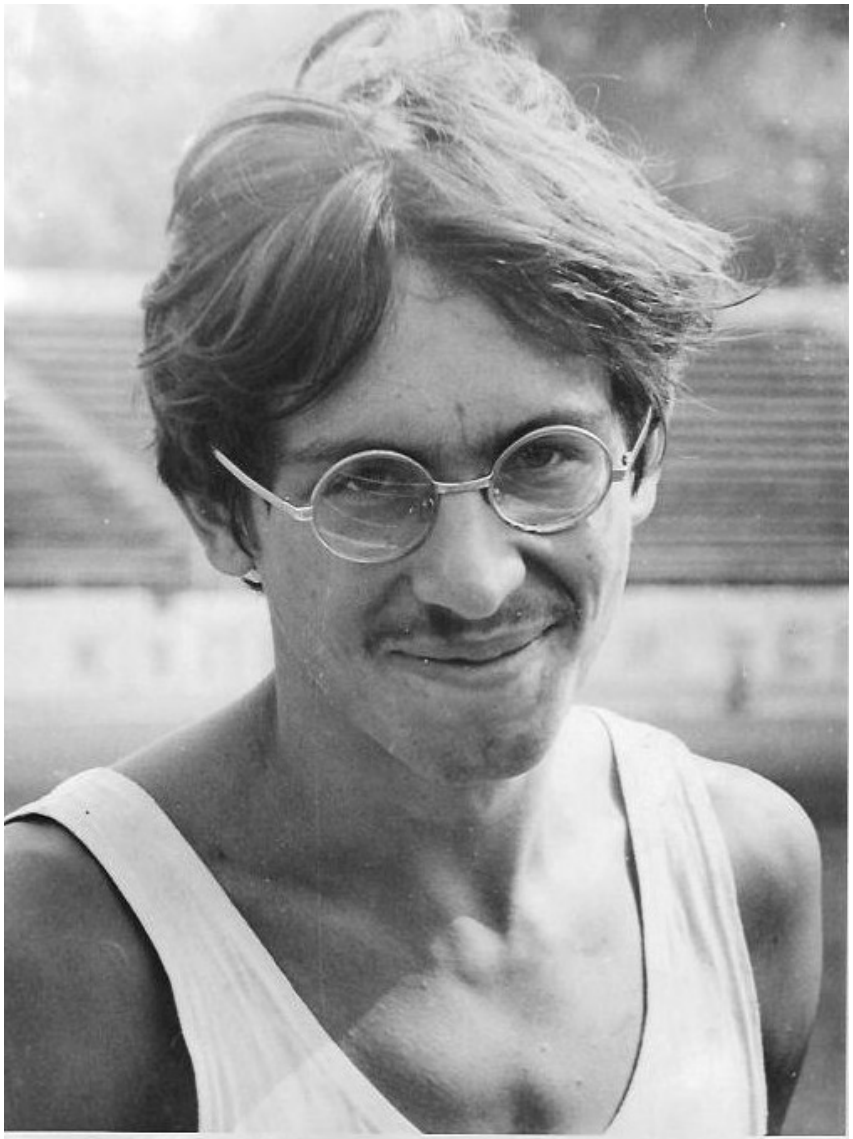
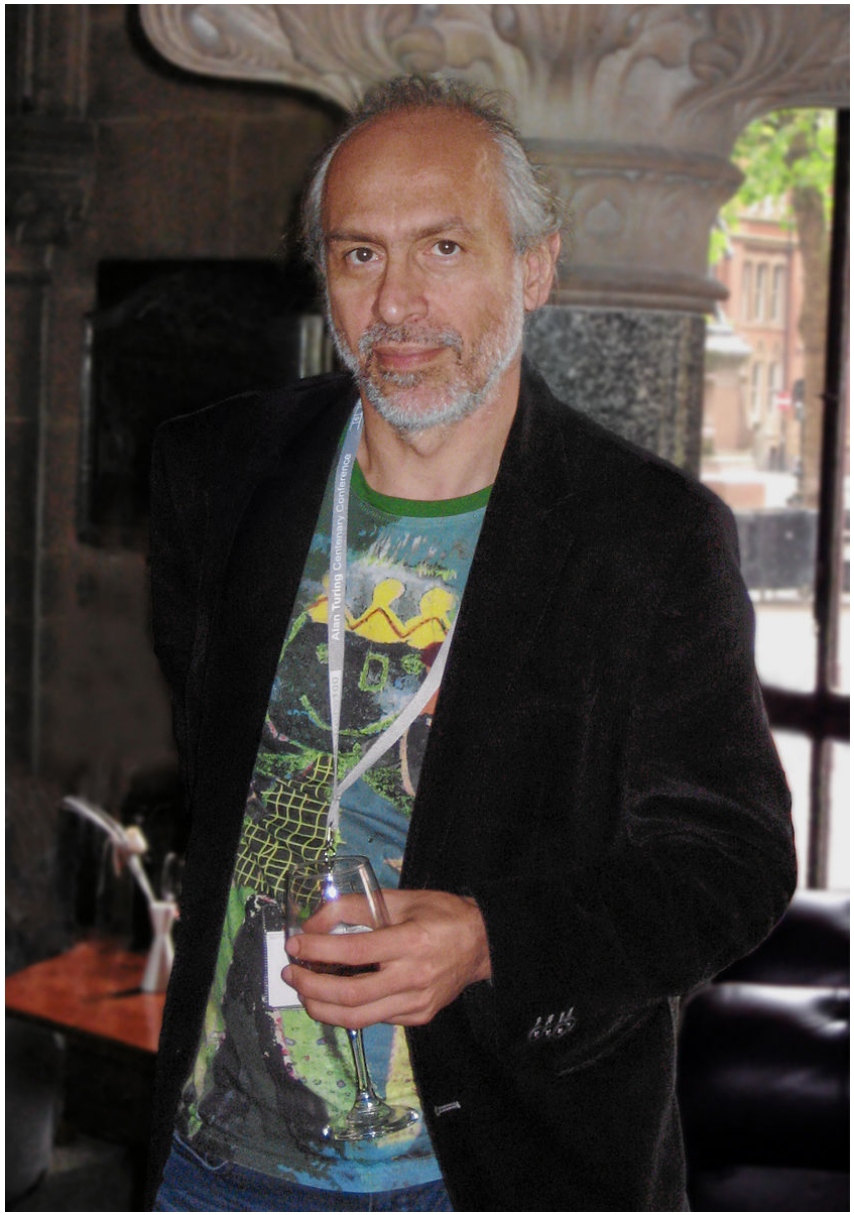
Higher-Order Automated Theorem Proving
– Why has there been such a Resistance? –

C. Benzmüller
FU Berlin & U Luxembourg

ANDREI-60, Tbilisi, Georgia, May 22, 2019

Andrei 60 — Happy Birthday!





And thank you! Competition is only fun if there are strong athletes to chase.

Leo-III

∀P.Leo III

with A. Steen, M. Wisniewski

What is Leo-III?

- ATP for classical HOL based on extensional higher-order paramodulation
- integrates SMT- & FO-ATPs
- supports all TPTP syntax dialects
- native support for **more than 120 modal logics** (normal quantified modal logics)
- including native support for **quantified SDL and DDL**
- Website: `http://page.mi.fu-berlin.de/lex/leo3/`

Recent Independent Evaluation:

- GRUNGE: A Grand Unified ATP Challenge [arXiv:1903.02539], accepted for CADE

∀P.Leo III

with A. Steen, M. Wisniewski

What is Leo-III?

- ATP for classical HOL based on extensional higher-order paramodulation
- integrates SMT- & FO-ATPs
- supports all TPTP syntax dialects
- native support for **more than 120 modal logics** (normal quantified modal logics)
- including native support for **quantified SDL and DDL**
- Website: `http://page.mi.fu-berlin.de/lex/leo3/`

Recent Independent Evaluation:

- GRUNGE: A Grand Unified ATP Challenge [arXiv:1903.02539], accepted for CADE

∀P.Leo III

with A. Steen, M. Wisniewski

What is Leo-III?

- ATP for classical HOL based on extensional higher-order paramodulation
- integrates SMT- & FO-ATPs
- supports all TPTP syntax dialects
- native support for **more than 120 modal logics** (normal quantified modal logics)
- including native support for **quantified SDL and DDL**
- Website: `http://page.mi.fu-berlin.de/lex/leo3/`

Recent Independent Evaluation:

- GRUNGE: A Grand Unified ATP Challenge [arXiv:1903.02539], accepted for CADE

Leo-III

Computer Science > Logic in Computer Science

GRUNGE: A Grand Unified ATP Challenge

Chad E. Brown, Thibault Gauthier, Cezary Kaliszyk, Geoff Sutcliffe, Josef Urban

(Submitted on 6 Mar 2019)

This paper describes a large set of related theorem proving problems obtained by translating theorems from the HOL4 standard library into multiple logical formalisms. The formalisms are in higher-order logic (with and without type variables) and first-order logic (possibly with multiple types, and possibly with type variables). The resultant problem sets allow us to run automated theorem provers that support different logical formats on corresponding problems, and compare their performances. This also results in a new "grand unified" large theory benchmark that emulates the ITP/ATP hammer setting, where systems and metasystems can use multiple ATP formalisms in complementary ways, and jointly learn from the accumulated knowledge.

Leo-III

System	TH1-I	TH0-I	TH0-II	TF1-I	TF0-I	TF0-II	FOF-I	FOF-II	Union
agsyHOL		1374	1187						1605
Beagle					2007	2047			2531
cocATP		899	599						1000
CSE_E							4251	3102	4480
CVC4					4851	3991			5252
E					4277	3622	4618	3844	5118
HOLyHammer	5059								5059
iProver							2778	2894	3355
iProverMo'					2435	1639			2699
LEO-II		2579	1923						3213
Leo-III	6668	5018	3485	3458	4032	3421			7062
Metis							2353	474	2356
Princess					3646	2138			3849
Prover9							2894	1742	3128
Satallax		2207	1292						2494
SPASS							2850	3349	3821
Vampire					4837	4693	4008	4928	5929
Zipperp'n		2252	2161	3771	3099	2576			4203
Union	6824	5209	3771	4608	5732	5073	5165	5108	**7377**

Table 2. Number of theorems proved, out of 12140.

What you should do in the next 60 years!

Let EasyChair make you rich!

But spent more time with Vampire

and also go in bed with Lambda's

What you should do in the next 60 years!

Let EasyChair make you rich!

But spent more time with Vampire

and also go in bed with Lambda's

What you should do in the next 60 years!

Let EasyChair make you rich!

But spent more time with Vampire

and also go in bed with Lambda's

Why Lambda's should be your friends?

Why Lambda's should be your friends?

$$\{x|\text{green}(x) \wedge \text{alien}(x)\}$$

corresponds to

$$\lambda x_\alpha.\text{green}_{\alpha\to o}x_\alpha \wedge \text{alien}_{\alpha\to o}x_\alpha$$

$\cup$ **is abreviation for** $\lambda\varphi_{\alpha\to o}.\lambda\psi_{\alpha\to o}.(\lambda x_\alpha.\varphi_{\alpha\to o}x_\alpha \vee \psi_{\alpha\to o}x_\alpha)$

$\cap$ **is abreviation for** $\lambda\varphi_{\alpha\to o}.\lambda\psi_{\alpha\to o}.(\lambda x_\alpha.\varphi_{\alpha\to o}x_\alpha \wedge \psi_{\alpha\to o}x_\alpha)$

$\emptyset$ **is abreviation for** $\lambda x_\alpha.\bot$

etc.

Why Lambda's should be your friends?

One important aspect:

$$\exists N_{\overline{\alpha^n}\to\beta}\ \forall \overline{z^n}\ N(z^1,\ldots,z^n) = B_\beta$$

versus

$$\lambda \overline{z^n}.B_\beta$$

And another one:

$$\forall x_\alpha.P(x)$$

versus

$$\Pi(\lambda x_\alpha.P\ x)$$

And there is more: . . .

Example: TPTP Problem SET171+3

Axiomatization in FO Set Theory

Assumptions:

$\forall B, C, x_{\bullet}(x \in (B \cup C) \Leftrightarrow x \in B \vee x \in C)$

$\forall B, C, x_{\bullet}(x \in (B \cap C) \Leftrightarrow x \in B \wedge x \in C)$

$\forall B, C_{\bullet}(B \subseteq C \Leftrightarrow \forall x_{\bullet} x \in B \Rightarrow x \in C)$

$\forall B, C_{\bullet}(B \cup C = C \cup B)$

$\forall B, C_{\bullet}(B \cap C = C \cap B)$

$\forall B, C_{\bullet}(B = C \Leftrightarrow B \subseteq C \wedge C \subseteq B)$

$\forall B, C_{\bullet}(B = C \Leftrightarrow \forall x_{\bullet} x \in B \Leftrightarrow x \in C)$

Proof Goal:

$\forall B, C, D_{\bullet}$

$$B \cup (C \cap D) = (B \cup C) \cap (B \cup D)$$

Performance: FO-ATPs

```
% SPASS---3.0
% Problem  : SET171+3
% SPASS beiseite: Ran out of time.

% E---0.999
% Problem  : SET171+3
% Failure: Resource limit exceeded
(time)

% Vampire---9.0
% Problem  : SET171+3
% Result   : Theorem 68.6s
```

Performance: LEO-II + E

```
Eureka --- Thanks to Corina!
Total Reasoning Time: 0.03s
LEO-II (Proof Found!)
```

Example: TPTP Problem SET171+3

Axiomatization in FO Set Theory

Assumptions:

$\forall B, C, x_{\bullet}(x \in (B \cup C) \Leftrightarrow x \in B \vee x \in C)$

$\forall B, C, x_{\bullet}(x \in (B \cap C) \Leftrightarrow x \in B \wedge x \in C)$

$\forall B, C_{\bullet}(B \subseteq C \Leftrightarrow \forall x_{\bullet} x \in B \Rightarrow x \in C)$

$\forall B, C_{\bullet}(B \cup C = C \cup B)$

$\forall B, C_{\bullet}(B \cap C = C \cap B)$

$\forall B, C_{\bullet}(B = C \Leftrightarrow B \subseteq C \wedge C \subseteq B)$

$\forall B, C_{\bullet}(B = C \Leftrightarrow \forall x_{\bullet} x \in B \Leftrightarrow x \in C)$

Proof Goal:

$\forall B, C, D_{\bullet}$

$$B \cup (C \cap D) = (B \cup C) \cap (B \cup D)$$

Performance: FO-ATPs

```
% SPASS---3.0
% Problem  : SET171+3
% SPASS beiseite: Ran out of time.

% E---0.999
% Problem  : SET171+3
% Failure: Resource limit exceeded
(time)

% Vampire---9.0
% Problem  : SET171+3
% Result   : Theorem 68.6s
```

Performance: LEO-II + E

```
Eureka --- Thanks to Corina!
Total Reasoning Time: 0.03s
LEO-II (Proof Found!)
```

Example: TPTP Problem SET171+3

Axiomatization in FO Set Theory

Assumptions:

$\forall B, C, x_{\bullet}(x \in (B \cup C) \Leftrightarrow x \in B \vee x \in C)$

$\forall B, C, x_{\bullet}(x \in (B \cap C) \Leftrightarrow x \in B \wedge x \in C)$

$\forall B, C_{\bullet}(B \subseteq C \Leftrightarrow \forall x_{\bullet} x \in B \Rightarrow x \in C)$

$\forall B, C_{\bullet}(B \cup C = C \cup B)$

$\forall B, C_{\bullet}(B \cap C = C \cap B)$

$\forall B, C_{\bullet}(B = C \Leftrightarrow B \subseteq C \wedge C \subseteq B)$

$\forall B, C_{\bullet}(B = C \Leftrightarrow \forall x_{\bullet} x \in B \Leftrightarrow x \in C)$

Proof Goal:

$\forall B, C, D_{\bullet}$

$$B \cup (C \cap D) = (B \cup C) \cap (B \cup D)$$

Performance: FO-ATPs

```
% SPASS---3.0
% Problem  : SET171+3
% SPASS beiseite: Ran out of time.

% E---0.999
% Problem  : SET171+3
% Failure: Resource limit exceeded
(time)

% Vampire---9.0
% Problem  : SET171+3
% Result   : Theorem 68.6s
```

Performance: LEO-II + E

```
Eureka --- Thanks to Corina!
Total Reasoning Time: 0.03s
LEO-II (Proof Found!)
```

Example: TPTP Problem SET171+3

Axiomatization in FO Set Theory

Assumptions:

$\forall B, C, x_{\bullet}(x \in (B \cup C) \Leftrightarrow x \in B \vee x \in C)$

$\forall B, C, x_{\bullet}(x \in (B \cap C) \Leftrightarrow x \in B \wedge x \in C)$

$\forall B, C_{\bullet}(B \subseteq C \Leftrightarrow \forall x_{\bullet} x \in B \Rightarrow x \in C)$

$\forall B, C_{\bullet}(B \cup C = C \cup B)$

$\forall B, C_{\bullet}(B \cap C = C \cap B)$

$\forall B, C_{\bullet}(B = C \Leftrightarrow B \subseteq C \wedge C \subseteq B)$

$\forall B, C_{\bullet}(B = C \Leftrightarrow \forall x_{\bullet} x \in B \Leftrightarrow x \in C)$

Proof Goal:

$\forall B, C, D_{\bullet}$

$$B \cup (C \cap D) = (B \cup C) \cap (B \cup D)$$

Performance: FO-ATPs

```
% SPASS---3.0
% Problem  : SET171+3
% SPASS beiseite: Ran out of time.

% E---0.999
% Problem  : SET171+3
% Failure: Resource limit exceeded
(time)

% Vampire---9.0
% Problem  : SET171+3
% Result   : Theorem 68.6s
```

Performance: LEO-II + E

```
Eureka --- Thanks to Corina!
Total Reasoning Time: 0.03s
LEO-II (Proof Found!)
```

Results

Problem	Vamp. 9.0	LEO+Vamp.	LEO-II+E
014+4	114.5	2.60	0.300
017+1	1.0	5.05	0.059
066+1	–	3.73	0.029
067+1	4.6	0.10	0.040
076+1	51.3	0.97	0.031
086+1	0.1	0.01	0.028
096+1	5.9	7.29	0.033
143+3	0.1	0.31	0.034
171+3	68.6	0.38	0.030
580+3	0.0	0.23	0.078
601+3	1.6	1.18	0.089
606+3	0.1	0.27	0.033
607+3	1.2	0.26	0.036
609+3	145.2	0.49	0.039
611+3	0.3	4.00	0.125
612+3	111.9	0.46	0.030
614+3	3.7	0.41	0.060
615+3	103.9	0.47	0.035
623+3	–	2.27	0.282
624+3	3.8	3.29	0.047
630+3	0.1	0.05	0.025
640+3	1.1	0.01	0.033
646+3	84.4	0.01	0.032
647+3	98.2	0.12	0.037
648+3	98.2	0.12	0.037
649+3	117.5	0.25	0.037
651+3	117.5	0.09	0.029
657+3	146.6	0.01	0.028
669+3	83.1	0.20	0.041
670+3	–	0.14	0.067
671+3	214.9	0.47	0.038
672+3	–	0.23	0.034
673+3	217.1	0.47	0.042
680+3	146.3	2.38	0.035
683+3	0.3	0.27	0.053
684+3	–	3.39	0.039
716+4	–	0.40	0.033
724+4	–	1.91	0.032
741+4	–	3.70	0.042
747+4	–	1.18	0.028
752+4	–	516.00	0.086
753+4	–	1.64	0.037
764+4	0.1	0.01	0.032

Vamp. 9.0: 2.80GHz, 1GB memory, 600s time limit
LEO+Vamp.: 2.40GHz, 4GB memory, 120s time limit
LEO-II+E: 1.60GHz, 1GB memory, 60s time limit

Test: May 2019

```
%------------------------------------------------------------------------------
% File     : SET171+3 : TPTP v7.2.0. Released v2.2.0.
% Domain   : Set Theory
% Problem  : Union distributes over intersection
%        (B union (C intersection D)) = ((B union C) intersection (B union C))

% END OF SYSTEM OUTPUT
% RESULT: SOT_7huDyR - E---2.3 says Timeout - CPU = 60.04 WC = 60.17
% OUTPUT: SOT_7huDyR - E---2.3 says None - CPU = 0 WC = 0

% END OF SYSTEM OUTPUT
% RESULT: SOT_7huDyR - SPASS---3.9 says Timeout - CPU = 60.03 WC = 60.16
% OUTPUT: SOT_7huDyR - SPASS---3.9 says None - CPU = 0 WC = 0

% END OF SYSTEM OUTPUT
% RESULT: SOT_7huDyR - Vampire---4.3 says Theorem - CPU = 2.29 WC = 2.26
% OUTPUT: SOT_7huDyR - Vampire---4.3 says Refutation - CPU = 2.29 WC = 2.26
%------------------------------------------------------------------------------
```

Test: May 2019

```
%------------------------------------------------------------------------------
% File     : SET623+3 : TPTP v7.2.0. Released v2.2.0.
% Domain   : Set Theory
% Problem  :  Associativity of symmetric difference
%           (X symDiff Y) symDiff Z = X symDiff (Y symDiff Z)

% END OF SYSTEM OUTPUT
% RESULT: SET623+3 - E---2.3 says Timeout - CPU = 60.07 WC = 60.20
% OUTPUT: SET623+3 - E---2.3 says None - CPU = 0 WC = 0

% END OF SYSTEM OUTPUT
% RESULT: SET623+3 - SPASS---3.9 says Timeout - CPU = 60.04 WC = 60.15
% OUTPUT: SET623+3 - SPASS---3.9 says None - CPU = 0 WC = 0

% END OF SYSTEM OUTPUT
% RESULT: SET623+3 - Vampire---4.3 says Timeout - CPU = 60.04 WC = 58.06
% OUTPUT: SET623+3 - Vampire---4.3 says None - CPU = 0 WC = 0
%------------------------------------------------------------------------------
```

Cantor's theorem: The set of all subsets of A, that is, the power set of A, has a strictly greater cardinality than A itself.

In HOL Cantor's theorem (surjective version) can be encoded as

$$\neg\exists G_{i\to(i\to o)}\forall F_{i\to o}\exists X_i.GX = F$$

HO ATPs can solve this problem very efficiently.

Solution includes the detection and application of the diagonalisation argument.

In the Leo (and TPS) provers this works with **primitive substitution**:

$$F \leftarrow \lambda X_i.\neg H_{i\to(i\to o)}XX$$

Further reading: [AndrewsEtAl., Automating higher-order logic, 1984]

Expressivity Matters — Boolos' Example — More on Comprehension

[George Boolos, A curious inference, J. Philosophical Logic, 16:1-12, 1987]

1. $\forall n\, f(n,1) = s(1)$
2. $\forall x\, f(1,s(x)) = s(s(f(1,x)))$
3. $\forall n \forall x\, f(s(n),s(x)) = f(n,f(s(n),x))$
4. $D(1)$
5. $\forall x\, D(x) \rightarrow D(s(x))$

   $\therefore$
6. $D(f(s(s(s(s(1))))),s(s(s(s(1))))))$

Induction proof: from (4) and (5), we get $\forall x D(x)$, hence $D(f(s(s(s(s(1))))),s(s(s(s(1))))))$ by $\forall$-elimination.

But induction is not given, hence the first order proof consists of brute force modus ponens applications: infeasible number of steps ($2^{(2\cdots 2)}$ with 64K '2s')

Expressivity Matters — Boolos' Example — More on Comprehension

[Boolos, A curious inference, J. Philosophical Logic, 16:1-12, 1987]

Comprehension axioms $\exists N_{\overline{\alpha^n}\to\beta}\ \forall \overline{z^n}\ N(z^1,\ldots,z^n) = B_\beta$

Can be avoided: use λ-binding construct to denote N

Instances of comprehension axioms in Boolos' proof:

$$\exists N \forall z N(z) \leftrightarrow (\forall X X(1) \land \forall y(X(y) \to X(s(y))) \to X(z))$$

$$\exists E \forall z E(z) \leftrightarrow (N(z) \land D(z))$$

Central idea: "assume the induction principle holds for number z – corresponding to $N(z)$ – then we can show for any predicate X a property $X(z)$ by induction."

The proof employs the following lemmata:

Lemma 1:
$N(1)$, $\forall y(N(y) \to N(s(y)))$, $N(s(s(s(s(1))))))$, $E(1)$, $\forall y(E(y) \to E(s(y)))$, $E(s(1))$

Lemma 2: $\forall n N(n) \to \forall x(N(x) \to E(f(n,x)))$

The theorem itself is then an easy application of the two lemmata.

Expressivity Matters — Boolos' Example — More on Comprehension

[1]By the comprehension principle of second order logic, $\exists N \forall z(Nz \leftrightarrow \forall X[X1 \& \forall y(Xy \rightarrow Xsy) \rightarrow Xz])$, [2]and then for some N, $\exists E \forall z(Ez \leftrightarrow Nz \& Dz)$.

[3]**Lemma 1:** [3.1]$N1$; [3.2]$\forall y(Ny \rightarrow Nsy)$; [3.3]$Nssss1$; [3.4]$E1$; [3.5]$\forall y(Ey \rightarrow Esy)$; [3.6]$Es1$;

[4]**Lemma 2:** $\forall n(Nn \rightarrow (\forall x(Nx \rightarrow Efnx)))$

Proof: [4.1]By comprehension, $\exists M \forall n(Mn \leftrightarrow \forall x(Nx \Rightarrow Efnx))$. [4.2]We want $\forall n(Nn \rightarrow Mn)$. [4.3]Enough to show [4.3.1]$M1$ and [4.3.2]$\forall n(Mn \rightarrow Msn)$, for then if [4.4]$Nn$, [4.5]$Mn$.

[4.3.1]$M1$: [4.3.1.1]Want $\forall x(Nx \rightarrow Ef1x)$. [4.3.1.2]By comprehension, $\exists Q \forall x(Qx \leftrightarrow Ef1x)$. [4.3.1.3]Want $\forall x(Nx \rightarrow Qx)$. [4.3.1.4]Enough to show [4.3.1.4.1]$Q1$ and [4.3.1.4.2]$\forall x(Qx \rightarrow Qsx)$.

