Typed λ-Calculus: Logical Constants

We gain expressive power by combining typed λ-calculus with logical constants.

Typed λ-Calculus: Logical Constants

We gain expressive power by combining typed λ-calculus with logical constants.

$\top_o$ – true

$\bot_o$ – false

$\neg_{oo}$ – negation

$\vee_{ooo}$ – disjunction

$\wedge_{ooo}$ – conjunction

$\supset_{ooo}$ – implication

$\equiv_{ooo}$ – equivalence

Typed λ-Calculus: Logical Constants

We gain expressive power by combining typed λ-calculus with logical constants.

$=^{\alpha}_{o\alpha\alpha}$ – equality at type α

$\Pi^{\alpha}_{o(o\alpha)}$ – universal quantification over type α

$\Sigma^{\alpha}_{o(o\alpha)}$ – existential quantification over type α

Intuition: $[\Sigma^{\alpha} . \lambda x_{\alpha} . C_o]$ is to true iff $\{x_{\alpha} | C\}$ is nonempty.

HOL: Abbreviations

$[A_o \lor B_o]$ means $[\lor_{ooo} A_o B_o]$

$[A_o \land B_o]$ means $[\land_{ooo} A_o B_o]$

$[A_o \supset B_o]$ means $[\supset_{ooo} A_o B_o]$

$[A_o \equiv B_o]$ means $[\equiv_{ooo} A_o B_o]$

$[A_\alpha =^\alpha B_\alpha]$ means $[=^\alpha_{o\alpha\alpha} A_\alpha B_\alpha]$

$[\forall x_\alpha . A_o]$ means $[\Pi^\alpha_{o(o\alpha)} . \lambda x_\alpha . A_o]$.

$[\exists x_\alpha . A_o]$ means $[\Sigma^\alpha_{o(o\alpha)} . \lambda x_\alpha . A_o]$.

HOL: Expressing Properties

$$[\lambda x_\iota . x^2 - 1]$$

HOL: Expressing Properties

$$[\lambda \mathsf{x}_\iota . \mathsf{x}^2 - 1]$$

$$[\lambda \mathsf{x}_\iota . [\mathsf{MINUS}_{\iota\iota\iota}\, [\mathsf{SQUARE}_{\iota\iota}\, \mathsf{x}] 1_\iota]]_{\iota\iota}$$

HOL: Expressing Properties

$$[\lambda x_\iota . x^2 - 1]$$

Term of type o expressing existence of an f with two roots:

$$[\exists f_{\iota\iota}.\exists n_\iota.\exists m_\iota.[[f\ n] =^\iota 0_\iota] \wedge [[f\ m] =^\iota 0_\iota] \wedge \neg[n =^\iota m]]_o$$

HOL: Expressing Properties

$$[\lambda \mathsf{x}_\iota . \mathsf{x}^2 - 1]$$

Term of type o expressing existence of an f with two roots:

$$[\underbrace{\exists \mathsf{f}_{\iota\iota}}_{\Sigma^{\iota\iota}\lambda \mathsf{f}_{\iota\iota}} . \exists \mathsf{n}_\iota . \exists \mathsf{m}_\iota . [[\mathsf{f}\ \mathsf{n}] =^\iota 0_\iota] \wedge [[\mathsf{f}\ \mathsf{m}] =^\iota 0_\iota] \wedge \neg [\mathsf{n} =^\iota \mathsf{m}]]_\mathsf{o}$$

HOL: Expressing Properties

$$[\lambda x_\iota . x^2 - 1]$$

Term of type o expressing existence of an f with two roots:

$$[\exists f_{\iota\iota} . \exists n_\iota . \exists m_\iota . \underbrace{[[f\ n] =^\iota 0_\iota]}_{[=^\iota [f\ n]\ 0]} \wedge [[f\ m] =^\iota 0_\iota] \wedge \neg[n =^\iota m]]_o$$

HOL: Expressing Properties

$$[\lambda x_\iota . x^2 - 1]$$

$$[\lambda x_\iota . [x^2 - 1] = 0]$$

HOL: Expressing Properties

$$[\lambda \mathsf{x}_\iota . \mathsf{x}^2 - 1]$$

$$[\lambda \mathsf{x}_\iota . [\mathsf{x}^2 - 1] = 0]$$

$$[\lambda \mathsf{x}_\iota . [=^\iota \ [\mathsf{MINUS}_{\iota\iota\iota} \ [\mathsf{SQUARE}_{\iota\iota} \ \mathsf{x}] 1_\iota] \ 0_\iota]]_{o\iota}$$

HOL: Expressing Properties

$$[\lambda x_\iota . x^2 - 1]$$

$$[\lambda x_\iota . [x^2 - 1] = 0]$$

Term of type o expressing existence of a set (characteristic function) P with two elements

$$[\exists P_{o\iota} . \exists m_\iota . \exists n_\iota . \, [P\, m] \wedge [P\, n] \wedge \neg [m = n]]_o$$

HOL: Expressing Properties

Suppose ι corresponds to real numbers.

