Sweet SIXTEEN

Automation via Embedding into Classical Higher-Order Logic[1]

Alexander Steen Christoph Benzmüller
Freie Universität Berlin

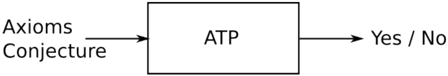
Non-Classical Logic. Theory and Applications. Seventh Edition.
Torun, Poland, 2015

[1] This work has been supported by the German National Research Foundation (DFG) under grant BE 2501/11-1 (Leo-III).

Freie Universität Berlin

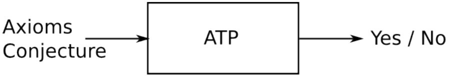
Why use Automation?

- Use automated deduction tools for assistance in verifying/refuting arguments
  - Especially interested in using higher-order ATP



Why use Automation?

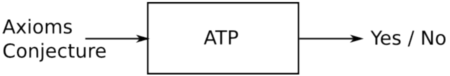
- Use automated deduction tools for assistance in verifying/refuting arguments
  - Especially interested in using higher-order ATP
- That should also be possible in non-classical logics
  - *Feasible automation often hard*



Freie Universität Berlin

Why use Automation?

- Use automated deduction tools for assistance in verifying/refuting arguments
  - Especially interested in using higher-order ATP
- That should also be possible in non-classical logics
  - *Feasible automation often hard*
- Applications in linguistics, computer science and philosophy
  - *Formalization of arguments in domain-specific logic*



Motivation

Can anything (non-trivial) be done with it?
Yes, at least to some extent

- Some reasoning tools reached maturity and offer fair usability, e.g. Isabelle [NipkowPaulsenWenzel,202]

- Proof of *Kepler's conjecture* (Flyspeck project) [Hales et al., 2014]

- Formal verification/inspection of *Gödel's ontological proof* and various versions of it [BenzmüllerWoltzenlogel Paleo, ECAI 2014]

Automation of Non-Classical Logics

We want

- Automation of non-classical logics
- Reasoning with uncertainty, vagueness

But often

- Proof calculi suited for automation scarce
- and so are tools

Automation of Non-Classical Logics

We want

- Automation of non-classical logics
- Reasoning with uncertainty, vagueness

But often

- Proof calculi suited for automation scarce
- and so are tools

Instead, we

- exploit the expressiveness of classical higher-order logic
- and encode the target logic within HOL

SIXTEEN

Here: Focus on SIXTEEN [ShramkoWansing,2011]

Truth-values given by

$$\begin{aligned}\mathbf{16} &= 2^{2^{\{T,F\}}}\\ &= 2^{\{\mathbf{N},\mathbf{T},\mathbf{F},\mathbf{B}\}}\\ &= \{\mathbf{N}, \boldsymbol{N}, \boldsymbol{T}, \boldsymbol{F}, \boldsymbol{B}, \mathbf{NT}, \mathbf{NF}, \ldots, \mathbf{NTFB}\}\end{aligned}$$

- Generalization of well-known Dunn/Belnap 4-valued system
- yielding *Trilattice* $(\mathbf{16}, \sqcap_i, \sqcup_i, \sqcap_t, \sqcup_t, \sqcap_f, \sqcup_f)$

SIXTEEN

Orders

by information:

$$x \leq_i y :\Leftrightarrow x \subseteq y$$

by truthhood:

$$x \leq_t y :\Leftrightarrow x^t \subseteq y^t \wedge y^{-t} \subseteq x^{-t}$$

by falsehood:

$$x \leq_f y :\Leftrightarrow x^f \subseteq y^f \wedge y^{-f} \subseteq x^{-f}$$

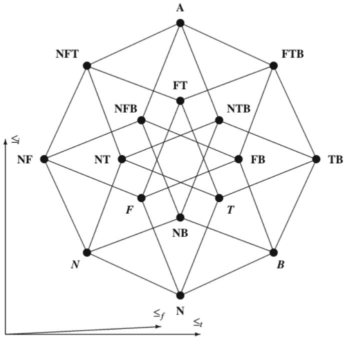


Image: Shramko, Wansing, *Truth and Falsehood*

Operations

for each (independent) axis

Meet	$\sqcup_\star$	Join	$\sqcap_\star$
Inversion	$-_\star$	for $\star \in \{i, t, f\}$	

Languages

$$\begin{array}{rl} \mathcal{L}_t: & A ::= x \mid \sim_t A \mid A \wedge_t A \mid A \vee_t A \\ \mathcal{L}_f: & A ::= x \mid \sim_f A \mid A \wedge_f A \mid A \vee_f A \\ \mathcal{L}_{tf}: & A ::= x \mid \sim_t A \mid \sim_f A \mid A \wedge_t A \mid A \vee_t A \mid A \wedge_f A \mid A \vee_f A \end{array}$$