[4.3.1.4.1]$Q1$: [4.3.1.4.1.1]Want $Ef11$. [4.3.1.4.1.2]But $f11 = s1$ by (1) and [4.3.1.4.1.3]$Es1$ by Lemma 1.

[4.3.1.4.2]$\forall x(Qx \rightarrow Qsx)$: [4.3.1.4.2.1]Suppose Qx, [4.3.1.4.2.2]i.e. Ef1x. [4.3.1.4.2.3]By (2) $f1sx = ssf1x$; [4.3.1.4.2.4]by Lemma 1 twice, $Ef1sx$. [4.3.1.4.2.5]Thus Qsx and [4.3.1.4.2.6]$M1$.

[4.3.2]$\forall n(Mn \rightarrow Msn)$: [4.3.2.1]Suppose Mn, [4.3.2.2]i.e. $\forall x(Nx \rightarrow Efnx)$. [4.3.2.3]Want Msn, [4.3.2.4]i.e. $\forall x(Nx \rightarrow Efsnx)$. [4.3.2.5]By comprehension, $\exists P \forall x(Px \leftrightarrow Efsnx)$. [4.3.2.6]Want $\forall x(Nx \rightarrow Px)$. [4.3.2.7]Enough to show [4.3.2.7.1]$P1$ and [4.3.2.7.2]$\forall x(Px \rightarrow Psx)$.

[4.3.2.7.1]$P1$: [4.3.2.7.1.1]Want $Efsn1$. [4.3.2.7.1.2]But $fsn1 = s1$ by (1) and [4.3.2.7.1.3]$Es1$ by Lemma 1.

[4.3.2.7.2]$\forall x(Px \rightarrow Psx)$: [4.3.2.7.2.1]Suppose Px, [4.3.2.7.2.2]i.e. $Efsnx$; [4.3.2.7.2.3]thus $Nfsnx$. [4.3.2.7.2.4]Want $Efsnsx$. [4.3.2.7.2.5]Since $Nfsnx$ and Mn, $Efnfsnx$. [4.3.2.7.2.6]But by (3) $fnfsnx = fsnsx$; [4.3.2.7.2.7]thus $Efsnsx$.

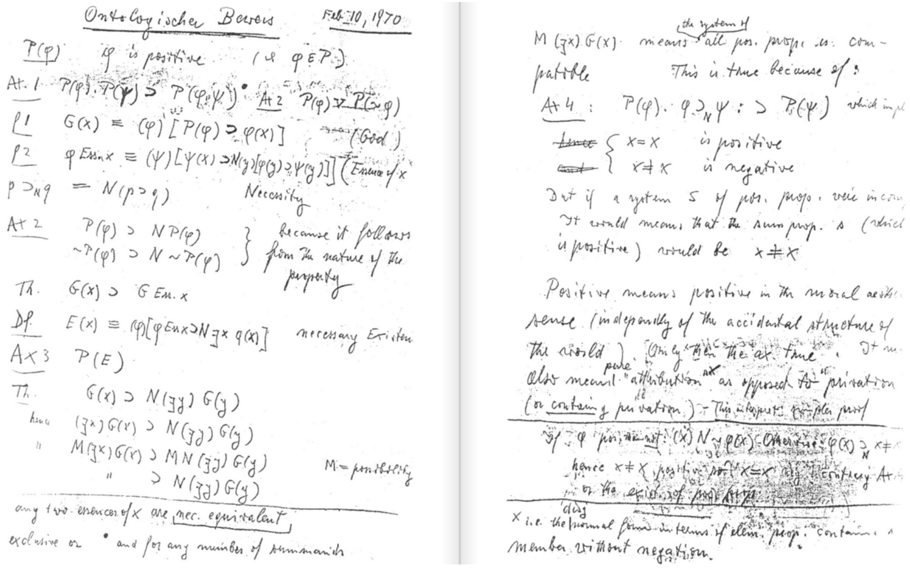
[5]By Lemma 1, $Nssss1$. [6]By Lemma 2, $Efssss1ssss1$. [7]Thus, $Dfssss1ssss1$, as desired.

Formalization in OMEGA and Mizar:

[BenzmüllerBrown, The curious inference of Boolos in MIZAR and OMEGA, Studies in Logic, Grammar, and Rhetoric, volume 10(23), pp. 299-388, 2007.]

Earlier paper: [BenzmüllerKerber, A Lost Proof, 2001].

Expressivity Matters — Inconsisency in Gödel's Ontological Argument



Expressivity Matters — Inconsisency in Gödel's Ontological Argument

Ontologischer Beweis Feb 10, 1970

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

Ax. 1 $P(\varphi) \cdot P(\psi) \supset P(\varphi \cdot \psi)$ · Ax 2 $P(\varphi) \vee P(\neg\varphi)$

D1 $G(x) \equiv (\varphi)[P(\varphi) \supset \varphi(x)]$ (God)

D2 $\varphi \, Ess_x \equiv (\psi)[\psi(x) \supset N(y)[\varphi(y) \supset \psi(y)]]$ (Essence of x

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

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

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

Df $E(x) \equiv (\varphi)[\varphi \, Ess x \supset N \exists x \, \varphi(x)]$ necessary Existen

Ax 3 $P(E)$

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

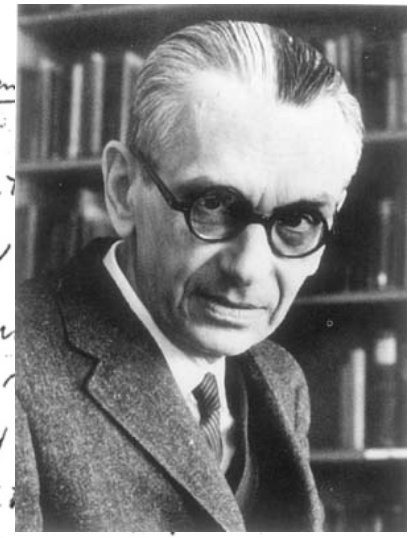
hence $(\exists x) G(x) \supset N(\exists y) G(y)$

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

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

any two essences of x are nec. equivalent

exclusive or • and for any number of summands



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

patible This is

Ax 4 : $P(\varphi) \cdot \varphi \supset_N \psi$

~~Thus~~ $x = x$ is p

~~not~~ $x \neq x$ is

But if a system S

It would mean that

is positive) would be $x \neq x$

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

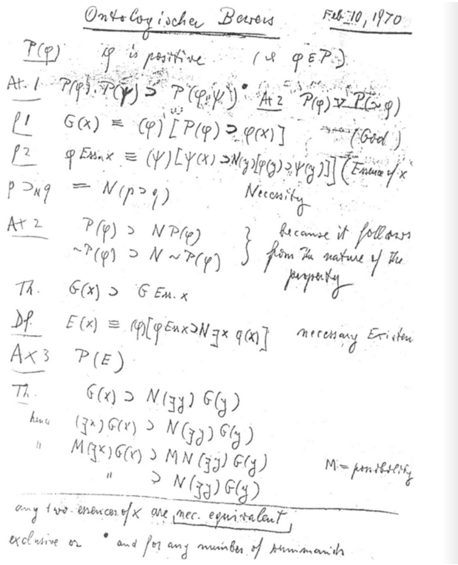
If φ positive: $(x) \, N \neg \varphi(x)$ Otherwise $\varphi(x) \supset_N x \neq x$

hence $x \neq x$, positive no $x = x$ by contrary Ax 4 or the [illegible]

dieg

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

Expressivity Matters — Inconsisency in Gödel's Ontological Argument



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 $P(\varphi) \vee P(\sim\varphi)$

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

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

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

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

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

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

Ax 3 $P(E)$

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

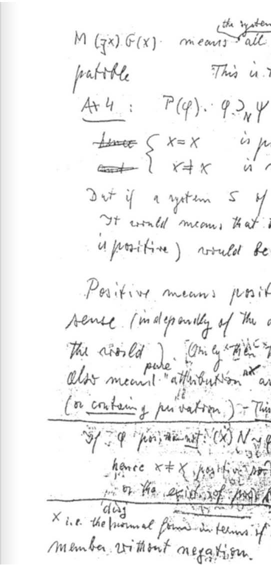
hence $(\exists x)\, G(x) \supset N(\exists y)\, G(y)$

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

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

any two essences of x are nec. equivalent

exclusive or • and for any number of summands



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

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

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

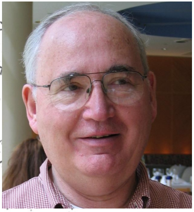
But if a system S of ...

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

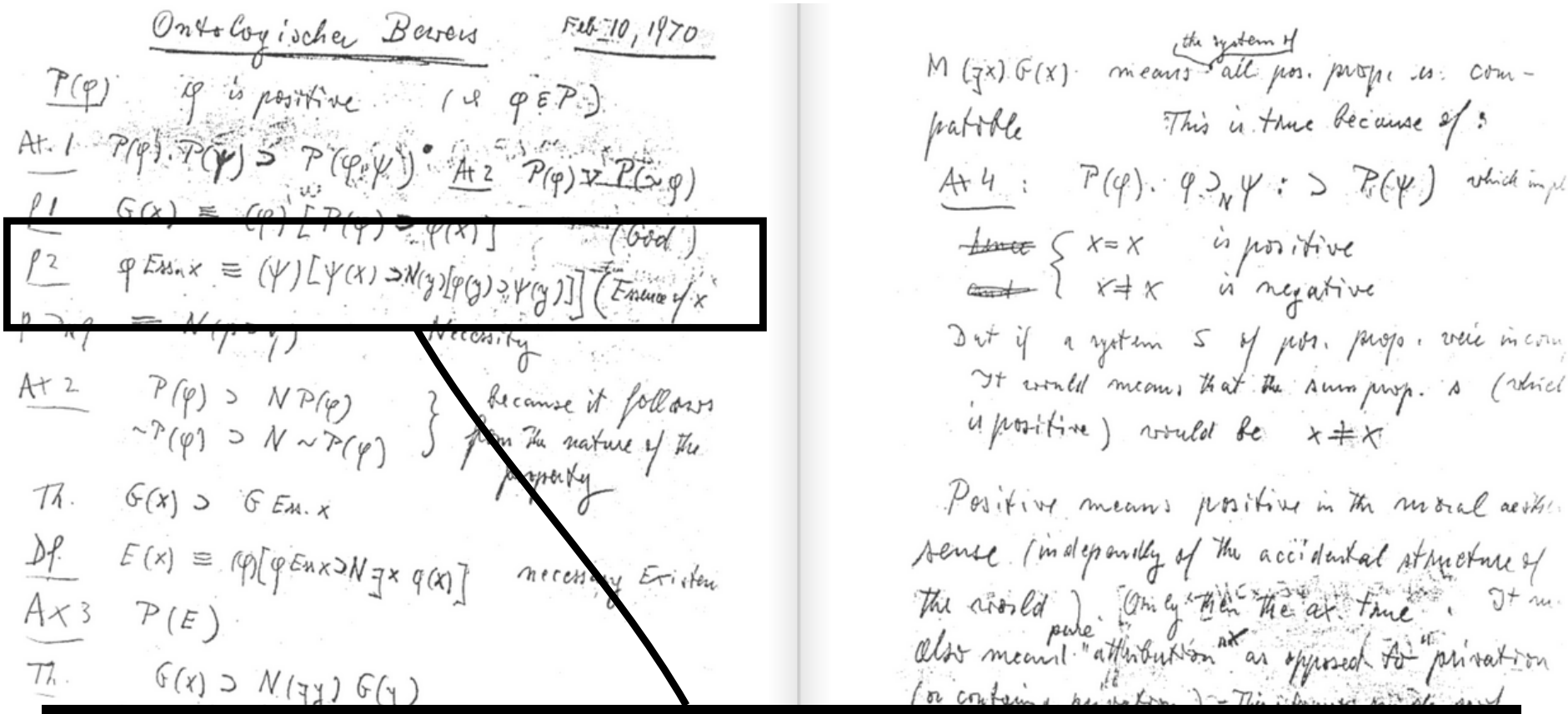
Positive means positive in the moral aesth. sense (independently of the accidental structure of the world). Only then the ax. true. It m... also means pure "attribution" as opposed to "privation" (or containing privation). This interpret. simpler proof

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

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



Expressivity Matters — Inconsisency in Gödel's Ontological Argument


Ontologischer Beweis Feb 10, 1970

P(φ) φ is positive (= φ ∈ P)

Ax. 1 P(φ) . P(ψ) ⊃ P(φ ∘ ψ) Ax 2 P(φ) ∨ P(¬φ)

D1 G(x) ≡ (φ)[P(φ) ⊃ φ(x)] (God)

D2 φ Ess. x ≡ (ψ)[ψ(x) ⊃N (y)[φ(y) ⊃ ψ(y)]] (Essence of x)

Necessity

Ax 2 P(φ) ⊃ NP(φ)
¬P(φ) ⊃ N¬P(φ)
} because it follows from the nature of the property

Th. G(x) ⊃ G Ess. x

Df. E(x) ≡ (φ)[φ Ess x ⊃N ∃x φ(x)] necessary Existen

Ax 3 P(E)

Th. G(x) ⊃ N(∃y) G(y)

M (∃x) G(x) means (the system of) all pos. propr. is: compatible This is true because of:

Ax 4: P(φ) . φ ⊃N ψ : ⊃ P(ψ) which imp

~~Axiom~~ { x = x is positive
~~Def~~ { x ≠ x is negative

But if a system S of pos. propr. were inconsistent it would mean that the sum prop. of (which is positive) would be x ≠ x

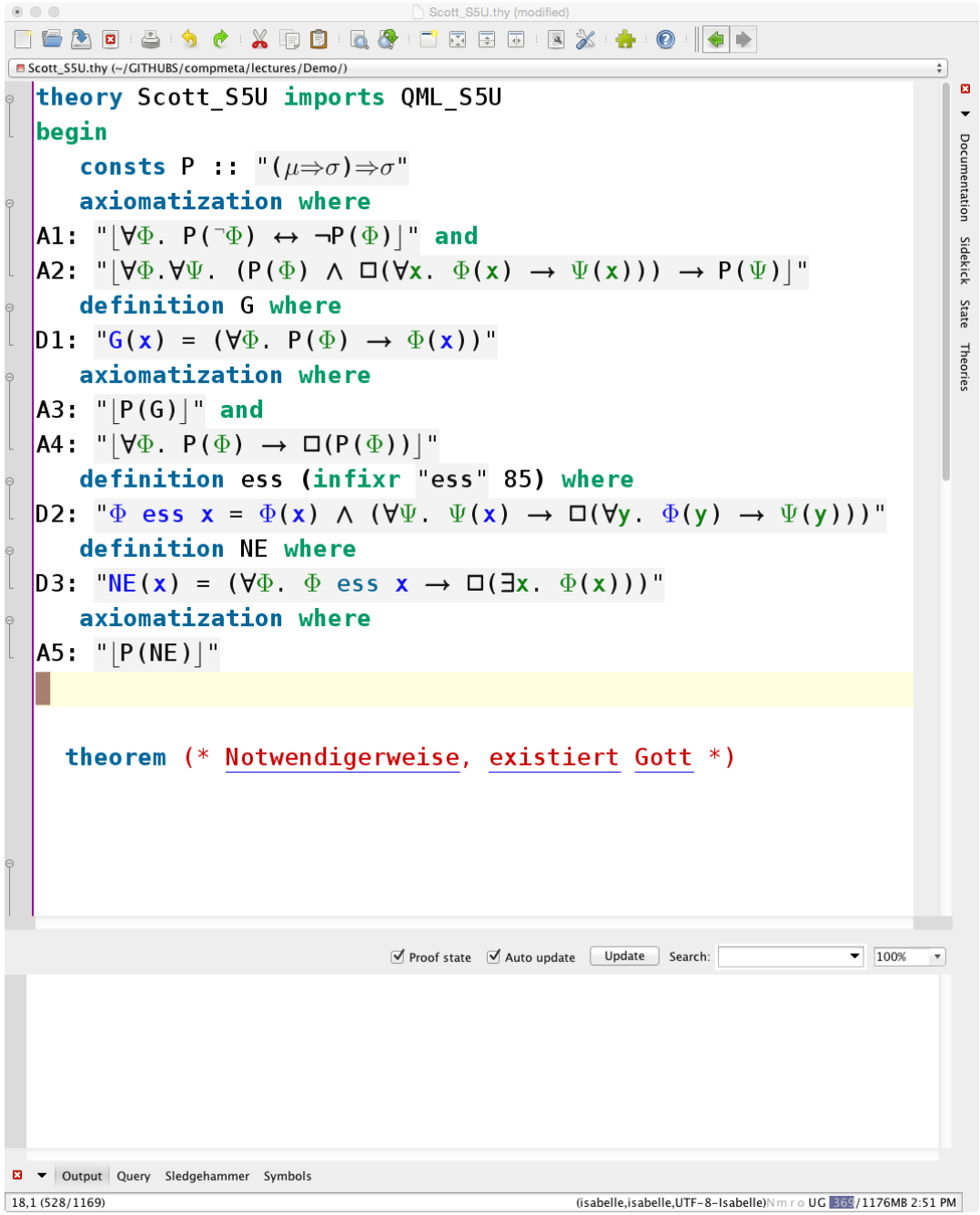
Positive means positive in the moral aesthetic sense (independently of the accidental structure of the world). Only then the Ax. true. It must also mean "attribution" as opposed to "privation" (or contains negation).

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

Gödel: Property E is Ess. of x iff all of x's properties are nec. entailed by E.

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

Expressivity Matters — Inconsisency in Gödel's Ontological Argument



Expressivity Matters — Inconsisency in Gödel's Ontological Argument

Expressivity Matters — Inconsisency in Gödel's Ontological Argument

[Benzmüller&Woltzenlogel-Paleo, IJCAI, 2016]

Def. D2* $\phi\ ess.\ x \leftrightarrow \cancel{\phi(x)} \forall\psi(\psi(x) \rightarrow \Box\forall y(\phi(y) \rightarrow \psi(y)))$

Lemma 1 The empty property is an essence of every entity. $\forall x\,(\emptyset\ ess.\ x)$

Theorem 1 Positive Properties are possibly exemplified. $\forall\phi[P(\phi) \rightarrow \Diamond\exists x\phi(x)]$

Axiom A5 $P(NE)$

- by T1, A5: $\Diamond\exists x[NE(x)]$

Def. D3 $NE(x) \leftrightarrow \forall\phi[\phi\ ess.\ x \rightarrow \Box\exists y\phi(y)]$

- $\Diamond\exists x[\forall\phi[\phi\ ess.\ x \rightarrow \Box\exists y[\phi(y)]]]$
- $\Diamond\exists x[\emptyset\ ess.\ x \rightarrow \Box\exists y[\emptyset(y)]]$
- by L1 $\Diamond\exists x[\top \rightarrow \Box\exists y[\emptyset(y)]]$
- by def. of $\emptyset$ $\Diamond\exists x[\top \rightarrow \Box\bot]$
- $\Diamond\exists x[\Box\bot]$
- $\Diamond\Box\bot$

Inconsistency $\bot$

The last step is not hard to justify semantically: we did this in the IJCAI-16 paper! A syntactical proof is not entirely trivial.

Expressivity Matters — Inconsisency in Gödel's Ontological Argument

[Benzmüller&Woltzenlogel-Paleo, IJCAI, 2016]

Def. D2* $\phi\ ess.\ x \leftrightarrow \cancel{\phi(x)} \forall\psi(\psi(x) \rightarrow \Box\forall y(\phi(y) \rightarrow \psi(y)))$

Lemma 1 The empty property is an essence of every entity. $\forall x\,(\emptyset\ ess.\ x)$

Theorem 1 Positive Properties are possibly exemplified. $\forall\phi[P(\phi) \rightarrow \Diamond\exists x\phi(x)]$

Axiom A5 $P(NE)$

- by T1, A5: $\Diamond\exists x[NE(x)]$

Def. D3 $NE(x) \leftrightarrow \forall\phi[\phi\ ess.\ x \rightarrow \Box\exists y\phi(y)]$

- $\Diamond\exists x[\forall\phi[\phi\ ess.\ x \rightarrow \Box\exists y[\phi(y)]]]$
- $\Diamond\exists x[\emptyset\ ess.\ x \rightarrow \Box\exists y[\emptyset(y)]]$
- by L1 $\Diamond\exists x[\top \rightarrow \Box\exists y[\emptyset(y)]]$
- by def. of $\emptyset$ $\Diamond\exists x[\top \rightarrow \Box\bot]$
- $\Diamond\exists x[\Box\bot]$
- $\Diamond\Box\bot$

Inconsistency $\bot$

The last step is not hard to justify semantically: we did this in the IJCAI-16 paper! A syntactical proof is not entirely trivial.

Expressivity Matters — Inconsisency in Gödel's Ontological Argument

[Benzmüller&Woltzenlogel-Paleo, IJCAI, 2016]

Def. D2* $\phi\ ess.\ x \leftrightarrow \cancel{\phi(x)} \forall\psi(\psi(x) \rightarrow \Box\forall y(\phi(y) \rightarrow \psi(y)))$

Lemma 1 The empty property is an essence of every entity. $\forall x\,(\emptyset\ ess.\ x)$

Theorem 1 Positive Properties are possibly exemplified. $\forall\phi[P(\phi) \rightarrow \Diamond\exists x\phi(x)]$

Axiom A5 $P(NE)$

- by T1, A5: $\Diamond\exists x[NE(x)]$

Def. D3 $NE(x) \leftrightarrow \forall\phi[\phi\ ess.\ x \rightarrow \Box\exists y\phi(y)]$

- $\Diamond\exists x[\forall\phi[\phi\ ess.\ x \rightarrow \Box\exists y[\phi(y)]]]$
- $\Diamond\exists x[\emptyset\ ess.\ x \rightarrow \Box\exists y[\emptyset(y)]]$
- by L1 $\Diamond\exists x[\top \rightarrow \Box\exists y[\emptyset(y)]]$
- by def. of $\emptyset$ $\Diamond\exists x[\top \rightarrow \Box\bot]$
- $\Diamond\exists x[\Box\bot]$
- $\Diamond\Box\bot$

Inconsistency $\bot$

The last step is not hard to justify semantically: we did this in the IJCAI-16 paper! A syntactical proof is not entirely trivial.

Expressivity Matters — Inconsisency in Gödel's Ontological Argument

[Benzmüller&Woltzenlogel-Paleo, IJCAI, 2016]

Def. D2* $\phi\ ess.\ x \leftrightarrow \cancel{\phi(x)} \forall\psi(\psi(x) \rightarrow \Box\forall y(\phi(y) \rightarrow \psi(y)))$

Lemma 1 The empty property is an essence of every entity. $\forall x\,(\emptyset\ ess.\ x)$

Theorem 1 Positive Properties are possibly exemplified. $\forall\phi[P(\phi) \rightarrow \Diamond\exists x\phi(x)]$

Axiom A5 $P(NE)$

- by T1, A5: $\Diamond\exists x[NE(x)]$

Def. D3 $NE(x) \leftrightarrow \forall\phi[\phi\ ess.\ x \rightarrow \Box\exists y\phi(y)]$

- $\Diamond\exists x[\forall\phi[\phi\ ess.\ x \rightarrow \Box\exists y[\phi(y)]]]$
- $\Diamond\exists x[\emptyset\ ess.\ x \rightarrow \Box\exists y[\emptyset(y)]]$
- by L1 $\Diamond\exists x[\top \rightarrow \Box\exists y[\emptyset(y)]]$
- by def. of $\emptyset$ $\Diamond\exists x[\top \rightarrow \Box\bot]$
- $\Diamond\exists x[\Box\bot]$
- $\Diamond\Box\bot$

Inconsistency $\bot$

The last step is not hard to justify semantically: we did this in the IJCAI-16 paper! A syntactical proof is not entirely trivial.

Expressivity Matters — Inconsisency in Gödel's Ontological Argument

[Benzmüller&Woltzenlogel-Paleo, IJCAI, 2016]

Def. D2* $\phi \; ess. \; x \leftrightarrow \cancel{\phi(x)} \; \forall \psi(\psi(x) \rightarrow \Box \forall y(\phi(y) \rightarrow \psi(y)))$

Lemma 1 The empty property is an essence of every entity. $\forall x \, (\emptyset \; ess. \; x)$

Theorem 1 Positive Properties are possibly exemplified. $\forall \phi [P(\phi) \rightarrow \Diamond \exists x \phi(x)]$

Axiom A5 $P(NE)$

- by T1, A5: $\Diamond \exists x [NE(x)]$

Def. D3 $NE(x) \leftrightarrow \forall \phi [\phi \; ess. \; x \rightarrow \Box \exists y \phi(y)]$

- $\Diamond \exists x [\forall \phi [\phi \; ess. \; x \rightarrow \Box \exists y [\phi(y)]]]$
- $\Diamond \exists x [\emptyset \; ess. \; x \rightarrow \Box \exists y [\emptyset(y)]]$
- by L1 $\Diamond \exists x [\top \rightarrow \Box \exists y [\emptyset(y)]]$
- by def. of $\emptyset$ $\Diamond \exists x [\top \rightarrow \Box \bot]$
- $\Diamond \exists x [\Box \bot]$
- $\Diamond \Box \bot$

Inconsistency $\bot$

The last step is not hard to justify semantically: we did this in the IJCAI-16 paper! A syntactical proof is not entirely trivial.