Given constants: $<_{o\iota\iota}$, $\mathsf{ABS}_{\iota\iota}$, $\mathsf{MINUS}_{\iota\iota\iota}$

We can give the usual $\epsilon - \delta$ definition of limits.

HOL: Expressing Properties

Suppose ι corresponds to real numbers.

Given constants: $<_{\mathsf{o}\iota\iota}$, $\mathsf{ABS}_{\iota\iota}$, $\mathsf{MINUS}_{\iota\iota\iota}$

We can give the usual $\epsilon - \delta$ definition of limits.

$\mathsf{LIM}_{\mathsf{o}\iota\iota(\iota\iota)}$:

HOL: Expressing Properties

Suppose ι corresponds to real numbers.
Given constants: $<_{o\iota\iota}$, $\mathsf{ABS}_{\iota\iota}$, $\mathsf{MINUS}_{\iota\iota\iota}$
We can give the usual $\epsilon - \delta$ definition of limits.
$\mathsf{LIM}_{o\iota\iota(\iota\iota)}$:

$$[\lambda \mathsf{f}_{\iota\iota}.\lambda \mathsf{a}_{\iota}.\lambda \mathsf{L}_{\iota}.\forall \epsilon_{\iota}.[\epsilon > 0] \supset .\exists \delta_{\iota}.[\delta > 0]$$
$$\wedge .\forall \mathsf{x}_{\iota}.[|\mathsf{x} - \mathsf{a}| < \delta] \supset [|[\mathsf{f}\,\mathsf{x}] - \mathsf{L}| < \epsilon]]$$

HOL: Expressing Properties

Suppose ι corresponds to real numbers.
Given constants: $<_{o\iota\iota}$, $\mathsf{ABS}_{\iota\iota}$, $\mathsf{MINUS}_{\iota\iota\iota}$
We can give the usual $\epsilon - \delta$ definition of limits.
$\mathsf{LIM}_{o\iota\iota(\iota\iota)}$:

$$[\lambda \mathsf{f}_{\iota\iota}.\lambda \mathsf{a}_{\iota}.\lambda \mathsf{L}_{\iota}.\forall \epsilon_{\iota}.\overbrace{[\epsilon > 0]}^{[<0\,\epsilon]} \supset .\exists \delta_{\iota}.[\delta > 0]$$
$$\wedge .\forall \mathsf{x}_{\iota}.[|\mathsf{x} - \mathsf{a}| < \delta] \supset [|[\mathsf{f}\,\mathsf{x}] - \mathsf{L}| < \epsilon]]$$

HOL: Expressing Properties

Suppose ι corresponds to real numbers.
Given constants: $<_{o\iota\iota}$, $\mathsf{ABS}_{\iota\iota}$, $\mathsf{MINUS}_{\iota\iota\iota}$
We can give the usual $\epsilon - \delta$ definition of limits.
$\mathsf{LIM}_{o\iota\iota(\iota\iota)}$:

$$[\lambda \mathsf{f}_{\iota\iota}.\lambda \mathsf{a}_{\iota}.\lambda \mathsf{L}_{\iota}.\forall \epsilon_{\iota}.[\epsilon > 0] \supset .\exists \delta_{\iota}.\overbrace{[\delta > 0]}^{[<0\,\delta]}$$
$$\wedge .\forall \mathsf{x}_{\iota}.[|\mathsf{x} - \mathsf{a}| < \delta] \supset [|[\mathsf{f}\,\mathsf{x}] - \mathsf{L}| < \epsilon]]$$

HOL: Expressing Properties

Suppose ι corresponds to real numbers.
Given constants: $<_{o\iota\iota}$, $\mathsf{ABS}_{\iota\iota}$, $\mathsf{MINUS}_{\iota\iota\iota}$
We can give the usual $\epsilon - \delta$ definition of limits.
$\mathsf{LIM}_{o\iota\iota(\iota\iota)}$:

$$[\lambda \mathsf{f}_{\iota\iota}.\lambda \mathsf{a}_{\iota}.\lambda \mathsf{L}_{\iota}.\forall \epsilon_{\iota}.[\epsilon > 0] \supset .\exists \delta_{\iota}.[\delta > 0]$$
$$\wedge .\forall \mathsf{x}_{\iota}.[|\mathsf{x} - \mathsf{a}| < \delta] \supset [|[\mathsf{f}\,\mathsf{x}] - \mathsf{L}| < \epsilon]]$$