Languages based on SIXTEEN

Languages

$$\begin{aligned}
\mathcal{L}_t:&\quad A ::= x \mid \sim_t A \mid A \wedge_t A \mid A \vee_t A\\
\mathcal{L}_f:&\quad A ::= x \mid \sim_f A \mid A \wedge_f A \mid A \vee_f A\\
\mathcal{L}_{tf}:&\quad A ::= x \mid \sim_t A \mid \sim_f A \mid A \wedge_t A \mid A \vee_t A \mid A \wedge_f A \mid A \vee_f A
\end{aligned}$$

Value of formulae

$$\begin{aligned}
v(A \wedge_\star B) &= v(A) \sqcap_\star v(B)\\
v(A \vee_\star B) &= v(A) \sqcup_\star v(B)\\
v(\sim_\star A) &= -_\star v(A)
\end{aligned}$$

Validity

$$A \models_\star B :\Leftrightarrow v(A) \leq_\star v(B) \text{ for all valuations } v$$

for $\star \in \{t, f\}$

Sadly, no deduction system is available!

- in fact, for many non-classical logics

To that end:

We embed $\mathcal{L}_\star$ in classical higher-order logic (HOL).

Higher Order Logic (HOL)

Due to Alonzo Church's "Simple type theory"
[Church, J.Symb.L., 1940]

- Simple types T generated by base types and $\to$
- Typically, base types are o and i

Higher Order Logic (HOL)

Due to Alonzo Church's "Simple type theory"
[Church, J.Symb.L., 1940]

- Simple types T generated by base types and $\rightarrow$
- Typically, base types are o and i

Type of truth-values

Higher Order Logic (HOL)

Due to Alonzo Church's "Simple type theory"
[Church, J.Symb.L., 1940]

- Simple types T generated by base types and $\rightarrow$
- Typically, base types are o and i

Type of indivinduals

Higher Order Logic (HOL)

Due to Alonzo Church's "Simple type theory"
[Church, J.Symb.L., 1940]

- Simple types T generated by base types and $\rightarrow$
- Typically, base types are o and i

- Terms defined by ($\alpha \in T$)

$$s, t ::= p_\alpha \mid X_\alpha$$

Higher Order Logic (HOL)

Due to Alonzo Church's "Simple type theory"
[Church, J.Symb.L., 1940]

- Simple types T generated by base types and $\rightarrow$
- Typically, base types are o and i

- Terms defined by ($\alpha \in T$)

$$
\begin{aligned}
s, t ::= & \, p_\alpha \mid X_\alpha \\
& \mid (\lambda X_\alpha . s_\beta)_{\alpha \rightarrow \beta} \mid (s_{\alpha \rightarrow \beta}\, t_\alpha)_\beta
\end{aligned}
$$

Higher Order Logic (HOL)

Due to Alonzo Church's "Simple type theory"
[Church, J.Symb.L., 1940]

- Simple types T generated by base types and $\rightarrow$
- Typically, base types are o and i

- Terms defined by ($\alpha \in T$)

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

Higher Order Logic (HOL)

Due to Alonzo Church's "Simple type theory"
[Church, J.Symb.L., 1940]

- Simple types T generated by base types and $\rightarrow$
- Typically, base types are o and i

- Terms defined by ($\alpha \in T$)

$$
\begin{aligned}
s, t ::= & \; p_\alpha \mid X_\alpha \\
& \mid (\lambda X_\alpha . s_\beta)_{\alpha\rightarrow\beta} \mid (s_{\alpha\rightarrow\beta} \, t_\alpha)_\beta \\
& \mid (\neg_{o\rightarrow o} s_o)_o \mid (s_o \vee_{o\rightarrow o\rightarrow o} t_o)_o \mid \Pi^\alpha{}_{(\alpha\rightarrow o)\rightarrow o} \, (\lambda X_\alpha . \, s_o)_{\alpha\rightarrow o}
\end{aligned}
$$

Higher Order Logic (HOL)

Due to Alonzo Church's "Simple type theory"
[Church, J.Symb.L., 1940]

- Simple types T generated by base types and $\rightarrow$
- Typically, base types are o and i

- Terms defined by ($\alpha \in T$)

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

- Formulae of HOL are those terms with type o

Semantics: Models

Definition (Models for HOL)

A *model* for HOL M is a tuple $M = (D, I)$ such that

- $D = \{D_\alpha\}_{\alpha \in T}$ is a frame, that is, a family of non-empty sets D_α s.t.
  - $D_o = \{T, F\}$ is the set of truth and falsehood
  - $D_{\alpha \to \beta}$ is a set of functions mapping D_α into D_β
- $I = \{I_\alpha\}_{\alpha \in T}$ is a family of typed interpretation functions mapping each p_α to an element of D_α

Semantics: Models

Definition (Models for HOL)