Expressivity Matters — Inconsisency in Gödel's Ontological Argument

[Benzmüller&Woltzenlogel-Paleo, IJCAI, 2016]

Def. D2* $\phi \; ess. \; x \leftrightarrow \cancel{\phi(x)} \; \forall \psi(\psi(x) \rightarrow \Box \forall y(\phi(y) \rightarrow \psi(y)))$

Lemma 1 The empty property is an essence of every entity. $\forall x \, (\emptyset \; ess. \; x)$

Theorem 1 Positive Properties are possibly exemplified. $\forall \phi [P(\phi) \rightarrow \Diamond \exists x \phi(x)]$

Axiom A5 $P(NE)$

- by T1, A5: $\Diamond \exists x [NE(x)]$

Def. D3 $NE(x) \leftrightarrow \forall \phi [\phi \; ess. \; x \rightarrow \Box \exists y \phi(y)]$

- $\Diamond \exists x [\forall \phi [\phi \; ess. \; x \rightarrow \Box \exists y [\phi(y)]]]$
- $\Diamond \exists x [\emptyset \; ess. \; x \rightarrow \Box \exists y [\emptyset(y)]]$
- by L1 $\Diamond \exists x [\top \rightarrow \Box \exists y [\emptyset(y)]]$
- by def. of $\emptyset$ $\Diamond \exists x [\top \rightarrow \Box \bot]$
- $\Diamond \exists x [\Box \bot]$
- $\Diamond \Box \bot$

Inconsistency $\bot$

The last step is not hard to justify semantically: we did this in the IJCAI-16 paper! A syntactical proof is not entirely trivial.

Expressivity Matters — Inconsisency in Gödel's Ontological Argument

[Benzmüller&Woltzenlogel-Paleo, IJCAI, 2016]

Def. D2* $\phi\ ess.\ x \leftrightarrow \cancel{\phi(x)} \forall\psi(\psi(x) \rightarrow \Box\forall y(\phi(y) \rightarrow \psi(y)))$

Lemma 1 The empty property is an essence of every entity. $\forall x\,(\emptyset\ ess.\ x)$

Theorem 1 Positive Properties are possibly exemplified. $\forall\phi[P(\phi) \rightarrow \Diamond\exists x\phi(x)]$

Axiom A5 $P(NE)$

- by T1, A5: $\Diamond\exists x[NE(x)]$

Def. D3 $NE(x) \leftrightarrow \forall\phi[\phi\ ess.\ x \rightarrow \Box\exists y\phi(y)]$

- $\Diamond\exists x[\forall\phi[\phi\ ess.\ x \rightarrow \Box\exists y[\phi(y)]]]$
- $\Diamond\exists x[\emptyset\ ess.\ x \rightarrow \Box\exists y[\emptyset(y)]]$
- by L1 $\Diamond\exists x[\top \rightarrow \Box\exists y[\emptyset(y)]]$
- by def. of $\emptyset$ $\Diamond\exists x[\top \rightarrow \Box\bot]$
- $\Diamond\exists x[\Box\bot]$
- $\Diamond\Box\bot$

Inconsistency $\bot$

The last step is not hard to justify semantically: we did this in the IJCAI-16 paper! A syntactical proof is not entirely trivial.

Expressivity Matters — Inconsisency in Gödel's Ontological Argument

[Benzmüller&Woltzenlogel-Paleo, IJCAI, 2016]

Def. D2* $\phi\ ess.\ x \leftrightarrow \cancel{\phi(x)} \forall\psi(\psi(x) \rightarrow \Box\forall y(\phi(y) \rightarrow \psi(y)))$

Lemma 1 The empty property is an essence of every entity. $\forall x\,(\emptyset\ ess.\ x)$

Theorem 1 Positive Properties are possibly exemplified. $\forall\phi[P(\phi) \rightarrow \Diamond\exists x\phi(x)]$

Axiom A5 $P(NE)$

- by T1, A5: $\Diamond\exists x[NE(x)]$

Def. D3 $NE(x) \leftrightarrow \forall\phi[\phi\ ess.\ x \rightarrow \Box\exists y\phi(y)]$

- $\Diamond\exists x[\forall\phi[\phi\ ess.\ x \rightarrow \Box\exists y[\phi(y)]]]$
- $\Diamond\exists x[\emptyset\ ess.\ x \rightarrow \Box\exists y[\emptyset(y)]]$
- by L1 $\Diamond\exists x[\top \rightarrow \Box\exists y[\emptyset(y)]]$
- by def. of $\emptyset$ $\Diamond\exists x[\top \rightarrow \Box\bot]$
- $\Diamond\exists x[\Box\bot]$
- $\Diamond\Box\bot$

Inconsistency $\bot$

The last step is not hard to justify semantically: we did this in the IJCAI-16 paper! A syntactical proof is not entirely trivial.

Expressivity Matters — Inconsisency in Gödel's Ontological Argument

[Benzmüller&Woltzenlogel-Paleo, IJCAI, 2016]

Def. D2*		$\phi \; ess. \; x \leftrightarrow \cancel{\phi(x)} \; \forall \psi(\psi(x) \rightarrow \Box \forall y(\phi(y) \rightarrow \psi(y)))$
Lemma 1	The empty property is an essence of every entity.	$\forall x \, (\emptyset \; ess. \; x)$
Theorem 1	Positive Properties are possibly exemplified.	$\forall \phi[P(\phi) \rightarrow \Diamond \exists x \phi(x)]$
Axiom A5		$P(NE)$
▶	by T1, A5:	$\Diamond \exists x[NE(x)]$
Def. D3		$NE(x) \leftrightarrow \forall \phi[\phi \; ess. \; x \rightarrow \Box \exists y \phi(y)]$
▶		$\Diamond \exists x[\forall \phi[\phi \; ess. \; x \rightarrow \Box \exists y[\phi(y)]]]$
▶		$\Diamond \exists x[\emptyset \; ess. \; x \rightarrow \Box \exists y[\emptyset(y)]]$
▶	by L1	$\Diamond \exists x[\top \rightarrow \Box \exists y[\emptyset(y)]]$
▶	by def. of $\emptyset$	$\Diamond \exists x[\top \rightarrow \Box \bot]$
▶		$\Diamond \exists x[\Box \bot]$
▶		$\Diamond \Box \bot$
Inconsistency		$\bot$

The last step is not hard to justify semantically: we did this in the IJCAI-16 paper!
A syntactical proof is not entirely trivial.

Expressivity Matters — Inconsisency in Gödel's Ontological Argument

[Benzmüller&Woltzenlogel-Paleo, IJCAI, 2016]

Def. D2*		$\phi \; ess. \; x \leftrightarrow \cancel{\phi(x)} \; \forall \psi(\psi(x) \rightarrow \Box \forall y(\phi(y) \rightarrow \psi(y)))$
Lemma 1	The empty property is an essence of every entity.	$\forall x \, (\emptyset \; ess. \; x)$
Theorem 1	Positive Properties are possibly exemplified.	$\forall \phi[P(\phi) \rightarrow \Diamond \exists x \phi(x)]$
Axiom A5		$P(NE)$
▶	by T1, A5:	$\Diamond \exists x[NE(x)]$
Def. D3		$NE(x) \leftrightarrow \forall \phi[\phi \; ess. \; x \rightarrow \Box \exists y \phi(y)]$
▶		$\Diamond \exists x[\forall \phi[\phi \; ess. \; x \rightarrow \Box \exists y[\phi(y)]]]$
▶		$\Diamond \exists x[\emptyset \; ess. \; x \rightarrow \Box \exists y[\emptyset(y)]]$
▶	by L1	$\Diamond \exists x[\top \rightarrow \Box \exists y[\emptyset(y)]]$
▶	by def. of $\emptyset$	$\Diamond \exists x[\top \rightarrow \Box \bot]$
▶		$\Diamond \exists x[\Box \bot]$
▶		$\Diamond \Box \bot$
Inconsistency		$\bot$

The last step is not hard to justify semantically: we did this in the IJCAI-16 paper! A syntactical proof is not entirely trivial.

Expressivity Matters — Inconsisency in Gödel's Ontological Argument

[Benzmüller&Woltzenlogel-Paleo, IJCAI, 2016]

Def. D2*		$\phi \; ess. \; x \leftrightarrow \cancel{\phi(x)} \; \forall \psi(\psi(x) \rightarrow \Box \forall y(\phi(y) \rightarrow \psi(y)))$
Lemma 1	The empty property is an essence of every entity.	$\forall x \, (\emptyset \; ess. \; x)$
Theorem 1	Positive Properties are possibly exemplified.	$\forall \phi [P(\phi) \rightarrow \Diamond \exists x \phi(x)]$
Axiom A5		$P(NE)$
▶	by T1, A5:	$\Diamond \exists x [NE(x)]$
Def. D3		$NE(x) \leftrightarrow \forall \phi [\phi \; ess. \; x \rightarrow \Box \exists y \phi(y)]$
▶		$\Diamond \exists x [\forall \phi [\phi \; ess. \; x \rightarrow \Box \exists y [\phi(y)]]]$
▶		$\Diamond \exists x [\emptyset \; ess. \; x \rightarrow \Box \exists y [\emptyset(y)]]$
▶	by L1	$\Diamond \exists x [\top \rightarrow \Box \exists y [\emptyset(y)]]$
▶	by def. of $\emptyset$	$\Diamond \exists x [\top \rightarrow \Box \bot]$
▶		$\Diamond \exists x [\Box \bot]$
▶		$\Diamond \Box \bot$
Inconsistency		$\bot$

The last step is not hard to justify semantically: we did this in the IJCAI-16 paper!
A syntactical proof is not entirely trivial.

Expressivity Matters — Inconsisency in Gödel's Ontological Argument

[Benzmüller&Woltzenlogel-Paleo, IJCAI, 2016]

Def. D2*		$\phi \; ess. \; x \leftrightarrow \cancel{\phi(x)} \; \forall \psi(\psi(x) \rightarrow \Box \forall y(\phi(y) \rightarrow \psi(y)))$
Lemma 1	The empty property is an essence of every entity.	$\forall x \, (\emptyset \; ess. \; x)$
Theorem 1	Positive Properties are possibly exemplified.	$\forall \phi [P(\phi) \rightarrow \Diamond \exists x \phi(x)]$
Axiom A5		$P(NE)$
▶	by T1, A5:	$\Diamond \exists x [NE(x)]$
Def. D3		$NE(x) \leftrightarrow \forall \phi [\phi \; ess. \; x \rightarrow \Box \exists y \phi(y)]$
▶		$\Diamond \exists x [\forall \phi [\phi \; ess. \; x \rightarrow \Box \exists y [\phi(y)]]]$
▶		$\Diamond \exists x [\emptyset \; ess. \; x \rightarrow \Box \exists y [\emptyset(y)]]$
▶	by L1	$\Diamond \exists x [\top \rightarrow \Box \exists y [\emptyset(y)]]$
▶	by def. of $\emptyset$	$\Diamond \exists x [\top \rightarrow \Box \bot]$
▶		$\Diamond \exists x [\Box \bot]$
▶		$\Diamond \Box \bot$
Inconsistency		$\bot$

The last step is not hard to justify semantically: we did this in the IJCAI-16 paper!
A syntactical proof is not entirely trivial.

Expressivity Matters — Inconsistency in Gödel's Ontological Argument

[Benzmüller&Woltzenlogel-Paleo, IJCAI, 2016]

Def. D2*		$\phi \; ess. \; x \leftrightarrow \cancel{\phi(x)} \; \forall \psi(\psi(x) \rightarrow \Box \forall y(\phi(y) \rightarrow \psi(y)))$
Lemma 1	The empty property is an essence of every entity.	$\forall x \, (\emptyset \; ess. \; x)$
Theorem 1	Positive Properties are possibly exemplified.	$\forall \phi [P(\phi) \rightarrow \Diamond \exists x \phi(x)]$
Axiom A5		$P(NE)$
▶	by T1, A5:	$\Diamond \exists x [NE(x)]$
Def. D3		$NE(x) \leftrightarrow \forall \phi [\phi \; ess. \; x \rightarrow \Box \exists y \phi(y)]$
▶		$\Diamond \exists x [\forall \phi [\phi \; ess. \; x \rightarrow \Box \exists y [\phi(y)]]]$
▶		$\Diamond \exists x [\emptyset \; ess. \; x \rightarrow \Box \exists y [\emptyset(y)]]$
▶	by L1	$\Diamond \exists x [\top \rightarrow \Box \exists y [\emptyset(y)]]$
▶	by def. of $\emptyset$	$\Diamond \exists x [\top \rightarrow \Box \bot]$
▶		$\Diamond \exists x [\Box \bot]$
▶		$\Diamond \Box \bot$
Inconsistency		$\bot$

The last step is not hard to justify semantically: we did this in the IJCAI-16 paper! A syntactical proof is not entirely trivial.

Expressivity Matters — Inconsisency in Gödel's Ontological Argument

$\Diamond\Box\bot$ leads to Inconsistency: Syntactical Proof in Modal Logic K

- Assume: $\Diamond\Box\bot$ (holds globally)
- Show: there exist a formula φ (in current world) s.t. φ and $\neg\varphi$
- Choose $\varphi := \Diamond\Diamond\bot$
- $\Diamond\Diamond\bot$ follows from $\Diamond\Box\bot$

- $\neg\Diamond\Diamond\bot$ holds: $\neg\Diamond\Diamond\bot \equiv \Box\Box\top$, which easily follows by necessitation
- Hence, $\Diamond\Diamond\bot$ and $\neg\Diamond\Diamond\bot$
- Hence, $\bot$

Expressivity Matters — Inconsisency in Gödel's Ontological Argument

$\Diamond\Box\bot$ leads to Inconsistency: Syntactical Proof in Modal Logic K

- Assume: $\Diamond\Box\bot$ (holds globally)
- Show: there exist a formula φ (in current world) s.t. φ and $\neg\varphi$
- Choose $\varphi := \Diamond\Diamond\bot$
- $\Diamond\Diamond\bot$ follows from $\Diamond\Box\bot$

- $\neg\Diamond\Diamond\bot$ holds: $\neg\Diamond\Diamond\bot \equiv \Box\Box\top$, which easily follows by necessitation
- Hence, $\Diamond\Diamond\bot$ and $\neg\Diamond\Diamond\bot$
- Hence, $\bot$

Expressivity Matters — Inconsisency in Gödel's Ontological Argument

$\Diamond\Box\bot$ leads to Inconsistency: Syntactical Proof in Modal Logic K

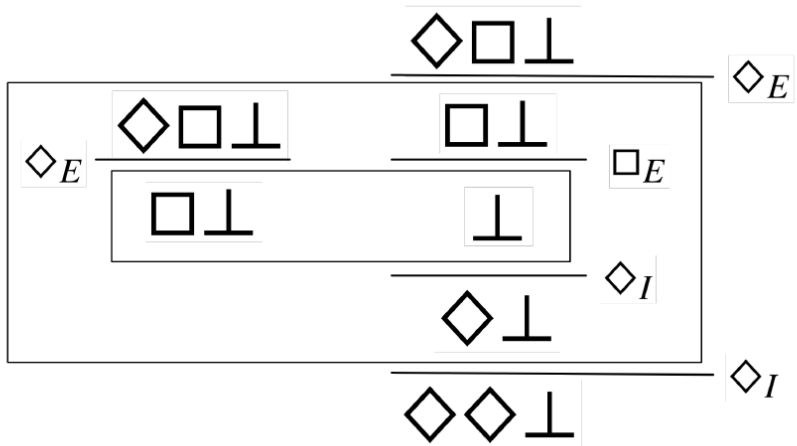
- Assume: $\Diamond\Box\bot$ (holds globally)
- Show: there exist a formula φ (in current world) s.t. φ and $\neg\varphi$
- Choose $\varphi := \Diamond\Diamond\bot$
- $\Diamond\Diamond\bot$ follows from $\Diamond\Box\bot$

- $\neg\Diamond\Diamond\bot$ holds: $\neg\Diamond\Diamond\bot \equiv \Box\Box\top$, which easily follows by necessitation
- Hence, $\Diamond\Diamond\bot$ and $\neg\Diamond\Diamond\bot$
- Hence, $\bot$

Expressivity Matters — Inconsisency in Gödel's Ontological Argument

$\Diamond\Box\bot$ leads to Inconsistency: Syntactical Proof in Modal Logic K

- Assume: $\Diamond\Box\bot$ (holds globally)
- Show: there exist a formula φ (in current world) s.t. φ and $\neg\varphi$
- Choose $\varphi := \Diamond\Diamond\bot$
- $\Diamond\Diamond\bot$ follows from $\Diamond\Box\bot$

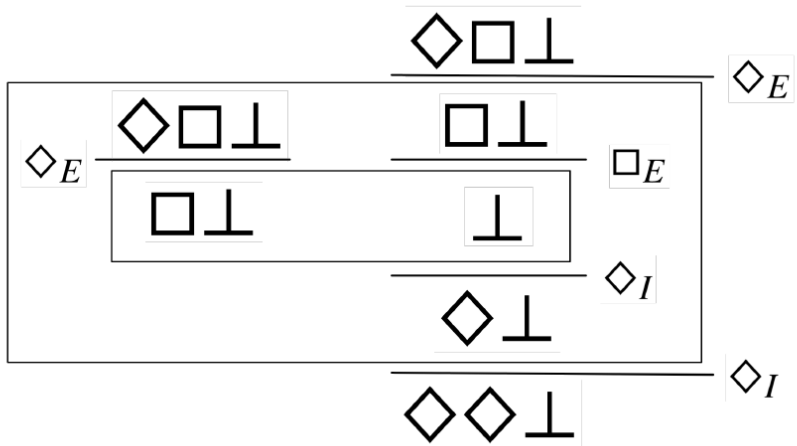


- $\neg\Diamond\Diamond\bot$ holds: $\neg\Diamond\Diamond\bot \equiv \Box\Box\top$, which easily follows by necessitation
- Hence, $\Diamond\Diamond\bot$ and $\neg\Diamond\Diamond\bot$
- Hence, $\bot$

Expressivity Matters — Inconsisency in Gödel's Ontological Argument

$\Diamond\Box\bot$ leads to Inconsistency: Syntactical Proof in Modal Logic K

- Assume: $\Diamond\Box\bot$ (holds globally)
- Show: there exist a formula φ (in current world) s.t. φ and $\neg\varphi$
- Choose $\varphi := \Diamond\Diamond\bot$
- $\Diamond\Diamond\bot$ follows from $\Diamond\Box\bot$

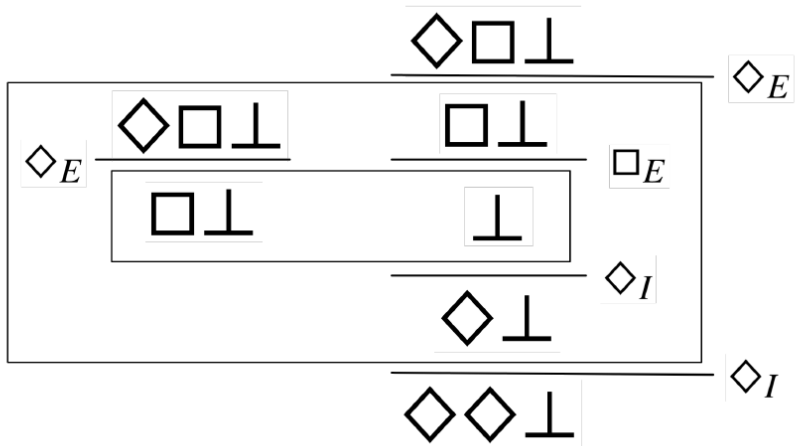


- $\neg\Diamond\Diamond\bot$ holds: $\neg\Diamond\Diamond\bot \equiv \Box\Box\top$, which easily follows by necessitation
- Hence, $\Diamond\Diamond\bot$ and $\neg\Diamond\Diamond\bot$
- Hence, $\bot$

Expressivity Matters — Inconsisency in Gödel's Ontological Argument

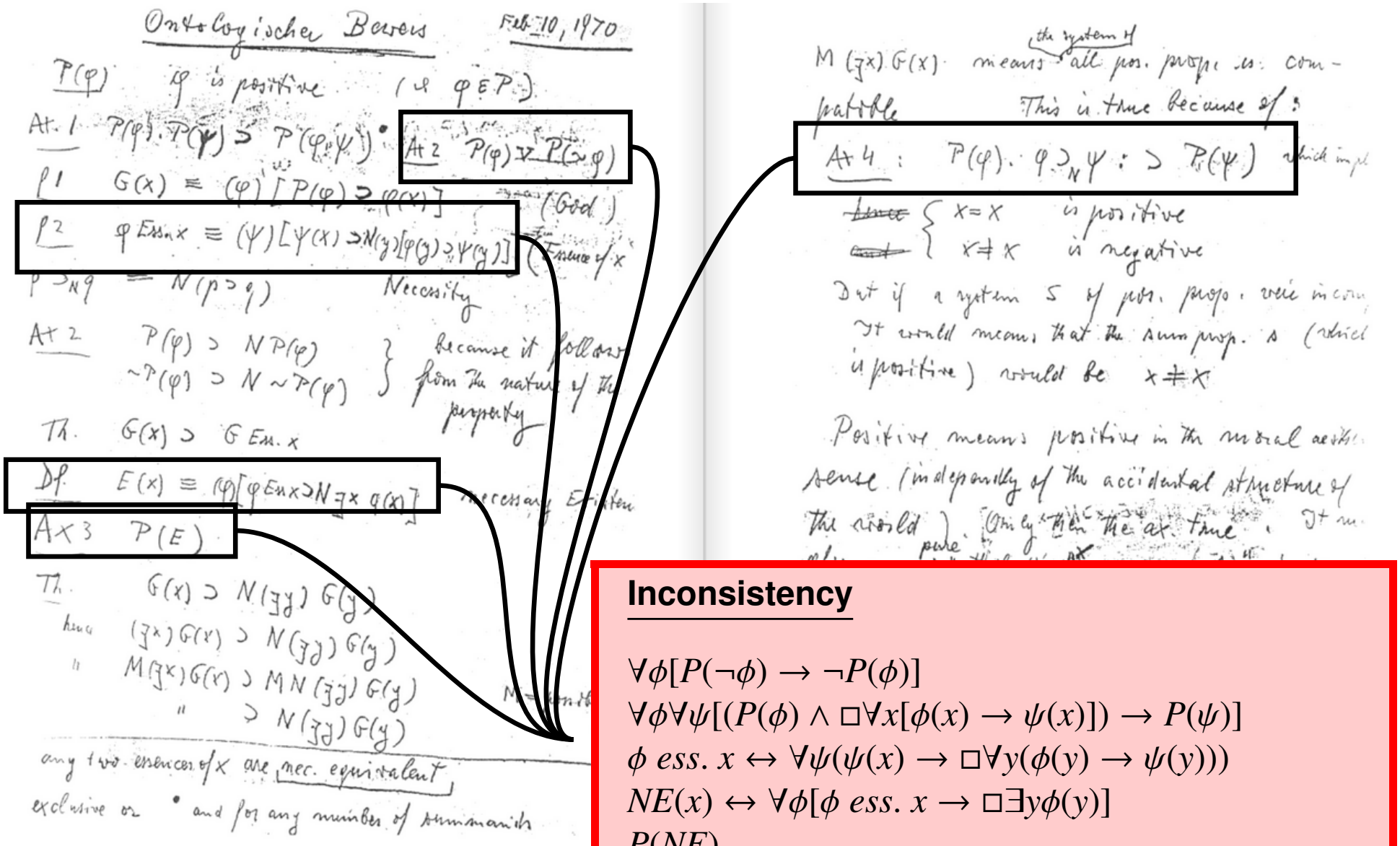
$\Diamond\Box\bot$ leads to Inconsistency: Syntactical Proof in Modal Logic K

- Assume: $\Diamond\Box\bot$ (holds globally)
- Show: there exist a formula φ (in current world) s.t. φ and $\neg\varphi$
- Choose $\varphi := \Diamond\Diamond\bot$
- $\Diamond\Diamond\bot$ follows from $\Diamond\Box\bot$



- $\neg\Diamond\Diamond\bot$ holds: $\neg\Diamond\Diamond\bot \equiv \Box\Box\top$, which easily follows by necessitation
- Hence, $\Diamond\Diamond\bot$ and $\neg\Diamond\Diamond\bot$
- Hence, $\bot$

Expressivity Matters — Inconsisency in Gödel's Ontological Argument



Inconsistency

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

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

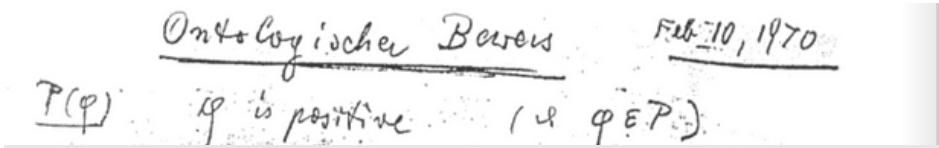
$\phi\ ess.\ x \leftrightarrow \forall\psi(\psi(x) \rightarrow \Box\forall y(\phi(y) \rightarrow \psi(y)))$