HOL: Expressing Properties

Suppose ι corresponds to real numbers.
Given constants: $<_{o\iota\iota}$, $\mathsf{ABS}_{\iota\iota}$, $\mathsf{MINUS}_{\iota\iota\iota}$
We can give the usual $\epsilon - \delta$ definition of limits.
$\mathsf{LIM}_{o\iota\iota(\iota\iota)}$:

$$[\lambda f_{\iota\iota}.\lambda a_{\iota}.\lambda L_{\iota}.\forall \epsilon_{\iota}.[\epsilon > 0] \supset .\exists \delta_{\iota}.[\delta > 0]$$

$$\wedge .\forall x_{\iota}.[|\underbrace{x - a}_{[\mathsf{MINUS}\, x\, a]}| < \delta] \supset [|[f\, x] - L| < \epsilon]]$$

HOL: Expressing Properties

Suppose ι corresponds to real numbers.
Given constants: $<_{o\iota\iota}$, $\mathsf{ABS}_{\iota\iota}$, $\mathsf{MINUS}_{\iota\iota\iota}$
We can give the usual $\epsilon - \delta$ definition of limits.
$\mathsf{LIM}_{o\iota\iota(\iota\iota)}$:

$$[\lambda \mathsf{f}_{\iota\iota}.\lambda \mathsf{a}_{\iota}.\lambda \mathsf{L}_{\iota}.\forall \epsilon_{\iota}.[\epsilon > 0] \supset .\exists \delta_{\iota}.[\delta > 0] \wedge .\forall \mathsf{x}_{\iota}.[\underbrace{|\mathsf{x} - \mathsf{a}|}_{[\mathsf{ABS}\,[\mathsf{MINUS}\,\mathsf{x}\,\mathsf{a}]]} < \delta] \supset [|[\mathsf{f}\,\mathsf{x}] - \mathsf{L}| < \epsilon]]$$

HOL: Expressing Properties

Suppose ι corresponds to real numbers.
Given constants: $<_{o\iota\iota}$, $\mathsf{ABS}_{\iota\iota}$, $\mathsf{MINUS}_{\iota\iota\iota}$
We can give the usual $\epsilon - \delta$ definition of limits.
$\mathsf{LIM}_{o\iota\iota(\iota\iota)}$:

$$[\lambda \mathsf{f}_{\iota\iota}.\lambda \mathsf{a}_{\iota}.\lambda \mathsf{L}_{\iota}.\forall \epsilon_{\iota}.[\epsilon > 0] \supset .\exists \delta_{\iota}.[\delta > 0]$$

$$\wedge .\forall \mathsf{x}_{\iota}. \quad \underbrace{[|\mathsf{x} - \mathsf{a}| < \delta]}_{[< [\mathsf{ABS}\, [\mathsf{MINUS}\, \mathsf{x}\, \mathsf{a}]]\, \delta]} \quad \supset [|[\mathsf{f}\, \mathsf{x}] - \mathsf{L}| < \epsilon]]$$

HOL: Expressing Properties

Suppose ι corresponds to real numbers.
Given constants: $<_{o\iota\iota}$, $\mathsf{ABS}_{\iota\iota}$, $\mathsf{MINUS}_{\iota\iota\iota}$
We can give the usual $\epsilon - \delta$ definition of limits.
$\mathsf{LIM}_{o\iota\iota(\iota\iota)}$:

$$[\lambda \mathsf{f}_{\iota\iota}.\lambda \mathsf{a}_{\iota}.\lambda \mathsf{L}_{\iota}.\forall \epsilon_{\iota}.[\epsilon > 0] \supset .\exists \delta_{\iota}.[\delta > 0]$$

$$\wedge .\forall \mathsf{x}_{\iota}.[|\mathsf{x} - \mathsf{a}| < \delta] \supset [|[\mathsf{f}\,\mathsf{x}] - \mathsf{L}| < \epsilon]]$$

HOL: Expressing Properties

Suppose ι corresponds to real numbers.
Given constants: $<_{o\iota\iota}$, $\mathsf{ABS}_{\iota\iota}$, $\mathsf{MINUS}_{\iota\iota\iota}$
We can give the usual $\epsilon - \delta$ definition of limits.
$\mathsf{LIM}_{o\iota\iota(\iota\iota)}$:

$$[\lambda \mathsf{f}_{\iota\iota}.\lambda \mathsf{a}_{\iota}.\lambda \mathsf{L}_{\iota}.\forall \epsilon_{\iota}.[\epsilon > 0] \supset .\exists \delta_{\iota}.[\delta > 0]$$
$$\wedge .\forall \mathsf{x}_{\iota}.[|\mathsf{x} - \mathsf{a}| < \delta] \supset [|[\mathsf{f}\,\mathsf{x}] - \mathsf{L}| < \epsilon]]$$

Similarly can define continuity, differentiation, etc.

