | Search: | 100%

Output | Query | Sledgehammer | Symbols

28,53 (1529/6107)   (isabelle,isabelle,UTF-8-Isabelle) | n m r o UG 195/587MB 6:25 PM

Documentation | Sidekick | State | Theories

DEMO: Anderson's Variant



DEMO: Anderson's Variant

Isabelle2018/HOL - AndersonProof.thy (modified)

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

```
abbreviation trivialConversion ("(_)") where "(φ) ≡ (λw. φ)"
(*Q ↓φ: the extension of a (possibly) non-rigid predicate φ is turned into a rigid intensional one,
 then Q is applied to the latter; ↓φ can be read as "the rigidly intensionalised predicate φ"*)
 abbreviation mextPredArg (infix "↓" 60) where "Q ↓φ ≡ λw. Q (λx. (φ x w)) w"

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

 lemma helpE: "∀w.¬((𝒫 ↓∅) w)" using T1 by blast
 lemma helpF: "∀φ ψ w.((𝒫 ↓φ) w ∧ (𝒫 ↓ψ) w) ⟶ ((𝒫 ↓(φ⊓ψ)) w)" by (smt GA_def T3 T5 symm)
 lemma helpG: "∀w.((𝒫 ↓φ)w ∨ (𝒫 ↓(⁻φ))w) ∧ ¬((𝒫 ↓φ)w ∧ (𝒫 ↓(⁻φ))w)" by (smt GA_def T3 T5 symm)
 lemma helpH: "∀w.((𝒫 ↓φ) w ∧ φ⊆ψ) ⟶ (𝒫 ↓ψ) w" by (metis GA_def T3 T5 symm)

 abbreviation "𝒫' φ ≡ (𝒫 ↓φ)" (*𝒫': the set of all rigidly intensionalised positive properties*)

(*U2*) theorem U2: "∀w. ultrafilter 𝒫' w" using helpE helpF helpG helpH by simp
(*U3*) theorem U3: "(𝒫' ⊆ 𝒫) ∧ (𝒫 ⊆ 𝒫')" nitpick[user_axioms] oops (*countermodel: 𝒫',𝒫 not equal
```

Proof state ✓ Auto update ✓ Update Search: 100%

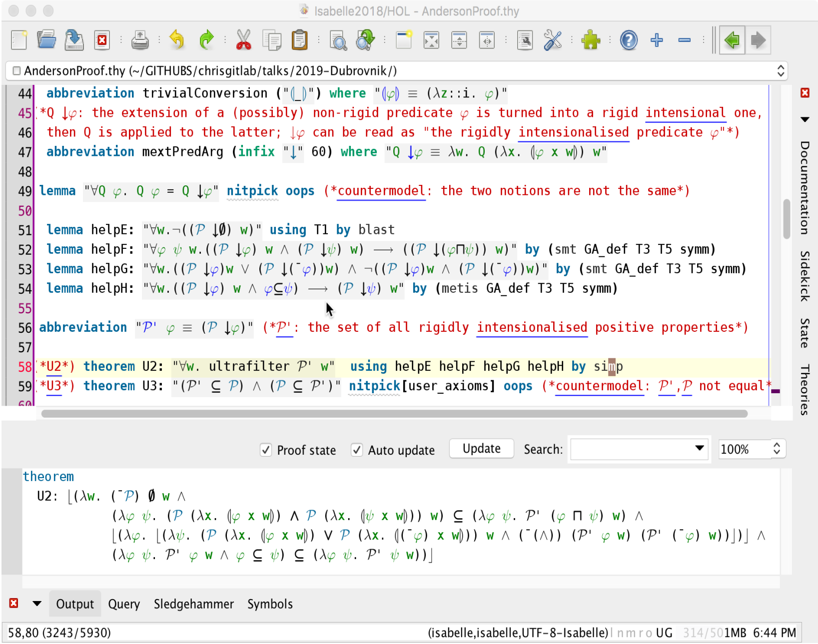
```
theorem
  U2: ⌊(λw. (⁻𝒫) (⦇False⦈) w ∧
          (λφ ψ. (𝒫 (λx. (φ x w)) ∧ 𝒫 (λx. (ψ x w))) w) ⊆ (λφ ψ. 𝒫' (φ ⊓ ψ) w) ∧
          ⌊(λφ. ⌊(𝒫 (λx. (φ x w)) ∨ 𝒫 (λx. ((⁻φ) x w))) w ∧ (⁻(∧)) (𝒫' φ w) (𝒫' (⁻φ) w)⦈⌋)⌋ ∧
          (λφ ψ. 𝒫' φ w ∧ φ ⊆ ψ) ⊆ (λψ. 𝒫' ψ w))⌋
```

Output Query Sledgehammer Symbols

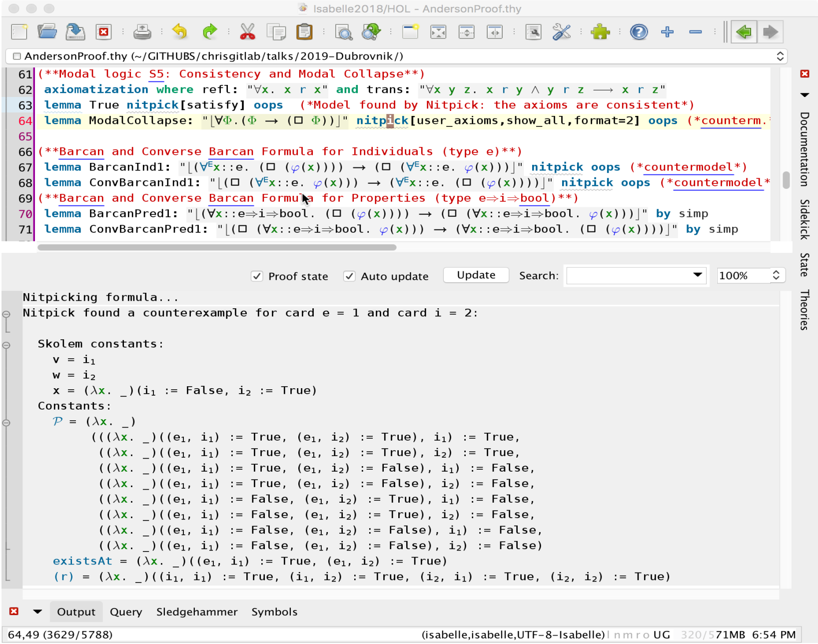
58,79 (3239/5927) (isabelle,isabelle,UTF-8-Isabelle) n m r o UG 193/500MB 6:41 PM