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

$P(NE)$

Expressivity Matters — Inconsisency in Gödel's Ontological Argument

The Heureka Step



DemoMaterial — bash — 166×52

```
@SV8)@SV3)=$false) | (((p@(^[SX0:mu,SX1:$i]: $false))@SV3)=$true))),inference(prim_subst,[status(thm)],[66:[bind(SV11,$thf(^[SV23:mu,SV24:$i]: $false))]])).
 thf(84,plain,(![SV22:(mu>($i>$o)),SV3:$i,SV8:(mu>($i>$o))]: ((((SV8@(((sK2_SY33@SV3)@(^[SX0:mu,SX1:$i]: (~ ((SV22@SX0)@SX1))))@SV8))@(((sK1_SY31@(^[SX0:mu,SX1:$i]: (
~ ((SV22@SX0)@SX1))))@SV8)@SV3))=$true) | (((p@SV8)@SV3)=$false) | (((p@(^[SX0:mu,SX1:$i]: (~ ((SV22@SX0)@SX1))))@SV3)=$true))),inference(prim_subst,[status(thm)],[66
:[bind(SV11,$thf(^[SV20:mu,SV21:$i]: (~ ((SV22@SV20)@SV21))))]])).
 thf(85,plain,(![SV4:$i,SV9:(mu>($i>$o))]: ((((p@(^[SY27:mu,SY28:$i]: (~ ((SV9@SY27)@SY28))))@SV4)=$false) | ((((p@SV9)@SV4) = ((p@(^[SY27:mu,SY28:$i]: (~ ((SV9@SY27)
@SY28))))@SV4))=$false))),inference(fac_restr,[status(thm)],[56])).
 thf(86,plain,(![SV4:$i,SV9:(mu>($i>$o))]: ((((p@(^[SY29:mu,SY30:$i]: (~ ((SV9@SY29)@SY30))))@SV4)=$true) | ((((p@SV9)@SV4) = ((p@(^[SY29:mu,SY30:$i]: (~ ((SV9@SY29)@
SY30))))@SV4))=$false))),inference(fac_restr,[status(thm)],[57])).
 thf(87,plain,(![SV4:$i,SV9:(mu>($i>$o))]: ((((~ (((p@SV9)@SV4) | ((p@(^[SY27:mu,SY28:$i]: (~ ((SV9@SY27)@SY28))))@SV4))) | (~ ((~ ((p@SV9)@SV4)) | (~ ((p@(^[SY27:mu,
SY28:$i]: (~ ((SV9@SY27)@SY28))))@SV4)))))=$false) | (((p@(^[SY27:mu,SY28:$i]: (~ ((SV9@SY27)@SY28))))@SV4)=$false))),inference(extcnf_equal_neg,[status(thm)],[85])).
 thf(89,plain,(![SV4:$i,SV9:(mu>($i>$o))]: ((((~ (((p@SV9)@SV4) | ((p@(^[SY29:mu,SY30:$i]: (~ ((SV9@SY29)@SY30))))@SV4))) | (~ ((~ ((p@SV9)@SV4)) | (~ ((p@(^[SY29:mu,
SY30:$i]: (~ ((SV9@SY29)@SY30))))@SV4)))))=$false) | (((p@(^[SY29:mu,SY30:$i]: (~ ((SV9@SY29)@SY30))))@SV4)=$true))),inference(extcnf_equal_neg,[status(thm)],[86])).
 thf(92,plain,(![SV4:$i,SV9:(mu>($i>$o))]: (((~ ((~ ((p@SV9)@SV4)) | (~ ((p@(^[SY27:mu,SY28:$i]: (~ ((SV9@SY27)@SY28))))@SV4))))=$false) | (((p@(^[SY27:mu,SY28:$i]: (
~ ((SV9@SY27)@SY28))))@SV4)=$false))),inference(extcnf_or_neg,[status(thm)],[87])).
 thf(93,plain,(![SV4:$i,SV9:(mu>($i>$o))]: (((~ (((p@SV9)@SV4) | ((p@(^[SY29:mu,SY30:$i]: (~ ((SV9@SY29)@SY30))))@SV4)))=$false) | (((p@(^[SY29:mu,SY30:$i]: (~ ((SV9@
SY29)@SY30))))@SV4)=$true))),inference(extcnf_or_neg,[status(thm)],[89])).
 thf(96,plain,(![SV4:$i,SV9:(mu>($i>$o))]: ((((~ ((p@SV9)@SV4)) | (~ ((p@(^[SY27:mu,SY28:$i]: (~ ((SV9@SY27)@SY28))))@SV4)))=$true) | (((p@(^[SY27:mu,SY28:$i]: (~ ((S
V9@SY27)@SY28))))@SV4)=$false))),inference(extcnf_not_neg,[status(thm)],[92])).
 thf(97,plain,(![SV4:$i,SV9:(mu>($i>$o))]: (((((p@SV9)@SV4) | ((p@(^[SY29:mu,SY30:$i]: (~ ((SV9@SY29)@SY30))))@SV4))=$true) | (((p@(^[SY29:mu,SY30:$i]: (~ ((SV9@SY29)
@SY30))))@SV4)=$true))),inference(extcnf_not_neg,[status(thm)],[93])).
 thf
 (~
 thf
9@SY
 thf(103,plain,(![SV4:$i,SV9:(mu>($i>$o))]: ((((p@SV9)@SV4)=$false) | ((~ ((p@(^[SY27:mu,SY28:$i]: (~ ((SV9@SY27)@SY28))))@SV4))=$true) | (((p@(^[SY27:mu,SY28:$i]: (~
 ((SV9@SY27)@SY28))))@SV4)=$false))),inference(extcnf_not_pos,[status(thm)],[100])).
 thf(105,plain,(![SV4:$i,SV9:(mu>($i>$o))]: ((((p@(^[SY27:mu,SY28:$i]: (~ ((SV9@SY27)@SY28))))@SV4)=$false) | (((p@SV9)@SV4)=$false) | (((p@(^[SY27:mu,SY28:$i]: (~ ((
SV9@SY27)@SY28))))@SV4)=$false))),inference(extcnf_not_pos,[status(thm)],[103])).
 thf(107,plain,(![SV8:(mu>($i>$o)),SV3:$i,SV22:(mu>($i>$o))]: ((((SV22@(((sK2_SY33@SV3)@(^[SX0:mu,SX1:$i]: (~ ((SV22@SX0)@SX1))))@SV8))@(((sK1_SY31@(^[SX0:mu,SX1:$i]:
 (~ ((SV22@SX0)@SX1))))@SV8)@SV3))=$true) | (((p@SV8)@SV3)=$false) | (((p@(^[SX0:mu,SX1:$i]: (~ ((SV22@SX0)@SX1))))@SV3)=$true))),inference(extcnf_not_neg,[status(thm
)],[78])).
 thf
SX1:
s(th
 thf
5])).
 thf(110,plain,(![SV4:$i,SV9:(mu>($i>$o))]: ((((p@SV9)@SV4)=$true) | (((p@(^[SY29:mu,SY30:$i]: (~ ((SV9@SY29)@SY30))))@SV4)=$true))),inference(sim,[status(thm)],[101]
)).
 thf(111,plain,(![SV3:$i,SV8:(mu>($i>$o))]: ((((p@SV8)@SV3)=$false) | (((p@(^[SX0:mu,SX1:$i]: $true))@SV3)=$true))),inference(sim,[status(thm)],[76])).
 thf(112,plain,(![SV11:(mu>($i>$o)),SV3:$i]: ((((p@(^[SX0:mu,SX1:$i]: $false))@SV3)=$false) | (((p@SV11)@SV3)=$true))),inference(sim,[status(thm)],[80])).
 thf(113,plain,(((($false)=$true)),inference(fo_atp_e,[status(thm)],[25,112,111,110,109,108,107,84,83,82,75,74,73,72,71,70,69,68,67,66,65,62,57,56,51,42,29])).
 thf(114,plain,($false),inference(solved_all_splits,[solved_all_splits(join,[])],[113])).
% SZS output end CNFRefutation

%%%%% End of derivation protocol %%%%%
%%%%% no. of clauses in derivation: 97 %%%%%
%%%%% clause counter: 113 %%%%%
% SZS status Unsatisfiable for ConsistencyWithoutFirstConjunctinD2.p : (rf:0,axioms:6,ps:3,u:6,ude:false,rLeibEQ:true,rAndEQ:true,use_choice:true,use_extuni:true,use_
```

Lemma 1 The empty property is an essence of every entity. $\forall x\,(\emptyset\ ess.\ x)$

In Leo: Primitive substitution with $\lambda X.X \neq X$ resp. $\lambda X.\bot$

Expressivity Matters — My Claim

’Heureka’ steps =?= primitive substitutions

Shouldn’t we focus more on exactly this challenge issue?

Universal (Meta-)Logical Reasoning

[Universal (Meta-)Logical Reasoning: Recent Successes, SCP, 2019]
[Universal (Meta-)Logical Reasoning: The Wise Men Puzzle, Data in Brief, 2019]
[Automating Free Logic in HOL, with an . . . , JAR, 2019]

Expressivity Matters — Higher-Order Modal Logic

$\Box P$

P is necessary

$\Diamond P$

P is possible

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

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

Expressivity Matters — Higher-Order Modal Logic

$\Box P$

P is necessary

$\Diamond P$

P is possible

□ and ◇ are not truth-functional

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

Expressivity Matters — Higher-Order Modal Logic

$\Box P$

P is necessary

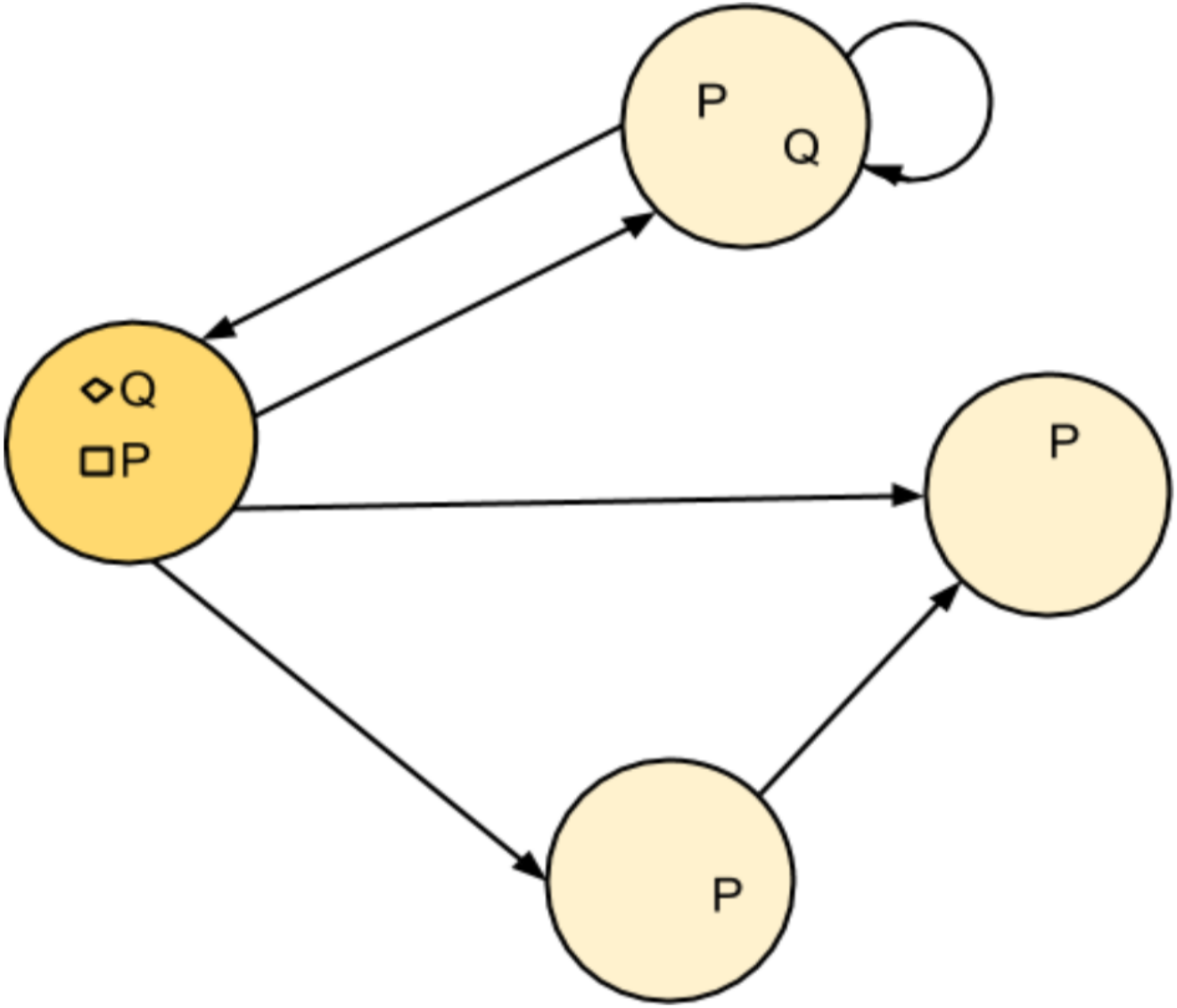
$\Diamond P$

P is possible

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

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

Expressivity Matters — Higher-Order Modal Logic in HOL



Expressivity Matters — Higher-Order Modal Logic in HOL — Lambda's at Work

Standard translation:

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

Standard translation extended to quantifiers:

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

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

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

Expressivity Matters — Higher-Order Modal Logic in HOL — Lambda's at Work

Standard translation:

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

Standard translation extended to quantifiers:

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

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

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

Expressivity Matters — Higher-Order Modal Logic in HOL — Lambda's at Work

Standard translation:

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

Standard translation extended to quantifiers:

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

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

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

Expressivity Matters — Higher-Order Modal Logic in HOL — Lambda's at Work

Standard translation:

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

Standard translation extended to quantifiers:

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

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

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

Expressivity Matters — Higher-Order Modal Logic in HOL — Lambda's at Work

Standard translation:

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

Standard translation extended to quantifiers:

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

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

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

Expressivity Matters — Higher-Order Modal Logic in HOL — Lambda's at Work

Standard translation:

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

Standard translation extended to quantifiers:

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

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

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

Expressivity Matters — Higher-Order Modal Logic in HOL — Lambda's at Work

Standard translation:

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

Standard translation extended to quantifiers:

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

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

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

Expressivity Matters — Higher-Order Modal Logic in HOL — Lambda's at Work

Standard translation:

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

Standard translation extended to quantifiers:

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

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

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

Expressivity Matters — Higher-Order Modal Logic in HOL — Lambda's at Work

Standard translation:

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

Standard translation extended to quantifiers:

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

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

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

Expressivity Matters — Higher-Order Modal Logic in HOL — Lambda's at Work

Standard translation:

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

Standard translation extended to quantifiers:

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

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

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

Expressivity Matters — Higher-Order Modal Logic in HOL — Lambda's at Work

Standard translation:

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

Standard translation extended to quantifiers:

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

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

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

Standard translation:

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

Standard translation extended to quantifiers:

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

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

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

Expressivity Matters — Higher-Order Modal Logic in HOL — Lambda's at Work

Standard translation:

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

Standard translation extended to quantifiers:

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

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

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

Expressivity Matters — Higher-Order Modal Logic in HOL

Isabelle2018/HOL - IHOML.thy

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

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

Expressivity Matters — Higher-Order Modal Logic in HOL

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

Axiom A property necessarily implied by a positive property is positive: $\forall\phi\forall\psi[(P(\phi) \wedge \Box\forall x[\phi(x) \rightarrow \psi(x)]) \rightarrow P(\psi)]$

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

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

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

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

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

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

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

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

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

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

Expressivity Matters — Higher-Order Modal Logic in HOL

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

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

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

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

Axiom $P(G)$

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

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

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

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

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

Axiom $P(NE)$

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

Expressivity Matters — Higher-Order Modal Logic in HOL

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

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

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

Axiom $P(G)$

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

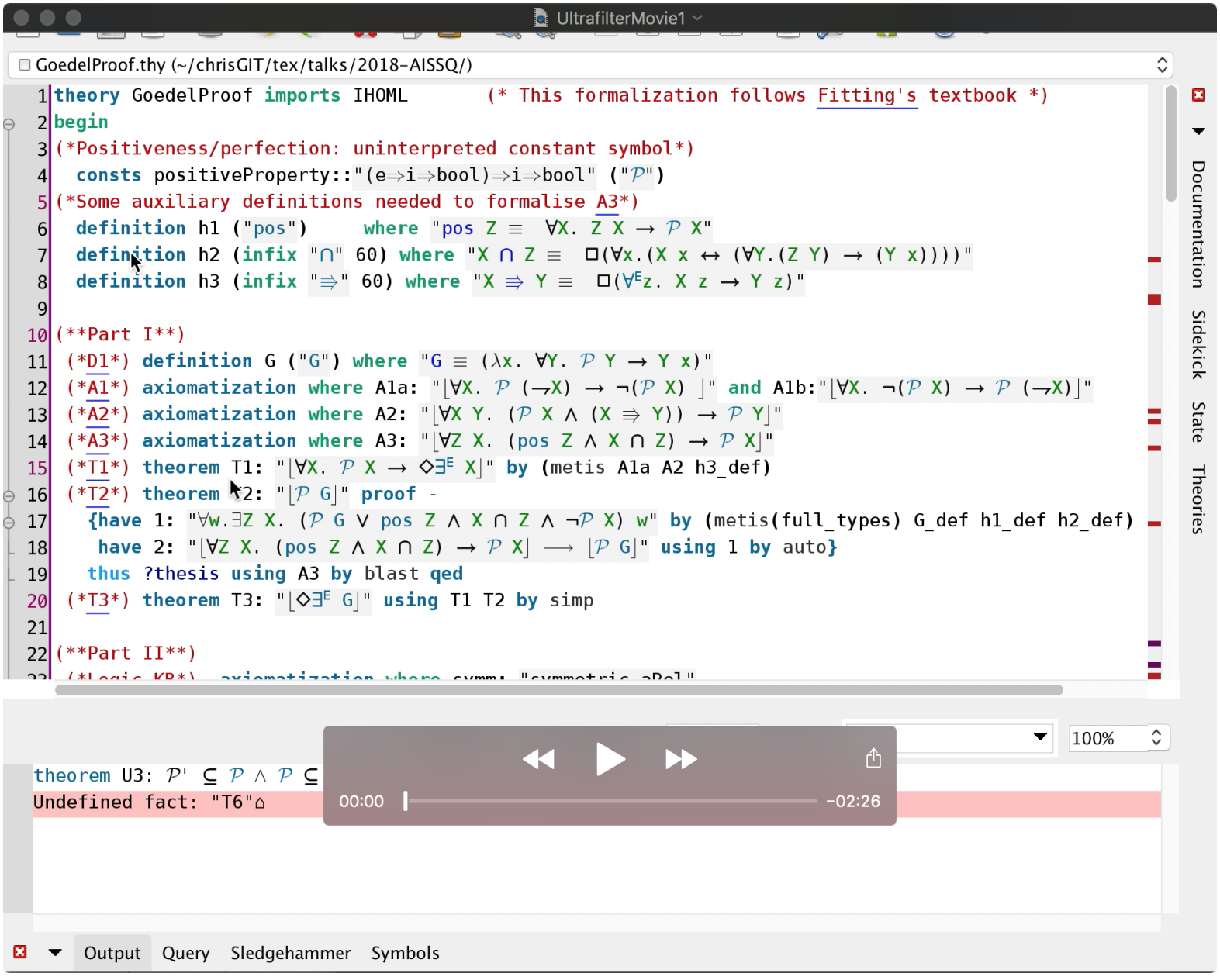
Def. $\phi\ ess.\ x \leftrightarrow \phi(x)\ \wedge\ \forall\psi(\psi(x) \rightarrow \Box\forall y(\phi(y) \rightarrow \psi(y)))$

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

Axiom $P(NE)$

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

Expressivity Matters — Higher-Order Modal Logic in HOL



Contents lists available at ScienceDirect

Data in brief

journal homepage: www.elsevier.com/locate/dib

Data Article

Universal (meta-)logical reasoning: The Wise Men Puzzle (Isabelle/HOL dataset)

Christoph Benzmüller [a, b, *]

[a] Freie Universität Berlin, Germany
[b] University of Luxembourg, Luxembourg

ARTICLE INFO

Article history:
Received 20 November 2018
Received in revised form 22 February 2019
Accepted 4 March 2019
Available online 20 March 2019

Keywords:
Universal logical reasoning
Automated theorem proving
Higher-order logic

ABSTRACT

The authors universal (meta-)logical reasoning approach is demonstrated and assessed with a prominent riddle in epistemic reasoning: the Wise Men Puzzle. The presented solution puts a particular emphasis on the adequate modeling of common knowledge and it illustrates the elegance and the practical relevance of the shallow semantical embedding approach when utilized within modern proof assistant systems such as Isabelle/HOL. The contributed dataset provides supporting evidence for claims made in the article "Universal (meta-)logical reasoning: Recent successes" (Benzmüller, 2019).

Expressivity Matters — Higher-Order Multimodal Logic in HOL

Contents lists available at ScienceDirect

Data in brief

journal homepage: www.elsevier.com/locate/dib

Data Article

Universal (meta-)logical reasoning: The Wise Men Puzzle (Isabelle/HOL dataset)

Christoph Benzmüller [a, b, *]

```
definition mbox :: "α⇒σ⇒σ" ("□_ _") where "□r φ ≡ (λw. ∀v. r w v ⟶ φ v)"
definition mdia :: "α⇒σ⇒σ" ("◇_ _") where "◇r φ ≡ (λw. ∃v. r w v ∧ φ v)"
```

ARTICLE INFO

Article history:
Received 20 November 2018

ABSTRACT

The authors universal (meta-)logical reasoning approach is

```
(*In HOL the transitive closure of a relation can be defined in a single line.*)
definition tc :: "α⇒α" where "tc R ≡ λx y.∀Q. transitive Q ⟶ (sub_rel R Q ⟶ Q x y)"
```

Universal logical reasoning
Automated theorem proving
Higher-order logic

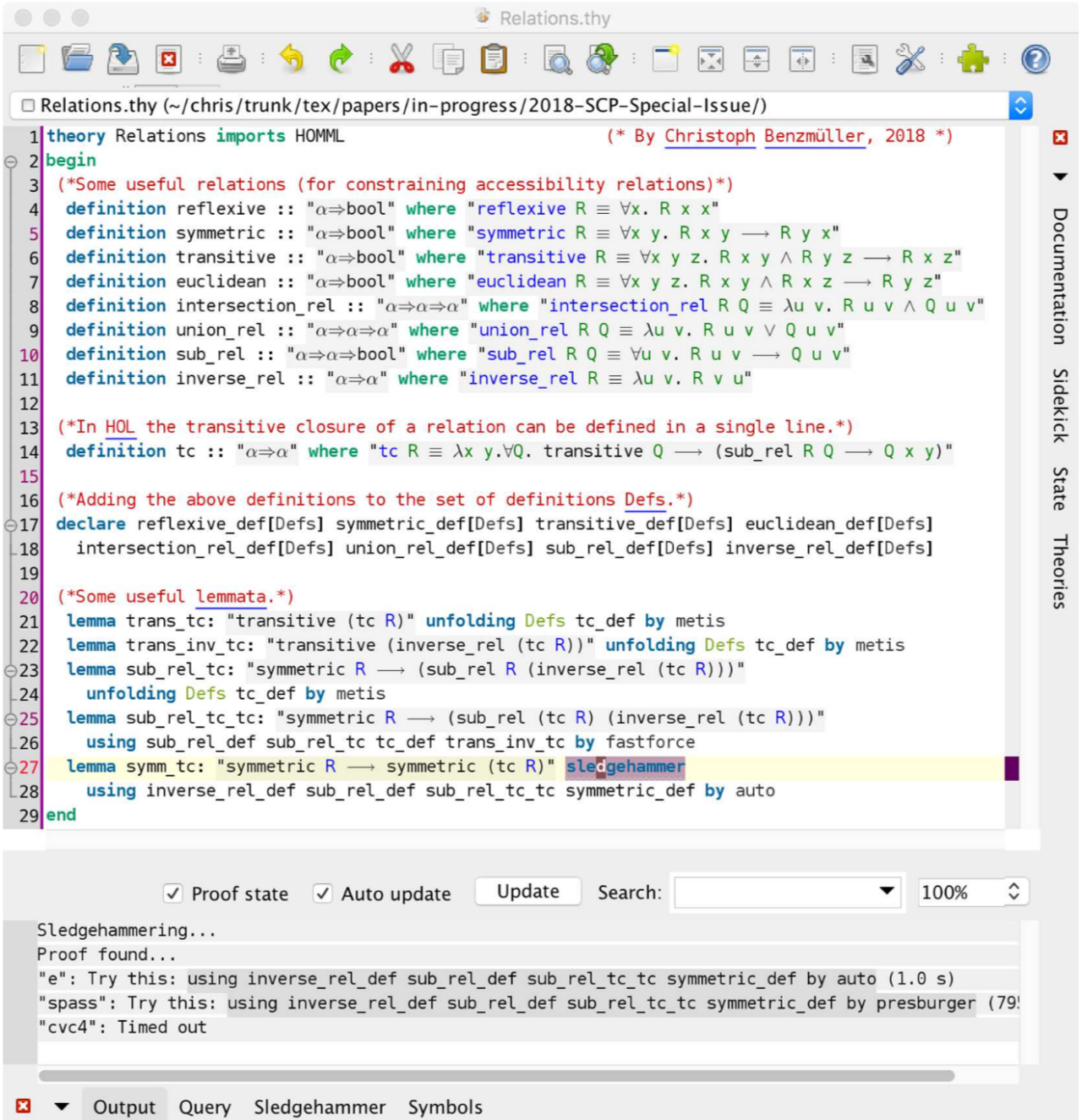
lized within modern proof assistant systems such as Isabelle/HOL. The contributed dataset provides supporting evidence for claims made in the article "Universal (meta-)logical reasoning: Recent successes" (Benzmüller, 2019).

