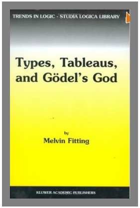
Gödel's God on the Computer

Christoph Benzmüller and **Bruno Woltzenlogel Paleo**

Invited Presentation, IWIL @ LPAR-2013
Stellenbosch, South Africa, December 14, 2013



$$\dfrac{\dfrac{\text{Axiom 3}}{P(G)} \qquad \dfrac{\dfrac{\text{Theorem 1}}{\forall\varphi.[P(\varphi) \to \Diamond\exists x.\varphi(x)]}}{P(G) \to \Diamond\exists x.G(x)}\,\forall_E}{\Diamond\exists x.G(x)}\,\to_E$$

A gift to **Priest Edvaldo** in Piracicaba, Brazil


Login | Registrierung

SPIEGEL ONLINE WISSENSCHAFT

Politik | Wirtschaft | Panorama | Sport | Kultur | Netzwelt | Wissenschaft | Gesundheit | einestages | Karriere | Uni | Schule | Reise | Auto

Nachrichten > Wissenschaft > Mensch > Mathematik > Formel von Kurt Gödel: Mathematiker bestätigen Gottesbeweis

Formel von Kurt Gödel: Mathematiker bestätigen Gottesbeweis

Von Tobias Hürter

picture-alliance/ Imagno/ Wiener Stadt- und Landesbibliothek

Kurt Gödel (um das Jahr 1935): Der Mathematiker hielt seinen Gottesbeweis jahrzehntelang geheim

Ein Wesen existiert, das alle positiven Eigenschaften in sich vereint. Das bewies der legendäre Mathematiker Kurt Gödel mit einem komplizierten Formelgebilde. Zwei Wissenschaftler haben diesen Gottesbeweis nun überprüft - und für gültig befunden.

Montag, 09.09.2013 – 12:03 Uhr

Drucken | Versenden | Merken

Jetzt sind die letzten Zweifel ausgeräumt: Gott existiert tatsächlich. Ein Computer hat es mit kalter Logik bewiesen - das MacBook des Computerwissenschaftlers Christoph Benzmüller von der Freien Universität Berlin.

Germany
- Telepolis & Heise
- Spiegel Online
- FAZ
- Die Welt
- Berliner Morgenpost
- Hamburger Abendpost
- . . .

Austria
- Die Presse
- Wiener Zeitung
- ORF
- . . .

Italy
- Repubblica
- Ilsussidario
- . . .

India
- DNA India
- Delhi Daily News
- India Today
- . . .

US
- ABC News
- . . .

International
- Spiegel International
- Yahoo Finance
- United Press Intl.
- . . .

Home | Video | Themen | Forum | English | DER SPIEGEL | SPIEGEL TV | Abo | Shop   RSS | Mobile | Newsletter

Sign in | Register

SPIEGEL ONLINE INTERNATIONAL

Front Page World Europe Germany Business Zeitgeist Newsletter

English Site > Germany > Science > Scientists Use Computer to Mathematically Prove Gödel God Theorem

Holy Logic: Computer Scientists 'Prove' God Exists

By David Knight

picture-alliance/ Imagno/ Wiener Stadt- und Landesbibliothek

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

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

Germany
- Telepolis & Heise
- Spiegel Online
- FAZ
- Die Welt
- Berliner Morgenpost
- Hamburger Abendpost
- ...

Austria
- Die Presse
- Wiener Zeitung
- ORF
- ...

Italy
- Repubblica
- Ilsussidario
- ...

India
- DNA India
- Delhi Daily News
- India Today
- ...

US
- ABC News
- ...

International
- Spiegel International
- Yahoo Finance
- United Press Intl.
- ...


DIE WELT

FREITAG, 18. OKTOBER 2013

Zippert zappt

Teuer erkaufte Harmonie

Mindestlohn lockt SPD in dritte Ehe mit der Union

Mittwoch soll über eine große Koalition verhandelt werden. Votum des SPD-Konvents steht aus

THEMEN

Sport
Bundestrainer Löw unterschreibt beim DFB bis 2016

Wirtschaft
Der Jumbojet wird bei Boeing zum Ladenhüter

Karriere
Stress im Job
So legen Sie sich einen dicken Panzer zu

Aus aller Welt
Mietprozess in Paris: Dürfen 15 Quadratmeter 330 Euro kosten?

Dax
Im Minus

Göttliche Mathematik

Haben Forscher mit einem Computerprogramm die Existenz Gottes bewiesen?

Austria
- Die Presse
- Wiener Zeitung
- ORF
- ...

Italy
- Repubblica
- Ilsussidario
- ...

India
- DNA India
- Delhi Daily News
- India Today
- ...

US
- ABC News
- ...

International
- Spiegel International
- Yahoo Finance
- United Press Intl.
- ...

SCIENCE NEWS

HOME / SCIENCE NEWS / RESEARCHERS SAY THEY USED MACBOOK TO PROVE GOEDEL'S GOD THEOREM

Researchers say they used MacBook to prove Goedel's God theorem

Oct. 23, 2013 | 8:14 PM | 1 comments

Are we in contact with Steve Jobs? No

Do you really need a MacBook to obtain the results? No

Did Apple send us some money? No

(but maybe they should)

SCIENCE NEWS

HOME / SCIENCE NEWS / RESEARCHERS SAY THEY USED MACBOOK TO PROVE GOEDEL'S GOD THEOREM

Researchers say they used MacBook to prove Goedel's God theorem

Oct. 23, 2013 | 8:14 PM | 1 comments

Are we in contact with Steve Jobs? No

Do you really need a MacBook to obtain the results? No

Did Apple send us some money? No (but maybe they should)

Def: Ontological Argument

* deductive argument
* for the existence of God
* starting from premises, which are justified by pure reasoning, i.e. they do not depend on observation of the world.

Existence of God: different types of arguments/proofs

— a posteriori (use experience/observation in the world)
—— teleological
—— cosmological
—— moral
—— ...
— a priori (based on pure reasoning, independent)
—— ontological argument
——— definitional
——— modal
——— ...
—— other a priori arguments

Def: Ontological Argument

* deductive argument
* for the existence of God
* starting from premises, which are justified by pure reasoning, i.e. they do not depend on observation of the world.

Existence of God: different types of arguments/proofs

- a posteriori (use experience/observation in the world)
  - teleological
  - cosmological
  - moral
  - ...
- a priori (based on pure reasoning, independent)
  - ontological argument
    - definitional
    - modal
    - ...
  - other a priori arguments

Wohl eine jede Philosophie kreist um den ontologischen Gottesbeweis

(Adorno, Th. W.: Negative Dialektik. Frankfurt a. M. 1966, p.378)

Rich history on ontological arguments (pros and cons)

. . . Anselm v. C. Gaunilo . . . Th. Aquinas . . . . . . Descartes Spinoza Leibniz . . . Hume Kant . . . Hegel . . . Frege . . . Hartshorne Malcolm Lewis Plantinga Gödel . . .

Anselm's notion of God:

"God is that, than which nothing greater can be conceived."

Gödel's notion of God:

"A God-like being possesses all 'positive' properties."

To show by logical reasoning:

"(Necessarily) God exists."

Rich history on ontological arguments (pros and cons)

… Anselm v. C. Gaunilo … Th. Aquinas …… Descartes Spinoza Leibniz … Hume Kant … Hegel … Frege … Hartshorne Malcolm Lewis Plantinga Gödel …

Anselm's notion of God:

"God is that, than which nothing greater can be conceived."

Gödel's notion of God:

"A God-like being possesses all 'positive' properties."

To show by logical reasoning:

"(Necessarily) God exists."

Different Interests in Ontological Arguments:

- Philosophical: Boundaries of Metaphysics & Epistemology
  - We talk about a metaphysical concept (God),
  - but we want to draw a conclusion for the real world.
- Theistic: Successful argument should convince atheists
- Ours: Can computers (theorem provers) be used ...
  - ...to formalize the definitions, axioms and theorems?
  - ...to verify the arguments step-by-step?
  - ...to fully automate (sub-)arguments?

“Computer-assisted Theoretical Philosophy”

Challenge: No provers for *Higher-order Quantified Modal Logic* (QML)

Our solution: Embedding in *Higher-order Classical Logic* (HOL)

[BenzmüllerPaulson, Logica Universalis, 2013]

What we did (rough outline for remaining presentation!):

A: Pen and paper: detailed natural deduction proof

B: Formalization: in classical higher-order logic (HOL)

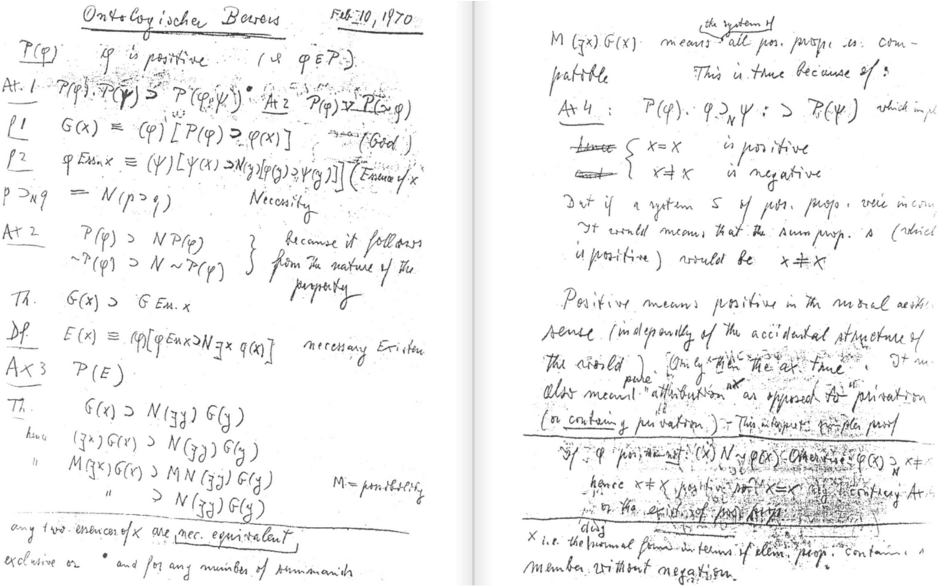
Automation: theorem provers LEO-II(**E**) and SATALLAX

Consistency: model finder NITPICK (NITROX)

C: Step-by-step verification: proof assistant COQ

D: Automation & verification: proof assistant ISABELLE

Did we get any new results? Yes — let's discuss this later!



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 the system of all pos. props. is compatible. This is true because of:

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

~~true~~ $x = x$ is positive

~~and~~ $x \neq x$ is negative

But if a system S of pos. props. were incons 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). This interpret. simpler proof

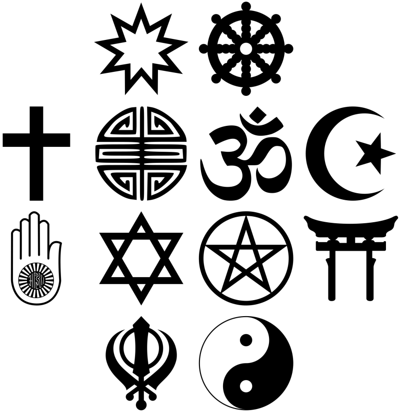
If φ pos. not: $(x)\, N \sim \varphi(x)$. Otherwise: $\varphi(x) \supset_N x \neq x$ hence $x \neq x$ positive not $x = x$ [illegible] At. [illegible] or the exist. of pos. [illegible]

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

Scott's Version of Gödel's Axioms, Definitions and Theorems

A1	Either a property or its negation is positive, but not both:	$\forall\phi[P(\neg\phi) \leftrightarrow \neg P(\phi)]$
A2	A property necessarily implied by a positive property is positive:	$\forall\phi\forall\psi[(P(\phi) \wedge \Box\forall x[\phi(x) \rightarrow \psi(x)]) \rightarrow P(\psi)]$
T1	Positive properties are possibly exemplified:	$\forall\varphi[P(\varphi) \rightarrow \Diamond\exists x\varphi(x)]$
D1	A *God-like* being possesses all positive properties:	$G(x) \leftrightarrow \forall\phi[P(\phi) \rightarrow \phi(x)]$
A3	The property of being God-like is positive:	$P(G)$
C	Possibly, God exists:	$\Diamond\exists x G(x)$
A4	Positive properties are necessarily positive:	$\forall\phi[P(\phi) \rightarrow \Box\, P(\phi)]$
D2	An *essence* of an individual is a property possessed by it and necessarily implying any of its properties:	$\phi\ ess.\ x \leftrightarrow \phi(x) \wedge \forall\psi(\psi(x) \rightarrow \Box\forall y(\phi(y) \rightarrow \psi(y)))$
T2	Being God-like is an essence of any God-like being:	$\forall x[G(x) \rightarrow G\ ess.\ x]$
D3	*Necessary existence* of an individual is the necessary exemplification of all its essences:	$NE(x) \leftrightarrow \forall\phi[\phi\ ess.\ x \rightarrow \Box\exists y\phi(y)]$
A5	Necessary existence is a positive property:	$P(NE)$
T3	Necessarily, God exists:	$\Box\exists x G(x)$

— see also the Isabelle/HOL handout —



Part A:
Proof Overview (ND style)

T3: $\Box \exists x. G(x)$

$$\frac{\textbf{C1: } \Diamond \exists z. G(z)}{\textbf{T3: } \Box \exists x. G(x)}$$

Proof Overview

$$\frac{\textbf{C1: } \Diamond \exists z.G(z) \qquad \textbf{L2: } \Diamond \exists z.G(z) \rightarrow \Box \exists x.G(x)}{\textbf{T3: } \Box \exists x.G(x)}$$

Proof Overview

$$\textbf{L2: } \Diamond \exists z. G(z) \rightarrow \Box \exists x. G(x)$$

$$\frac{\textbf{C1: } \Diamond \exists z. G(z) \qquad \textbf{L2: } \Diamond \exists z. G(z) \rightarrow \Box \exists x. G(x)}{\textbf{T3: } \Box \exists x. G(x)}$$

Proof Overview

$$\dfrac{\dfrac{\textbf{S5}}{\forall \xi.[\Diamond\Box\xi \rightarrow \Box\xi]}}{\textbf{L2: } \Diamond\exists z.G(z) \rightarrow \Box\exists x.G(x)}$$

$$\dfrac{\textbf{C1: } \Diamond\exists z.G(z) \qquad \textbf{L2: } \Diamond\exists z.G(z) \rightarrow \Box\exists x.G(x)}{\textbf{T3: } \Box\exists x.G(x)}$$

Proof Overview

$$\dfrac{\Diamond\exists z.G(z) \rightarrow \Diamond\Box\exists x.G(x) \qquad \dfrac{\textbf{S5}}{\overline{\forall}\xi.[\overline{\Diamond}\overline{\Box}\xi \mathbin{\overline{\rightarrow}} \overline{\Box}\xi]}}{\textbf{L2: } \Diamond\exists z.G(z) \rightarrow \Box\exists x.G(x)}$$

$$\dfrac{\textbf{C1: } \Diamond\exists z.G(z) \qquad \textbf{L2: } \Diamond\exists z.G(z) \rightarrow \Box\exists x.G(x)}{\textbf{T3: } \Box\exists x.G(x)}$$

$$\dfrac{\dfrac{\textbf{L1: } \exists z.G(z) \rightarrow \Box\exists x.G(x)}{\Diamond\exists z.G(z) \rightarrow \Diamond\Box\exists x.G(x)} \qquad \dfrac{\textbf{S5}}{\forall\xi.[\Diamond\Box\xi \rightarrow \Box\xi]}}{\textbf{L2: } \Diamond\exists z.G(z) \rightarrow \Box\exists x.G(x)}$$

$$\dfrac{\textbf{C1: } \Diamond\exists z.G(z) \qquad \textbf{L2: } \Diamond\exists z.G(z) \rightarrow \Box\exists x.G(x)}{\textbf{T3: } \Box\exists x.G(x)}$$

Proof Overview

$$\textbf{D1: } G(x) \equiv \forall\varphi.[P(\varphi) \to \varphi(x)]$$

$$\dfrac{\dfrac{\textbf{L1: } \exists z.G(z) \to \Box\exists x.G(x)}{\Diamond\exists z.G(z) \to \Diamond\Box\exists x.G(x)} \qquad \dfrac{\textbf{S5}}{\forall\xi.[\Diamond\Box\xi \to \Box\xi]}}{\textbf{L2: } \Diamond\exists z.G(z) \to \Box\exists x.G(x)}$$

$$\dfrac{\textbf{C1: } \Diamond\exists z.G(z) \qquad \textbf{L2: } \Diamond\exists z.G(z) \to \Box\exists x.G(x)}{\textbf{T3: } \Box\exists x.G(x)}$$