HOL: Prefix Polymorphism

Some definitions are naturally expressed using type variables:

HOL: Prefix Polymorphism

Some definitions are naturally expressed using type variables:

Consider the notion of subset:

For each type α we can define $\subseteq_{\mathsf{o}(\mathsf{o}\alpha)(\mathsf{o}\alpha)}$ to be:

$$\lambda \mathsf{X}_{\mathsf{o}\alpha}.\, \lambda \mathsf{Y}_{\mathsf{o}\alpha}.\, [\forall \mathsf{z}_{\alpha}.\, [\mathsf{X}\ \mathsf{z}] \supset [\mathsf{Y}\ \mathsf{z}]]$$

HOL: Prefix Polymorphism

Some definitions are naturally expressed using type variables:
Consider the notion of subset:
For each type α we can define $\subseteq_{\mathsf{o(o\alpha)(o\alpha)}}$ to be:

$$\lambda \mathsf{X}_{\mathsf{o}\alpha}.\, \lambda \mathsf{Y}_{\mathsf{o}\alpha}.\, [\forall \mathsf{z}_\alpha.\, [\mathsf{X}\ \mathsf{z}] \supset [\mathsf{Y}\ \mathsf{z}]]$$

We can think of α as a type variable and $\subseteq_{\mathsf{o(o\alpha)(o\alpha)}}$ to be polymorphic.

HOL: Prefix Polymorphism

Some definitions are naturally expressed using type variables:
Consider the notion of subset:
For each type α we can define $\subseteq_{o(o\alpha)(o\alpha)}$ to be:

$$\lambda X_{o\alpha}.\,\lambda Y_{o\alpha}.\,[\forall z_{\alpha}.\,[X\ z] \supset [Y\ z]]$$

We can think of α as a type variable and $\subseteq_{o(o\alpha)(o\alpha)}$ to be polymorphic. In any particular occurrence of $\subseteq_{o(o\alpha)(o\alpha)}$ we should be able to instantiate the type variable α.

HOL: Prefix Polymorphism

Some definitions are naturally expressed using type variables:
Consider the notion of subset:
For each type α we can define $\subseteq_{o(o\alpha)(o\alpha)}$ to be:

$$\lambda X_{o\alpha}.\, \lambda Y_{o\alpha}.\, [\forall z_\alpha.\, [X\ z] \supset [Y\ z]]$$

We can think of α as a type variable and $\subseteq_{o(o\alpha)(o\alpha)}$ to be polymorphic. In any particular occurrence of $\subseteq_{o(o\alpha)(o\alpha)}$ we should be able to instantiate the type variable α.
Example: (using infix notation)

$$[\lambda U_{o\iota}.\, [U \subseteq_{o(o\iota)(o\iota)} X_{o\iota}]] \subseteq_{o(o(o\iota))(o(o\iota))} [\lambda U_{o\iota}.\, [U \subseteq_{o(o\iota)(o\iota)} Y_{o\iota}]]$$

HOL: Cantor's Theorem

There is no surjection from a set A onto the power set $\mathcal{P}(A)$ of A.

HOL: Cantor's Theorem

There is no surjection from a set A onto the power set $\mathcal{P}(\mathrm{A})$ of A.

- Suppose A corresponds to type ι.

HOL: Cantor's Theorem

There is no surjection from a set A onto the power set $\mathcal{P}(\mathrm{A})$ of A.

- Suppose A corresponds to type ι.

- Then $\mathcal{P}(\mathrm{A})$ corresponds to type $(\mathrm{o}\iota)$.

HOL: Cantor's Theorem

There is no surjection from a set A onto the power set $\mathcal{P}(A)$ of A.

- Suppose A corresponds to type ι.

- Then $\mathcal{P}(A)$ corresponds to type $(o\iota)$.

$$\neg \exists\, g_{o\iota\iota} . \forall\, f_{o\iota} . \exists\, x_\iota \,.\, g\, x =^{o\iota} f$$

HOL: Standard Higher-Order Model

$\mathcal{D}_\iota$ (individuals)

HOL: Standard Higher-Order Model

$\mathcal{P}(\mathcal{D}_\iota)$	(all sets)
$\mathcal{D}_\iota$	(individuals)

HOL: Standard Higher-Order Model

(all sets of sets)

$\mathcal{P}(\mathcal{P}(\mathcal{D}_\iota))$

$\mathcal{P}(\mathcal{D}_\iota)$ (all sets)