Expressivity Matters — Higher-Order Multimodal Logic in HOL

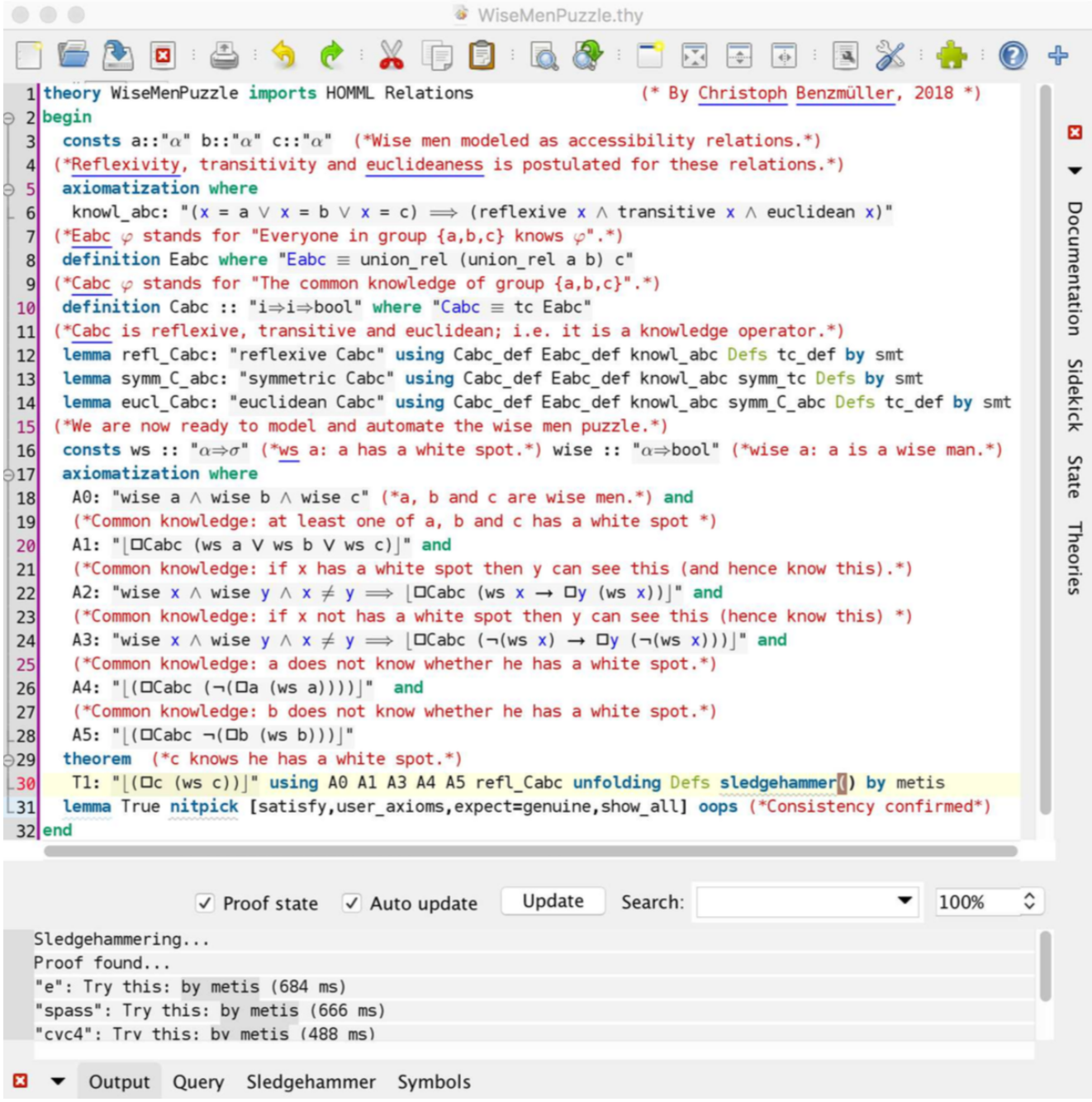
HOMML.thy

```
theory HOMML imports Main                    (* By Christoph Benzmüller, 2018 *)

begin (*An Embedding of Higher-Order Multi-Modal Logic (HOMML) in HOL.*)

  typedecl i (*Type of possible worlds.*) typedecl μ (*Type of individuals.*)
  type_synonym σ="(i⇒bool)" (*Type of world depended formulas (truth sets).*)
  type_synonym α="(i⇒i⇒bool)" (*Type of accessibility relations between worlds.*)

 (*Lifted HOMML connectives: they operate on world depended formulas (truth sets).*)
  definition mtop :: "σ" ("⊤") where "⊤ ≡ λw. True"
  definition mbot :: "σ" ("⊥") where "⊥ ≡ λw. False"
  definition mneg :: "σ⇒σ" ("¬_"[52]53)  where "¬φ ≡ λw. ¬φ(w)"
  definition mand :: "σ⇒σ⇒σ" (infixr"∧"51) where "φ∧ψ ≡ λw. φ(w)∧ψ(w)"
  definition mor  :: "σ⇒σ⇒σ" (infixr"∨"50) where "φ∨ψ ≡ λw. φ(w)∨ψ(w)"
  definition mimp :: "σ⇒σ⇒σ" (infixr"→"49) where "φ→ψ ≡ λw. φ(w)⟶ψ(w)"
  definition mequ :: "σ⇒σ⇒σ" (infixr"↔"48) where "φ↔ψ ≡ λw. φ(w)⟷ψ(w)"
  definition mall :: "('a⇒σ)⇒σ" ("∀") where "∀Φ ≡ λw.∀x. Φ(x)(w)"
  definition mallB:: "('a⇒σ)⇒σ" (binder"∀"[8]9) where "∀x. φ(x) ≡ ∀φ"
  definition mexi :: "('a⇒σ)⇒σ" ("∃") where "∃Φ ≡ λw.∃x. Φ(x)(w)"
  definition mexiB:: "('a⇒σ)⇒σ" (binder"∃"[8]9) where "∃x. φ(x) ≡ ∃φ"
  definition mbox :: "α⇒σ⇒σ" ("□_ _") where "□r φ ≡ (λw. ∀v. r w v ⟶ φ v)"
  definition mdia :: "α⇒σ⇒σ" ("◇_ _") where "◇r φ ≡ (λw. ∃v. r w v ∧ φ v)"

 (*Global and local validity of lifted formulas*)
  definition global_valid :: "σ ⇒ bool" ("⌊_⌋"[7]8) where "⌊p⌋ ≡ ∀w. p w"
  consts cw :: i (*Current world; uninterpreted constant of type i*)
  definition local_valid :: "σ ⇒ bool" ("⌊_⌋cw"[9]10) where "⌊p⌋cw ≡ p cw"

 (*Introducing "Defs" as the set of the above definitions; useful for convenient unfolding.*)
 named_theorems Defs declare mtop_def[Defs] mbot_def[Defs] mneg_def[Defs] mand_def[Defs]
  mor_def[Defs] mimp_def[Defs] mequ_def[Defs] mall_def[Defs] mallB_def[Defs] mexi_def[Defs]
  mexiB_def[Defs] mbox_def[Defs] mdia_def[Defs] global_valid_def[Defs] local_valid_def[Defs]
end
```

Expressivity Matters — Higher-Order Multimodal Logic in HOL



Expressivity Matters — Higher-Order Multimodal Logic in HOL

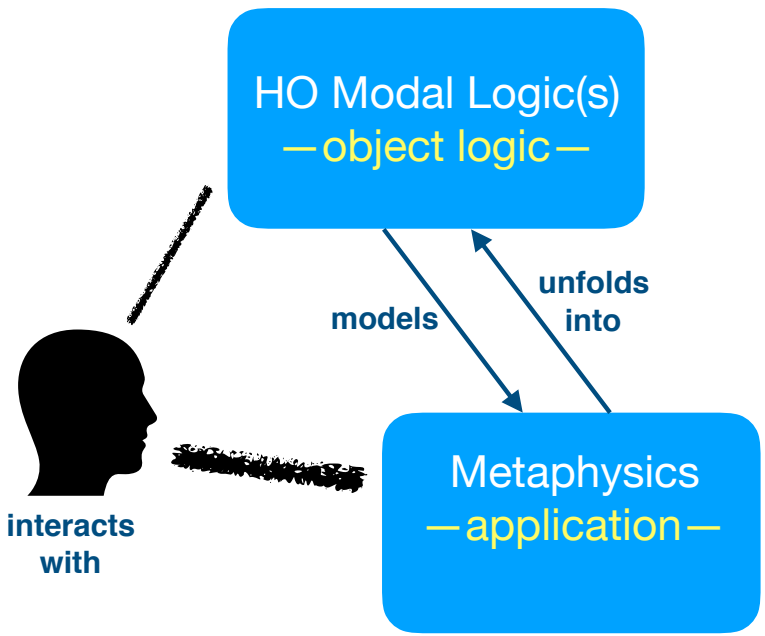


Universal (Meta-)Logical Reasoning Framework [SCP, 2019]

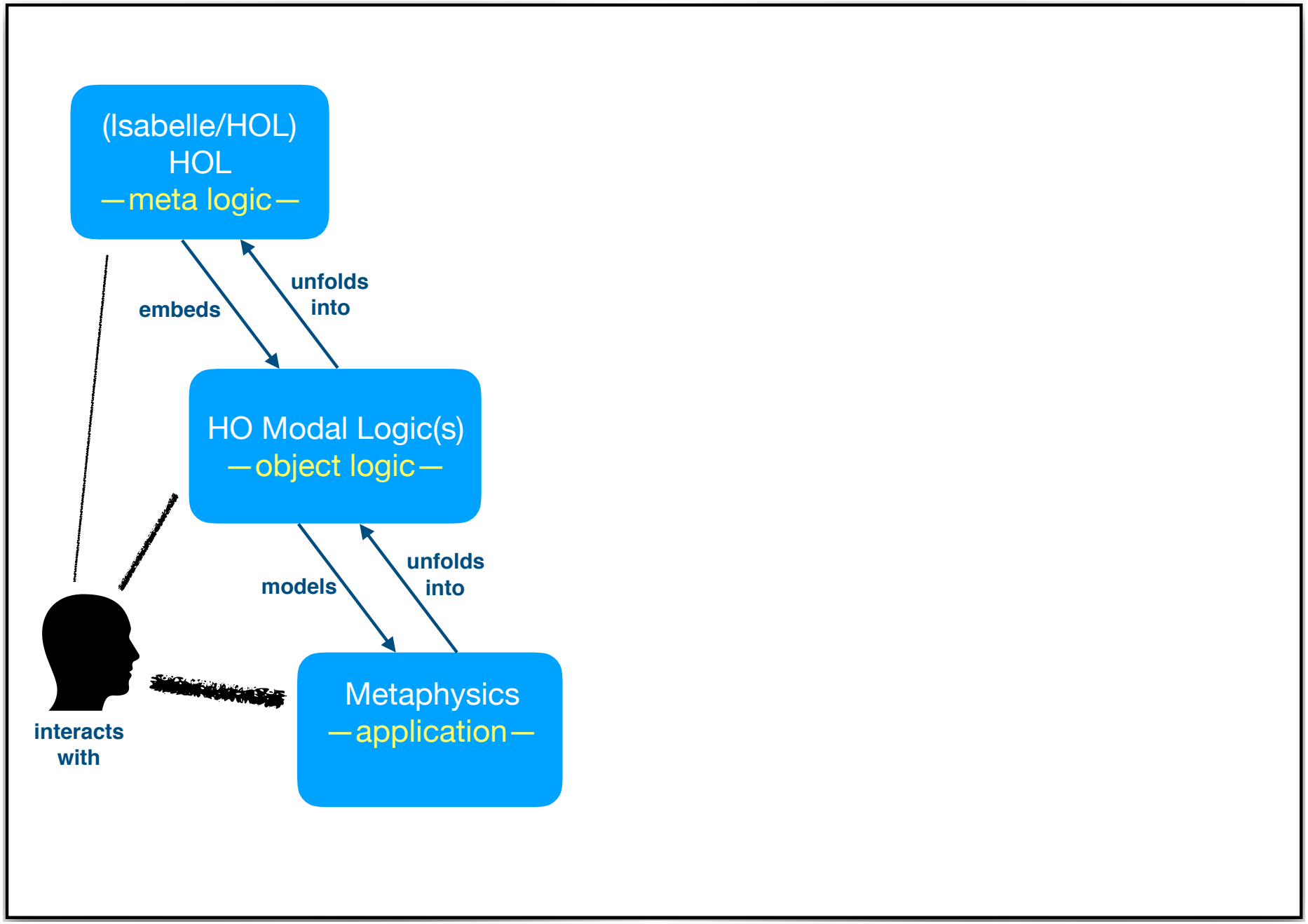
interacts with

Metaphysics
—application—

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

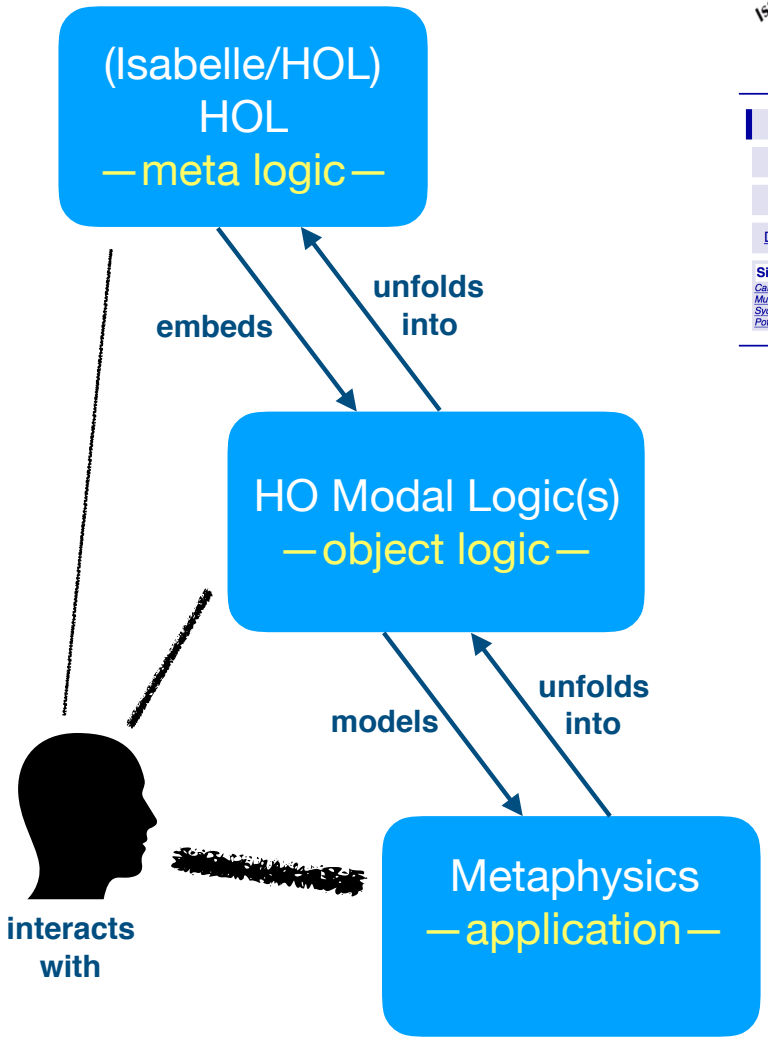


Universal (Meta-)Logical Reasoning Framework [SCP, 2019]



Universal (Meta-)Logical Reasoning Framework

[SCP, 2019]



Isabelle

UNIVERSITY OF CAMBRIDGE Computer Laboratory

TECHNISCHE UNIVERSITÄT MÜNCHEN

Home
Overview
Installation
Documentation

Site Mirrors:
Cambridge (.uk)
Munich (.de)
Sydney (.au)
Potsdam, NY (.us)

What is Isabelle?

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

Now available: Isabelle2017 (October 2017)

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

Some notable changes:

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

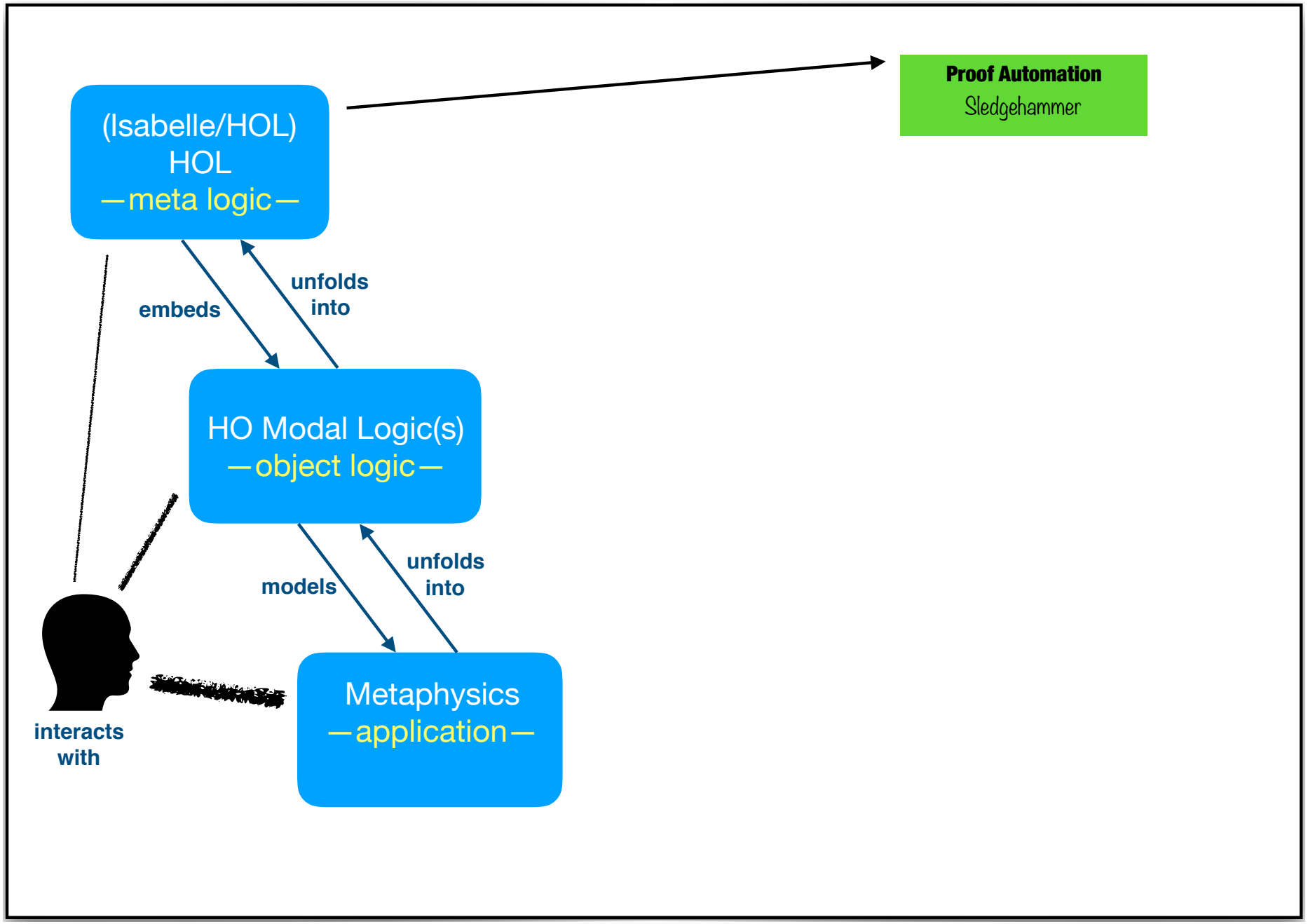
See also the cumulative NEWS.

Distribution & Support

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

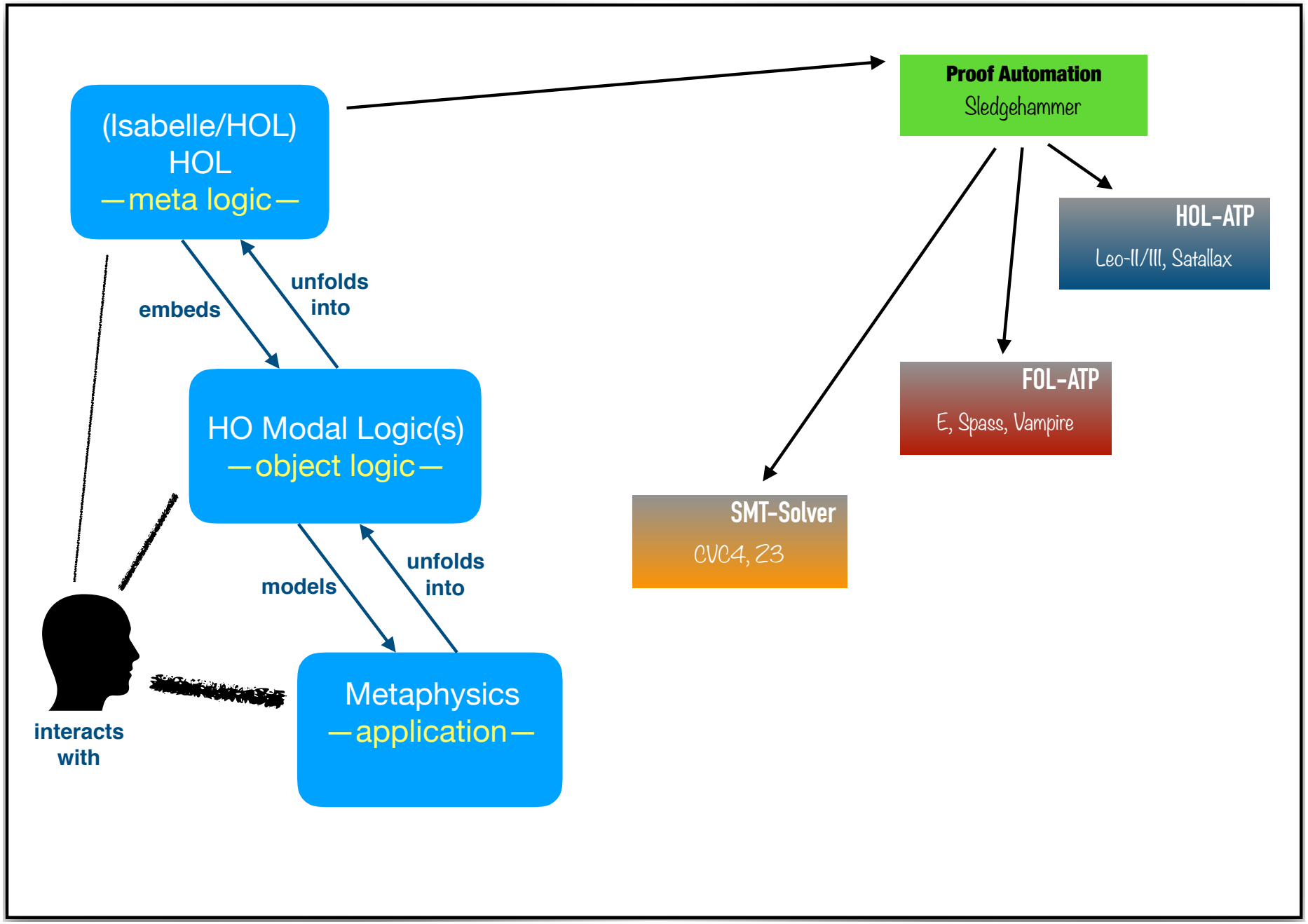
Universal (Meta-)Logical Reasoning Framework

[SCP, 2019]



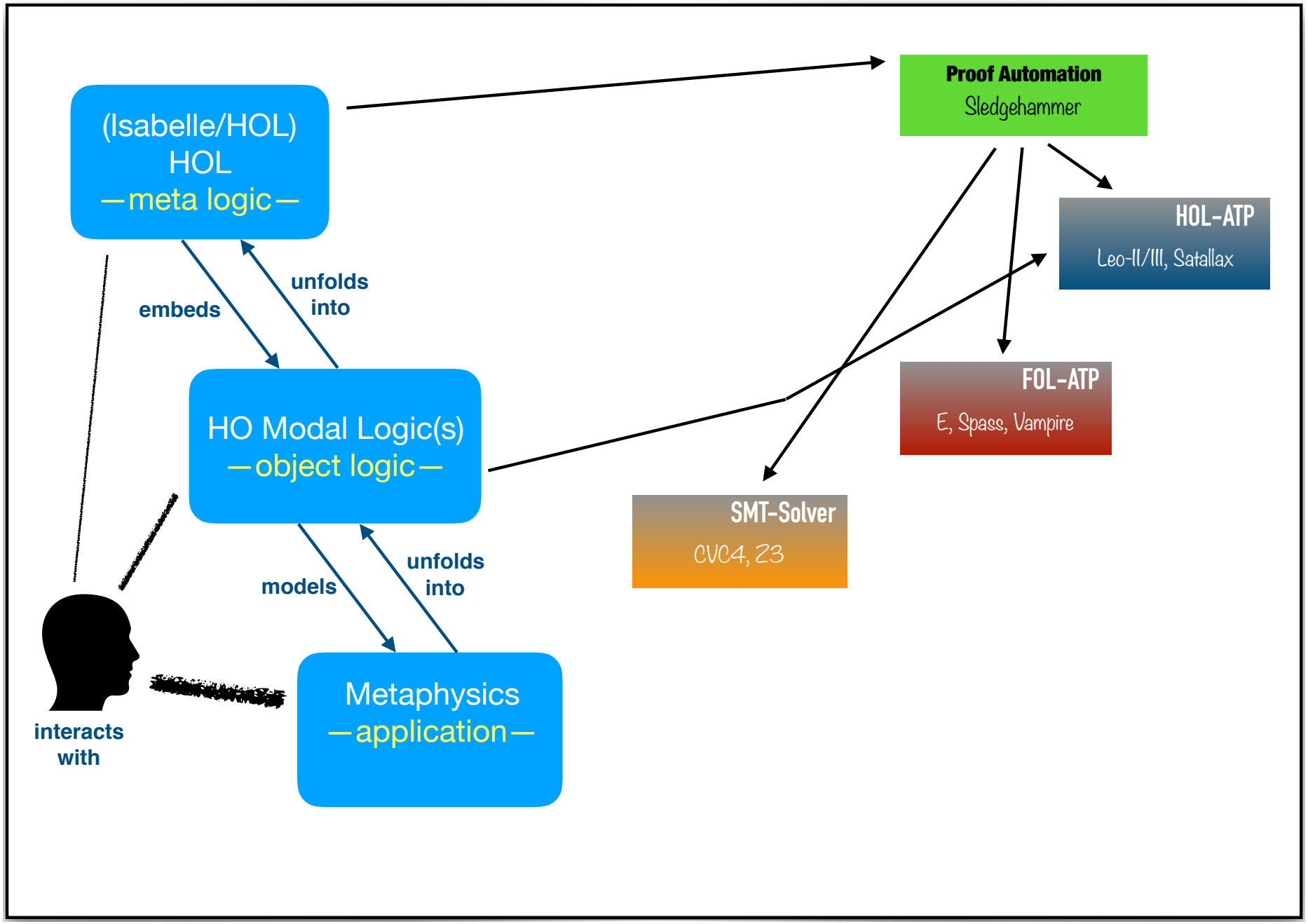
Universal (Meta-)Logical Reasoning Framework

[SCP, 2019]