Proof Overview

$$\textbf{D1: } G(x) \equiv \forall\varphi.[P(\varphi) \rightarrow \varphi(x)]$$

$$\textbf{D3*: } NE(x) \equiv \Box\exists y.G(y)$$

$$\dfrac{\dfrac{\dfrac{P(NE)}{\textbf{L1: } \exists z.G(z) \rightarrow \Box\exists x.G(x)}}{\Diamond\exists z.G(z) \rightarrow \Diamond\Box\exists x.G(x)} \qquad \dfrac{\textbf{S5}}{\forall\xi.[\Diamond\Box\xi \rightarrow \Box\xi]}}{\textbf{L2: } \Diamond\exists z.G(z) \rightarrow \Box\exists x.G(x)}$$

$$\dfrac{\textbf{C1: } \Diamond\exists z.G(z) \qquad \textbf{L2: } \Diamond\exists z.G(z) \rightarrow \Box\exists x.G(x)}{\textbf{T3: } \Box\exists x.G(x)}$$

$$\textbf{D1:}\ G(x) \equiv \forall\varphi.[P(\varphi) \to \varphi(x)]$$

D3*: $NE(x) \equiv \Box\exists y.G(y)$ (cheating!)

$$\dfrac{\dfrac{\dfrac{P(NE)}{\textbf{L1:}\ \exists z.G(z) \to \Box\exists x.G(x)}}{\Diamond\exists z.G(z) \to \Diamond\Box\exists x.G(x)} \qquad \dfrac{\textbf{S5}}{\forall\xi.[\Diamond\Box\xi \to \Box\xi]}}{\textbf{L2:}\ \Diamond\exists z.G(z) \to \Box\exists x.G(x)}$$

$$\dfrac{\textbf{C1:}\ \Diamond\exists z.G(z) \qquad \textbf{L2:}\ \Diamond\exists z.G(z) \to \Box\exists x.G(x)}{\textbf{T3:}\ \Box\exists x.G(x)}$$

Proof Overview

$$\textbf{D1: } G(x) \equiv \forall\varphi.[P(\varphi) \rightarrow \varphi(x)]$$

$$\textbf{D3*: } NE(x) \equiv \Box\exists y.G(y) \qquad \textbf{D3: } NE(x) \equiv \forall\varphi.[\varphi\ ess\ x \rightarrow \Box\exists y.\varphi(y)]$$

$$\dfrac{\dfrac{\dfrac{\textbf{T2: } \forall y.[G(y) \rightarrow G\ ess\ y] \qquad P(NE)}{\textbf{L1: } \exists z.G(z) \rightarrow \Box\exists x.G(x)}}{\Diamond\exists z.G(z) \rightarrow \Diamond\Box\exists x.G(x)} \qquad \dfrac{\textbf{S5}}{\forall\xi.[\Diamond\Box\xi \rightarrow \Box\xi]}}{\textbf{L2: } \Diamond\exists z.G(z) \rightarrow \Box\exists x.G(x)}$$

$$\dfrac{\textbf{C1: } \Diamond\exists z.G(z) \qquad \textbf{L2: } \Diamond\exists z.G(z) \rightarrow \Box\exists x.G(x)}{\textbf{T3: } \Box\exists x.G(x)}$$

Proof Overview

$$\textbf{D1:}\ G(x) \equiv \forall \varphi.[P(\varphi) \rightarrow \varphi(x)]$$

$$\textbf{D3*:}\ NE(x) \equiv \Box \exists y.G(y) \qquad\qquad \textbf{D3:}\ NE(x) \equiv \forall \varphi.[\varphi\ ess\ x \rightarrow \Box \exists y.\varphi(y)]$$

$$\dfrac{\dfrac{\dfrac{\textbf{T2:}\ \forall y.[G(y) \rightarrow G\ ess\ y] \qquad \dfrac{\textbf{A5}}{P(NE)}}{\textbf{L1:}\ \exists z.G(z) \rightarrow \Box \exists x.G(x)}}{\Diamond \exists z.G(z) \rightarrow \Diamond \Box \exists x.G(x)} \qquad \dfrac{\textbf{S5}}{\forall \xi.[\Diamond \Box \xi \rightarrow \Box \xi]}}{\textbf{L2:}\ \Diamond \exists z.G(z) \rightarrow \Box \exists x.G(x)}$$

$$\dfrac{\textbf{C1:}\ \Diamond \exists z.G(z) \qquad \textbf{L2:}\ \Diamond \exists z.G(z) \rightarrow \Box \exists x.G(x)}{\textbf{T3:}\ \Box \exists x.G(x)}$$

D1: $G(x) \equiv \forall \varphi.[P(\varphi) \rightarrow \varphi(x)]$

D3*: $NE(x) \equiv \Box \exists y.G(y)$

D3: $NE(x) \equiv \forall \varphi.[\varphi \; ess \; x \rightarrow \Box \exists y.\varphi(y)]$

$$\dfrac{\dfrac{\dfrac{\textbf{T2: } \forall y.[G(y) \rightarrow G \; ess \; y] \qquad \dfrac{\textbf{A5}}{P(NE)}}{\textbf{L1: } \exists z.G(z) \rightarrow \Box \exists x.G(x)}}{\Diamond \exists z.G(z) \rightarrow \Diamond \Box \exists x.G(x)} \qquad \dfrac{\textbf{S5}}{\forall \xi.[\Diamond \Box \xi \rightarrow \Box \xi]}}{\textbf{L2: } \Diamond \exists z.G(z) \rightarrow \Box \exists x.G(x)}$$

$$\dfrac{\textbf{C1: } \Diamond \exists z.G(z) \qquad \textbf{L2: } \Diamond \exists z.G(z) \rightarrow \Box \exists x.G(x)}{\textbf{T3: } \Box \exists x.G(x)}$$

Proof Overview

$$\textbf{D1:}\ G(x) \equiv \forall\varphi.[P(\varphi) \to \varphi(x)]$$

$$\textbf{D2:}\ \varphi\ ess\ x \equiv \varphi(x) \wedge \forall\psi.(\psi(x) \to \Box\forall x.(\varphi(x) \to \psi(x)))$$

$$\textbf{D3*:}\ NE(x) \equiv \Box\exists y.G(y) \qquad \textbf{D3:}\ NE(x) \equiv \forall\varphi.[\varphi\ ess\ x \to \Box\exists y.\varphi(y)]$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\mathbf{A1b}}{\forall\varphi.[\neg P(\varphi) \to P(\neg\varphi)]} \qquad \dfrac{\mathbf{A4}}{\forall\varphi.[P(\varphi) \to \Box P(\varphi)]}}{\textbf{T2:}\ \forall y.[G(y) \to G\ ess\ y]} \qquad \dfrac{\mathbf{A5}}{P(NE)}}{\textbf{L1:}\ \exists z.G(z) \to \Box\exists x.G(x)}}{\Diamond\exists z.G(z) \to \Diamond\Box\exists x.G(x)} \qquad \dfrac{\mathbf{S5}}{\forall\xi.[\Diamond\Box\xi \to \Box\xi]}}{\textbf{L2:}\ \Diamond\exists z.G(z) \to \Box\exists x.G(x)}$$

$$\dfrac{\textbf{C1:}\ \Diamond\exists z.G(z) \qquad \textbf{L2:}\ \Diamond\exists z.G(z) \to \Box\exists x.G(x)}{\textbf{T3:}\ \Box\exists x.G(x)}$$

D1: $G(x) \equiv \forall\varphi.[P(\varphi) \rightarrow \varphi(x)]$

D2: $\varphi\ ess\ x \equiv \varphi(x) \wedge \forall\psi.(\psi(x) \rightarrow \Box\forall x.(\varphi(x) \rightarrow \psi(x)))$

D3*: $NE(x) \equiv \Box\exists y.G(y)$

D3: $NE(x) \equiv \forall\varphi.[\varphi\ ess\ x \rightarrow \Box\exists y.\varphi(y)]$

C1: $\Diamond\exists z.G(z)$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\overline{\forall\varphi.[\neg P(\varphi) \rightarrow P(\neg\varphi)]}^{\textbf{A1b}} \quad \overline{\forall\varphi.[P(\varphi) \rightarrow \Box P(\varphi)]}^{\textbf{A4}}}{\textbf{T2: } \forall y.[G(y) \rightarrow G\ ess\ y]} \quad \overline{P(NE)}^{\textbf{A5}}}{\textbf{L1: } \exists z.G(z) \rightarrow \Box\exists x.G(x)}}{\Diamond\exists z.G(z) \rightarrow \Diamond\Box\exists x.G(x)} \quad \overline{\forall\xi.[\Diamond\Box\xi \rightarrow \Box\xi]}^{\textbf{S5}}}{\textbf{L2: } \Diamond\exists z.G(z) \rightarrow \Box\exists x.G(x)}$$

$$\dfrac{\textbf{C1: } \Diamond\exists z.G(z) \qquad \textbf{L2: } \Diamond\exists z.G(z) \rightarrow \Box\exists x.G(x)}{\textbf{T3: } \Box\exists x.G(x)}$$

D1: $G(x) \equiv \forall\varphi.[P(\varphi) \rightarrow \varphi(x)]$

D2: $\varphi\ ess\ x \equiv \varphi(x) \wedge \forall\psi.(\psi(x) \rightarrow \Box\forall x.(\varphi(x) \rightarrow \psi(x)))$

D3*: $NE(x) \equiv \Box\exists y.G(y)$ **D3:** $NE(x) \equiv \forall\varphi.[\varphi\ ess\ x \rightarrow \Box\exists y.\varphi(y)]$

$$\frac{P(G)}{\textbf{C1: } \Diamond\exists z.G(z)}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\forall\varphi.[\neg P(\varphi) \rightarrow P(\neg\varphi)]}\,\textbf{A1b} \quad \dfrac{}{\forall\varphi.[P(\varphi) \rightarrow \Box P(\varphi)]}\,\textbf{A4}}{\textbf{T2: } \forall y.[G(y) \rightarrow G\ ess\ y]} \quad \dfrac{}{P(NE)}\,\textbf{A5}}{\dfrac{\textbf{L1: } \exists z.G(z) \rightarrow \Box\exists x.G(x)}{\Diamond\exists z.G(z) \rightarrow \Diamond\Box\exists x.G(x)}} \quad \dfrac{}{\forall\xi.[\Diamond\Box\xi \rightarrow \Box\xi]}\,\textbf{S5}}{\textbf{L2: } \Diamond\exists z.G(z) \rightarrow \Box\exists x.G(x)}$$

$$\frac{\textbf{C1: } \Diamond\exists z.G(z) \qquad \textbf{L2: } \Diamond\exists z.G(z) \rightarrow \Box\exists x.G(x)}{\textbf{T3: } \Box\exists x.G(x)}$$

$$\textbf{D1:}\ G(x) \equiv \forall\varphi.[P(\varphi) \to \varphi(x)]$$

$$\textbf{D2:}\ \varphi\ ess\ x \equiv \varphi(x) \wedge \forall\psi.(\psi(x) \to \Box\forall x.(\varphi(x) \to \psi(x)))$$

$$\textbf{D3*:}\ NE(x) \equiv \Box\exists y.G(y) \qquad \textbf{D3:}\ NE(x) \equiv \forall\varphi.[\varphi\ ess\ x \to \Box\exists y.\varphi(y)]$$

$$\dfrac{\overset{\textbf{A3}}{P(G)}}{\textbf{C1:}\ \Diamond\exists z.G(z)}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\overset{\textbf{A1b}}{\forall\varphi.[\neg P(\varphi) \to P(\neg\varphi)]} \qquad \overset{\textbf{A4}}{\forall\varphi.[P(\varphi) \to \Box\, P(\varphi)]}}{\textbf{T2:}\ \forall y.[G(y) \to G\ ess\ y]} \qquad \overset{\textbf{A5}}{P(NE)}}{\textbf{L1:}\ \exists z.G(z) \to \Box\exists x.G(x)}}{\Diamond\exists z.G(z) \to \Diamond\Box\exists x.G(x)} \qquad \overset{\textbf{S5}}{\forall\xi.[\Diamond\Box\xi \to \Box\xi]}}{\textbf{L2:}\ \Diamond\exists z.G(z) \to \Box\exists x.G(x)}$$

$$\dfrac{\textbf{C1:}\ \Diamond\exists z.G(z) \qquad \textbf{L2:}\ \Diamond\exists z.G(z) \to \Box\exists x.G(x)}{\textbf{T3:}\ \Box\exists x.G(x)}$$

$$\textbf{D1: } G(x) \equiv \forall\varphi.[P(\varphi) \rightarrow \varphi(x)]$$

$$\textbf{D2: } \varphi\ ess\ x \equiv \varphi(x) \wedge \forall\psi.(\psi(x) \rightarrow \Box\forall x.(\varphi(x) \rightarrow \psi(x)))$$

$$\textbf{D3*: } NE(x) \equiv \Box\exists y.G(y) \qquad\qquad \textbf{D3: } NE(x) \equiv \forall\varphi.[\varphi\ ess\ x \rightarrow \Box\exists y.\varphi(y)]$$

$$\dfrac{\dfrac{\textbf{A3}}{P(G)} \qquad \textbf{T1: } \forall\varphi.[P(\varphi) \rightarrow \Diamond\exists x.\varphi(x)]}{\textbf{C1: } \Diamond\exists z.G(z)}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\textbf{A1b}}{\forall\varphi.[\neg P(\varphi) \rightarrow P(\neg\varphi)]} \qquad \dfrac{\textbf{A4}}{\forall\varphi.[P(\varphi) \rightarrow \Box P(\varphi)]}}{\textbf{T2: } \forall y.[G(y) \rightarrow G\ ess\ y]} \qquad \dfrac{\textbf{A5}}{P(NE)}}{\textbf{L1: } \exists z.G(z) \rightarrow \Box\exists x.G(x)}}{\Diamond\exists z.G(z) \rightarrow \Diamond\Box\exists x.G(x)} \qquad \dfrac{\textbf{S5}}{\forall\xi.[\Diamond\Box\xi \rightarrow \Box\xi]}}{\textbf{L2: } \Diamond\exists z.G(z) \rightarrow \Box\exists x.G(x)}$$

$$\dfrac{\textbf{C1: } \Diamond\exists z.G(z) \qquad \textbf{L2: } \Diamond\exists z.G(z) \rightarrow \Box\exists x.G(x)}{\textbf{T3: } \Box\exists x.G(x)}$$

Proof Overview

$$\textbf{D1:}\ G(x) \equiv \forall\varphi.[P(\varphi) \to \varphi(x)]$$

$$\textbf{D2:}\ \varphi\ ess\ x \equiv \varphi(x) \wedge \forall\psi.(\psi(x) \to \Box\forall x.(\varphi(x) \to \psi(x)))$$

$$\textbf{D3*:}\ NE(x) \equiv \Box\exists y.G(y) \qquad\qquad \textbf{D3:}\ NE(x) \equiv \forall\varphi.[\varphi\ ess\ x \to \Box\exists y.\varphi(y)]$$

$$\dfrac{\dfrac{\textbf{A3}}{P(G)} \qquad \dfrac{\dfrac{\textbf{A2}}{\forall\varphi.\forall\psi.[(P(\varphi) \wedge \Box\forall x.[\varphi(x) \to \psi(x)]) \to P(\psi)]} \qquad \dfrac{\textbf{A1a}}{\forall\varphi.[P(\neg\varphi) \to \neg P(\varphi)]}}{\textbf{T1:}\ \forall\varphi.[P(\varphi) \to \Diamond\exists x.\varphi(x)]}}{\textbf{C1:}\ \Diamond\exists z.G(z)}$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\textbf{A1b}}{\forall\varphi.[\neg P(\varphi) \to P(\neg\varphi)]} \qquad \dfrac{\textbf{A4}}{\forall\varphi.[P(\varphi) \to \Box P(\varphi)]}}{\textbf{T2:}\ \forall y.[G(y) \to G\ ess\ y]} \qquad \dfrac{\textbf{A5}}{P(NE)}}{\textbf{L1:}\ \exists z.G(z) \to \Box\exists x.G(x)}}{\Diamond\exists z.G(z) \to \Diamond\Box\exists x.G(x)} \qquad \dfrac{\textbf{S5}}{\forall\xi.[\Diamond\Box\xi \to \Box\xi]}}{\textbf{L2:}\ \Diamond\exists z.G(z) \to \Box\exists x.G(x)}$$

$$\dfrac{\textbf{C1:}\ \Diamond\exists z.G(z) \qquad \textbf{L2:}\ \Diamond\exists z.G(z) \to \Box\exists x.G(x)}{\textbf{T3:}\ \Box\exists x.G(x)}$$