Documentation Sidekick State Theories

DEMO: Anderson’s Variant



DEMO: Anderson’s Variant



DEMO: Anderson’s Variant

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

(* Some tests *)
consts aw::i peter::e mary::e supreme_being::e loves::"e⇒e⇒i⇒bool"
axiomatization where t1: "⌊𝒫 (λx. loves x mary)⌋" and
 t2: "¬(peter = mary)" and t3: "¬(peter = supreme_being)" and t4: "¬(mary = supreme_being)" and
 t5: "¬(Gᴬ peter aw)" and t6:  "¬(Gᴬ mary aw)" and t7: "¬(loves peter mary aw)"
consts P_prime::"(e⇒i⇒bool)⇒i⇒bool"
 axiomatization where  t8: "P_prime = 𝒫'"

lemma "(ultrafilter 𝒫' aw) ∧ ¬(ultrafilter 𝒫 aw)"
       nitpick[        user_axioms,show_all,format=3] (*counterm.*)
       nitpick[satisfy,user_axioms,show_all,format=3,timeout=100]  oops
```

```
Proof state   Auto update   Update   Search:   100%

Constants:
  P_prime =
    (λx. _)
    (((λx. _)((e₁, i₁) := True, (e₁, i₂) := True, (e₂, i₁) := True, (e₂, i₂) := True, (e₃, i₁) := True, (e₃, i₂) := True), i₁) := True,
     ((λx. _)((e₁, i₁) := True, (e₁, i₂) := True, (e₂, i₁) := True, (e₂, i₂) := True, (e₃, i₁) := True, (e₃, i₂) := True), i₂) := True,
     ((λx. _)((e₁, i₁) := True, (e₁, i₂) := True, (e₂, i₁) := True, (e₂, i₂) := True, (e₃, i₁) := True, (e₃, i₂) := False), i₁) := True,
     ((λx. _)((e₁, i₁) := True, (e₁, i₂) := True, (e₂, i₁) := True, (e₂, i₂) := True, (e₃, i₁) := True, (e₃, i₂) := False), i₂) := True,
     ((λx. _)((e₁, i₁) := True, (e₁, i₂) := True, (e₂, i₁) := True, (e₂, i₂) := True, (e₃, i₁) := False, (e₃, i₂) := True), i₁) := True,
     ((λx. _)((e₁, i₁) := True, (e₁, i₂) := True, (e₂, i₁) := True, (e₂, i₂) := True, (e₃, i₁) := False, (e₃, i₂) := True), i₂) := True,
     ((λx. _)((e₁, i₁) := True, (e₁, i₂) := True, (e₂, i₁) := True, (e₂, i₂) := True, (e₃, i₁) := False, (e₃, i₂) := False), i₁) := True,
     ((λx. _)((e₁, i₁) := True, (e₁, i₂) := True, (e₂, i₁) := True, (e₂, i₂) := True, (e₃, i₁) := False, (e₃, i₂) := False), i₂) := True,
     ((λx. _)((e₁, i₁) := True, (e₁, i₂) := True, (e₂, i₁) := True, (e₂, i₂) := False, (e₃, i₁) := True, (e₃, i₂) := True), i₁) := True,
     ((λx. _)((e₁, i₁) := True, (e₁, i₂) := True, (e₂, i₁) := True, (e₂, i₂) := False, (e₃, i₁) := True, (e₃, i₂) := True), i₂) := False,
     ((λx. _)((e₁, i₁) := True, (e₁, i₂) := True, (e₂, i₁) := True, (e₂, i₂) := False, (e₃, i₁) := True, (e₃, i₂) := False), i₁) := True,
     ((λx. _)((e₁, i₁) := True, (e₁, i₂) := True, (e₂, i₁) := True, (e₂, i₂) := False, (e₃, i₁) := True, (e₃, i₂) := False), i₂) := False,
     ((λx. _)((e₁, i₁) := True, (e₁, i₂) := True, (e₂, i₁) := True, (e₂, i₂) := False, (e₃, i₁) := False, (e₃, i₂) := True), i₁) := True,
     ((λx. _)((e₁, i₁) := True, (e₁, i₂) := True, (e₂, i₁) := True, (e₂, i₂) := False, (e₃, i₁) := False, (e₃, i₂) := True), i₂) := False,
     ((λx. _)((e₁, i₁) := True, (e₁, i₂) := True, (e₂, i₁) := True, (e₂, i₂) := False, (e₃, i₁) := False, (e₃, i₂) := False), i₁) := True,
     ((λx. _)((e₁, i₁) := True, (e₁, i₂) := True, (e₂, i₁) := True, (e₂, i₂) := False, (e₃, i₁) := False, (e₃, i₂) := False), i₂) := False,

Output  Query  Sledgehammer  Symbols
83,43 (4896/5398)   (isabelle,isabelle,UTF-8-Isabelle) nmro UG  133/500MB 7:05 PM
```

DEMO: Anderson’s Variant