$\mathcal{D}_\iota$ (individuals)

HOL: Standard Higher-Order Model

$$\vdots$$

$$\mathcal{P}(\mathcal{P}(\mathcal{D}_\iota))$$

$$\mathcal{P}(\mathcal{D}_\iota)$$

$$\mathcal{D}_\iota$$

HOL: Henkin-Style Model

$\mathcal{D}_{o\iota} \subseteq \mathcal{P}(\mathcal{D}_\iota)$ (some sets)

$\mathcal{D}_\iota$ (individuals)

HOL: Henkin-Style Model

(some sets of sets)

$\mathcal{D}_{o(o\iota)} \subseteq \mathcal{P}(\mathcal{D}_{o\iota})$

$\mathcal{D}_{o\iota} \subseteq \mathcal{P}(\mathcal{D}_{\iota})$ (some sets)

$\mathcal{D}_{\iota}$ (individuals)

HOL: Henkin-Style Model

$$\vdots$$

$$\mathcal{D}_{o(o\iota)} \subseteq \mathcal{P}(\mathcal{D}_{o\iota})$$

$$\mathcal{D}_{o\iota} \subseteq \mathcal{P}(\mathcal{D}_{\iota})$$

$$\mathcal{D}_{\iota}$$

Classical Higher-Order Logic (HOL)

(Church's Type Theory)

Classical Higher-Order Logic (HOL)

Expressivity	FOL	HOL	Example
Quantification over			
- Individuals	$\checkmark$	$\checkmark$	$\forall X\ p(f(X))$
- Functions	−	$\checkmark$	$\forall F\ p(F(a))$
- Predicates/Sets/Rels	−	$\checkmark$	$\forall P\ P(f(a))$
Unnamed			
- Functions	−	$\checkmark$	$(\lambda X\ X)$
- Predicates/Sets/Rels	−	$\checkmark$	$(\lambda X\ X \neq a)$
Statements about			
- Functions	−	$\checkmark$	$\mathit{continuous}(\lambda X\ X)$
- Predicates/Sets/Rels	−	$\checkmark$	$\mathit{reflexive}(=)$
Powerful abbreviations	−	$\checkmark$	$\mathit{reflexive} = \lambda R\ \lambda X\ R(X, X)$

Classical Higher-Order Logic (HOL)

Expressivity	FOL	HOL	Example
Quantification over			
- Individuals	✓	✓	$\forall X_\iota\, p_{\iota\to o}(f_{\iota\to\iota}(X_\iota))$
- Functions	–	✓	$\forall F_{\iota\to\iota}\, p_{\iota\to o}(F_{\iota\to o}(a_\iota))$
- Predicates/Sets/Rels	–	✓	$\forall P_{\iota\to o}\, P_{\iota\to o}(f_{\iota\to\iota}(a_\iota))$
Unnamed			
- Functions	–	✓	$(\lambda X_\iota\, X_\iota)$
- Predicates/Sets/Rels	–	✓	$(\lambda X_{\iota\to\iota}\, X_{\iota\to\iota} \neq_{\iota\to\iota\to p} a)_\iota)$
Statements about			
- Functions	–	✓	$continuous_{(\iota\to\iota)\to o}(\lambda X_\iota\, X_\iota)$
- Predicates/Sets/Rels	–	✓	$reflexive_{(\iota\to\iota\to o)\to o}(=_{\iota\to\iota\to o})$
Powerful abbreviations	–	✓	$reflexive_{(\iota\to\iota\to o)\to o} = \lambda R_{(\iota\to\iota\to o)}\, \lambda X_\iota\, R(X,X)$

Simple Types: Prevent Paradoxes and Inconsistencies

Classical Higher-Order Logic (HOL) / Church's Simple Type Theory

- Simple Types $\alpha ::= \iota \mid o \mid \alpha_1 \to \alpha_2$

Classical Higher-Order Logic (HOL) / Church's Simple Type Theory

- Simple Types $\alpha ::= \iota \mid o \mid \alpha_1 \to \alpha_2$

  Individuals

  Booleans (True and False)

  Functions

Classical Higher-Order Logic (HOL) / Church's Simple Type Theory

- Simple Types $\alpha ::= \mu \mid \iota \mid o \mid \alpha_1 \rightarrow \alpha_2$

Possible worlds

Individuals

Booleans (True and False)