Proof Overview

$$\textbf{D1: } G(x) \equiv \forall\varphi.[P(\varphi) \rightarrow \varphi(x)]$$

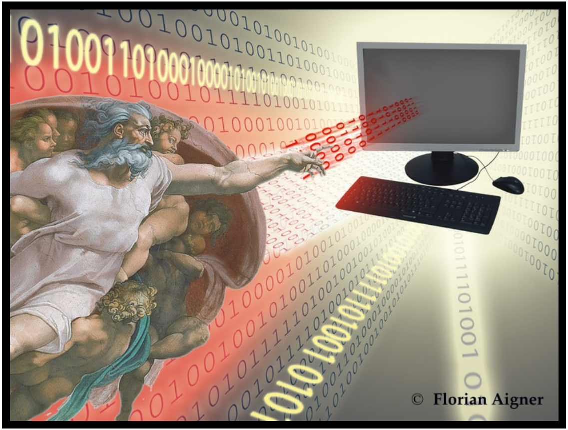
$$\textbf{D2: } \varphi \; ess \; x \equiv \varphi(x) \wedge \forall\psi.(\psi(x) \rightarrow \Box\forall x.(\varphi(x) \rightarrow \psi(x)))$$

$$\textbf{D3: } NE(x) \equiv \forall\varphi.[\varphi \; ess \; x \rightarrow \Box\exists y.\varphi(y)]$$

$$\cfrac{\cfrac{\textbf{A3}}{P(G)} \qquad \cfrac{\cfrac{\textbf{A2}}{\forall\varphi.\forall\psi.[(P(\varphi) \wedge \Box\forall x.[\varphi(x) \rightarrow \psi(x)]) \rightarrow P(\psi)]} \qquad \cfrac{\textbf{A1a}}{\forall\varphi.[P(\neg\varphi) \rightarrow \neg P(\varphi)]}}{\textbf{T1: } \forall\varphi.[P(\varphi) \rightarrow \Diamond\exists x.\varphi(x)]}}{\textbf{C1: } \Diamond\exists z.G(z)}$$

$$\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{\textbf{A1b}}{\forall\varphi.[\neg P(\varphi) \rightarrow P(\neg\varphi)]} \qquad \cfrac{\textbf{A4}}{\forall\varphi.[P(\varphi) \rightarrow \Box P(\varphi)]}}{\textbf{T2: } \forall y.[G(y) \rightarrow G \; ess \; y]} \qquad \cfrac{\textbf{A5}}{P(NE)}}{\textbf{L1: } \exists z.G(z) \rightarrow \Box\exists x.G(x)}}{\Diamond\exists z.G(z) \rightarrow \Diamond\Box\exists x.G(x)} \qquad \cfrac{\textbf{S5}}{\forall\xi.[\Diamond\Box\xi \rightarrow \Box\xi]}}{\textbf{L2: } \Diamond\exists z.G(z) \rightarrow \Box\exists x.G(x)}$$

$$\cfrac{\textbf{C1: } \Diamond\exists z.G(z) \qquad \textbf{L2: } \Diamond\exists z.G(z) \rightarrow \Box\exists x.G(x)}{\textbf{T3: } \Box\exists x.G(x)}$$



Part B:

Formalization:	in classical higher-order logic (HOL)
Automation:	theorem provers LEO-II and SATALLAX
Consistency:	model finder NITPICK (NITROX)

Challenge: No provers for *Higher-order Quantified Modal Logic* (QML)

Our solution: Embedding in *Higher-order Classical Logic* (HOL)
Then use existing HOL theorem provers for reasoning in QML
[BenzmüllerPaulson, Logica Universalis, 2013]

Previous empiricial findings:

Embedding of *First-order Modal Logic* in HOL works well
[BenzmüllerOttenRaths, ECAI, 2012]
[BenzmüllerRaths, LPAR, 2013]

QML $\quad \varphi, \psi \quad ::= \quad \ldots \mid \neg\varphi \mid \varphi \wedge \psi \mid \varphi \rightarrow \psi \mid \Box\varphi \mid \Diamond\varphi \mid \forall x\, \varphi \mid \exists x\, \varphi \mid \forall P\, \varphi$

- Kripke style semantics (possible world semantics)

HOL $\quad s, t \quad ::= \quad C \mid x \mid \lambda x s \mid s\, t \mid \neg s \mid s \vee t \mid \forall x\, t$

- meanwhile very well understood
- **Henkin semantics** vs. standard semantics
- various theorem provers do exists

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

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

QML $\varphi, \psi \quad ::= \quad \ldots \mid \neg\varphi \mid \varphi \wedge \psi \mid \varphi \rightarrow \psi \mid \Box\varphi \mid \Diamond\varphi \mid \forall x\, \varphi \mid \exists x\, \varphi \mid \forall P\, \varphi$

HOL $s, t \quad ::= \quad C \mid x \mid \lambda x s \mid s\, t \mid \neg s \mid s \vee t \mid \forall x\, t$

QML in HOL: QML formulas φ are mapped to HOL predicates $\varphi_{\iota\rightarrow o}$

$$
\begin{array}{lcl}
\neg & = & \lambda\varphi_{\iota\rightarrow o}\lambda s_\iota \neg\varphi s \\
\wedge & = & \lambda\varphi_{\iota\rightarrow o}\lambda\psi_{\iota\rightarrow o}\lambda s_\iota(\varphi s \wedge \psi s) \\
\rightarrow & = & \lambda\varphi_{\iota\rightarrow o}\lambda\psi_{\iota\rightarrow o}\lambda s_\iota(\neg\varphi s \vee \psi s) \\
\Box & = & \lambda\varphi_{\iota\rightarrow o}\lambda s_\iota \forall u_\iota\, (\neg rsu \vee \varphi u) \\
\Diamond & = & \lambda\varphi_{\iota\rightarrow o}\lambda s_\iota \exists u_\iota\, (rsu \wedge \varphi u) \\
\forall & = & \lambda h_{\mu\rightarrow(\iota\rightarrow o)}\lambda s_\iota \forall d_\mu\, hds \\
\exists & = & \lambda h_{\mu\rightarrow(\iota\rightarrow o)}\lambda s_\iota \exists d_\mu\, hds \\
\forall & = & \lambda H_{(\mu\rightarrow(\iota\rightarrow o))\rightarrow(\iota\rightarrow o)}\lambda s_\iota \forall d_\mu\, Hds \\
 & & \\
\text{valid} & = & \lambda\varphi_{\iota\rightarrow o}\forall w_\iota \varphi w
\end{array}
$$

Ax

The equations in Ax are given as axioms to the HOL provers!

(Remark: Note that we are here dealing with constant domain quantification)

Formalization in HOL

QML $\quad \varphi, \psi \quad ::= \quad \ldots \mid \neg\varphi \mid \varphi \wedge \psi \mid \varphi \rightarrow \psi \mid \Box\varphi \mid \Diamond\varphi \mid \forall x\, \varphi \mid \exists x\, \varphi \mid \forall P\, \varphi$

HOL $\quad s, t \quad ::= \quad C \mid x \mid \lambda x s \mid s\, t \mid \neg s \mid s \vee t \mid \forall x\, t$

QML in HOL: QML formulas φ are mapped to HOL predicates $\varphi_{\iota\rightarrow o}$

$$
\begin{array}{lcl}
\neg & = & \lambda\varphi_{\iota\rightarrow o}\lambda s_\iota \neg\varphi s \\
\wedge & = & \lambda\varphi_{\iota\rightarrow o}\lambda\psi_{\iota\rightarrow o}\lambda s_\iota(\varphi s \wedge \psi s) \\
\rightarrow & = & \lambda\varphi_{\iota\rightarrow o}\lambda\psi_{\iota\rightarrow o}\lambda s_\iota(\neg\varphi s \vee \psi s) \\
\Box & = & \lambda\varphi_{\iota\rightarrow o}\lambda s_\iota \forall u_\iota\, (\neg r s u \vee \varphi u) \\
\Diamond & = & \lambda\varphi_{\iota\rightarrow o}\lambda s_\iota \exists u_\iota\, (r s u \wedge \varphi u) \\
\forall & = & \lambda h_{\mu\rightarrow(\iota\rightarrow o)}\lambda s_\iota \forall d_\mu\, h d s \\
\exists & = & \lambda h_{\mu\rightarrow(\iota\rightarrow o)}\lambda s_\iota \exists d_\mu\, h d s \\
\forall & = & \lambda H_{(\mu\rightarrow(\iota\rightarrow o))\rightarrow(\iota\rightarrow o)}\lambda s_\iota \forall d_\mu\, H d s \\
 & & \\
\text{valid} & = & \lambda\varphi_{\iota\rightarrow o}\forall w_\iota \varphi w
\end{array}
$$

Ax

The equations in Ax are given as axioms to the HOL provers!

(Remark: Note that we are here dealing with constant domain quantification)

Formalization in HOL

QML $\varphi, \psi \quad ::= \quad \ldots \mid \neg\varphi \mid \varphi \wedge \psi \mid \varphi \rightarrow \psi \mid \Box\varphi \mid \Diamond\varphi \mid \forall x\, \varphi \mid \exists x\, \varphi \mid \forall P\, \varphi$

HOL $s, t \quad ::= \quad C \mid x \mid \lambda x s \mid s\, t \mid \neg s \mid s \vee t \mid \forall x\, t$

QML in HOL: QML formulas φ are mapped to HOL predicates $\varphi_{\iota\rightarrow o}$

$$
\begin{array}{lcl}
\neg & = & \lambda\varphi_{\iota\rightarrow o}\lambda s_\iota \neg\varphi s \\
\wedge & = & \lambda\varphi_{\iota\rightarrow o}\lambda\psi_{\iota\rightarrow o}\lambda s_\iota(\varphi s \wedge \psi s) \\
\rightarrow & = & \lambda\varphi_{\iota\rightarrow o}\lambda\psi_{\iota\rightarrow o}\lambda s_\iota(\neg\varphi s \vee \psi s) \\
\Box & = & \lambda\varphi_{\iota\rightarrow o}\lambda s_\iota \forall u_\iota\, (\neg r s u \vee \varphi u) \\
\Diamond & = & \lambda\varphi_{\iota\rightarrow o}\lambda s_\iota \exists u_\iota\, (r s u \wedge \varphi u) \\
\forall & = & \lambda h_{\mu\rightarrow(\iota\rightarrow o)}\lambda s_\iota \forall d_\mu\, h d s \\
\exists & = & \lambda h_{\mu\rightarrow(\iota\rightarrow o)}\lambda s_\iota \exists d_\mu\, h d s \\
\forall & = & \lambda H_{(\mu\rightarrow(\iota\rightarrow o))\rightarrow(\iota\rightarrow o)}\lambda s_\iota \forall d_\mu\, H d s \\
 & & \\
\text{valid} & = & \lambda\varphi_{\iota\rightarrow o}\forall w_\iota \varphi w
\end{array}
$$

Ax

The equations in Ax are given as axioms to the HOL provers!

(Remark: Note that we are here dealing with constant domain quantification)

Example

QML formula	$\Diamond \exists x G(x)$
QML formula in HOL	valid $(\Diamond \exists x G(x))_{\iota \to o}$
expansion, $\beta\eta$-conversion	$\forall w_\iota (\Diamond \exists x G(x))_{\iota \to o}\, w$
expansion, $\beta\eta$-conversion	$\forall w_\iota \exists u_\iota (rwu \wedge (\exists x G(x))_{\iota \to o} u)$
expansion, $\beta\eta$-conversion	$\forall w_\iota \exists u_\iota (rwu \wedge \exists x Gxu)$

What are we doing?

In order to prove that φ is valid in QML,
–> we instead prove that valid $\varphi_{\iota \to o}$ can be derived from Ax in HOL.

This can be done with interactive or automated HOL theorem provers.

Expansion: user or prover may flexibly choose expansion depth

Example

QML formula	$\Diamond \exists x G(x)$
QML formula in HOL	valid $(\Diamond \exists x G(x))_{\iota \to o}$
expansion, $\beta\eta$-conversion	$\forall w_\iota (\Diamond \exists x G(x))_{\iota \to o}\, w$
expansion, $\beta\eta$-conversion	$\forall w_\iota \exists u_\iota (rwu \wedge (\exists x G(x))_{\iota \to o} u)$
expansion, $\beta\eta$-conversion	$\forall w_\iota \exists u_\iota (rwu \wedge \exists x Gxu)$

What are we doing?

In order to prove that φ is valid in QML,
–> we instead prove that valid $\varphi_{\iota \to o}$ can be derived from Ax in HOL.

This can be done with interactive or automated HOL theorem provers.

Expansion: user or prover may flexibly choose expansion depth

Example

QML formula	$\Diamond \exists x G(x)$
QML formula in HOL	valid $(\Diamond \exists x G(x))_{\iota \to o}$
expansion, $\beta\eta$-conversion	$\forall w_\iota (\Diamond \exists x G(x))_{\iota \to o}\, w$
expansion, $\beta\eta$-conversion	$\forall w_\iota \exists u_\iota (rwu \wedge (\exists x G(x))_{\iota \to o} u)$
expansion, $\beta\eta$-conversion	$\forall w_\iota \exists u_\iota (rwu \wedge \exists x Gxu)$

What are we doing?

In order to prove that φ is valid in QML,
–> we instead prove that valid $\varphi_{\iota \to o}$ can be derived from Ax in HOL.

This can be done with interactive or automated HOL theorem provers.

Expansion: user or prover may flexibly choose expansion depth

Example

QML formula	$\Diamond \exists x G(x)$
QML formula in HOL	valid $(\Diamond \exists x G(x))_{\iota \to o}$
expansion, $\beta\eta$-conversion	$\forall w_\iota (\Diamond \exists x G(x))_{\iota \to o}\, w$
expansion, $\beta\eta$-conversion	$\forall w_\iota \exists u_\iota (rwu \wedge (\exists x G(x))_{\iota \to o} u)$
expansion, $\beta\eta$-conversion	$\forall w_\iota \exists u_\iota (rwu \wedge \exists x Gxu)$

What are we doing?

In order to prove that φ is valid in QML,
–> we instead prove that valid $\varphi_{\iota \to o}$ can be derived from Ax in HOL.

This can be done with interactive or automated HOL theorem provers.

Expansion: user or prover may flexibly choose expansion depth

Example

QML formula	$\Diamond \exists x G(x)$
QML formula in HOL	valid $(\Diamond \exists x G(x))_{\iota\rightarrow o}$
expansion, $\beta\eta$-conversion	$\forall w_\iota (\Diamond \exists x G(x))_{\iota\rightarrow o}\, w$
expansion, $\beta\eta$-conversion	$\forall w_\iota \exists u_\iota (rwu \wedge (\exists x G(x))_{\iota\rightarrow o} u)$
expansion, $\beta\eta$-conversion	$\forall w_\iota \exists u_\iota (rwu \wedge \exists x Gxu)$

What are we doing?

In order to prove that φ is valid in QML,
–> we instead prove that valid $\varphi_{\iota\rightarrow o}$ can be derived from Ax in HOL.

This can be done with interactive or automated HOL theorem provers.

Expansion: user or prover may flexibly choose expansion depth

Example

QML formula	$\Diamond \exists x G(x)$
QML formula in HOL	valid $(\Diamond \exists x G(x))_{\iota \to o}$
expansion, $\beta\eta$-conversion	$\forall w_\iota (\Diamond \exists x G(x))_{\iota \to o}\, w$
expansion, $\beta\eta$-conversion	$\forall w_\iota \exists u_\iota (rwu \wedge (\exists x G(x))_{\iota \to o} u)$
expansion, $\beta\eta$-conversion	$\forall w_\iota \exists u_\iota (rwu \wedge \exists x G x u)$

What are we doing?

In order to prove that φ is valid in QML,
–> we instead prove that valid $\varphi_{\iota \to o}$ can be derived from Ax in HOL.

This can be done with interactive or automated HOL theorem provers.

Expansion: user or prover may flexibly choose expansion depth

Example

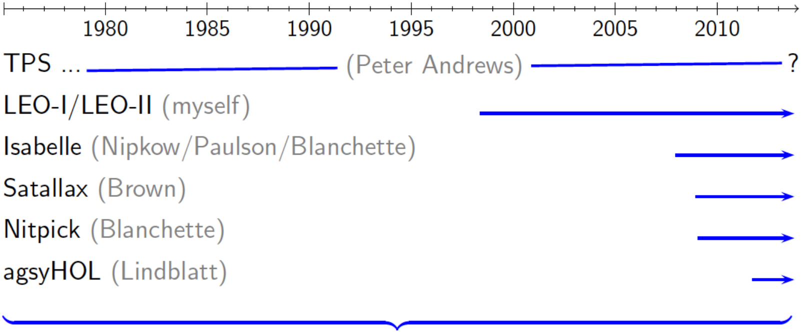
QML formula	$\Diamond \exists x G(x)$
QML formula in HOL	valid $(\Diamond \exists x G(x))_{\iota \to o}$
expansion, $\beta\eta$-conversion	$\forall w_\iota (\Diamond \exists x G(x))_{\iota \to o}\, w$
expansion, $\beta\eta$-conversion	$\forall w_\iota \exists u_\iota (rwu \wedge (\exists x G(x))_{\iota \to o} u)$
expansion, $\beta\eta$-conversion	$\forall w_\iota \exists u_\iota (rwu \wedge \exists x Gxu)$

What are we doing?