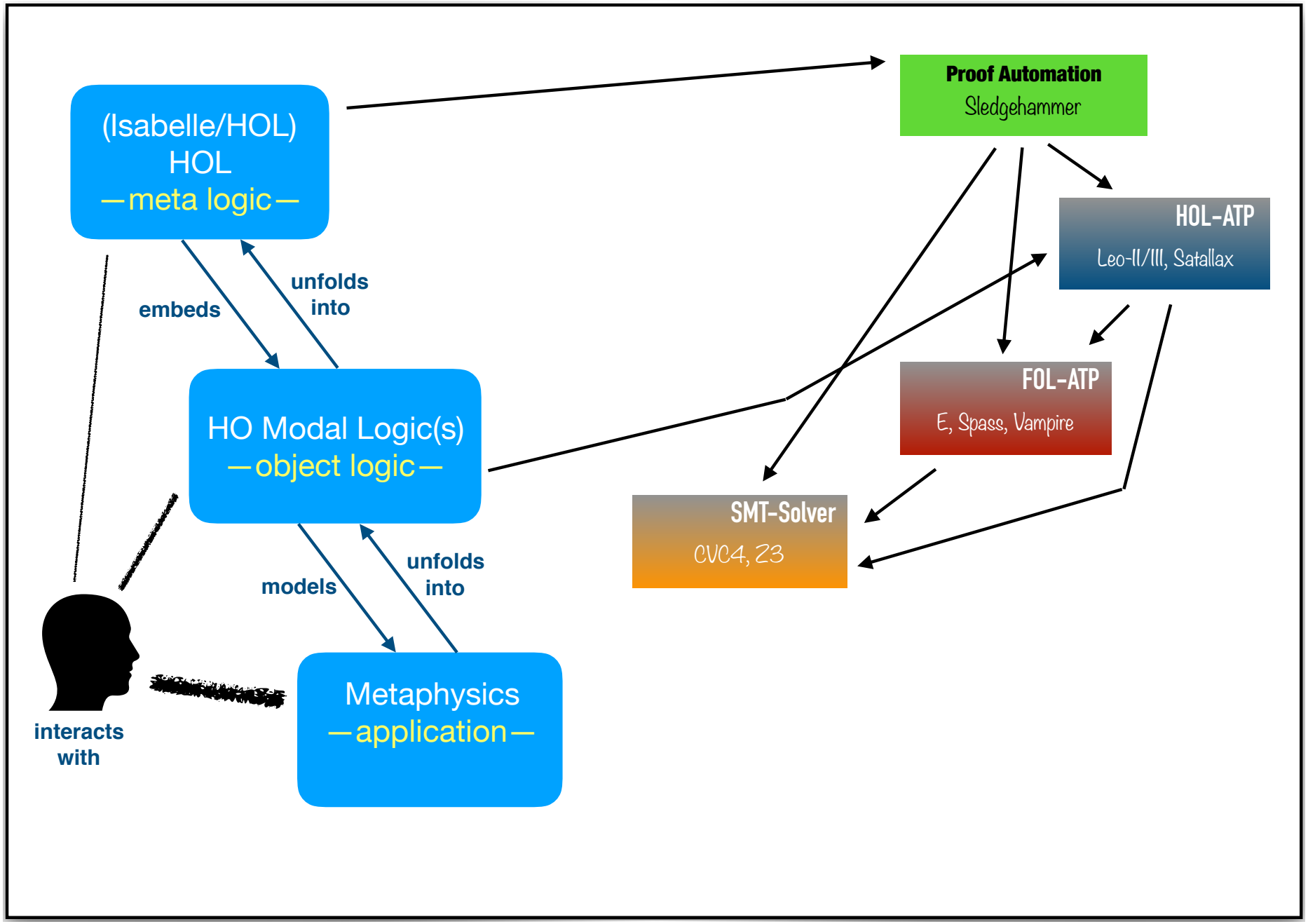
Universal (Meta-)Logical Reasoning Framework

[SCP, 2019]



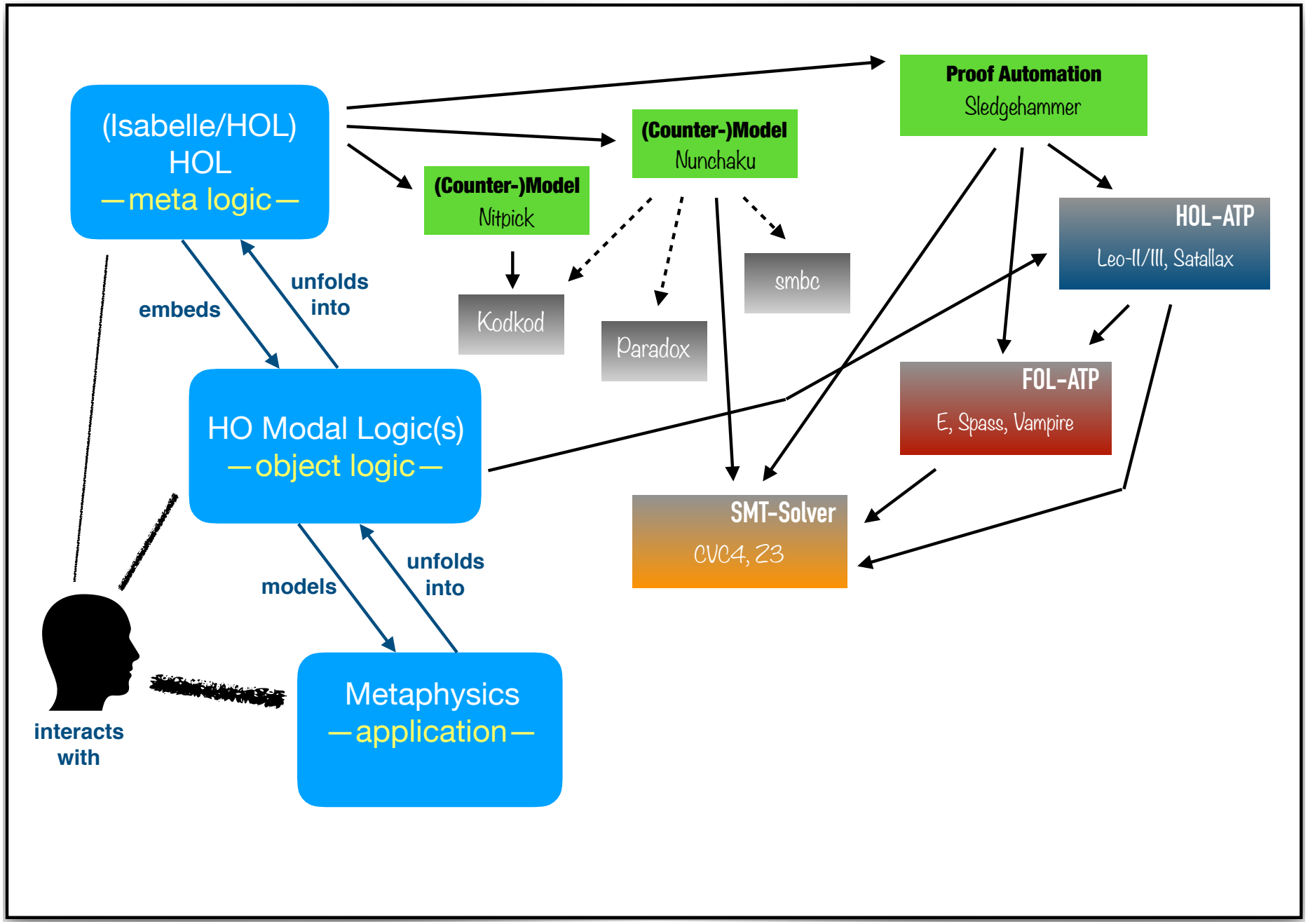
Universal (Meta-)Logical Reasoning Framework

[SCP, 2019]



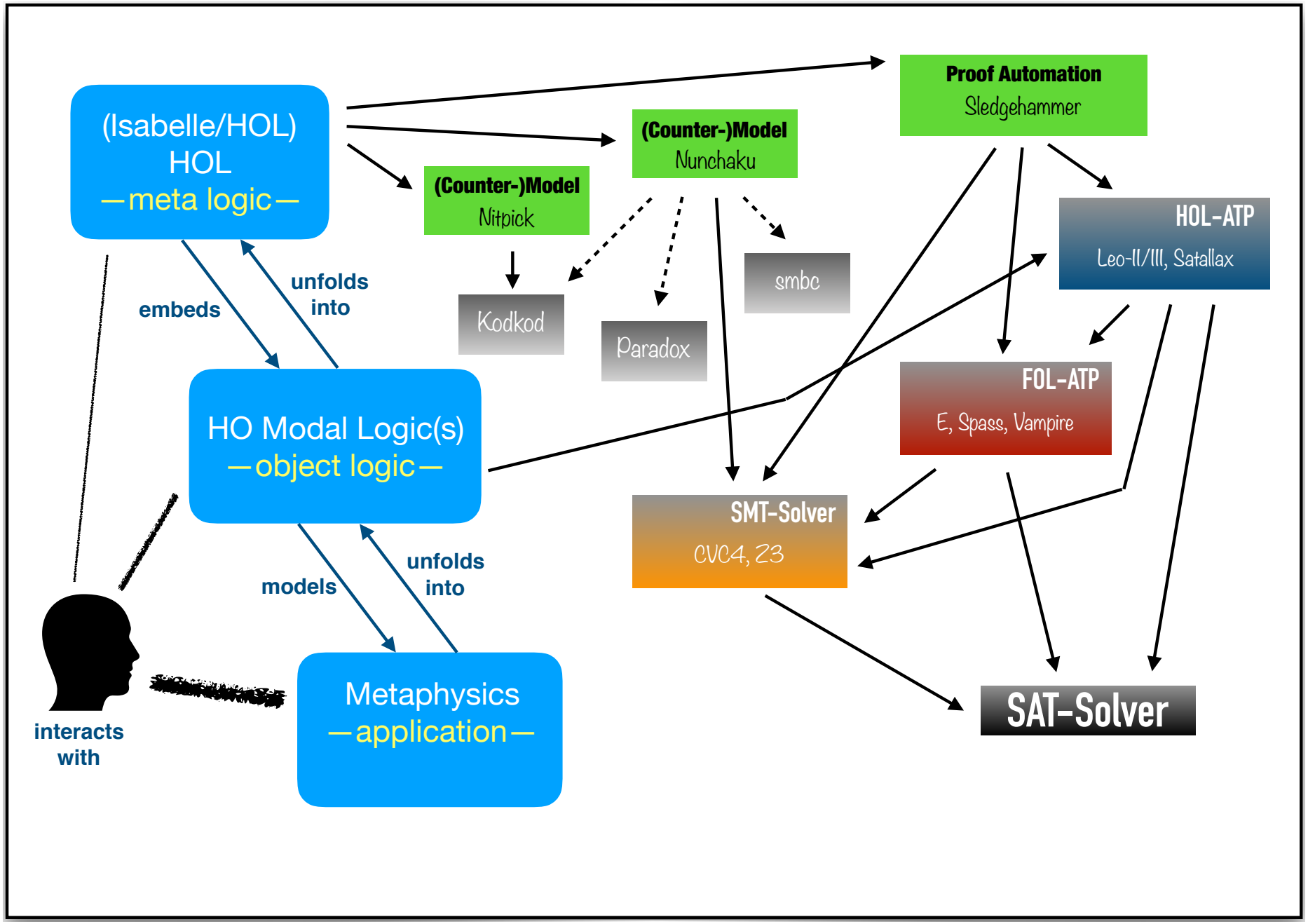
Universal (Meta-)Logical Reasoning Framework

[SCP, 2019]



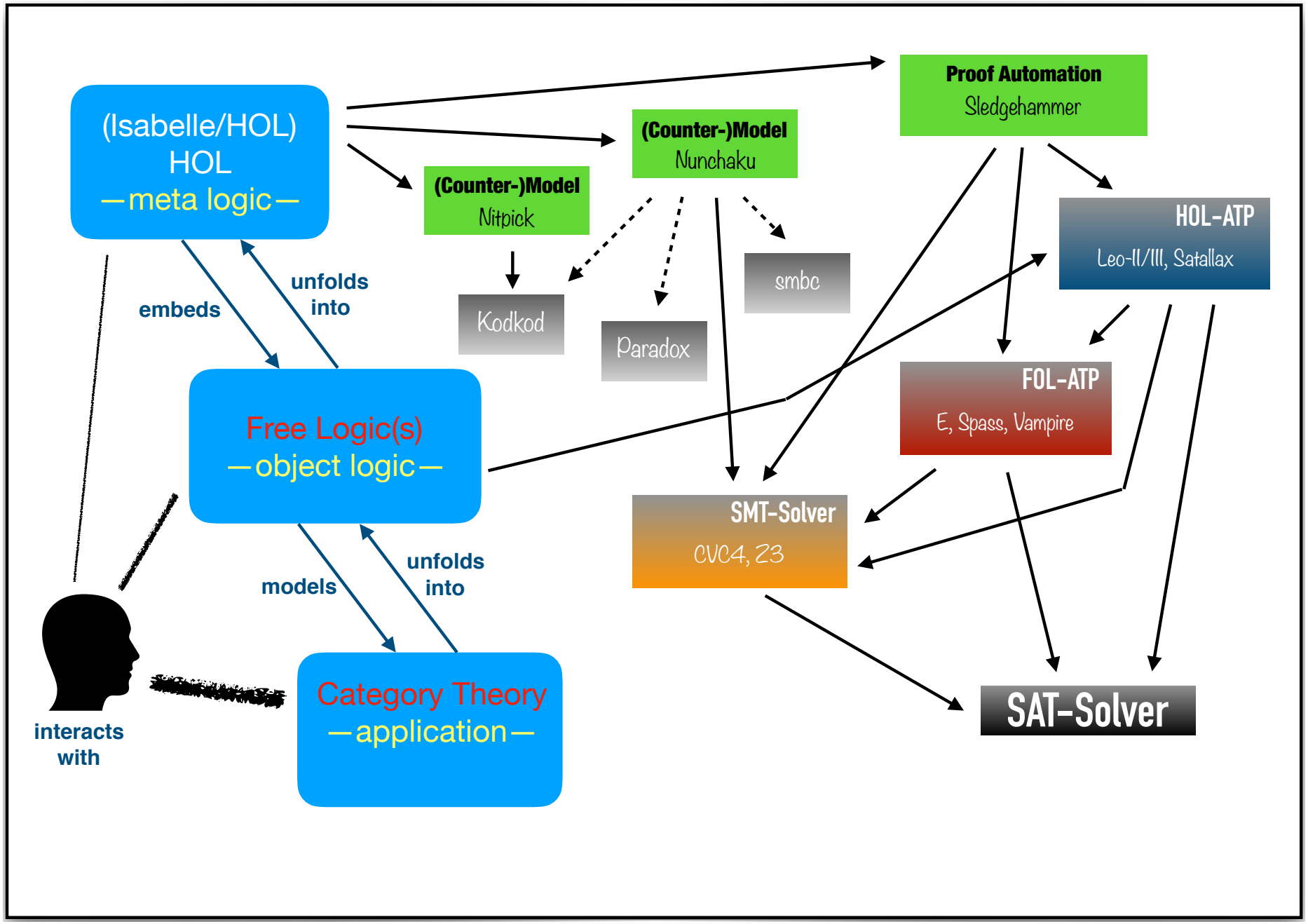
Universal (Meta-)Logical Reasoning Framework

[SCP, 2019]



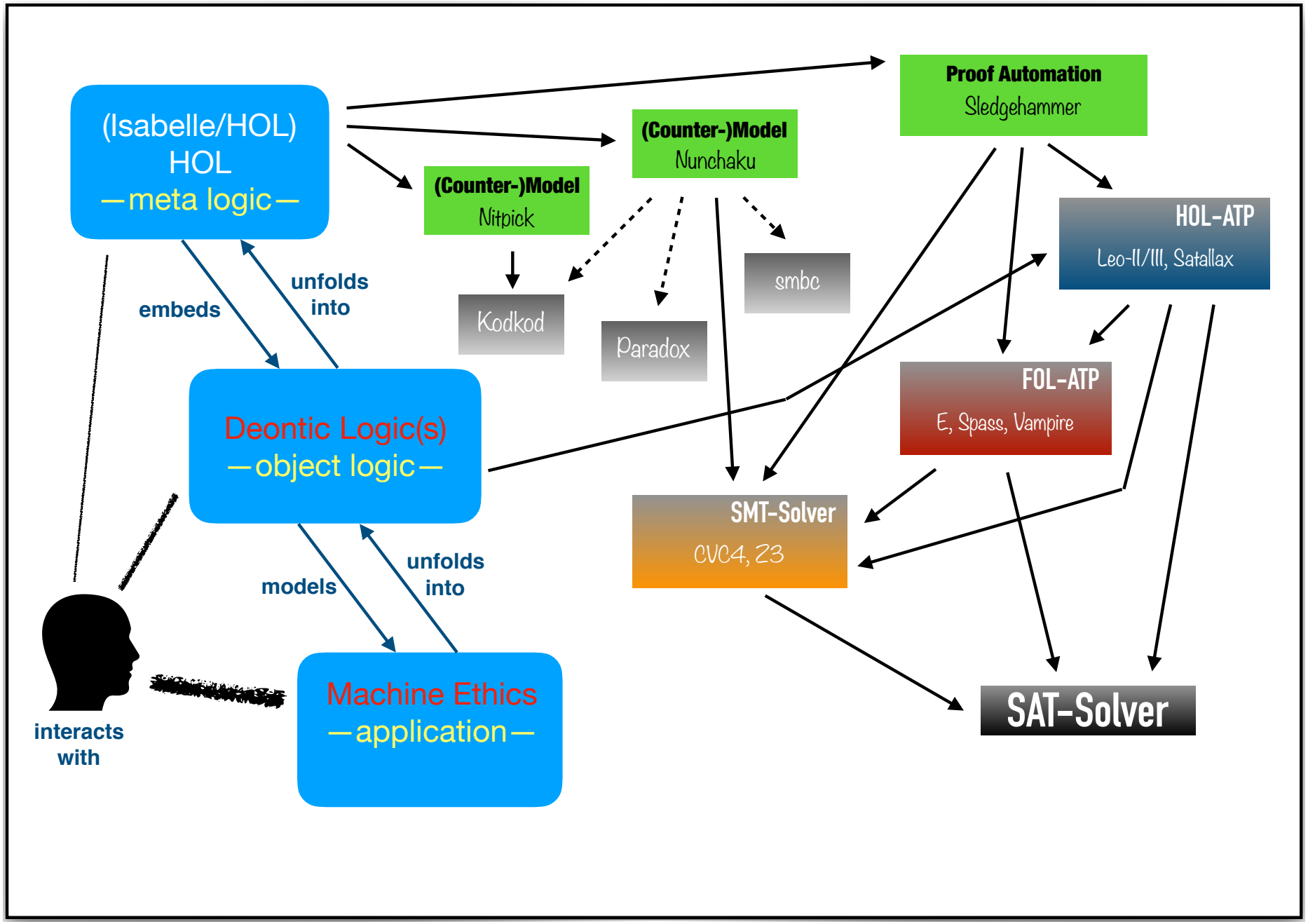
Universal (Meta-)Logical Reasoning Framework

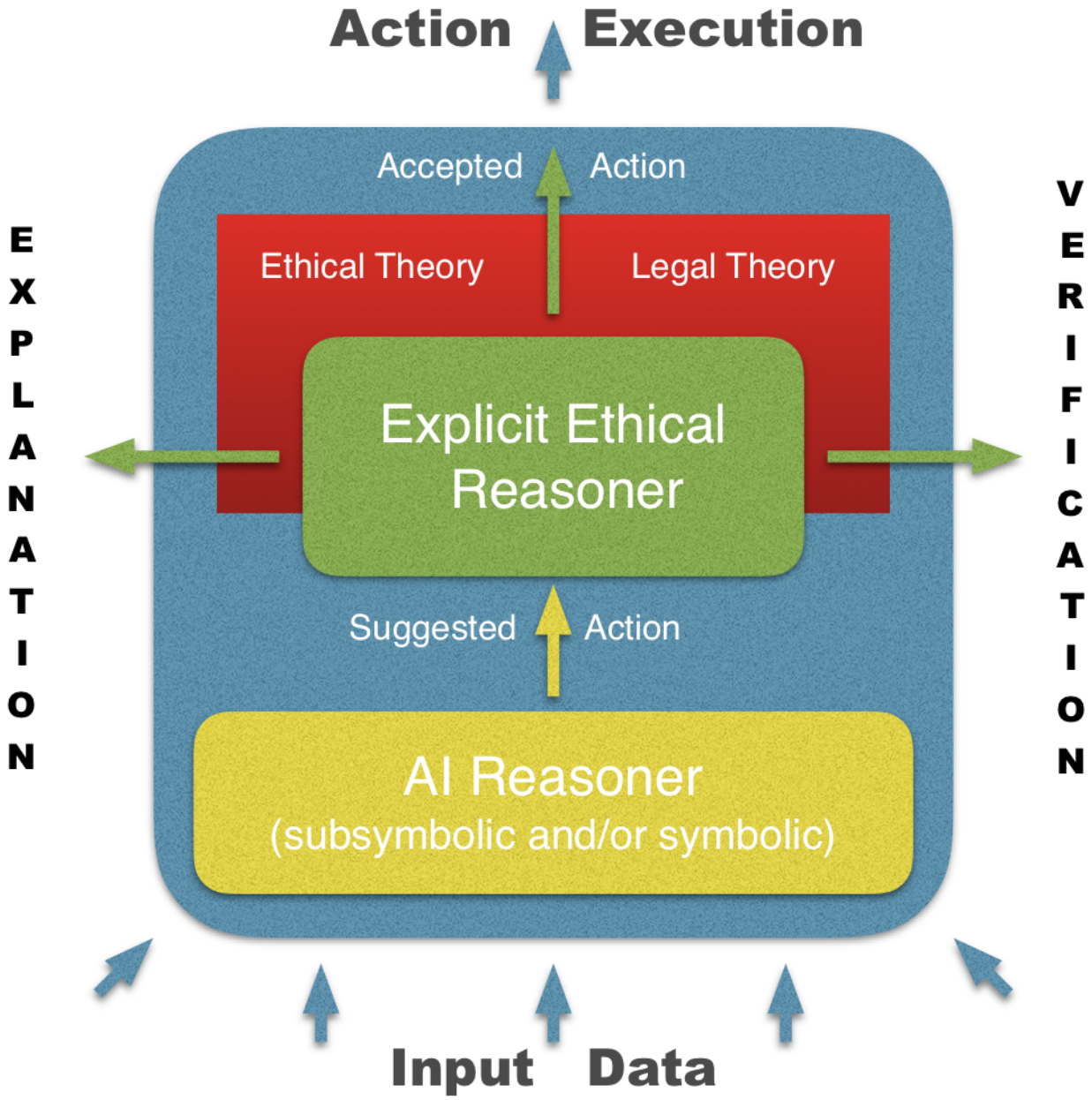
[SCP, 2019]

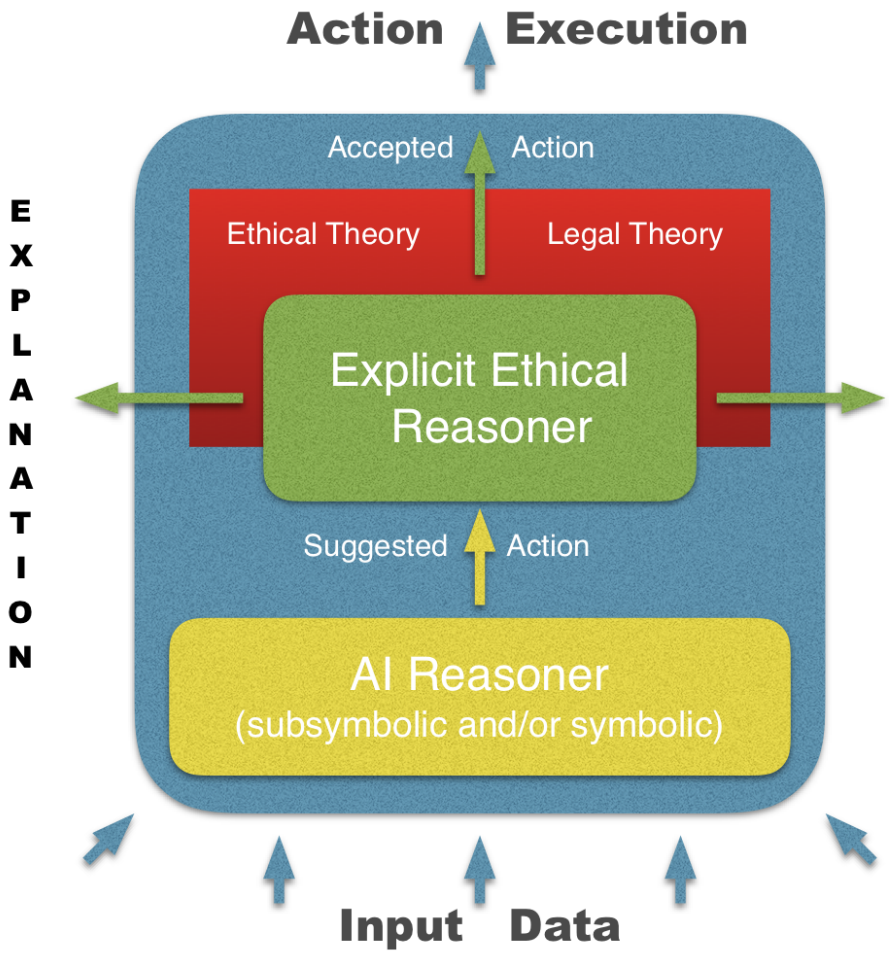


Universal (Meta-)Logical Reasoning Framework

[SCP, 2019]