Functions

Classical Higher-Order Logic (HOL) / Church's Simple Type Theory

- Simple Types $\alpha ::= \mu \mid \iota \mid o \mid \alpha_1 \to \alpha_2$
- HOL Syntax

$$
\begin{aligned}
s, t \quad ::= \quad & c_\alpha \mid X_\alpha \mid (\lambda X_\alpha \, s_\beta)_{\alpha\to\beta} \mid (s_{\alpha\to\beta} \, t_\alpha)_\beta \\
& \mid (\neg_{o\to o} \, s_o)_o \mid (s_o \vee_{o\to o\to o} t_o)_o \mid (\forall X_\alpha \, t_o)_o
\end{aligned}
$$

Constant Symbols
Variable Symbols

Classical Higher-Order Logic (HOL) / Church's Simple Type Theory

- Simple Types $\alpha ::= \mu \mid \iota \mid o \mid \alpha_1 \rightarrow \alpha_2$
- HOL Syntax

$$
\begin{aligned}
s, t \quad ::= \quad & c_\alpha \mid X_\alpha \mid (\lambda X_\alpha\, s_\beta)_{\alpha\rightarrow\beta} \mid (s_{\alpha\rightarrow\beta}\, t_\alpha)_\beta \\
& \mid (\neg_{o\rightarrow o}\, s_o)_o \mid (s_o \vee_{o\rightarrow o\rightarrow o}\, t_o)_o \mid (\forall X_\alpha\, t_o)_o
\end{aligned}
$$

Constant Symbols
Variable Symbols
Abstraction
Application

Classical Higher-Order Logic (HOL) / Church's Simple Type Theory

- Simple Types $\alpha ::= \mu \mid \iota \mid o \mid \alpha_1 \rightarrow \alpha_2$
- HOL Syntax

$$
\begin{array}{rcl}
s, t & ::= & c_\alpha \mid X_\alpha \mid (\lambda X_\alpha\, s_\beta)_{\alpha\rightarrow\beta} \mid (s_{\alpha\rightarrow\beta}\, t_\alpha)_\beta \\
 & & \mid (\neg_{o\rightarrow o}\, s_o)_o \mid (s_o \vee_{o\rightarrow o\rightarrow o}\, t_o)_o \mid (\forall X_\alpha\, t_o)_o
\end{array}
$$

Constant Symbols
Variable Symbols
Abstraction
Application
Logical Connectives

Classical Higher-Order Logic (HOL) / Church's Simple Type Theory

- Simple Types $\alpha ::= \mu \mid \iota \mid o \mid \alpha_1 \rightarrow \alpha_2$
- HOL Syntax

$$
\begin{array}{rcl}
s, t & ::= & c_\alpha \mid X_\alpha \mid (\lambda X_\alpha \, s_\beta)_{\alpha \rightarrow \beta} \mid (s_{\alpha \rightarrow \beta} \, t_\alpha)_\beta \\
 & & \mid (\neg_{o \rightarrow o} \, s_o)_o \mid (s_o \vee_{o \rightarrow o \rightarrow o} t_o)_o \mid \underbrace{(\forall X_\alpha \, t_o)_o}_{(\Pi_{(\alpha \rightarrow o) \rightarrow o} (\lambda X_\alpha \, t_o))_o}
\end{array}
$$

Classical Higher-Order Logic (HOL) / Church's Simple Type Theory

- Simple Types $\alpha ::= \mu \mid \iota \mid o \mid \alpha_1 \rightarrow \alpha_2$
- HOL Syntax

$$\begin{array}{rcl} s, t & ::= & c_\alpha \mid X_\alpha \mid (\lambda X_\alpha\, s_\beta)_{\alpha\rightarrow\beta} \mid (s_{\alpha\rightarrow\beta}\, t_\alpha)_\beta \\ & & \mid (\neg_{o\rightarrow o}\, s_o)_o \mid (s_o \vee_{o\rightarrow o\rightarrow o} t_o)_o \mid (\Pi_{(\alpha\rightarrow o)\rightarrow o}\, (\lambda X_\alpha\, t_o))_o \end{array}$$