In order to prove that φ is valid in QML,
–> we instead prove that valid $\varphi_{\iota \to o}$ can be derived from Ax in HOL.

This can be done with interactive or automated HOL theorem provers.

Expansion: user or prover may flexibly choose expansion depth



- all accept TPTP THF Syntax [SutcliffeBenzmüller, J.Form.Reas, 2009]
  - can be called remotely via SystemOnTPTP at Miami
  - they significantly gained in strength over the last years
  - they can be bundled into a combined prover HOL-P

Exploit HOL with Henkin semantics as metalogic
Automate other logics (& combinations) via semantic embeddings
— HOL-P becomes a Universal Reasoner —

```
Terminal — bash — 125×32
MacBook-Chris %
MacBook-Chris %
MacBook-Chris % ./call_tptp.sh T3.p

Asking various HOL-ATPs in Miami remotely (thanks to Geoff Sutcliffe)

MacBook-Chris % agsyHOL---1.0 : T3.p ++++++ RESULT: SOT_7L4x_Y - agsyHOL---1.0 says Unknown - CPU = 0.00 WC = 0.02
LEO-II---1.6.0 : T3.p ++++++ RESULT: SOT_E4SCha - LEO-II---1.6.0 says Theorem - CPU = 0.03 WC = 0.09
Satallax---2.7 : T3.p ++++++ RESULT: SOT_kVZ1cB - Satallax---2.7 says Theorem - CPU = 0.00 WC = 0.14
Isabelle---2013 : T3.p ++++++ RESULT: SOT_xa0gEp - Isabelle---2013 says Theorem - CPU = 14.06 WC = 17.73 SolvedBy = auto
TPS---3.120601S1b : T3.p ++++++ RESULT: SOT_R0Egsg - TPS---3.120601S1b says Unknown - CPU = 33.56 WC = 41.57
Nitrox---2013 : T3.p ++++++ RESULT: SOT_WGY1Tx - Nitrox---2013 says Unknown - CPU = 75.55 WC = 49.24

MacBook-Chris %
MacBook-Chris % ./call_tptp.sh Consistency.p

Asking various HOL-ATPs in Miami remotely (thanks to Geoff Sutcliffe)

MacBook-Chris % agsyHOL---1.0 : Consistency.p ++++++ RESULT: SOT_ZtY_7o - agsyHOL---1.0 says Unknown - CPU = 0.00 WC = 0.00
Nitrox---2013 : Consistency.p ++++++ RESULT: SOT_HUz10C - Nitrox---2013 says Satisfiable - CPU = 6.56 WC = 8.50
TPS---3.120601S1b : Consistency.p ++++++ RESULT: SOT_fpJxTM - TPS---3.120601S1b says Unknown - CPU = 43.00 WC = 49.42
Isabelle---2013 : Consistency.p ++++++ RESULT: SOT_6Tpp9i - Isabelle---2013 says Unknown - CPU = 69.96 WC = 72.62
LEO-II---1.6.0 : Consistency.p ++++++ RESULT: SOT_dY10sj - LEO-II---1.6.0 says Timeout - CPU = 90 WC = 89.86
Satallax---2.7 : Consistency.p ++++++ RESULT: SOT_Q9WSLf - Satallax---2.7 says Timeout - CPU = 90 WC = 90.50

MacBook-Chris %
```

– see THF files at: `https://github.com/FormalTheology/GoedelGod/blob/master/Formalizations/THF/` –

Provers are called remotely in Miami — no local installation needed!

The TPI Scripting Language

Cf. previous talk of Geoff Sutcliffe!

- Command and control instruction language for TPTP infrastructure
- Specify problems, avoiding repetitions
- Process the logical formulae: prove theorems, check consistency, etc.
- Report results; exploit TPTP SZS ontology
- Fully automatic
- Very useful for reproducing experiments

Productive collaboration with Geoff: new features, debugging and testing

```
Defining the embedding of quantified modal logics in HOL.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
tpi(com,write,'%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%').
tpi(com,write,'%%% 1. Introducing the embedding, signature, definitions, axioms, and theorems.').

tpi(com,start_group,embedding).
  thf(mu_type,type,(mu:$tType)).
  thf(mnot_type,type,(mnot:($i>$o)>$i>$o)).
  thf(mnot,definition,(mnot=(^[Phi:$i>$o,W:$i]:~(Phi@W)))).
  thf(mor_type,type,(mor:($i>$o)>($i>$o)>$i>$o)).
  thf(mor,definition,(mor=(^[Phi:$i>$o,Psi:$i>$o,W:$i]:((Phi@W)|(Psi@W))))).
  thf(mand_type,type,(mand:($i>$o)>($i>$o)>$i>$o)).
  thf(mand,definition,(mand=(^[Phi:$i>$o,Psi:$i>$o,W:$i]:((Phi@W)&(Psi@W))))).
  thf(mimplies_type,type,(mimplies:($i>$o)>($i>$o)>$i>$o)).
  thf(mimplies,definition,(mimplies=(^[Phi:$i>$o,Psi:$i>$o,W:$i]:((Phi@W)=>(Psi@W))))).
  thf(mequiv_type,type,(mequiv:($i>$o)>($i>$o)>$i>$o)).
  thf(mequiv,definition,(mequiv=(^[Phi:$i>$o,Psi:$i>$o,W:$i]:((Phi@W)<=>(Psi@W))))).
  thf(mforall_ind_type,type,(mforall_ind:(mu>$i>$o)>$i>$o)).
  thf(mforall_ind,definition,(mforall_ind=(^[Phi:mu>$i>$o,W:$i]:![X:mu]:(Phi@X@W)))).
  thf(mforall_indset_type,type,(mforall_indset:((mu>$i>$o)>$i>$o)>$i>$o)).
  thf(mforall_indset,definition,(mforall_indset=(^[Phi:(mu>$i>$o)>$i>$o,W:$i]:![X:mu>$i>$o]:(Phi@X@W)))).
  thf(mexists_ind_type,type,(mexists_ind:(mu>$i>$o)>$i>$o)).
  thf(mexists_ind,definition,(mexists_ind=(^[Phi:mu>$i>$o,W:$i]:?[X:mu]:(Phi@X@W)))).
  thf(mequals_type,type,(mequals:mu>mu>$i>$o)).
  thf(mequals,definition,(mequals=(^[X:mu,Y:mu,W:$i]:(X=Y)))).
  thf(rel_type,type,(rel:$i>$i>$o)).
  thf(mbox_type,type,(mbox:($i>$o)>$i>$o)).
  thf(mbox,definition,(mbox=(^[Phi:$i>$o,W:$i]:![V:$i]:((rel@W@V)=>(Phi@V))))).
  thf(mdia_type,type,(mdia:($i>$o)>$i>$o)).
  thf(mdia,definition,(mdia=(^[Phi:$i>$o,W:$i]:?[V:$i]:((rel@W@V)&(Phi@V))))).
  thf(mvalid_type,type,(v:($i>$o)>$o)).
  thf(mvalid,definition,(v=(^[Phi:$i>$o]:![W:$i]:(Phi@W)))).
tpi(com,end_group,embedding).
```

```
Defining the embedding of quantified modal logics in HOL.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
tpi(com,write,'%%%%%%%%%%%%%%%%%%%%%%%
tpi(com,write,'%%% 1. Introducing the e

tpi(com,start_group,embedding).
  thf(mu_type,type,(mu:$tType)).
  thf(mnot_type,type,(mnot:($i>$o)>$i>$o)).
  thf(mnot,definition,(mnot=(^[Phi:$i>$o,W:$i]:~(Phi@W)))).
  thf(mor_type,type,(mor:($i>$o)>($i>$o)>$i>$o)).
  thf(mor,definition,(mor=(^[Phi:$i>$o,Psi:$i>$o,W:$i]:((Phi@W)|(Psi@W))))).
  thf(mand_type,type,(mand:($i>$o)>($i>$o)>$i>$o)).
  thf(mand,definition,(mand=(^[Phi:$i>$o,Psi:$i>$o,W:$i]:((Phi@W)&(Psi@W))))).
  thf(mimplies_type,type,(mimplies:($i>$o)>($i>$o)>$i>$o)).
  thf(mimplies,definition,(mimplies=(^[Phi:$i>$o,Psi:$i>$o,W:$i]:((Phi@W)=>(Psi@W))))).
  thf(mequiv_type,type,(mequiv:($i>$o)>($i>$o)>$i>$o)).
  thf(mequiv,definition,(mequiv=(^[Phi:$i>$o,Psi:$i>$o,W:$i]:((Phi@W)<=>(Psi@W))))).
  thf(mforall_ind_type,type,(mforall_ind:(mu>$i>$o)>$i>$o)).
  thf(mforall_ind,definition,(mforall_ind=(^[Phi:mu>$i>$o,W:$i]:![X:mu]:(Phi@X@W)))).
  thf(mforall_indset_type,type,(mforall_indset:((mu>$i>$o)>$i>$o)>$i>$o)).
  thf(mforall_indset,definition,(mforall_indset=(^[Phi:(mu>$i>$o)>$i>$o,W:$i]:![X:mu>$i>$o]:(Phi@X@W)))).
  thf(mexists_ind_type,type,(mexists_ind:(mu>$i>$o)>$i>$o)).
  thf(mexists_ind,definition,(mexists_ind=(^[Phi:mu>$i>$o,W:$i]:?[X:mu]:(Phi@X@W)))).
  thf(mequals_type,type,(mequals:mu>mu>$i>$o)).
  thf(mequals,definition,(mequals=(^[X:mu,Y:mu,W:$i]:(X=Y)))).
  thf(rel_type,type,(rel:$i>$i>$o)).
  thf(mbox_type,type,(mbox:($i>$o)>$i>$o)).
  thf(mbox,definition,(mbox=(^[Phi:$i>$o,W:$i]:![V:$i]:((rel@W@V)=>(Phi@V))))).
  thf(mdia_type,type,(mdia:($i>$o)>$i>$o)).
  thf(mdia,definition,(mdia=(^[Phi:$i>$o,W:$i]:?[V:$i]:((rel@W@V)&(Phi@V))))).
  thf(mvalid_type,type,(v:($i>$o)>$o)).
  thf(mvalid,definition,(v=(^[Phi:$i>$o]:![W:$i]:(Phi@W)))).
tpi(com,end_group,embedding).
```

A (lean) QML prover in HOL

```
tpi(com,start_group,symmetry).
  thf(sym,axiom,(![S:$i,T:$i]:((rel@S@T)=>(rel@T@S)))).
tpi(com,end_group,symmetry).

tpi(com,start_group,sig).
  thf(p_tp,type,(p:(mu>$i>$o)>$i>$o)).
  thf(g_tp,type,(g:mu>$i>$o)).
  thf(ess_tp,type,(ess:(mu>$i>$o)>mu>$i>$o)).
  thf(ne_tp,type,(ne:mu>$i>$o)).
tpi(com,end_group,sig).

tpi(com,start_group,d1).
  thf(defD1,definition,(g=(^[X:mu]:(mforall_indset@^[Phi:mu>$i>$o]:(mimplies@(p@Phi)@(Phi@X)))))).
tpi(com,end_group,d1).

tpi(com,start_group,d2).
  thf(defD2,definition,(ess=(^[Phi:mu>$i>$o,X:mu]:
   (mand@(Phi@X)@(mforall_indset@^[Psi:mu>$i>$o]:
                (mimplies@(Psi@X)@(mbox@(mforall_ind@^[Y:mu]:(mimplies@(Phi@Y)@(Psi@Y)))))))))).
tpi(com,end_group,d2).

tpi(com,start_group,d3).
  thf(defD3,definition,(ne=(^[X:mu]:(mforall_indset@^[Phi:mu>$i>$o]:
   (mimplies@(ess@Phi@X)@(mbox@(mexists_ind@^[Y:mu]:(Phi@Y))))))))).
tpi(com,end_group,d3).

tpi(com,start_group,a1a).
  thf(axA1a,axiom,(v@(mforall_indset@^[Phi:mu>$i>$o]:(mimplies@(p@^[X:mu]:(mnot@(Phi@X)))@(mnot@(p@Phi)))))).
tpi(com,end_group,a1a).

tpi(com,start_group,a1b).
  thf(axA1b,axiom,(v@(mforall_indset@^[Phi:mu>$i>$o]:(mimplies@(mnot@(p@Phi))@(p@^[X:mu]:(mnot@(Phi@X))))))).
tpi(com,end_group,a1b).
```

```
tpi(com,start_group,symmetry).
  thf(sym,axiom,(![S:$i,T:$i]:((rel@S@T)=>(rel@T@S)))).
tpi(com,end_group,symmetry).

tpi(com,start_group,sig).
  thf(p_tp,type,(p:(mu>$i>$o)>$i>$o)).
  thf(g_tp,type,(g:mu>$i>$o)).
  thf(ess_tp,type,(ess:(mu>$i>$o)>mu>$i
  thf(ne_tp,type,(ne:mu>$i>$o)).
tpi(com,end_group,sig).

tpi(com,start_group,d1).
  thf(defD1,definition,(g=(^[X:mu]:(mforall_indset@^[Phi:mu>$i>$o]:(mimplies@(p@Phi)@(Phi@X)))))).
tpi(com,end_group,d1).

tpi(com,start_group,d2).
  thf(defD2,definition,(ess=(^[Phi:mu>$i>$o,X:mu]:
   (mand@(Phi@X)@(mforall_indset@^[Psi:mu>$i>$o]:
                   (mimplies@(Psi@X)@(mbox@(mforall_ind@^[Y:mu]:(mimplies@(Phi@Y)@(Psi@Y)))))))))).
tpi(com,end_group,d2).

tpi(com,start_group,d3).
  thf(defD3,definition,(ne=(^[X:mu]:(mforall_indset@^[Phi:mu>$i>$o]:
   (mimplies@(ess@Phi@X)@(mbox@(mexists_ind@^[Y:mu]:(Phi@Y)))))))).
tpi(com,end_group,d3).

tpi(com,start_group,a1a).
  thf(axA1a,axiom,(v@(mforall_indset@^[Phi:mu>$i>$o]:(mimplies@(p@^[X:mu]:(mnot@(Phi@X)))@(mnot@(p@Phi)))))).
tpi(com,end_group,a1a).

tpi(com,start_group,a1b).
  thf(axA1b,axiom,(v@(mforall_indset@^[Phi:mu>$i>$o]:(mimplies@(mnot@(p@Phi))@(p@^[X:mu]:(mnot@(Phi@X))))))).
tpi(com,end_group,a1b).
```

Axiom B (symmetry)

Gödel's God as THF TPI Script

```
tpi(com,start_group,symmetry).
  thf(sym,axiom,(![S:$i,T:$i]:((rel@S@T)=>(rel@T@S)))).
tpi(com,end_group,symmetry).

tpi(com,start_group,sig).
  thf(p_tp,type,(p:(mu>$i>$o)>$i>$o)).
  thf(g_tp,type,(g:mu>$i>$o)).
  thf(ess_tp,type,(ess:(mu>$i>$o)>mu>$i>$o)).
  thf(ne_tp,type,(ne:mu>$i>$o)).
tpi(com,end_group,sig).

tpi(com,start_group,d1).
  thf(defD1,definition,(g=(^[X:mu]:(mf
tpi(com,end_group,d1).

tpi(com,start_group,d2).
  thf(defD2,definition,(ess=(^[Phi:mu>$i>$o,X:mu]:
   (mand@(Phi@X)@(mforall_indset@^[Psi:mu>$i>$o]:
                   (mimplies@(Psi@X)@(mbox@(mforall_ind@^[Y:mu]:(mimplies@(Phi@Y)@(Psi@Y)))))))))).
tpi(com,end_group,d2).

tpi(com,start_group,d3).
  thf(defD3,definition,(ne=(^[X:mu]:(mforall_indset@^[Phi:mu>$i>$o]:
   (mimplies@(ess@Phi@X)@(mbox@(mexists_ind@^[Y:mu]:(Phi@Y)))))))).
tpi(com,end_group,d3).

tpi(com,start_group,a1a).
  thf(axA1a,axiom,(v@(mforall_indset@^[Phi:mu>$i>$o]:(mimplies@(p@^[X:mu]:(mnot@(Phi@X)))@(mnot@(p@Phi)))))).
tpi(com,end_group,a1a).

tpi(com,start_group,a1b).
  thf(axA1b,axiom,(v@(mforall_indset@^[Phi:mu>$i>$o]:(mimplies@(mnot@(p@Phi))@(p@^[X:mu]:(mnot@(Phi@X))))))).
tpi(com,end_group,a1b).
```