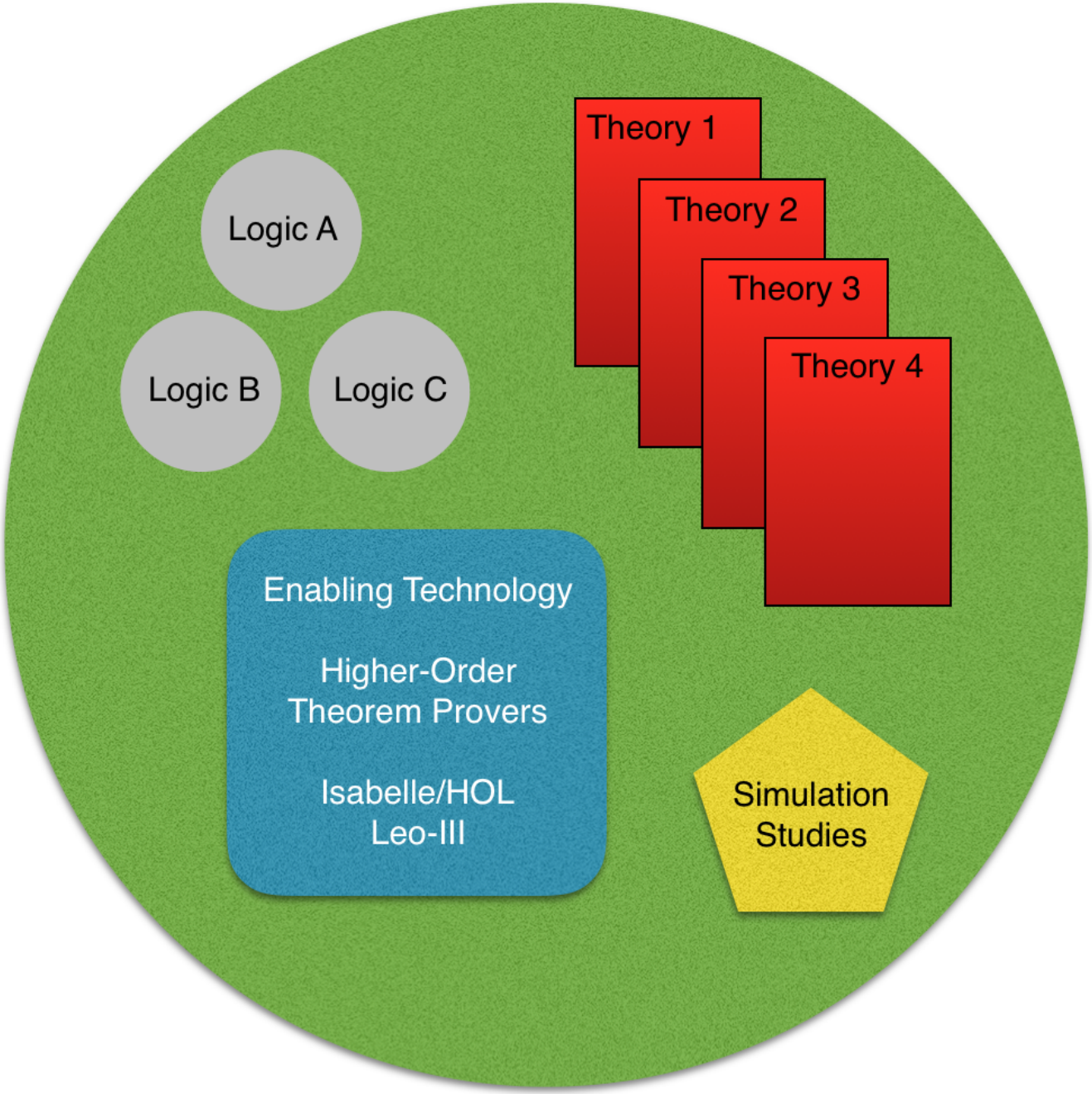

Action Execution
Accepted Action
Ethical Theory
Legal Theory
Explicit Ethical Reasoner
Suggested Action
AI Reasoner
(subsymbolic and/or symbolic)
Input Data
EXPLANATION
VERIFICATION



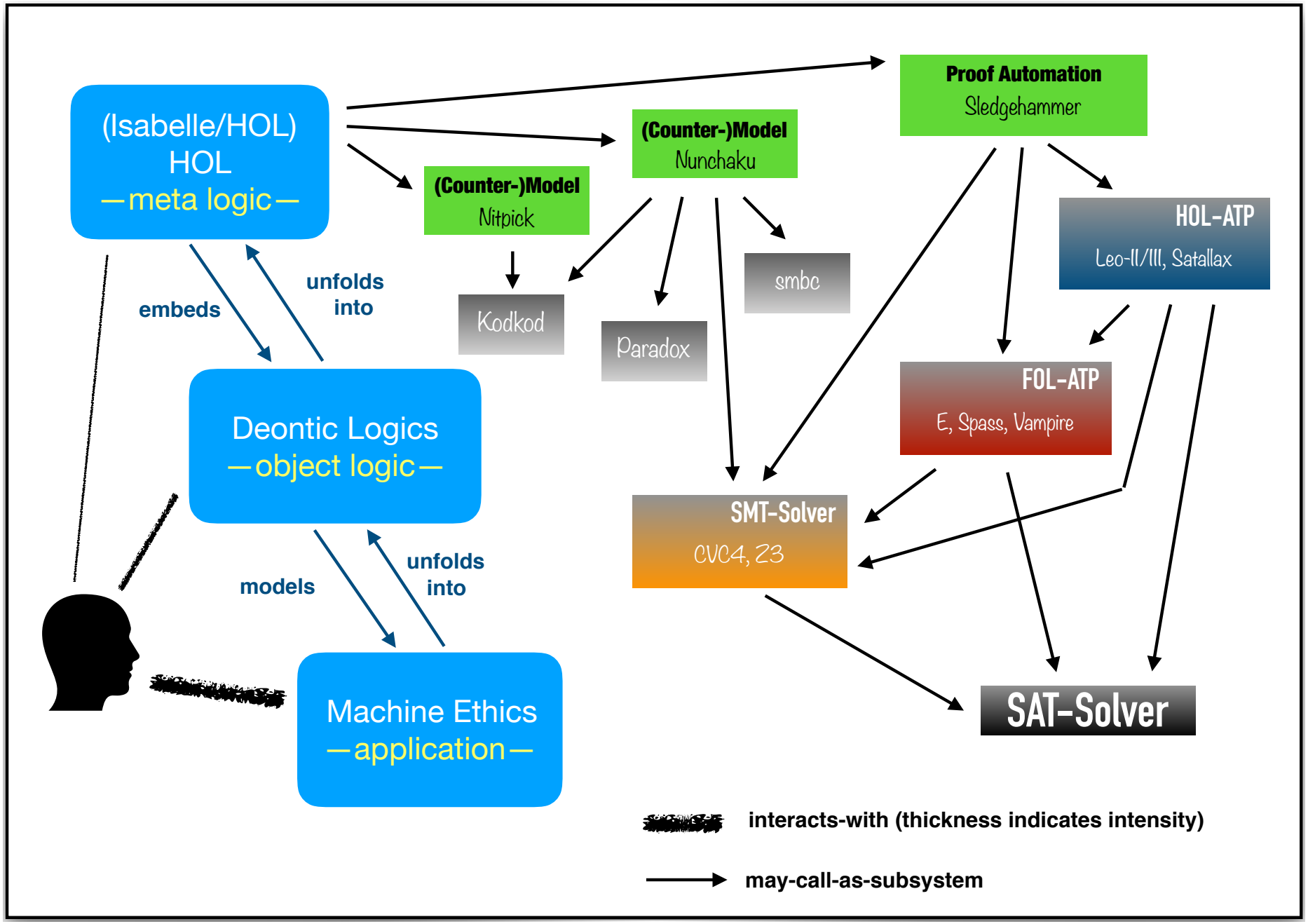
Related Work

- Artificial Moral Agents
  - [Wallach&Allen, 2008]
- Ethical Governors
  - [ArkinEtAl., 2009, 2012]
  - [Dennis&Fisher, 2017]
- Ethical Deliberation in ART
  - [Dignum, 2017]
- Programming Machine Ethics
  - [Pereira&Saptawijaya, 2016]
- . . .

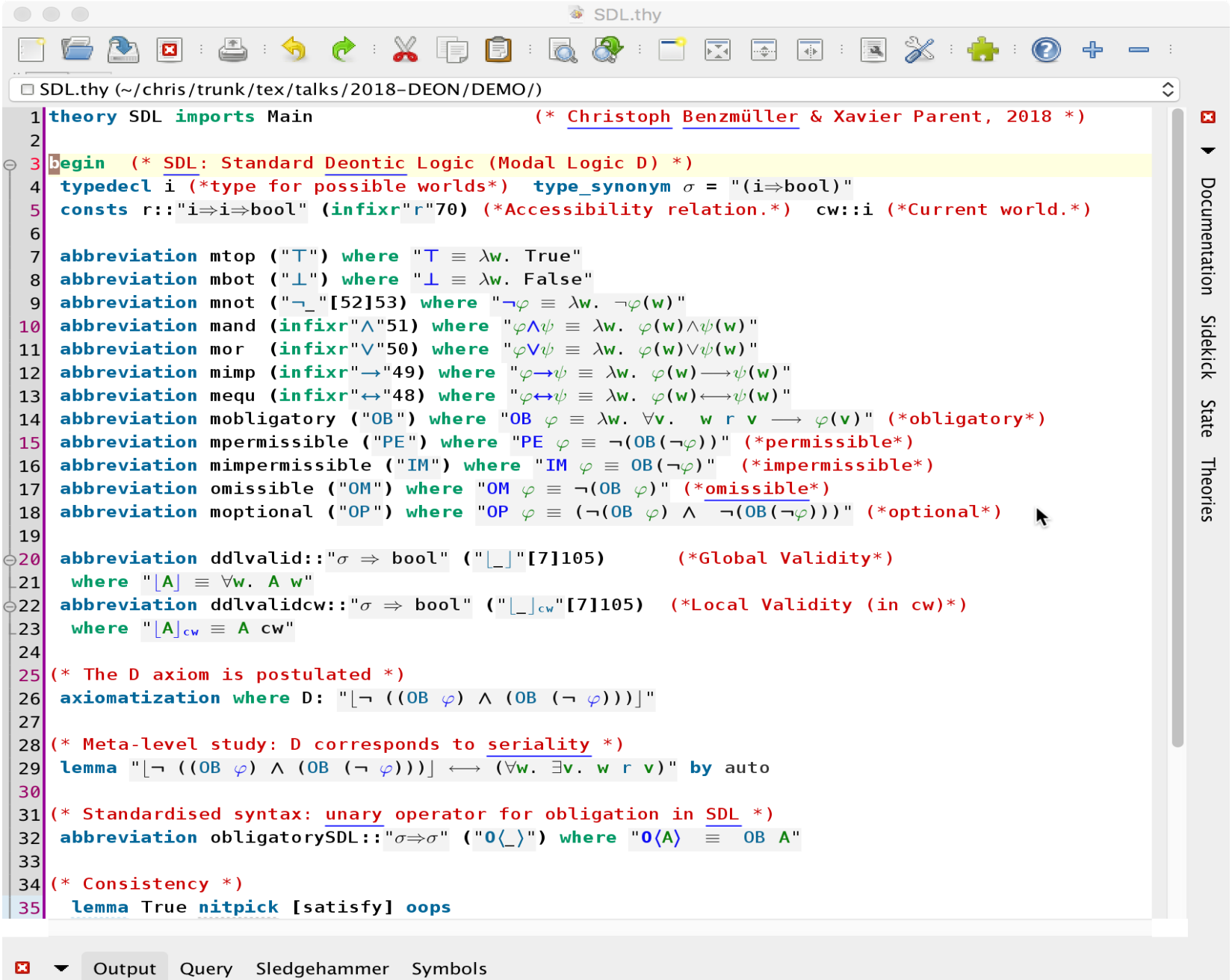
Normative Reasoning Experimentation Platform (Deontic Logics are required!)



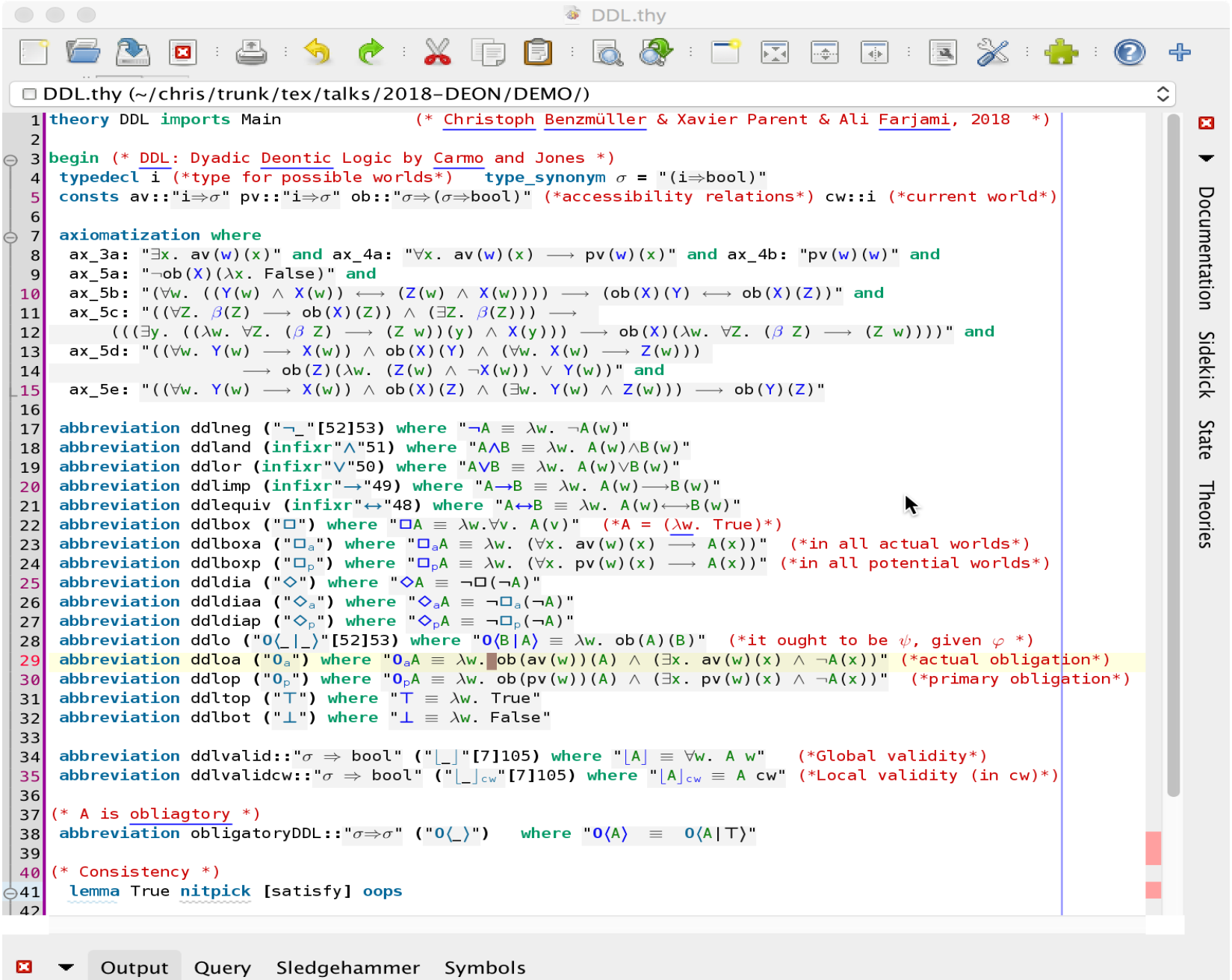
Universal (Meta-)Logical Reasoning Framework (supports multiple Deontic Logics)



Standard Deontic Logic (SDL) in Isabelle/HOL

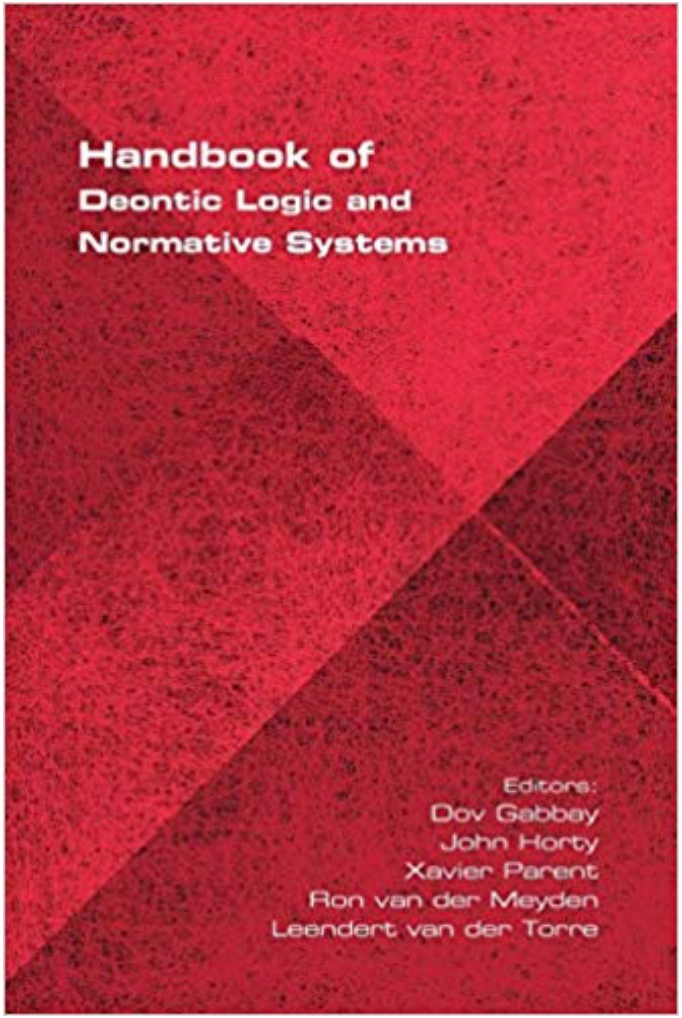


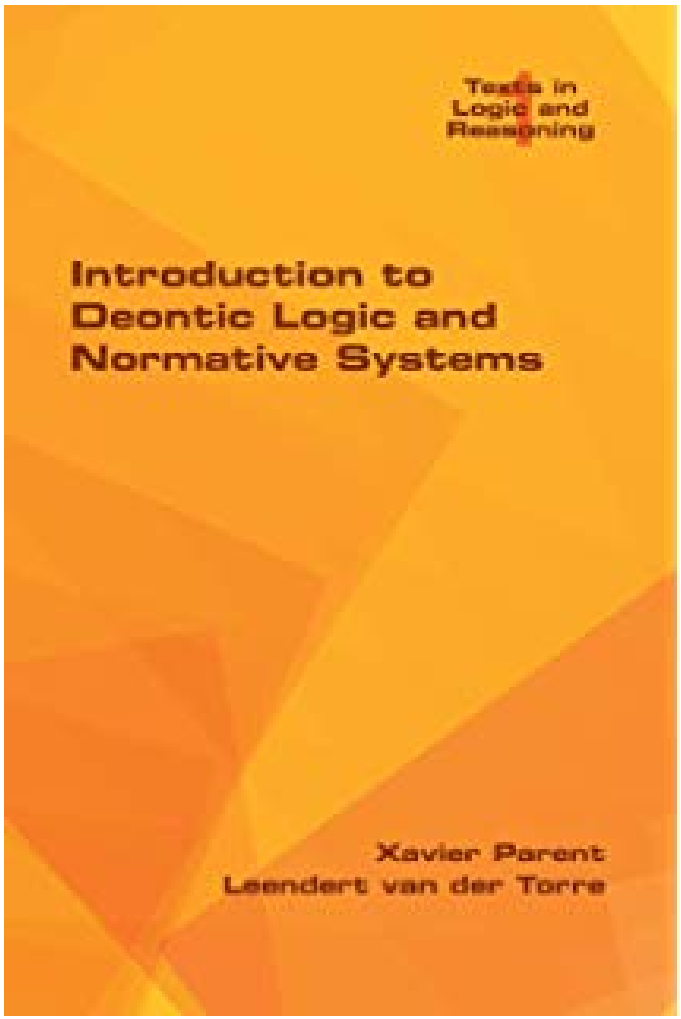
Dyadic Deontic Logic (DDL) in Isabelle/HOL




```
theory DDL imports Main            (* Christoph Benzmüller & Xavier Parent & Ali Farjami, 2018  *)

begin (* DDL: Dyadic Deontic Logic by Carmo and Jones *)
 typedecl i (*type for possible worlds*)   type_synonym σ = "(i⇒bool)"
 consts av::"i⇒σ" pv::"i⇒σ" ob::"σ⇒(σ⇒bool)" (*accessibility relations*) cw::i (*current world*)

 axiomatization where
  ax_3a: "∃x. av(w)(x)" and ax_4a: "∀x. av(w)(x) ⟶ pv(w)(x)" and ax_4b: "pv(w)(w)" and
  ax_5a: "¬ob(X)(λx. False)" and
  ax_5b: "(∀w. ((Y(w) ∧ X(w)) ⟷ (Z(w) ∧ X(w)))) ⟶ (ob(X)(Y) ⟷ ob(X)(Z))" and
  ax_5c: "((∀Z. β(Z) ⟶ ob(X)(Z)) ∧ (∃Z. β(Z))) ⟶
      (((∃y. ((λw. ∀Z. (β Z) ⟶ (Z w))(y) ∧ X(y))) ⟶ ob(X)(λw. ∀Z. (β Z) ⟶ (Z w))))" and
  ax_5d: "((∀w. Y(w) ⟶ X(w)) ∧ ob(X)(Y) ∧ (∀w. X(w) ⟶ Z(w)))
                  ⟶ ob(Z)(λw. (Z(w) ∧ ¬X(w)) ∨ Y(w))" and
  ax_5e: "((∀w. Y(w) ⟶ X(w)) ∧ ob(X)(Z) ∧ (∃w. Y(w) ∧ Z(w))) ⟶ ob(Y)(Z)"

 abbreviation ddlneg ("¬_"[52]53) where "¬A ≡ λw. ¬A(w)"
 abbreviation ddland (infixr"∧"51) where "A∧B ≡ λw. A(w)∧B(w)"
 abbreviation ddlor (infixr"∨"50) where "A∨B ≡ λw. A(w)∨B(w)"
 abbreviation ddlimp (infixr"→"49) where "A→B ≡ λw. A(w)⟶B(w)"
 abbreviation ddlequiv (infixr"↔"48) where "A↔B ≡ λw. A(w)⟷B(w)"
 abbreviation ddlbox ("□") where "□A ≡ λw.∀v. A(v)"  (*A = (λw. True)*)
 abbreviation ddlboxa ("□a") where "□aA ≡ λw. (∀x. av(w)(x) ⟶ A(x))"  (*in all actual worlds*)
 abbreviation ddlboxp ("□p") where "□pA ≡ λw. (∀x. pv(w)(x) ⟶ A(x))" (*in all potential worlds*)
 abbreviation ddldia ("◇") where "◇A ≡ ¬□(¬A)"
 abbreviation ddldiaa ("◇a") where "◇aA ≡ ¬□a(¬A)"
 abbreviation ddldiap ("◇p") where "◇pA ≡ ¬□p(¬A)"
 abbreviation ddlo ("O⟨_|_⟩"[52]53) where "O⟨B|A⟩ ≡ λw. ob(A)(B)"  (*it ought to be ψ, given φ *)
 abbreviation ddloa ("Oa") where "OaA ≡ λw. ob(av(w))(A) ∧ (∃x. av(w)(x) ∧ ¬A(x))" (*actual obligation*)
 abbreviation ddlop ("Op") where "OpA ≡ λw. ob(pv(w))(A) ∧ (∃x. pv(w)(x) ∧ ¬A(x))"  (*primary obligation*)
 abbreviation ddltop ("⊤") where "⊤ ≡ λw. True"
 abbreviation ddlbot ("⊥") where "⊥ ≡ λw. False"

 abbreviation ddlvalid::"σ ⇒ bool" ("⌊_⌋"[7]105) where "⌊A⌋ ≡ ∀w. A w"   (*Global validity*)
 abbreviation ddlvalidcw::"σ ⇒ bool" ("⌊_⌋cw"[7]105) where "⌊A⌋cw ≡ A cw" (*Local validity (in cw)*)

(* A is obliagtory *)
 abbreviation obligatoryDDL::"σ⇒σ" ("O⟨_⟩")   where "O⟨A⟩  ≡  O⟨A|⊤⟩"

(* Consistency *)
  lemma True nitpick [satisfy] oops
```