- Terms of type o: formulas

Classical Higher-Order Logic (HOL) / Church's Simple Type Theory

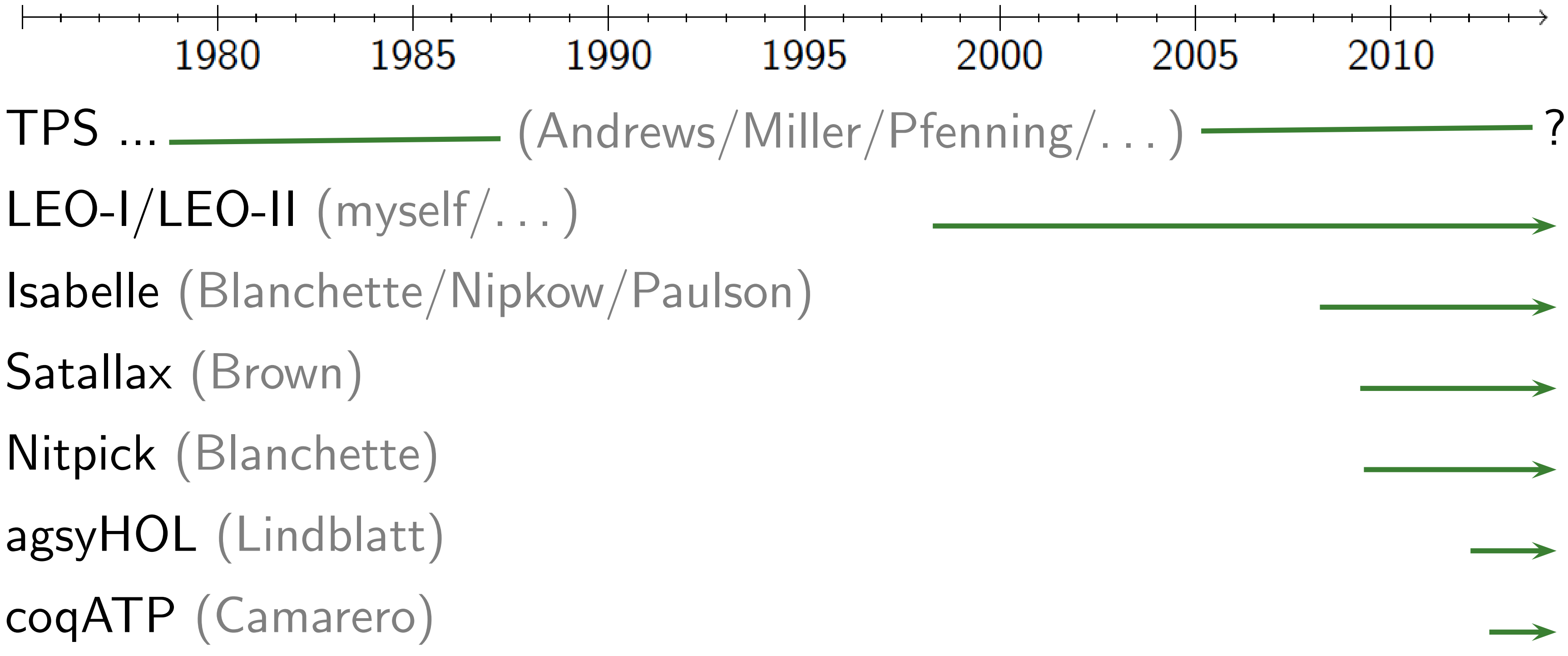
- Simple Types $\alpha ::= \mu \mid \iota \mid o \mid \alpha_1 \to \alpha_2$
- HOL Syntax

$$\begin{aligned} s, t \quad ::= \quad & c_\alpha \mid X_\alpha \mid (\lambda X_\alpha\, s_\beta)_{\alpha\to\beta} \mid (s_{\alpha\to\beta}\, t_\alpha)_\beta \\ & \mid (\neg_{o\to o}\, s_o)_o \mid (s_o \vee_{o\to o\to o} t_o)_o \mid (\Pi_{(\alpha\to o)\to o}\, (\lambda X_\alpha\, t_o))_o \end{aligned}$$

- Terms of type o: formulas
- HOL is (meanwhile) well understood
  - Origin [Church, J.Symb.Log., 1940]
  - Henkin-Semantics [Henkin, J.Symb.Log., 1950]
    [Andrews, J.Symb.Log., 1971, 1972]
  - Extens./Intens. [BenzmüllerEtAl., J.Symb.Log., 2004]
    [Muskens, J.Symb.Log., 2007]
- HOL with Henkin-Semantics: semi-decidable & compact (like FOL)

Higher-Order Automated Theorem Provers
(HOL-ATPs)

HOL-ATPs



- all accept TPTP THF0 syntax
- can be called remotely via SystemOnTPTP at Miami
- they significantly gained in strength over the last years
- they can be bundled into a combined prover HOL-P

HOL-ATPs

EU FP7 Project THFTPTP

- Collaboration with Geoff Sutcliffe and others (Chad Brown, Florian Rabe, Nik Sultana, Jasmin Blanchette, Frank Theiss, . . .)
- Results
  - THF0 syntax for HOL (with Choice; Henkin Semantics)
  - library with example problems (e.g. entire TPS library) and results
  - international CASC competition for HOL-ATP
  - online access to provers
  - various tools

More information: [SutcliffeBenzmüller, J.FormalizedReasoning, 2010]