Signature

```
tpi(com,start_group,symmetry).
  thf(sym,axiom,(![S:$i,T:$i]:((rel@S@T)=>(rel@T@S)))).
tpi(com,end_group,symmetry).

tpi(com,start_group,sig).
  thf(p_tp,type,(p:(mu>$i>$o)>$i>$o)).
  thf(g_tp,type,(g:mu>$i>$o)).
  thf(ess_tp,type,(ess:(mu>$i>$o)>mu>$i
  thf(ne_tp,type,(ne:mu>$i>$o)).
tpi(com,end_group,sig).
```

Definitions D1-D3 and Axiom A1(a/b)

```
tpi(com,start_group,d1).
  thf(defD1,definition,(g=(^[X:mu]:(mforall_indset@^[Phi:mu>$i>$o]:(mimplies@(p@Phi)@(Phi@X)))))).
tpi(com,end_group,d1).

tpi(com,start_group,d2).
  thf(defD2,definition,(ess=(^[Phi:mu>$i>$o,X:mu]:
   (mand@(Phi@X)@(mforall_indset@^[Psi:mu>$i>$o]:
                (mimplies@(Psi@X)@(mbox@(mforall_ind@^[Y:mu]:(mimplies@(Phi@Y)@(Psi@Y)))))))))).
tpi(com,end_group,d2).

tpi(com,start_group,d3).
  thf(defD3,definition,(ne=(^[X:mu]:(mforall_indset@^[Phi:mu>$i>$o]:
   (mimplies@(ess@Phi@X)@(mbox@(mexists_ind@^[Y:mu]:(Phi@Y)))))))).
tpi(com,end_group,d3).

tpi(com,start_group,a1a).
  thf(axA1a,axiom,(v@(mforall_indset@^[Phi:mu>$i>$o]:(mimplies@(p@^[X:mu]:(mnot@(Phi@X)))@(mnot@(p@Phi)))))).
tpi(com,end_group,a1a).

tpi(com,start_group,a1b).
  thf(axA1b,axiom,(v@(mforall_indset@^[Phi:mu>$i>$o]:(mimplies@(mnot@(p@Phi))@(p@^[X:mu]:(mnot@(Phi@X))))))).
tpi(com,end_group,a1b).
```

```
tpi(com,start_group,a2).
  thf(axA2,axiom,(v@(mforall_indset@^[Phi:mu>$i>$o]:
   (mforall_indset@^[Psi:mu>$i>$o]:
     (mimplies@(mand@(p@Phi)@(mbox@(mforall_ind@^[X:mu]:(mimplies@(Phi@X)@(Psi@X)))))@(p@Psi)))))).
tpi(com,end_group,a2).

tpi(com,start_group,a3).
  thf(axA3,axiom,(v@(p@g))).
tpi(com,end_group,a3).

tpi(com,start_group,a4).
  thf(axA4,axiom,(v@(mforall_indset@^[Phi:mu>$i>$o]:(mimplies@(p@Phi)@(mbox@(p@Phi)))))).
tpi(com,end_group,a4).

tpi(com,start_group,a5).
 thf(axA5,axiom,(v@(p@ne))).
tpi(com,end_group,a5).

tpi(com,write,'%%%    Done.').
tpi(com,write,'%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%').
tpi(com,write,'').

% Checking asynchroneously for satisfiability of Axioms.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
tpi(com,execute_async,'SZS_AXIOM_STATUS' = 'Nitrox---2013 60 $getgroups(tpi)').

% Checking asynchroneously for unsatisfiability of Goedel's original Axioms (modified definition D2).
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
tpi(com,start_group,d2orig).
  thf(defD2orig,definition,(ess=(^[Phi:mu>$i>$o,X:mu]:(mforall_indset@^[Psi:mu>$i>$o]:
   (mimplies@(Psi@X)@(mbox@(mforall_ind@^[Y:mu]:(mimplies@(Phi@Y)@(Psi@Y))))))))).
tpi(com,end_group,d2orig).
tpi(com,execute_async,'SZS_STATUS_D2orig' =
    'LEO-II---1.6.0 120 $getgroups(embedding,sig,symmetry,d2orig,a1a,a2,d3,a5)').
```

Gödel's God as THF TPI Script

```
tpi(com,start_group,a2).
  thf(axA2,axiom,(v@(mforall_indset@^[Phi:mu>$i>$o]:
   (mforall_indset@^[Psi:mu>$i>$o]:
     (mimplies@(mand@(p@Phi)@(mbox@(mforall_ind@^[X:mu]:(mimplies@(Phi@X)@(Psi@X)))))@(p@Psi)))))).
tpi(com,end_group,a2).

tpi(com,start_group,a3).
  thf(axA3,axiom,(v@(p@g))).
tpi(com,end_group,a3).

tpi(com,start_group,a4).
  thf(axA4,axiom,(v@(mforall_indset@^[Phi:mu>$i>$o]:(mimplies@(p@Phi)@(mbox@(p@Phi)))))).
tpi(com,end_group,a4).

tpi(com,start_group,a5).
 thf(axA5,axiom,(v@(p@ne))).
tpi(com,end_group,a5).

tpi(com,write,'%%%    Done.').
tpi(com,write,'%%%%%%%%%%%%%%%%%%%%%%%
tpi(com,write,'').

% Checking asynchroneously for satisfiability of Axioms.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
tpi(com,execute_async,'SZS_AXIOM_STATUS' = 'Nitrox---2013 60 $getgroups(tpi)').

% Checking asynchroneously for unsatisfiability of Goedel's original Axioms (modified definition D2).
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
tpi(com,start_group,d2orig).
  thf(defD2orig,definition,(ess=(^[Phi:mu>$i>$o,X:mu]:(mforall_indset@^[Psi:mu>$i>$o]:
   (mimplies@(Psi@X)@(mbox@(mforall_ind@^[Y:mu]:(mimplies@(Phi@Y)@(Psi@Y))))))))).
tpi(com,end_group,d2orig).
tpi(com,execute_async,'SZS_STATUS_D2orig' =
    'LEO-II---1.6.0 120 $getgroups(embedding,sig,symmetry,d2orig,a1a,a2,d3,a5)').
```

Axioms A2-A5

```
tpi(com,start_group,a2).
  thf(axA2,axiom,(v@(mforall_indset@^[Phi:mu>$i>$o]:
   (mforall_indset@^[Psi:mu>$i>$o]:
     (mimplies@(mand@(p@Phi)@(mbox@(mforall_ind@^[X:mu]:(mimplies@(Phi@X)@(Psi@X)))))@(p@Psi)))))).
tpi(com,end_group,a2).

tpi(com,start_group,a3).
  thf(axA3,axiom,(v@(p@g))).
tpi(com,end_group,a3).

tpi(com,start_group,a4).
  thf(axA4,axiom,(v@(mforall_indset@^[Phi:mu>$i>$o]:(mimplies@(p@Phi)@(mbox@(p@Phi)))))).
tpi(com,end_group,a4).

tpi(com,start_group,a5).
 thf(axA5,axiom,(v@(p@ne))).
tpi(com,end_group,a5).

tpi(com,write,'%%%    Done.').
tpi(com,write,'%%%%%%%%%%%%%%%%%%%%%%%
tpi(com,write,'').

% Checking asynchroneously for satisfiability of Axioms.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
tpi(com,execute_async,'SZS_AXIOM_STATUS' = 'Nitrox---2013 60 $getgroups(tpi)').

% Checking asynchroneously for unsatisfiability of Goedel's original Axioms (modified definition D2).
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
tpi(com,start_group,d2orig).
  thf(defD2orig,definition,(ess=(^[Phi:mu>$i>$o,X:mu]:(mforall_indset@^[Psi:mu>$i>$o]:
   (mimplies@(Psi@X)@(mbox@(mforall_ind@^[Y:mu]:(mimplies@(Phi@Y)@(Psi@Y))))))))).
tpi(com,end_group,d2orig).
tpi(com,execute_async,'SZS_STATUS_D2orig' =
    'LEO-II---1.6.0 120 $getgroups(embedding,sig,symmetry,d2orig,a1a,a2,d3,a5)').
```

Start checking consistency (asynchronously, Nitpick)

Gödel's God as THF TPI Script

```
tpi(com,start_group,a2).
  thf(axA2,axiom,(v@(mforall_indset@^[Phi:mu>$i>$o]:
   (mforall_indset@^[Psi:mu>$i>$o]:
     (mimplies@(mand@(p@Phi)@(mbox@(mforall_ind@^[X:mu]:(mimplies@(Phi@X)@(Psi@X)))))@(p@Psi)))))).
tpi(com,end_group,a2).

tpi(com,start_group,a3).
  thf(axA3,axiom,(v@(p@g))).
tpi(com,end_group,a3).

tpi(com,start_group,a4).
  thf(axA4,axiom,(v@(mforall_indset@^[Phi:mu>$i>$o]:(mimplies@(p@Phi)@(mbox@(p@Phi)))))).
tpi(com,end_group,a4).

tpi(com,start_group,a5).
 thf(axA5,axiom,(v@(p@ne))).
tpi(com,end_group,a5).

tpi(com,write,'%%%    Done.').
tpi(com,write,'%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%').
tpi(com,write,'').

% Checking asynchroneously for satisfiability of Axioms.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
tpi(com,execute_async,'SZS_AXIOM_STATUS

% Checking asynchroneously for unsatis
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
tpi(com,start_group,d2orig).
  thf(defD2orig,definition,(ess=(^[Phi:mu>$i>$o,X:mu]:(mforall_indset@^[Psi:mu>$i>$o]:
   (mimplies@(Psi@X)@(mbox@(mforall_ind@^[Y:mu]:(mimplies@(Phi@Y)@(Psi@Y))))))))).
tpi(com,end_group,d2orig).
tpi(com,execute_async,'SZS_STATUS_D2orig' =
    'LEO-II---1.6.0 120 $getgroups(embedding,sig,symmetry,d2orig,a1a,a2,d3,a5)').
```

Start checking consistency (asynchronously, LEO-II)
Gödel's original definition of D2

```
% 2. Analysing theorem t1.
%%%%%%%%%%%%%%%%%%%%%%%%%%
tpi(com,start_group,t1).
  thf(thmT1_con,conjecture,
   (v@(mforall_indset@^[Phi:mu>$i>$o]:(mimplies@(p@Phi)@(mdia@(mexists_ind@^[X:mu]:(Phi@X))))))).
tpi(com,end_group,t1).
tpi(com,execute,'SZS_STATUS_t1' = 'LEO-II---1.6.0 20 $getgroups(embedding,sig,a1a,a2,t1)').
tpi(com,write,'%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%').
tpi(com,write,'%%% 2. Analysing theorem t1.').
tpi(com,write,'%%%    Checking a1a,a2 |- t1 (using LEO-II).').
tpi(com,output,stdout = a1a).
tpi(com,output,stdout = a2).
tpi(com,output,stdout = t1).
tpi(com,write,'%%%    SZS_STATUS for t1 is ' & '$getenv(SZS_STATUS_t1)').
tpi(com,assert,$getenv('SZS_STATUS_t1') = 'Theorem').
tpi(com,set_role,thmT1_con = lemma).
tpi(com,write,'%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%').
tpi(com,write,'').

% 3. Analysing corollary c.
%%%%%%%%%%%%%%%%%%%%%%%%%%%
tpi(com,start_group,c).
  thf(corC_con,conjecture,(v@(mdia@(mexists_ind@^[X:mu]:(g@X))))).
tpi(com,end_group,c).
tpi(com,execute,'SZS_STATUS_c' = 'LEO-II---1.6.0 20 $getgroups(embedding,sig,d1,a3,t1,c)').
tpi(com,write,'%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%').
tpi(com,write,'%%% 3. Analysing corollary c.').
tpi(com,write,'%%%    Checking d1,a3,t1 |- c (using LEO-II).').
tpi(com,output,stdout = d1).
tpi(com,output,stdout = a3).
tpi(com,output,stdout = t1).
tpi(com,output,stdout = c).
tpi(com,write,'%%%    SZS_STATUS for c is ' & '$getenv(SZS_STATUS_c)').
tpi(com,assert,$getenv('SZS_STATUS_c') = 'Theorem').
tpi(com,set_role,corC_con = lemma).
tpi(com,write,'%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%').
tpi(com,write,'').
```

Gödel's God as THF TPI Script

```
% 2. Analysing theorem t1.
%%%%%%%%%%%%%%%%%%%%%%%%%%
tpi(com,start_group,t1).
  thf(thmT1_con,conjecture,
   (v@(mforall_indset@^[Phi:mu>$i>$o]:(mimplies@(p@Phi)@(mdia@(mexists_ind@^[X:mu]:(Phi@X))))))).
tpi(com,end_group,t1).
tpi(com,execute,'SZS_STATUS_t1' = 'LEO-II---1.6.0 20 $getgroups(embedding,sig,a1a,a2,t1)').
tpi(com,write,'%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%').
tpi(com,write,'%%% 2. Analysing theorem t1.').
tpi(com,write,'%%%    Checking a1a,a2 |- t1 (using LEO-II).').
tpi(com,output,stdout = a1a).
tpi(com,output,stdout = a2).
tpi(com,output,stdout = t1).
tpi(com,write,'%%%    SZS_STATUS for t1 is ' & '$getenv(SZS_STATUS_t1)').
tpi(com,assert,$getenv('SZS_STATUS_t1') = 'Theorem').
tpi(com,set_role,thmT1_con = lemma).
tpi(com,write,'%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%').
tpi(com,write,'').

% 3. Analysing corollary c.
%%%%%%%%%%%%%%%%%%%%%%%%%%%
tpi(com,start_group,c).
  thf(corC_con,conjecture,(v@(mdia@(me
tpi(com,end_group,c).
tpi(com,execute,'SZS_STATUS_c' = 'LEO-II---1.6.0 20 $getgroups(embedding,sig,d1,a3,t1,c)').
tpi(com,write,'%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%').
tpi(com,write,'%%% 3. Analysing corollary c.').
tpi(com,write,'%%%    Checking d1,a3,t1 |- c (using LEO-II).').
tpi(com,output,stdout = d1).
tpi(com,output,stdout = a3).
tpi(com,output,stdout = t1).
tpi(com,output,stdout = c).
tpi(com,write,'%%%    SZS_STATUS for c is ' & '$getenv(SZS_STATUS_c)').
tpi(com,assert,$getenv('SZS_STATUS_c') = 'Theorem').
tpi(com,set_role,corC_con = lemma).
tpi(com,write,'%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%').
tpi(com,write,'').
```

Proving: T1 is a theorem (LEO-II)

```
% 2. Analysing theorem t1.
%%%%%%%%%%%%%%%%%%%%%%%%%
tpi(com,start_group,t1).
  thf(thmT1_con,conjecture,
   (v@(mforall_indset@^[Phi:mu>$i>$o]:(mimplies@(p@Phi)@(mdia@(mexists_ind@^[X:mu]:(Phi@X))))))).
tpi(com,end_group,t1).
tpi(com,execute,'SZS_STATUS_t1' = 'LEO-II---1.6.0 20 $getgroups(embedding,sig,a1a,a2,t1)').
tpi(com,write,'%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%').
tpi(com,write,'%%% 2. Analysing theorem t1.').
tpi(com,write,'%%%    Checking a1a,a2 |- t1 (using LEO-II).').
tpi(com,output,stdout = a1a).
tpi(com,output,stdout = a2).
tpi(com,output,stdout = t1).
tpi(com,write,'%%%    SZS_STATUS for t1 is ' & '$getenv(SZS_STATUS_t1)').
tpi(com,assert,$getenv('SZS_STATUS_t1') = 'Theorem').
tpi(com,set_role,thmT1_con = lemma).
tpi(com,write,'%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%').
tpi(com,write,'').

% 3. Analysing corollary c.
%%%%%%%%%%%%%%%%%%%%%%%%%%
tpi(com,start_group,c).
  thf(corC_con,conjecture,(v@(mdia@(mexists_ind@^[X:mu]:(g@X))))).
tpi(com,end_group,c).
tpi(com,execute,'SZS_STATUS_c' = 'LEO-II---1.6.0 20 $getgroups(embedding,sig,d1,a3,t1,c)').
tpi(com,write,'%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%').
tpi(com,write,'%%% 3. Analysing corollary c.').
tpi(com,write,'%%%    Checking d1,a3,t1 |- c (using LEO-II).').
tpi(com,output,stdout = d1).
tpi(com,output,stdout = a3).
tpi(com,output,stdout = t1).
tpi(com,output,stdout = c).
tpi(com,write,'%%%    SZS_STATUS for c is ' & '$getenv(SZS_STATUS_c)').
tpi(com,assert,$getenv('SZS_STATUS_c') = 'Theorem').
tpi(com,set_role,corC_con = lemma).
tpi(com,write,'%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%').
tpi(com,write,'').
```