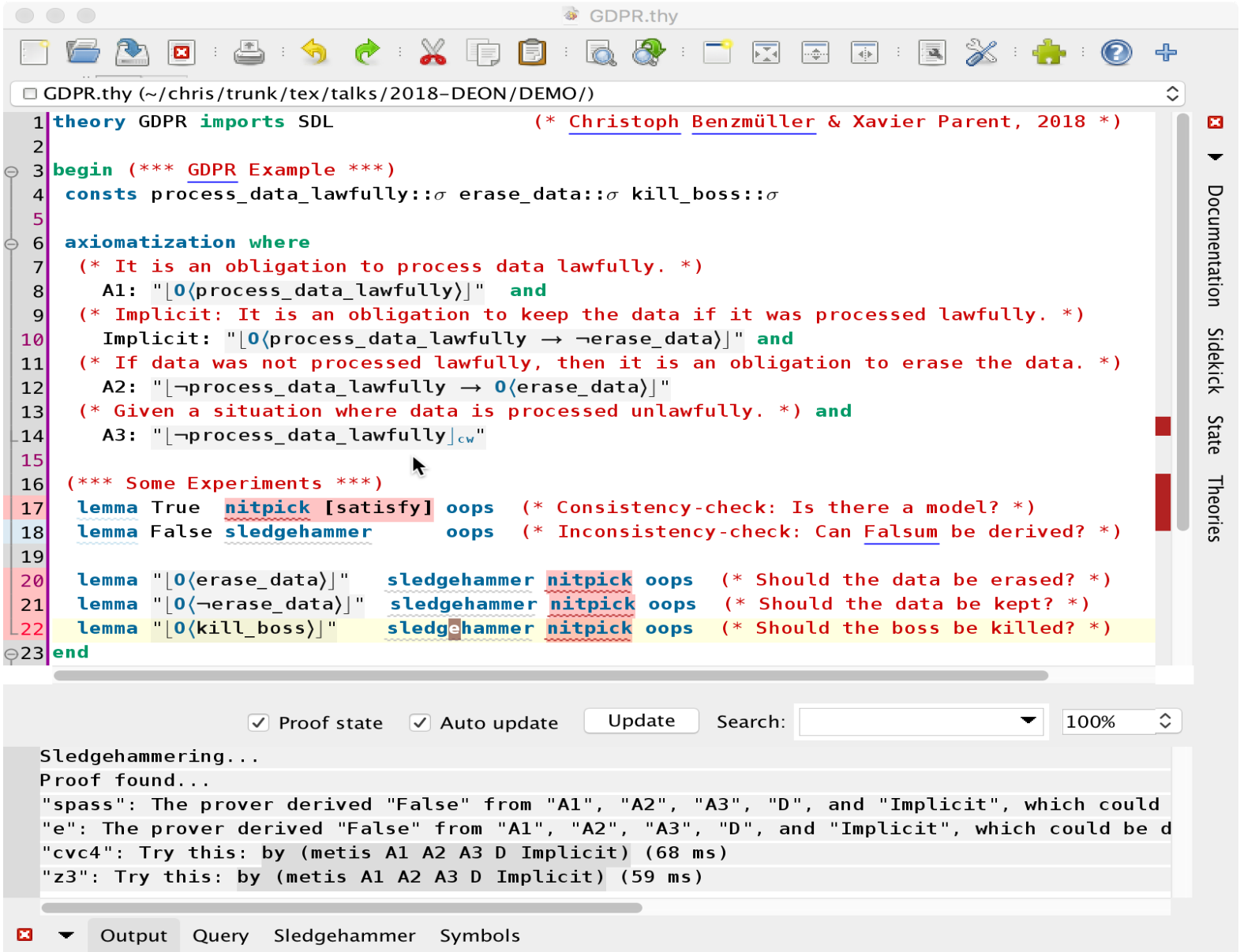
One SDL $\Longrightarrow$ Many Post-SDL's $\Longrightarrow$ One HOL (Meta-Logic)





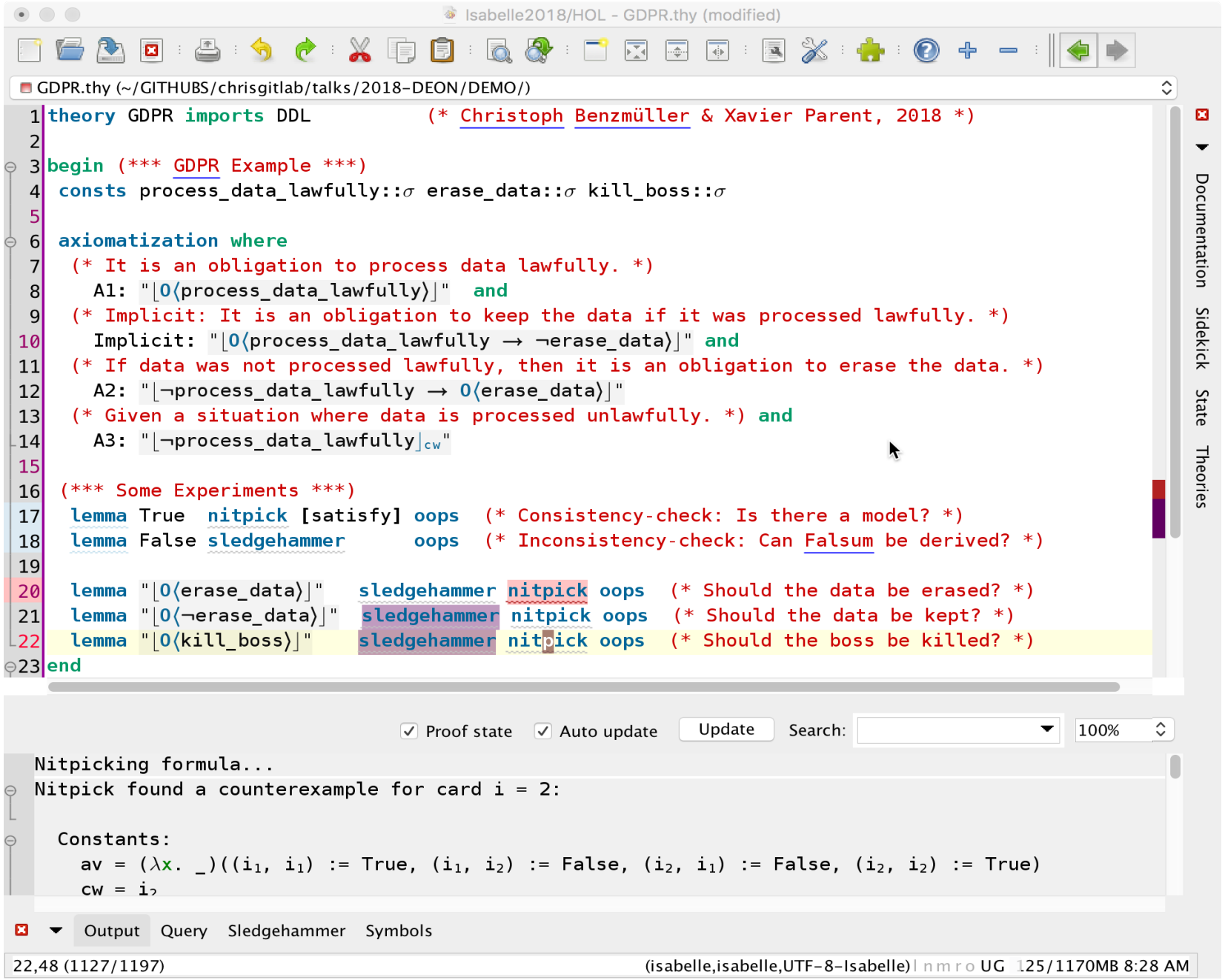
- Translation to First-Order Logic (FOL): supports propositional non-classical logics
- Embedding in Higher-Order Logic (HOL): additionally supports **quantified** non-classical logics
  (furthermore: **competitive automation** and **intuitive user-interaction**)

Experimenting with Deontic Logics in Isabelle/HOL



Our framework scales for first-order and higher-order deontic logics!!!

Experimenting with Deontic Logics in Isabelle/HOL



Input/Output-Logic in Isabelle/HOL

Input/Output (I/O) Logic

[Makinson, JPL, 2000], [GabbayHortyParentEtAl-Handbook, 2013]

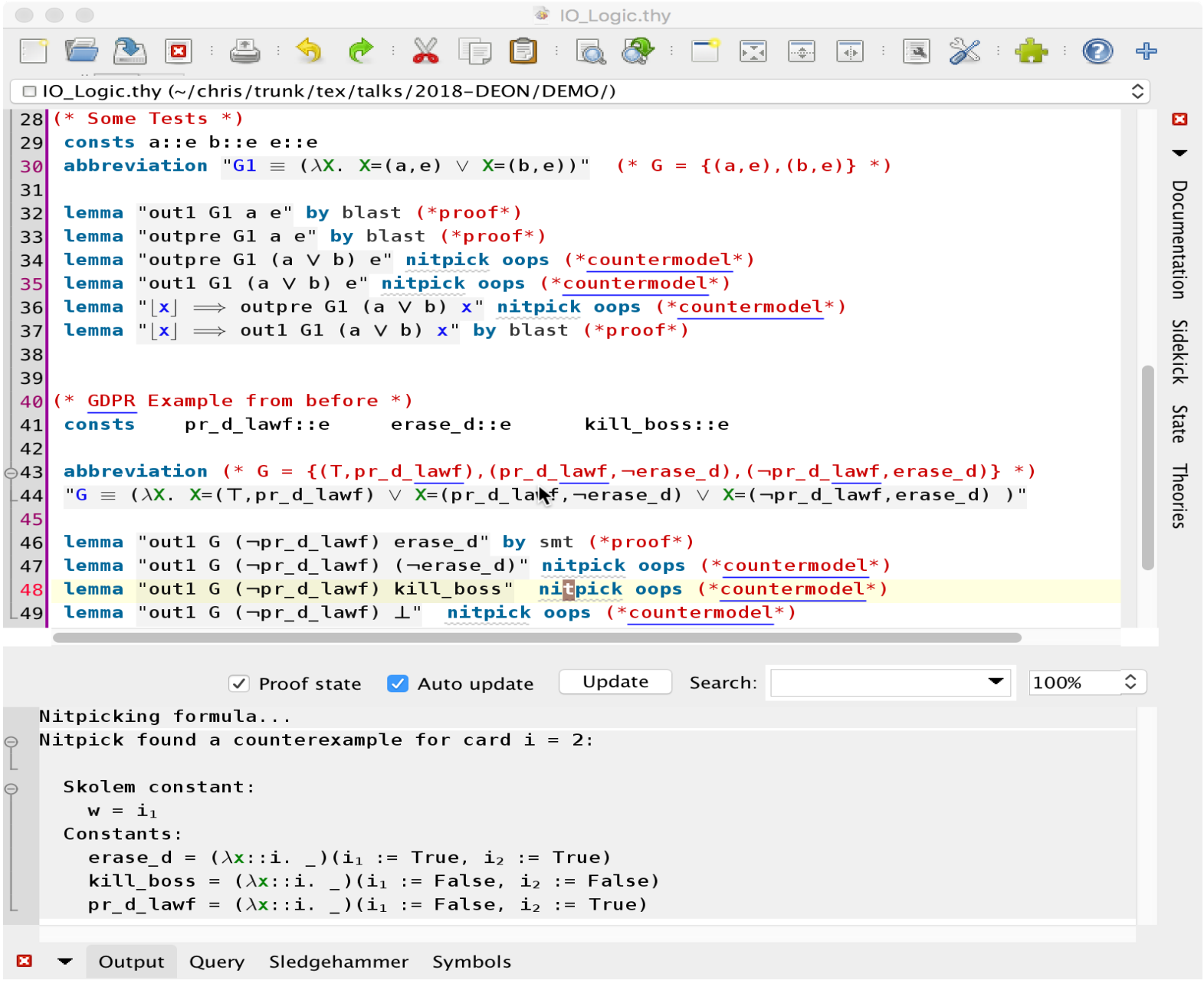
- I/O-operators, such as out1 (simple-minded output), accept set G of conditional norms as argument
- Conditional norms: pairs (a,x) with input “a” (condition) and output “x” (obligation)
- Pairs (a,x) are not given a truth-functional semantics in I/O logic

Semantics of out1

- $out1(G, A) := Cn(G(Cn(A)))$
- where $Cn(X) := \{s \mid X \models s\}$ and $G(X) := \{s \mid \exists a \in X.\ (a, s) \in G\}$.

```
(*IO logic in HOL*)
typedecl i -- "type for possible worlds"     type_synonym e = "(i⇒bool)"
abbreviation ktop   :: "e"     ("⊤")  where "⊤ ≡ λw. True"
abbreviation kbot   :: "e"     ("⊥")  where "⊥ ≡ λw. False"
abbreviation knot   :: "e⇒e"     ("¬_"[52]53)  where "¬φ ≡ λw. ¬φ(w)"
abbreviation kor    :: "e⇒e⇒e" (infixr"∨"50) where "φ∨ψ ≡ λw. φ(w)∨ψ(w)"
abbreviation kand   :: "e⇒e⇒e" (infixr"∧"51) where "φ∧ψ ≡ λw. φ(w)∧ψ(w)"
abbreviation kimp   :: "e⇒e⇒e" (infixr"⊃"49) where "φ⊃ψ ≡ λw. φ(w)⟶ψ(w)"
abbreviation kvalid :: "e⇒bool" ("⌊_⌋"[8]109)  where "⌊p⌋ ≡ ∀w. p w"

abbreviation "outpre ≡ λG.λa.λy::e. ∃f. ⌊a ⊃ f⌋ ∧ G (f,y)"

abbreviation "out1 ≡  λG.λa.λx. ⌊x⌋ ∨
   (∃i j k. outpre G a i ∧ outpre G a j ∧ outpre G a k ∧ ⌊(i ∧ j ∧ k) ⊃ x⌋)"
```

I/O-Logic in Isabelle/HOL



Preference-based DDL in Isabelle/HOL

Journal of Philosophical Logic / Vol. 43, No. 6, December 2014 / Maximality vs. Optim...

X. Parent

PrefDDL.thy

PrefDDL.thy (~/chris/trunk/tex/talks/2018-DEON/DEMO/)

```
246  (* axioms of proof theory for E, check for soundness *)
247  lemma classical: "A ⟹ ⌊(λw. A)⌋" by simp -- "all classical tautologies"
248
249  lemma "□M": "⌊□A ⟶ A⌋" by simp -- "part of S5 schema for □"
250  lemma "□5": "⌊◇A ⟶ □(◇A)⌋" by simp  -- "part of S5 schema for □"
251
252  lemma DfP: "⌊P⟨B|A⟩ ⟷ ¬(O⟨(¬B)|A⟩)⌋" by (simp add: prefDDLBase.truthSet_def)
253  lemma COK: "⌊O⟨(B ⟶ C)|A⟩ ⟶ (O⟨B|A⟩ ⟶ O⟨C|A⟩)⌋" by (simp add: prefDDLBase.truthSet_def)
254  lemma abs: "⌊O⟨B|A⟩ ⟶ □(O⟨B|A⟩)⌋" by simp
255  lemma nec: "⌊□A ⟶ O⟨A|B⟩⌋" by (simp add: prefDDLBase.truthSet_def)
256  lemma ext: "⌊□(A ⟷ B) ⟶ (O⟨C|A⟩ ⟷ O⟨C|B⟩)⌋" by (simp add: prefDDLBase.truthSet_def)
257  lemma id: "⌊O⟨A|A⟩⌋" by (simp add: optChoice)
258  lemma Sh: "⌊O⟨C|(A ∧ B)⟩ ⟶ O⟨(B ⟶ C)|A⟩⌋" by (smt optBest.optChoice optBest_axioms prefDDLBase.
259
260  (* soundness of inference rules *)
261  lemma MP: "⌊A⌋ ⟹ ⌊A ⟶ B⌋ ⟹ ⌊B⌋" by simp
262  lemma N: "⌊A⌋ ⟹ ⌊□A⌋" by simp
263
264  (* D* should hold in F, this can be verfied: *)
265  lemma "D*": "⌊◇A ⟶ (O⟨B|A⟩ ⟶ P⟨B|A⟩)⌋"
266    by (metis FOpt.opt_limitedness FOpt_axioms truthSet_def)
267
268  (* (CM) should not be provable in system F but only as of system F+CM, verified by nitpick *)
269  lemma CM: "⌊(O⟨B|A⟩ ∧ O⟨C|A⟩) ⟶ O⟨C|(A ∧ B)⟩⌋" nitpick oops
```

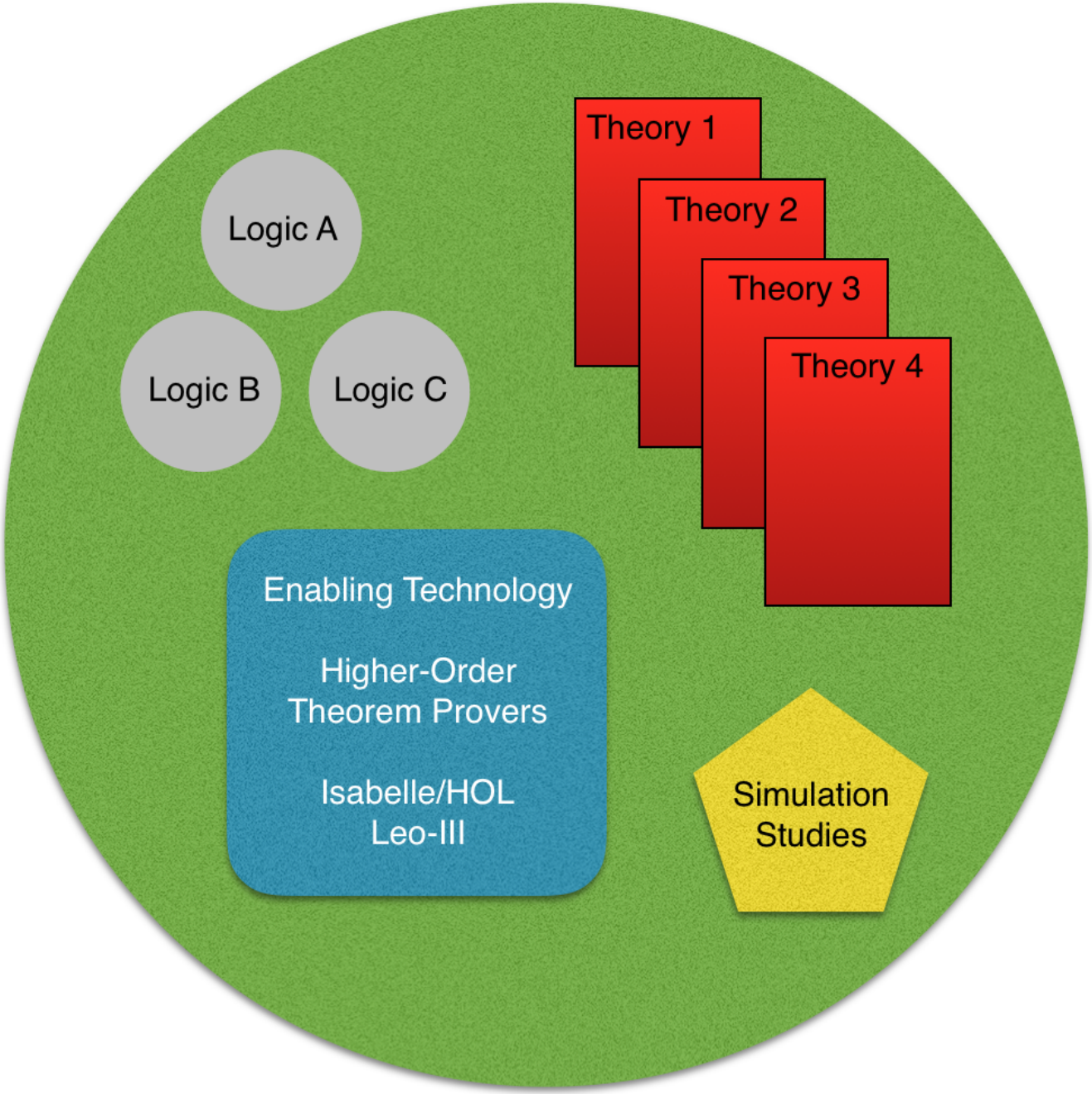
Documentation Sidekick State Theories

Proof state Auto update Update Search: 100%

```
Nitpicking formula...
Nitpick found a counterexample for card 'w = 3:

  Free variables:
    A = (λx. _)(w₁ := True, w₂ := True, w₃ := True)
    B = (λx. _)(w₁ := True, w₂ := True, w₃ := False)
    C = (λx. _)(w₁ := False, w₂ := True, w₃ := False)
    op ≳ =
      (λx. _)
      (w₁ := (λx. _)(w₁ := True, w₂ := True, w₃ := False),
       w₂ := (λx. _)(w₁ := True, w₂ := True, w₃ := True),
       w₃ := (λx. _)(w₁ := True, w₂ := False, w₃ := True))
    opt = (λx. _)
```

Output Query Sledgehammer Symbols

Normative Reasoning Experimentation Platform (Deontic Logics are required!)



What you should do in the next 60 years!

Let EasyChair make you rich!

But spent more time with Vampire

and also go in bed with Lambda's

What you should do in the next 60 years!

Let EasyChair make you rich!

But spent more time with Vampire

and also go in bed with Lambda's

What you should do in the next 60 years!

Let EasyChair make you rich!

But spent more time with Vampire

and also go in bed with Lambda's

Conclusion: HOL and Universal Reasoning deserves more attention!

- **Heureka steps:** can be expected in HOL, but less so in FOL or PL
- **Lean and elegant** approach to integrate and combine heterogeneous logics
- **Intuitive user interaction** at abstract level (in modern proof assistants)
- **Reuse** of existing systems and tools
- Important **link** between different **research communities**
- Approach very well suited for (interdisciplinary) **teaching of logics**

Leo-III — Universal Higher-Order Theorem Prover

- The Higher-Order Prover Leo-III, IJCAR 2018, Springer LNCS, 2018.
- Theorem Provers for Every Normal Modal Logic, LPAR 2017, Springer LNCS, 2018.

Universal (Meta-)Logical Reasoning

- Universal (Meta-)Logical Reasoning: Recent Successes, SCP, 2019.
- Universal (Meta-)Logical Reasoning: The Wise Men Puzzle, Data in Brief, 2019.
- Automating Free Logic in HOL, with an Experimental Application in Category Theory, JAR, 2019.

Deontic Logic Reasoning Infrastructure

- Designing Normative Theories of Ethical Reasoning, submitted, Url (preprint): here
- A Dyadic Deontic Logic in HOL, DEON 2018, 2018. (Best Paper Award)
- A Deontic Logic Reasoning Infrastructure, CiE 2018, Springer LNCS, 2018.
- Aqvist's Dyadic Deontic Logic E in HOL, Journal of Applied Logics, in print.
- I/O Logic in HOL, Journal of Applied Logics, in print.

Conclusion: HOL and Universal Reasoning deserves more attention!

- **Heureka steps:** can be expected in HOL, but less so in FOL or PL
- **Lean and elegant** approach to integrate and combine heterogeneous logics
- **Intuitive user interaction** at abstract level (in modern proof assistants)
- **Reuse** of existing systems and tools
- Important **link** between different **research communities**
- Approach very well suited for (interdisciplinary) **teaching of logics**

Leo-III — Universal Higher-Order Theorem Prover

- The Higher-Order Prover Leo-III, IJCAR 2018, Springer LNCS, 2018.
- Theorem Provers for Every Normal Modal Logic, LPAR 2017, Springer LNCS, 2018.

Universal (Meta-)Logical Reasoning

- Universal (Meta-)Logical Reasoning: Recent Successes, SCP, 2019.
- Universal (Meta-)Logical Reasoning: The Wise Men Puzzle, Data in Brief, 2019.
- Automating Free Logic in HOL, with an Experimental Application in Category Theory, JAR, 2019.

Deontic Logic Reasoning Infrastructure

- Designing Normative Theories of Ethical Reasoning, submitted, Url (preprint): here
- A Dyadic Deontic Logic in HOL, DEON 2018, 2018. (Best Paper Award)
- A Deontic Logic Reasoning Infrastructure, CiE 2018, Springer LNCS, 2018.
- Aqvist's Dyadic Deontic Logic E in HOL, Journal of Applied Logics, in print.
- I/O Logic in HOL, Journal of Applied Logics, in print.

Machine Ethics — Gewirth's Principle of Generic Consistency (PGC) in HOL

Emendation of the Golden Rule:

> "Act in accord with the generic rights of your recipients as well as of yourself. I shall call this the Principle of Generic Consistency (PGC), since it combines the formal consideration of consistency with the material consideration of rights to the generic features or goods of action." **(Alan Gewirth, Reason and Morality, 1978)**

- **Gewirth's PGC has**
  - stirred much controversy in moral philosophy
  - been discussed as means to bound the impact of artificial general intelligence (AGI)
- **Idea (in a nutshell):**
  - devise a safety mechanism of a mathematical (deductive) nature
  - to ensure that an AGI respects human's freedom and well-being
  - mechanism is based on assumption that it is able to recognize itself, as well as us humans, as agents (prospective purposive agents, PPA) which
    - act voluntarily on self-chosen purposes, and
    - reason rationally
- **References:**
  - A. Gewirth. Reason and morality. U of Chicago Press, 1978.
  - D. Beyleveld. The dialectical necessity of morality: An analysis and defense of Alan Gewirth's argument to the principle of generic consistency. U of Chicago Press, 1991.
  - A. Kornai. Bounding the impact of AGI. J. Experimental & Theoretical AI, 2014.
- **See our Automation of the PGC in Isabelle/HOL:**
  `https://www.isa-afp.org/entries/GewirthPGCProof.html`

But what if we don't want Full Comprehension?

This has been the case for Ed Zalta's

Principia-Logico Metaphysica

Zalta's Theory of Abstract Objects (https://mally.stanford.edu/theory.html)

Starting point:

- Higher-order modal logic S5
- Relational type-theory as foundation
- Functional type theory (HOL) not directly suited: comes with full comprehension which leads to paradoxes

Example of paradoxical term to exclude:

$$\lambda x \exists F(xF \wedge \neg Fx)$$

See: [Oppenheimer & Zalta, Relations versus functions at the foundations of logic: Type-theoretic considerations, Journal of Logic and Computation, 2011]

Challenge:

- Encoding of relational higher-order modal logic S5 in HOL
- Restricted comprehension to avoid paradoxes

Our solution in HOL: explicitly model denoting and non-denoting terms

[Computer Science and Metaphysics: A Cross-Fertilization, submitted]

[Mechanizing Principia Logico-Metaphysica in Functional Type Theory, submitted]

Russell discovered the well-known paradox in Cantor's set theory.

Similar construction possible in a naive representation of AOT in HOL

Assume that term $[\lambda x\ \exists F\,(xF\ \wedge\ \neg Fx)]$ denotes a valid property K. (property of being x, such that there exists a property that x encodes, but does not exemplify)

Comprehension axiom then ensures that there is an abstract object which encodes and only encodes K. The question whether this abstract object exemplifies K leads to paradox.

The relational formulation of AOT tries to avoid this issue by restricting the matrix of λ-expressions to so called *propositional formulas* only, that is, to formulas without encoding subformulas. This way the term $[\lambda x\ \exists F\,(xF\ \wedge\ \neg Fx)]$ is no longer well-formed and the construction of the paradox fails.

Solution in HOL: explicitly model denoting and non-denoting terms

[Kirchner, Benzmüller & Zalta, Computer Science and Metaphysics: A Cross-Fertilization, submitted, preprint: see my website]

[Kirchner, Benzmüller & Zalta, Mechanizing Principia Logico-Metaphysica in Functional Type Theory, submitted, preprint: see my website]