`http://cordis.europa.eu/result/report/rcn/45614_en.html`

HOL-ATPs: CASC Competitions since 2009

- **2009:** Winner TPS
- **2010:** Winner LEO-II 1.2 solved 56% more (than previous winner)
- **2011:** Winner Satallax 2.1 solved 21% more
- **2012:** Winner Isabelle-HOT-2012 solved 35% more
- **2013:** Winner Satallax-MaLeS solved 21% more

Some Applications in
Mathematics & Philosophy & AI

Some Applications: Mathematics

ATPs as external reasoners in Interactive Proof Assistants

[KaliszykUrban, Learning-Assisted Automated Reasoning with Flyspeck, JAR, 2014]

- Flyspeck project: formal proof (in HOL-light) of Kepler's Conjecture
- automation of 14185 theorems studied by Kaliszyk and Urban
- they developed AI architecture employing various external ATPs in which 39 % of the theorems could be proved in a push-button mode in 30 seconds of real time on a fourteen-CPU workstation
- subset of 1419 theorems extracted from Flyspeck theorems
- **next slide:** performance of THF0 provers on these 1419 problems

Some Applications: Mathematics

Table 7 All ATP re-proving with 30s time limit on 10 % of problems

Prover	Theorem (%)	Unique	SOTAC	Σ-SOTAC	CounterSat (%)	Processed
Isabelle	587 (41.3)	39	0.201	118.09	0 (0.0)	1419
Epar	545 (38.4)	9	0.131	71.18	0 (0.0)	1419
Z3	513 (36.1)	17	0.149	76.49	0 (0.0)	1419
E 1.6	463 (32.6)	0	0.101	46.69	0 (0.0)	1419
LEO2-po1	441 (31.0)	1	0.106	46.85	0 (0.0)	1419
Vampire	434 (30.5)	3	0.107	46.44	0 (0.0)	1419
CVC3	411 (28.9)	4	0.111	45.76	0 (0.0)	1419
Satallax	383 (26.9)	7	0.130	49.69	1 (0.0)	1419
Yices	360 (25.3)	0	0.097	35.06	0 (0.0)	1419
iProver	348 (24.5)	0	0.088	30.50	9 (0.6)	1419
Prover9	345 (24.3)	0	0.087	30.07	0 (0.0)	1419
Metis	331 (23.3)	0	0.085	28.23	0 (0.0)	1419
SPASS	326 (22.9)	0	0.081	26.46	0 (0.0)	1419
leanCoP	305 (21.4)	1	0.092	27.96	0 (0.0)	1419

Some Applications: Philosophy

Theoretical Philosophy and Metaphysics

[Benzmüller&Woltzenlogel-Paleo, AutomatingGödel'sOntologicalProof, ECAI, 2014]

- First-time verification/automation of a modern ontological argument

Gödel's/Scott's proof of the existence of God

- Remember Leibniz: Two debating philosophers . . . Calculemus!
- Gödel's argument employs Higher-Order Modal Logic

Login | Registrierung

SPIEGEL ONLINE WISSENSCHAFT

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

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

Formel von Kurt Gödel: Mathematiker bestätigen Gottesbeweis

Von Tobias Hürter

picture-alliance/ Imagno/ Wiener Stadt- und Landesbibliothek

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

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

Montag, 09.09.2013 – 12:03 Uhr

Drucken | Versenden | Merken

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

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

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

Italy
- Repubblica
- Ilsussidario
- . . .

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

US
- ABC News
- . . .

International
- Spiegel International
- United Press Intl.
- . . .

Many more links at: `https://github.com/FormalTheology/GoedelGod`

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

Sign In | Register

SPIEGEL ONLINE INTERNATIONAL

Front Page World Europe Germany Business Zeitgeist Newsletter

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

Holy Logic: Computer Scientists 'Prove' God Exists

By David Knight

picture-alliance/ Imagno/ Wiener Stadt- und Landesbibliothek

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

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

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

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

Italy
- Repubblica
- Ilsussidario
- . . .

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

US
- ABC News
- . . .

International
- Spiegel International
- United Press Intl.
- . . .

Many more links at: `https://github.com/FormalTheology/GoedelGod`

Some Applications: Artificial Intelligence

Quantified Conditional Logics (QCLs)

[Benzmüller, AutomatingQuantifiedConditionalLogicsInHOL, IJCAI, 2013]

- known as logics of normality or typicality
- many applications: action planning, counterfactual reasoning, default reasoning, deontic reasoning, reasoning about knowledge, ...
- examples [Delgrande, Artif.Intell., 1998]:
  "Birds normally fly, penguins normally do not fly and all penguins are necessarily birds."
- not yet widely studied
- no direct provers implemented so far

- automation of QCLs possible in HOL (via semantic embedding)
- cut-elimination as a side result