Proving: C is a theorem (LEO-II)

```
% 4. Analysing theorem t2.
%%%%%%%%%%%%%%%%%%%%%%%%%%
tpi(com,start_group,t2).
  thf(thmT2_con,conjecture,(v@(mforall_ind@^[X:mu]:(mimplies@(g@X)@(ess@g@X))))).
tpi(com,end_group,t2).
tpi(com,execute,'SZS_STATUS_t2' = 'LEO-II---1.6.0 60 $getgroups(embedding,symmetry,sig,d1,d2,a1b,a4,t2)').
tpi(com,write,'%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%').
tpi(com,write,'%%% 4. Analysing theorem t2.').
tpi(com,write,'%%%    Checking d1,d2,a1b,a4 |- t2 (using LEO-II).').
tpi(com,output,stdout = d1).
tpi(com,output,stdout = d2).
tpi(com,output,stdout = a1b).
tpi(com,output,stdout = a4).
tpi(com,output,stdout = t2).
tpi(com,write,'%%%    SZS_STATUS for t2 is ' & '$getenv(SZS_STATUS_t2)').
tpi(com,assert,$getenv('SZS_STATUS_t2') = 'Theorem').
tpi(com,set_role,thmT2_con = lemma).
tpi(com,write,'%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%').
tpi(com,write,'').

% 5. Analysing theorem t3.
%%%%%%%%%%%%%%%%%%%%%%%%%%
tpi(com,start_group,t3).
  thf(thmT3_con,conjecture,(v@(mbox@(mexists_ind@^[X:mu]:(g@X))))).
tpi(com,end_group,t3).
tpi(com,execute,'SZS_STATUS_t3' = 'LEO-II---1.6.0 20 $getgroups(embedding,symmetry,sig,d1,d3,c,t2,a5,t3)').
tpi(com,write,'%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%').
tpi(com,write,'%%% 5. Analysing theorem t3.').
tpi(com,write,'%%%    Checking sym,d1,d3,c,t2,a5 |- t3 (using LEO-II).').
tpi(com,output,stdout = d1).
tpi(com,output,stdout = d3).
tpi(com,output,stdout = c).
tpi(com,output,stdout = t2).
tpi(com,output,stdout = a5).
tpi(com,output,stdout = t3).
tpi(com,write,'%%%    SZS_STATUS for t3 is ' & '$getenv(SZS_STATUS_t3)').
tpi(com,assert,$getenv('SZS_STATUS_t3') = 'Theorem').
tpi(com,set_role,thmT3_con = lemma).
tpi(com,write,'%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%').
```

```
% 4. Analysing theorem t2.
%%%%%%%%%%%%%%%%%%%%%%%%%%
tpi(com,start_group,t2).
  thf(thmT2_con,conjecture,(v@(mforall_ind@^[X:mu]:(mimplies@(g@X)@(ess@g@X))))).
tpi(com,end_group,t2).
tpi(com,execute,'SZS_STATUS_t2' = 'LEO-II---1.6.0 60 $getgroups(embedding,symmetry,sig,d1,d2,a1b,a4,t2)').
tpi(com,write,'%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%').
tpi(com,write,'%%% 4. Analysing theorem t2.').
tpi(com,write,'%%%    Checking d1,d2,a1b,a4 |- t2 (using LEO-II).').
tpi(com,output,stdout = d1).
tpi(com,output,stdout = d2).
tpi(com,output,stdout = a1b).
tpi(com,output,stdout = a4).
tpi(com,output,stdout = t2).
tpi(com,write,'%%%    SZS_STATUS for t2 is ' & '$getenv(SZS_STATUS_t2)').
tpi(com,assert,$getenv('SZS_STATUS_t2') = 'Theorem').
tpi(com,set_role,thmT2_con = lemma).
tpi(com,write,'%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%').
tpi(com,write,'').

% 5. Analysing theorem t3.
%%%%%%%%%%%%%%%%%%%%%%%%%%
tpi(com,start_group,t3).
  thf(thmT3_con,conjecture,(v@(mbox@(me
tpi(com,end_group,t3).
tpi(com,execute,'SZS_STATUS_t3' = 'LEO-II---1.6.0 20 $getgroups(embedding,symmetry,sig,d1,d3,c,t2,a5,t3)').
tpi(com,write,'%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%').
tpi(com,write,'%%% 5. Analysing theorem t3.').
tpi(com,write,'%%%    Checking sym,d1,d3,c,t2,a5 |- t3 (using LEO-II).').
tpi(com,output,stdout = d1).
tpi(com,output,stdout = d3).
tpi(com,output,stdout = c).
tpi(com,output,stdout = t2).
tpi(com,output,stdout = a5).
tpi(com,output,stdout = t3).
tpi(com,write,'%%%    SZS_STATUS for t3 is ' & '$getenv(SZS_STATUS_t3)').
tpi(com,assert,$getenv('SZS_STATUS_t3') = 'Theorem').
tpi(com,set_role,thmT3_con = lemma).
tpi(com,write,'%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%').
```

Proving: T2 is a theorem (LEO-II)

```
% 4. Analysing theorem t2.
%%%%%%%%%%%%%%%%%%%%%%%%%%
tpi(com,start_group,t2).
  thf(thmT2_con,conjecture,(v@(mforall_ind@^[X:mu]:(mimplies@(g@X)@(ess@g@X))))).
tpi(com,end_group,t2).
tpi(com,execute,'SZS_STATUS_t2' = 'LEO-II---1.6.0 60 $getgroups(embedding,symmetry,sig,d1,d2,a1b,a4,t2)').
tpi(com,write,'%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%').
tpi(com,write,'%%% 4. Analysing theorem t2.').
tpi(com,write,'%%%    Checking d1,d2,a1b,a4 |- t2 (using LEO-II).').
tpi(com,output,stdout = d1).
tpi(com,output,stdout = d2).
tpi(com,output,stdout = a1b).
tpi(com,output,stdout = a4).
tpi(com,output,stdout = t2).
tpi(com,write,'%%%    SZS_STATUS for t2 is ' & '$getenv(SZS_STATUS_t2)').
tpi(com,assert,$getenv('SZS_STATUS_t2') = 'Theorem').
tpi(com,set_role,thmT2_con = lemma).
tpi(com,write,'%%%%%%%%%%%%%%%%%%%%%%%
tpi(com,write,'').

% 5. Analysing theorem t3.
%%%%%%%%%%%%%%%%%%%%%%%%%%
tpi(com,start_group,t3).
  thf(thmT3_con,conjecture,(v@(mbox@(mexists_ind@^[X:mu]:(g@X))))).
tpi(com,end_group,t3).
tpi(com,execute,'SZS_STATUS_t3' = 'LEO-II---1.6.0 20 $getgroups(embedding,symmetry,sig,d1,d3,c,t2,a5,t3)').
tpi(com,write,'%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%').
tpi(com,write,'%%% 5. Analysing theorem t3.').
tpi(com,write,'%%%    Checking sym,d1,d3,c,t2,a5 |- t3 (using LEO-II).').
tpi(com,output,stdout = d1).
tpi(com,output,stdout = d3).
tpi(com,output,stdout = c).
tpi(com,output,stdout = t2).
tpi(com,output,stdout = a5).
tpi(com,output,stdout = t3).
tpi(com,write,'%%%    SZS_STATUS for t3 is ' & '$getenv(SZS_STATUS_t3)').
tpi(com,assert,$getenv('SZS_STATUS_t3') = 'Theorem').
tpi(com,set_role,thmT3_con = lemma).
tpi(com,write,'%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%').
```

Proving: T3 is a theorem (LEO-II)

```
% 6. Analysing theorem c2.
%%%%%%%%%%%%%%%%%%%%%%%%%%
tpi(com,start_group,c2).
  thf(corC2_con,conjecture,(v@(mexists_ind@^[X:mu]:(g@X)))).
tpi(com,end_group,c2).
tpi(com,execute,'SZS_STATUS_c2' =
    'LEO-II---1.6.0 20 $getgroups(embedding,symmetry,sig,d1,t1,t3,c2)').
tpi(com,write,'%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%').
tpi(com,write,'%%% 6. Analysing corollary c2.').
tpi(com,write,'%%%    Checking  sym,d1,t1,t3 |- c2 (using LEO-II).').
tpi(com,output,stdout = d1).
tpi(com,output,stdout = t1).
tpi(com,output,stdout = t3).
tpi(com,output,stdout = c2).
tpi(com,write,'%%%    SZS_STATUS for c2 is ' & '$getenv(SZS_STATUS_c2)').
tpi(com,assert,$getenv('SZS_STATUS_c2') = 'Theorem').
tpi(com,set_role,corC2_con = lemma).
tpi(com,write,'%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%').
tpi(com,write,'').

% 7a. Report on consistency of axioms.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
tpi(com,waitenv,'SZS_AXIOM_STATUS').
tpi(com,write,'%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%').
tpi(com,write,'%%% 7a. Checking satisfiability of all assumptions (using Nitpick).').
tpi(com,write,'%%%    SZS_AXIOM_STATUS for assumptions is '& '$getenv(SZS_AXIOM_STATUS)').
tpi(com,assert,$getenv('SZS_AXIOM_STATUS') = 'Satisfiable').
tpi(com,write,'%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%').
tpi(com,write,'').
```

```
% 6. Analysing theorem c2.
%%%%%%%%%%%%%%%%%%%%%%%%%%
tpi(com,start_group,c2).
  thf(corC2_con,conjecture,(v@(mexists_ind@^[X:mu]:(g@X)))).
tpi(com,end_group,c2).
tpi(com,execute,'SZS_STATUS_c2' =
    'LEO-II---1.6.0 20 $getgroups(embedding,symmetry,sig,d1,t1,t3,c2)').
tpi(com,write,'%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%').
tpi(com,write,'%%% 6. Analysing corollary c2.').
tpi(com,write,'%%%    Checking  sym,d1,t1,t3 |- c2 (using LEO-II).').
tpi(com,output,stdout = d1).
tpi(com,output,stdout = t1).
tpi(com,output,stdout = t3).
tpi(com,output,stdout = c2).
tpi(com,write,'%%%    SZS_STATUS for c2 is ' & '$getenv(SZS_STATUS_c2)').
tpi(com,assert,$getenv('SZS_STATUS_c2') = 'Theorem').
tpi(com,set_role,corC2_con = lemma).
tpi(com,write,'%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%').
tpi(com,write,'').

% 7a. Report on consistency of axioms.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
tpi(com,waitenv,'SZS_AXIOM_STATUS').
tpi(com,write,'%%%%%%%%%%%%%%%%%%%%%%%%
tpi(com,write,'%%% 7a. Checking satisfiability of all assumptions (using Nitpick).').
tpi(com,write,'%%%    SZS_AXIOM_STATUS for assumptions is '& '$getenv(SZS_AXIOM_STATUS)').
tpi(com,assert,$getenv('SZS_AXIOM_STATUS') = 'Satisfiable').
tpi(com,write,'%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%').
tpi(com,write,'').
```

Proving: C2 is a theorem (LEO-II)

```
% 6. Analysing theorem c2.
%%%%%%%%%%%%%%%%%%%%%%%%%%%
tpi(com,start_group,c2).
  thf(corC2_con,conjecture,(v@(mexists_ind@^[X:mu]:(g@X)))).
tpi(com,end_group,c2).
tpi(com,execute,'SZS_STATUS_c2' =
    'LEO-II---1.6.0 20 $getgroups(embedding,symmetry,sig,d1,t1,t3,c2)').
tpi(com,write,'%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%').
tpi(com,write,'%%% 6. Analysing corollary c2.').
tpi(com,write,'%%%    Checking  sym,d1,t1,t3 |- c2 (using LEO-II).').
tpi(com,output,stdout = d1).
tpi(com,output,stdout = t1).
tpi(com,output,stdout = t3).
tpi(com,output,stdout = c2).
tpi(com,write,'%%%    SZS_STATUS for c2 is ' & '$getenv(SZS_STATUS_c2)').
tpi(com,assert,$getenv('SZS_STATUS_c2') = 'Theorem').
tpi(com,set_role,corC2_con = lemma).
tpi(com,write,'%%%%%%%%%%%%%%%%%%%%%%%%
tpi(com,write,'').

% 7a. Report on consistency of axioms.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
tpi(com,waitenv,'SZS_AXIOM_STATUS').
tpi(com,write,'%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%').
tpi(com,write,'%%% 7a. Checking satisfiability of all assumptions (using Nitpick).').
tpi(com,write,'%%%    SZS_AXIOM_STATUS for assumptions is '& '$getenv(SZS_AXIOM_STATUS)').
tpi(com,assert,$getenv('SZS_AXIOM_STATUS') = 'Satisfiable').
tpi(com,write,'%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%').
tpi(com,write,'').
```

Checking: Axioms are consistent (Nitpick)

```
% 7b. Report on Inconsistency of Goedel's original axioms.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
tpi(com,waitenv,'SZS_STATUS_D2orig').
tpi(com,write,'%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%').
tpi(com,write,'%%% 7b. Proving inconsistency of Goedel‘s original axioms (d2orig).').
tpi(com,write,'%%%     Checking  d2orig,a1a,a2,d3,a5  (using LEO-II).').
tpi(com,output,stdout = d2orig).
tpi(com,write,'%%%     SZS_STATUS_D2orig for d2orig is ' & '$getenv(SZS_STATUS_D2orig)').
tpi(com,assert,$getenv('SZS_STATUS_D2orig') = 'Unsatisfiable').
tpi(com,write,'%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%').
tpi(com,write,'').

% Analysing modal collapse (asynchroneously).
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
tpi(com,start_group,mc).
  thf(phi_type,type,(phi:$i>$o)).
  thf(thmMC_con,conjecture,(v@(mimplies@phi@(mbox@phi)))).
tpi(com,end_group,mc).
tpi(com,execute_async,'SZS_STATUS_MC' =
    'LEO-II---1.6.0 40 $getgroups(embedding,symmetry,sig,d2,t2,t3,mc)').

% Analysing flawlessness (asynchroneously).
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
tpi(com,start_group,fg).
  thf(thmFG_con,conjecture,(v@(mforall_indset@^[Phi:mu>$i>$o]:(mforall_ind@^[X:mu]:
                              (mimplies@(g@X)@(mimplies@(mnot@(p@Phi))@(mnot@(Phi@X)))))))).
tpi(com,end_group,fg).
tpi(com,execute_async,'SZS_STATUS_FG' = 'LEO-II---1.6.0 40 $getgroups(embedding,sig,a1b,d1,fg)').
```

```
% 7b. Report on Inconsistency of Goedel's original axioms.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
tpi(com,waitenv,'SZS_STATUS_D2orig').
tpi(com,write,'%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%').
tpi(com,write,'%%% 7b. Proving inconsistency of Goedel‘s original axioms (d2orig).').
tpi(com,write,'%%%     Checking  d2orig,a1a,a2,d3,a5  (using LEO-II).').
tpi(com,output,stdout = d2orig).
tpi(com,write,'%%%     SZS_STATUS_D2orig for d2orig is ' & '$getenv(SZS_STATUS_D2orig)').
tpi(com,assert,$getenv('SZS_STATUS_D2orig') = 'Unsatisfiable').
tpi(com,write,'%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%').
tpi(com,write,'').

% Analysing modal collapse (asynchrone
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
tpi(com,start_group,mc).
  thf(phi_type,type,(phi:$i>$o)).
  thf(thmMC_con,conjecture,(v@(mimplies
tpi(com,end_group,mc).
tpi(com,execute_async,'SZS_STATUS_MC' =
    'LEO-II---1.6.0 40 $getgroups(embedding,symmetry,sig,d2,t2,t3,mc)').

% Analysing flawlessness (asynchroneously).
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
tpi(com,start_group,fg).
  thf(thmFG_con,conjecture,(v@(mforall_indset@^[Phi:mu>$i>$o]:(mforall_ind@^[X:mu]:
                                (mimplies@(g@X)@(mimplies@(mnot@(p@Phi))@(mnot@(Phi@X)))))))).
tpi(com,end_group,fg).
tpi(com,execute_async,'SZS_STATUS_FG' = 'LEO-II---1.6.0 40 $getgroups(embedding,sig,a1b,d1,fg)').
```

Checking: Gödel’s original axioms are inconsistent (LEO-II)

```
% 7b. Report on Inconsistency of Goedel's original axioms.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
tpi(com,waitenv,'SZS_STATUS_D2orig').
tpi(com,write,'%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%').
tpi(com,write,'%%% 7b. Proving inconsistency of Goedel's original axioms (d2orig).').
tpi(com,write,'%%%     Checking  d2orig,a1a,a2,d3,a5  (using LEO-II).').
tpi(com,output,stdout = d2orig).
tpi(com,write,'%%%     SZS_STATUS_D2orig for d2orig is ' & '$getenv(SZS_STATUS_D2orig)').
tpi(com,assert,$getenv('SZS_STATUS_D2or
tpi(com,write,'%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
tpi(com,write,'').

% Analysing modal collapse (asynchroneously).
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
tpi(com,start_group,mc).
  thf(phi_type,type,(phi:$i>$o)).
  thf(thmMC_con,conjecture,(v@(mimplies@phi@(mbox@phi)))).
tpi(com,end_group,mc).
tpi(com,execute_async,'SZS_STATUS_MC' =
    'LEO-II---1.6.0 40 $getgroups(embedding,symmetry,sig,d2,t2,t3,mc)').

% Analysing flawlessness (asynchroneously).
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
tpi(com,start_group,fg).
  thf(thmFG_con,conjecture,(v@(mforall_indset@^[Phi:mu>$i>$o]:(mforall_ind@^[X:mu]:
                                 (mimplies@(g@X)@(mimplies@(mnot@(p@Phi))@(mnot@(Phi@X)))))))).
tpi(com,end_group,fg).
tpi(com,execute_async,'SZS_STATUS_FG' = 'LEO-II---1.6.0 40 $getgroups(embedding,sig,a1b,d1,fg)').
```

Start checking modal collapse (asynchronously, LEO-II)

```
% 7b. Report on Inconsistency of Goedel's original axioms.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
tpi(com,waitenv,'SZS_STATUS_D2orig').
tpi(com,write,'%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%').
tpi(com,write,'%%% 7b. Proving inconsistency of Goedel`s original axioms (d2orig).').
tpi(com,write,'%%%     Checking  d2orig,a1a,a2,d3,a5  (using LEO-II).').
tpi(com,output,stdout = d2orig).
tpi(com,write,'%%%     SZS_STATUS_D2orig for d2orig is ' & '$getenv(SZS_STATUS_D2orig)').
tpi(com,assert,$getenv('SZS_STATUS_D2orig') = 'Unsatisfiable').
tpi(com,write,'%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%').
tpi(com,write,'').

% Analysing modal collapse (asynchroneously).
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
tpi(com,start_group,mc).
  thf(phi_type,type,(phi:$i>$o)).
  thf(thmMC_con,conjecture,(v@(mimplies@phi@(mbox@phi)))).
tpi(com,end_group,mc).
tpi(com,execute_async,'SZS_STATUS_MC' =
    'LEO-II---1.6.0 40 $getgroups(embed

% Analysing flawlessness (asynchroneously).
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
tpi(com,start_group,fg).
  thf(thmFG_con,conjecture,(v@(mforall_indset@^[Phi:mu>$i>$o]:(mforall_ind@^[X:mu]:
                                (mimplies@(g@X)@(mimplies@(mnot@(p@Phi))@(mnot@(Phi@X)))))))).
tpi(com,end_group,fg).
tpi(com,execute_async,'SZS_STATUS_FG' = 'LEO-II---1.6.0 40 $getgroups(embedding,sig,a1b,d1,fg)').
```

Start checking ’flawlessness’ of God (LEO-II)

```
% 8. Report on modal collapse.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
tpi(com,waitenv,'SZS_STATUS_MC').
tpi(com,write,'%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%').
tpi(com,write,'%%% 8. Analysing modal collapse.').
tpi(com,write,'%%%    Checking  sym,d2,t2,t3 |- mc (using LEO-II).').
tpi(com,output,stdout = mc).
tpi(com,write,'%%%    SZS_STATUS_MC for mc is ' & '$getenv(SZS_STATUS_MC)').
tpi(com,assert,$getenv('SZS_STATUS_MC') = 'Theorem').
tpi(com,set_role,thmMC_con = lemma).
tpi(com,write,'%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%').
tpi(com,write,'').

% 9. Report on flawlessness.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
tpi(com,waitenv,'SZS_STATUS_FG').
tpi(com,write,'%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%').
tpi(com,write,'%%% 9. Analysing flawless god.').
tpi(com,write,'%%%    Checking  a1b,d1 |- fg (using LEO-II).').
tpi(com,output,stdout = fg).
tpi(com,write,'%%%    SZS_STATUS_FG for fg is ' & '$getenv(SZS_STATUS_FG)').
tpi(com,assert,$getenv('SZS_STATUS_FG') = 'Theorem').
tpi(com,set_role,thmFG_con = lemma).
tpi(com,write,'%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%').
tpi(com,write,'').
```

```
% 8. Report on modal collapse.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
tpi(com,waitenv,'SZS_STATUS_MC').
tpi(com,write,'%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%').
tpi(com,write,'%%% 8. Analysing modal collapse.').
tpi(com,write,'%%%    Checking  sym,d2,t2,t3 |- mc (using LEO-II).').
tpi(com,output,stdout = mc).
tpi(com,write,'%%%    SZS_STATUS_MC for mc is ' & '$getenv(SZS_STATUS_MC)').
tpi(com,assert,$getenv('SZS_STATUS_MC') = 'Theorem').
tpi(com,set_role,thmMC_con = lemma).
tpi(com,write,'%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%').
tpi(com,write,'').

% 9. Report on flawlessness.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
tpi(com,waitenv,'SZS_STATUS_FG').
tpi(com,write,'%%%%%%%%%%%%%%%%%%%%%%%%
tpi(com,write,'%%% 9. Analysing flawless god.').
tpi(com,write,'%%%    Checking  a1b,d1 |- fg (using LEO-II).').
tpi(com,output,stdout = fg).
tpi(com,write,'%%%    SZS_STATUS_FG for fg is ' & '$getenv(SZS_STATUS_FG)').
tpi(com,assert,$getenv('SZS_STATUS_FG') = 'Theorem').
tpi(com,set_role,thmFG_con = lemma).
tpi(com,write,'%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%').
tpi(com,write,'').
```

Checking: modal collapse holds (LEO-II)

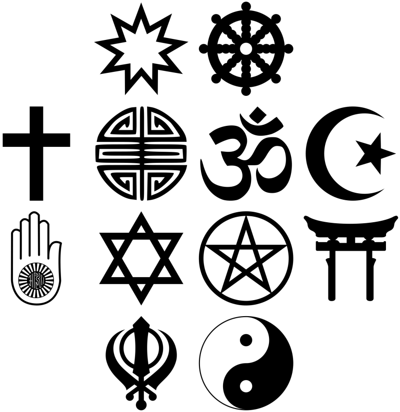
```
% 8. Report on modal collapse.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
tpi(com,waitenv,'SZS_STATUS_MC').
tpi(com,write,'%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%').
tpi(com,write,'%%% 8. Analysing modal collapse.').
tpi(com,write,'%%%    Checking  sym,d2,t2,t3 |- mc (using LEO-II).').
tpi(com,output,stdout = mc).
tpi(com,write,'%%%    SZS_STATUS_MC for mc is ' & '$getenv(SZS_STATUS_MC)').
tpi(com,assert,$getenv('SZS_STATUS_MC') = 'Theorem').
tpi(com,set_role,thmMC_con = lemma).
tpi(com,write,'%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
tpi(com,write,'').

% 9. Report on flawlessness.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
tpi(com,waitenv,'SZS_STATUS_FG').
tpi(com,write,'%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%').
tpi(com,write,'%%% 9. Analysing flawless god.').
tpi(com,write,'%%%    Checking  a1b,d1 |- fg (using LEO-II).').
tpi(com,output,stdout = fg).
tpi(com,write,'%%%    SZS_STATUS_FG for fg is ' & '$getenv(SZS_STATUS_FG)').
tpi(com,assert,$getenv('SZS_STATUS_FG') = 'Theorem').
tpi(com,set_role,thmFG_con = lemma).
tpi(com,write,'%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%').
tpi(com,write,'').
```

Checking: ’flawlessness’ of God (LEO-II)

```
% 10. Analysing monotheism.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%
tpi(com,start_group,mt).
  thf(thmMT_con,conjecture,(v@(mforall_ind@^[X:mu]:(mforall_ind@^[Y:mu]:
                               (mimplies@(g@X)@(mimplies@(g@Y)@(mequals@X@Y))))))).
tpi(com,end_group,mt).
tpi(com,execute,'SZS_STATUS_MT' = 'TPS---3.120601S1b 60 $getgroups(embedding,sig,fg,d1,mt)').
tpi(com,write,'%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%').
tpi(com,write,'%%% 10. Analysing monotheism.').
tpi(com,write,'%%%     Checking  fg,d1 |- mt (using TPS).').
%tpi(com,output,stdout = tpi_premises).
%tpi(com,output,stdout = tpi_conjectures).
tpi(com,output,stdout = mt).
tpi(com,write,'%%%     SZS_STATUS_MT for mt is ' & '$getenv(SZS_STATUS_MT)').
tpi(com,assert,$getenv('SZS_STATUS_MT') = 'Theorem').
tpi(com,write,'%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%').
tpi(com,write,'').

tpi(end,exit,exit).
```

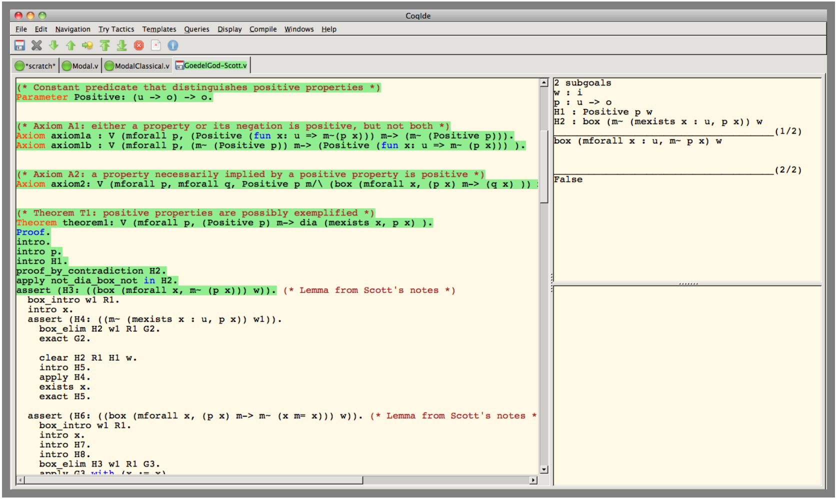
```
% 10. Analysing monotheism.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%
tpi(com,start_group,mt).
  thf(thmMT_con,conjecture,(v@(mforall_ind@^[X:mu]:(mforall_ind@^[Y:mu]:
                              (mimplies@(g@X)@(mimplies@(g@Y)@(mequals@X@Y))))))).
tpi(com,end_group,mt).
tpi(com,execute,'SZS_STATUS_MT' = 'TPS---3.120601S1b 60 $getgroups(embedding,sig,fg,d1,mt)').
tpi(com,write,'%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%').
tpi(com,write,'%%% 10. Analysing monotheism.').
tpi(com,write,'%%%     Checking  fg,d1 |- mt (using TPS).').
%tpi(com,output,stdout = tpi_premises).
%tpi(com,output,stdout = tpi_conjectures).
tpi(com,output,stdout = mt).
tpi(com,write,'%%%     SZS_STATUS_MT for mt is ' & '$getenv(SZS_STATUS_MT)').
tpi(com,assert,$getenv('SZS_STATUS_MT') = 'Theorem').
tpi(com,write,'%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%').
tpi(com,write,'').

tpi(end,exit,exit).
```

Proving:Monotheism (TPS)



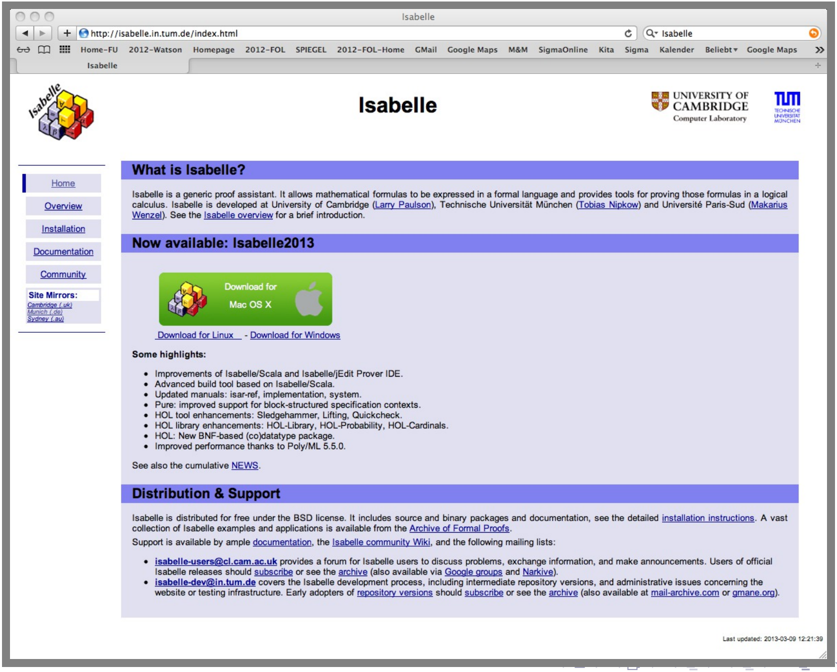
Part C:
Formalization and Verification in Coq

- Goal: verification of the natural deduction proof
  - Step-by-step formalization
  - Almost no automation (intentionally!)
- Interesting facts:
  - Embedding is transparent to the user
  - Embedding gives labeled calculus for free


CoqIde
File Edit Navigation Try Tactics Templates Queries Display Compile Windows Help
scratch Modal.v ModalClassical.v GoedelGod-Scott.v
(* Constant predicate that distinguishes positive properties *)
Parameter Positive: (u -> o) -> o.
(* Axiom A1: either a property or its negation is positive, but not both *)
Axiom axiom1a : V (mforall p, (Positive (fun x: u => m~(p x))) m-> (m~ (Positive p))).
Axiom axiom1b : V (mforall p, (m~ (Positive p)) m-> (Positive (fun x: u => m~ (p x)))).
(* Axiom A2: a property necessarily implied by a positive property is positive *)
Axiom axiom2: V (mforall p, mforall q, Positive p m/\ (box (mforall x, (p x) m-> (q x)))
(* Theorem T1: positive properties are possibly exemplified *)
Theorem theorem1: V (mforall p, (Positive p) m-> dia (mexists x, p x)).
Proof.
intro.
intro p.
intro H1.
proof_by_contradiction H2.
apply not_dia_box_not in H2.
assert (H3: ((box (mforall x, m~ (p x))) w)). (* Lemma from Scott's notes *)
box_intro w1 R1.
intro x.
assert (H4: ((m~ (mexists x : u, p x)) w1)).
box_elim H2 w1 R1 G2.
exact G2.
clear H2 R1 H1 w.
intro H5.
apply H4.
exists x.
exact H5.
assert (H6: ((box (mforall x, (p x) m-> m~ (x m= x))) w)). (* Lemma from Scott's notes *
box_intro w1 R1.
intro x.
intro H7.
intro H8.
box_elim H3 w1 R1 G3.
2 subgoals
w : i
p : u -> o
H1 : Positive p w
H2 : box (m~ (mexists x : u, p x)) w
(1/2)
box (mforall x : u, m~ p x) w
(2/2)
False

Part D:

Automation and Verification in ISABELLE/HOL



Isabelle

- Home
- Overview
- Installation
- Documentation
- Community

Site Mirrors:
Cambridge (.uk)
Munich (.de)
Sydney (.au)

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 is developed at University of Cambridge (Larry Paulson), Technische Universität München (Tobias Nipkow) and Université Paris-Sud (Makarius Wenzel). See the Isabelle overview for a brief introduction.

Now available: Isabelle2013

Download for Mac OS X

Download for Linux - Download for Windows

Some highlights:

- Improvements of Isabelle/Scala and Isabelle/jEdit Prover IDE.
- Advanced build tool based on Isabelle/Scala.
- Updated manuals: isar-ref, implementation, system.
- Pure: improved support for block-structured specification contexts.
- HOL tool enhancements: Sledgehammer, Lifting, Quickcheck.
- HOL library enhancements: HOL-Library, HOL-Probability, HOL-Cardinals.
- HOL: New BNF-based (co)datatype package.
- Improved performance thanks to Poly/ML 5.5.0.

See also the cumulative NEWS.

Distribution & Support

Isabelle is distributed for free under the BSD license. It includes 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.

Support is available by ample documentation, the Isabelle community Wiki, and the following mailing lists:

- **isabelle-users@cl.cam.ac.uk** provides a forum for Isabelle users to discuss problems, exchange information, and make announcements. Users of official Isabelle releases should subscribe or see the archive (also available via Google groups and Narkive).
- **isabelle-dev@in.tum.de** covers the Isabelle development process, including intermediate repository versions, and administrative issues concerning the website or testing infrastructure. Early adopters of repository versions should subscribe or see the archive (also available at mail-archive.com or gmane.org).

Last updated: 2013-03-09 12:21:39

Isabelle/HOL (Cambridge University/TU Munich)

- HOL instance of the generic ISABELLE proof assistant
- User interaction and proof automation
- Automation is supported by SLEDGEHAMMER tool
- Verification of the proofs in ISABELLE/HOL's small proof kernel

What we did?

- Proof automation of Gödel's proof script (Scott's version)
- SLEDGEHAMMER makes calls to remote THF provers in Miami
- These calls the suggest respective calls to the METIS prover
- METIS proofs are verified in ISABELLE/HOL's proof kernel

— see the handout (generated from the Isabelle source file) —

GoedelGod.thy (modified)

GoedelGod.thy (~/GoedelGod/Formalizations/Isabelle/GoedelGodSession/)

```
  corollary C: "[◇ (∃ G)]"
  sledgehammer [provers = remote_leo2] by (metis A3 T1)

text {* Axiom @{text "A4"} is added: $\all \phi [P(\phi) \to \Box \; P(\phi)]$
(Positive properties are necessarily positive). *}

  axiomatization where A4:  "[Π (λΦ. P Φ m⇒ □ (P Φ))]"

text {* Symbol @{text "ess"} for `Essence' is introduced and defined as
$\ess{\phi}{x} \biimp \phi(x) \wedge \all \psi (\psi(x) \imp \nec \all y (\phi(y)
\imp \psi(y)))$ (An \emph{essence} of an individual is a property possessed by it
and necessarily implying any of its properties). *}

  definition ess :: "(μ ⇒ σ) ⇒ μ ⇒ σ" (infixr "ess" 85) where
    "Φ ess x = Φ x m∧ Π (λψ. ψ x m⇒ □ (∀ (λy. Φ y m⇒ ψ y)))"

text {* Next, Sledgehammer and Metis prove theorem @{text "T2"}: $\all x [G(x) \imp \ess{G}{x}]$
(Being God-like is an essence of any God-like being). *}

  theorem T2: "[∀ (λx. G x m⇒ G ess x)]"
  sledgehammer [provers = remote_leo2] by (metis A1b A4 G_def ess_def)

text {* Symbol @{text "NE"}, for `Necessary Existence', is introduced and
defined as $\NE(x) \biimp \all \phi [\ess{\phi}{x} \imp \nec \ex y \phi(y)]$ (Necessary
existence of an individual is the necessary exemplification of all its essences). *}

  definition NE :: "μ ⇒ σ" where "NE = (λx. Π (λΦ. Φ ess x m⇒ □ (∃ Φ)))"
```

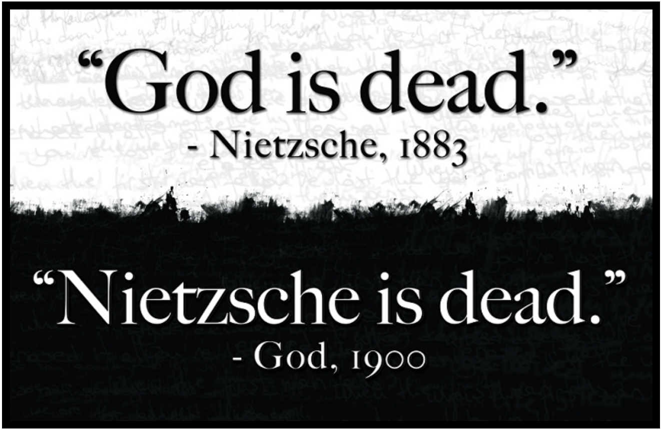
100% ☑ Auto update Update Detach

```
Sledgehammering...
```

Output README Symbols

139,39 (6613/8211) (isabelle,sidekick,UTF-8-Isabelle) Nmro UG 157/183MB 16:02

Sidekick Theories



Part E:
Criticisms

$$\forall P.[\Diamond\Box P \rightarrow \Box P]$$

If something is possibly necessary, then it is necessary.

$$\Diamond\Box(A \vee \neg A) \qquad \Box(A \vee \neg A)$$

S5 usually considered adequate

(But KB is sufficient! — shown by HOL ATPs)

Criticisms: Modal logic S5 is too strong

$$\forall P.[\Diamond\Box P \rightarrow \Box P]$$

If something is possibly necessary, then it is necessary.

$\Diamond\Box(A \vee \neg A)$ $\Box(A \vee \neg A)$

S5 usually considered adequate

(But KB is sufficient! — shown by HOL ATPs)

Criticisms: Modal logic S5 is too strong

$$\forall P.[\Diamond\Box P \rightarrow \Box P]$$

If something is possibly necessary, then it is necessary.

$$\Diamond\Box(A \vee \neg A) \qquad \Box(A \vee \neg A)$$

S5 usually considered adequate

(But KB is sufficient! — shown by HOL ATPs)

$$\forall P.[\Diamond\Box P \rightarrow \Box P]$$

If something is possibly necessary, then it is necessary.

$$\Diamond\Box(A \vee \neg A) \qquad \Box(A \vee \neg A)$$

S5 usually considered adequate

(But KB is sufficient! — shown by HOL ATPs)

$$\forall P.[\Diamond\Box P \rightarrow \Box P]$$

If something is possibly necessary, then it is necessary.

$$\Diamond\Box(A \vee \neg A) \qquad \Box(A \vee \neg A)$$

S5 usually considered adequate

(But KB is sufficient! — shown by HOL ATPs)

Criticisms: Modal logic S5 is too strong

$$\forall P.[\Diamond\Box P \rightarrow \Box P]$$

If something is possibly necessary, then it is necessary.

$$\Diamond\Box(A \vee \neg A) \qquad \Box(A \vee \neg A)$$

S5 usually considered adequate

(But KB is sufficient! — shown by HOL ATPs)

Criticisms: Modal logic S5 is too strong

$$\forall P.[\Diamond\Box P \rightarrow \Box P]$$

If something is possibly necessary, then it is necessary.

$$\Diamond\Box(A \vee \neg A) \qquad \Box(A \vee \neg A)$$

S5 usually considered adequate

(But KB is sufficient! — shown by HOL ATPs)

Criticisms: Modal logic S5 is too strong

$$\forall P.[\Diamond\Box P \rightarrow \Box P]$$

If something is possibly necessary, then it is necessary.

$$\Diamond_c\Box_c(A \vee \neg A) \qquad \Box_c(A \vee \neg A)$$

S5 usually considered adequate

(But KB is sufficient! — shown by HOL ATPs)

Criticisms: Gödel's Axioms imply Modal Collapse

$$\forall P.[P \rightarrow \Box P]$$

Everything that is the case is so necessarily.

Follows from T2, T3 and D2 (as shown by HOL ATPs).

There are no contingent "truths".
Everything is determined.
There is no free will.

Many proposed solutions: Anderson, Fitting, Hájek, . . .

Criticisms: Gödel's Axioms imply Modal Collapse

$$\forall P.[P \rightarrow \Box P]$$

Everything that is the case is so necessarily.

Follows from T2, T3 and D2 (as shown by HOL ATPs).

There are no contingent "truths".
Everything is determined.
There is no free will.

Many proposed solutions: Anderson, Fitting, Hájek, ...

Criticisms: Gödel's Axioms imply Modal Collapse

$$\forall P.[P \rightarrow \Box P]$$

Everything that is the case is so necessarily.

Follows from T2, T3 and D2 (as shown by HOL ATPs).

There are no contingent "truths".
Everything is determined.
There is no free will.

Many proposed solutions: Anderson, Fitting, Hájek, . . .

$$\forall P.[P \rightarrow \Box P]$$

Everything that is the case is so necessarily.

Follows from T2, T3 and D2 (as shown by HOL ATPs).

There are no contingent "truths".
Everything is determined.
There is no free will.

Many proposed solutions: Anderson, Fitting, Hájek, ...

Criticisms: Gödel's Axioms imply Modal Collapse

$$\forall P.[P \rightarrow \Box P]$$

Everything that is the case is so necessarily.

Follows from T2, T3 and D2 (as shown by HOL ATPs).

There are no contingent "truths".
Everything is determined.
There is no free will.

Many proposed solutions: Anderson, Fitting, Hájek, . . .

$$\forall P.[P \rightarrow \Box P]$$

Everything that is the case is so necessarily.

Follows from T2, T3 and D2 (as shown by HOL ATPs).

There are no contingent "truths".
Everything is determined.
There is no free will.

Many proposed solutions: Anderson, Fitting, Hájek, ...

Criticisms: Gödel's Axioms imply Modal Collapse

$$\forall P.[P \rightarrow \Box P]$$

Everything that is the case is so necessarily.

Follows from T2, T3 and D2 (as shown by HOL ATPs).

There are no contingent "truths".
Everything is determined.
There is no free will.

Many proposed solutions: Anderson, Fitting, Hájek, ...

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

Either a property is positive or its negation is (but never both)

Are the following properties positive or negative?

$\lambda x.G(x)$ $\quad$ $\lambda x.NE(x)$ $\quad$ $\lambda x.x = x$ $\quad$ $\lambda x.\top$

$\lambda x.blue(x)$ $\quad$ $\lambda x.punishing(x)$ $\quad$ $\lambda x.human(x)$

Solution:
"…positive in the moral aesthetic sense (independently of the accidental structure of the world). Only then the ax. true. …"
- Gödel, 1970

See also my extended Isabelle formalization at:
`https://github.com/FormalTheology/GoedelGod/blob/master/Formalizations/Isabelle/DivineVersion/GoedelGodDivine.thy`

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

Either a property is positive or its negation is (but never both)

Are the following properties positive or negative?

$\lambda x.G(x)$ $\quad$ $\lambda x.NE(x)$ $\quad$ $\lambda x.x = x$ $\quad$ $\lambda x.\top$

$\lambda x.blue(x)$ $\quad$ $\lambda x.punishing(x)$ $\quad$ $\lambda x.human(x)$

Solution:
"...positive in the moral aesthetic sense (independently of the accidental structure of the world). Only then the ax. true. ..."
- Gödel, 1970

See also my extended Isabelle formalization at:
`https://github.com/FormalTheology/GoedelGod/blob/master/Formalizations/Isabelle/DivineVersion/GoedelGodDivine.thy`

Criticisms: No Neutral Properties

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

Either a property is positive or its negation is (but never both)

Are the following properties positive or negative?

$\lambda x.G(x)$ $\lambda x.NE(x)$ $\lambda x.x = x$ $\lambda x.\top$

$\lambda x.blue(x)$ $\lambda x.punishing(x)$ $\lambda x.human(x)$

Solution:

"...positive in the moral aesthetic sense (independently of the accidental structure of the world). Only then the ax. true. ..."

- Gödel, 1970

See also my extended Isabelle formalization at:
https://github.com/FormalTheology/GoedelGod/blob/master/Formalizations/Isabelle/DivineVersion/GoedelGodDivine.thy

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

Either a property is positive or its negation is (but never both)

Are the following properties positive or negative?

$\lambda x.G(x)$ $\lambda x.NE(x)$ $\lambda x.x = x$ $\lambda x.\top$

$\lambda x.blue(x)$ $\lambda x.punishing(x)$ $\lambda x.human(x)$

Solution:
"...positive in the moral aesthetic sense (independently of the accidental structure of the world). Only then the ax. true. ..."
- Gödel, 1970

See also my extended Isabelle formalization at:
https://github.com/FormalTheology/GoedelGod/blob/master/Formalizations/Isabelle/DivineVersion/GoedelGodDivine.thy

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

Either a property is positive or its negation is (but never both)

Are the following properties positive or negative?

$\lambda x.G(x)$ $\lambda x.NE(x)$ $\lambda x.x = x$ $\lambda x.\top$

$\lambda x.blue(x)$ $\lambda x.punishing(x)$ $\lambda x.human(x)$

Solution:

"…positive in the moral aesthetic sense (independently of the accidental structure of the world). Only then the ax. true. …"

- Gödel, 1970

See also my extended Isabelle formalization at:
https://github.com/FormalTheology/GoedelGod/blob/master/Formalizations/Isabelle/DivineVersion/GoedelGodDivine.thy

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

Either a property is positive or its negation is (but never both)

Are the following properties positive or negative?

$$\lambda x.G(x) \qquad \lambda x.NE(x) \qquad \lambda x.x = x \qquad \lambda x.\top$$

$$\lambda x.blue(x) \qquad \lambda x.punishing(x) \qquad \lambda x.human(x)$$

Solution:
"…positive in the moral aesthetic sense (independently of the accidental structure of the world). Only then the ax. true. …"
- Gödel, 1970

See also my extended Isabelle formalization at:
https://github.com/FormalTheology/GoedelGod/blob/master/Formalizations/Isabelle/DivineVersion/GoedelGodDivine.thy

Part F:
Conclusions

The (new) insights we gained from experiments include:

- Logic K sufficient for T1, C and T2
- Logic S5 not needed for T3
- Logic KB sufficient for T3 (not well known)
- We found a simpler new proof of C
- Gödel's axioms (without conjunct $\phi(x)$ in D2) are inconsistent
- Scott's axioms are consistent
- For T1, only half of A1 (A1a) is needed
- For T2, the other half (A1b) is needed

Our novel contributions to the theorem proving community include

- Powerful infrastructure for reasoning with QML
- A new natural deduction calculus for higher-order modal logic
- Difficult new benchmarks problems for HOL provers
- Huge media attention

What have we achieved

- Verification of Gödel's ontological argument with HOL provers
  - exact parameters known: constant domain quantification, Henkin Semantics
  - experiments with different parameters could be performed
- Gained some novel results and insights
- Major step towards Computer-assisted Theoretical Philosophy
  - see also Ed Zalta's *Computational Metaphysics* project at Stanford University
  - see also John Rushby's recent verification of Anselm's proof in PVS
  - remember Leibniz' dictum — *Calculemus!*
- Interesting bridge between CS, Philosophy and Theology

Ongoing and future work

- Formalize and verify literature on ontological arguments
  - ... in particular the criticism and improvements to Gödel
- Own contributions — supported by theorem provers

What have we achieved

- Verification of Gödel's ontological argument with HOL provers
  - exact parameters known: constant domain quantification, Henkin Semantics
  - experiments with different parameters could be performed
- Gained some novel results and insights
- Major step towards Computer-assisted Theoretical Philosophy
  - see also Ed Zalta's *Computational Metaphysics* project at Stanford University
  - see also John Rushby's recent verification of Anselm's proof in PVS
  - remember Leibniz' dictum — *Calculemus!*
- Interesting bridge between CS, Philosophy and Theology

Ongoing and future work

- Formalize and verify literature on ontological arguments
  - . . . in particular the criticism and improvements to Gödel
- Own contributions — supported by theorem provers

I'm sure that God would be impressed with your proof, if only he existed :-)

Larry

Die Philosophen können so schön staunen.

Sie packen Dinge in Begriffe (gucken dabei in die Luft) werfen die Begriffe dann in ihre Philosophiekiste, schütteln ganz dolle, und freuen sich, dass ganz genau rauskommt, was sie vorher reingetan haben. Und das geht sogar, wenn eine Maschine die Kiste schüttelt.

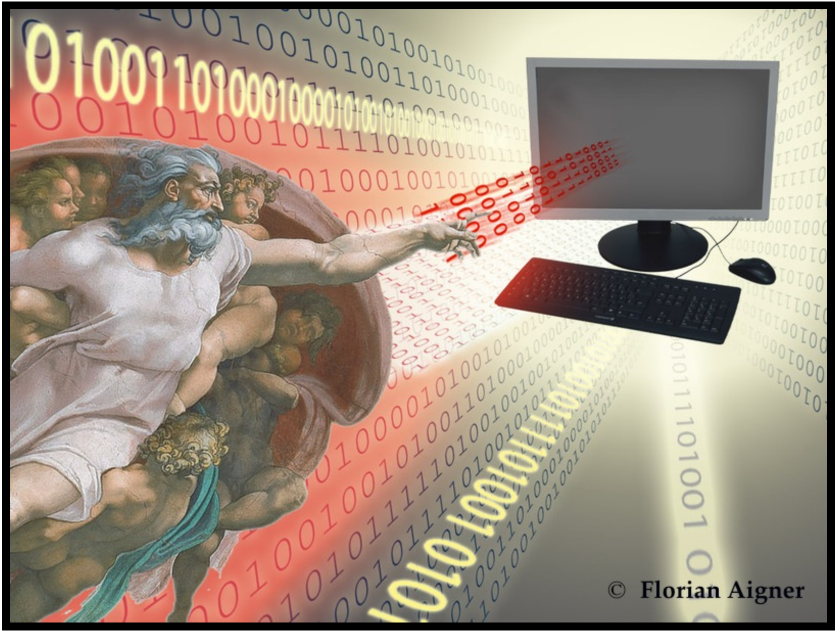
Unerstaunt
2017cp

60. Suchlauf

souveränsatt 09.09.2013

man kann auch auf andere Weise in diesem Zusammenhang methodisch vorgehen: bei einer längeren Autofahrt das Radio auf automatischen Suchlauf stellen. Nach zwei Tagen sieht man Gott

...find more on the internet ...


© Florian Aigner

Licenses