A *model* for HOL M is a tuple $M = (D, I)$ such that

- $D = \{D_\alpha\}_{\alpha \in T}$ is a frame, that is, a family of non-empty sets D_α s.t.
  - $D_o = \{T, F\}$ is the set of truth and falsehood
  - $D_{\alpha \to \beta}$ is a set of functions mapping D_α into D_β
- $I = \{I_\alpha\}_{\alpha \in T}$ is a family of typed interpretation functions mapping each p_α to an element of D_α

- Value of a statement s denoted by $\|s\|^{M,g}$ $\left(\begin{array}{l}\text{wrt. model } M \text{ and} \\ \text{var. assignment } g\end{array}\right)$

Definition (Models for HOL)

A *model* for HOL M is a tuple $M = (D, I)$ such that

- $D = \{D_\alpha\}_{\alpha \in T}$ is a frame, that is, a family of non-empty sets D_α s.t.
  - $D_o = \{T, F\}$ is the set of truth and falsehood
  - $D_{\alpha \to \beta}$ is a set of functions mapping D_α into D_β
- $I = \{I_\alpha\}_{\alpha \in T}$ is a family of typed interpretation functions mapping each p_α to an element of D_α

- Value of a statement s denoted by $\|s\|^{M,g}$ $\left(\begin{array}{l}\text{wrt. model } M \text{ and}\\ \text{var. assignment } g\end{array}\right)$

- M is a *standard model* iff $D_{\alpha \to \beta} = D_\beta^{D_\alpha}$
- ... a *Henkin model* iff $D_{\alpha \to \beta} \subseteq D_\beta^{D_\alpha}$ and $\|.\|^{M,g}$ is a total function

Semantics: Models

Definition (Models for HOL)

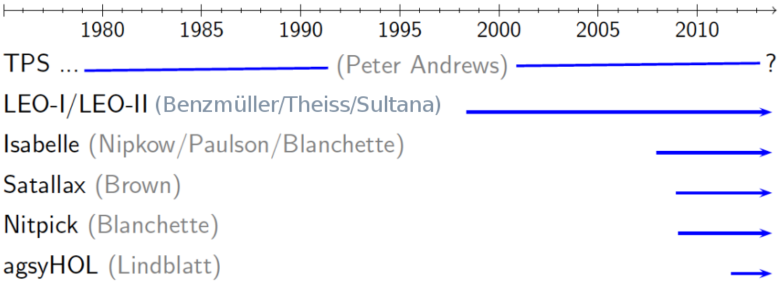
A *model* for HOL M is a tuple $M = (D, I)$ such that

- $D = \{D_\alpha\}_{\alpha \in T}$ is a frame, that is, a family of non-empty sets D_α s.t.
  - $D_o = \{T, F\}$ is the set of truth and falsehood
  - $D_{\alpha \to \beta}$ is a set of functions mapping D_α into D_β
- $I = \{I_\alpha\}_{\alpha \in T}$ is a family of typed interpretation functions mapping each p_α to an element of D_α

- Value of a statement s denoted by $\|s\|^{M,g}$ $\begin{pmatrix} \text{wrt. model } M \text{ and} \\ \text{var. assignment } g \end{pmatrix}$
- M is a *standard model* iff $D_{\alpha \to \beta} = D_\beta^{D_\alpha}$
- ... a *Henkin model* iff $D_{\alpha \to \beta} \subseteq D_\beta^{D_\alpha}$ and $\|.\|^{M,g}$ is a total function
- Validity $\models^{HOL}$ is then defined as $M, g \models^{HOL} s_o :\Leftrightarrow \|s_o\|^{M,g} = T$

Automation of HOL

- Quite a number of systems available

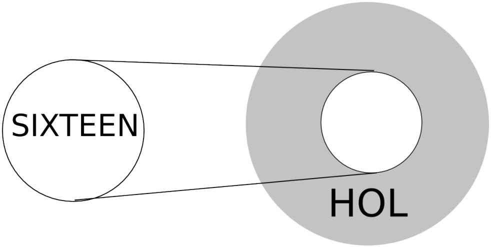


- TPTP THF Syntax [SutcliffeBenzmüller, J. Form. Reas., 2009]
- Can be called remotely via SystemOnTPTP at Miami [Sutcliffe, J. Autom. Reas., 2009]

Encoding of SIXTEEN in HOL

We now encode ($\star \in \{t, f\}$)

- the truth-degrees of SIXTEEN
- the orderings $\leq_\star$
- the operations $\sqcup_\star, \sqcap_\star, -_\star$



Truth-degree mapping

Sets in HOL

Sets M are represented by their characteristic function

$$f_M(x) = \begin{cases} T & \text{, iff } x \in M \\ F & \text{ otherwise} \end{cases}$$

Truth-degree mapping

Sets in HOL

Sets M are represented by their characteristic function

$$f_M(x) = \begin{cases} T & \text{, iff } x \in M \\ F & \text{otherwise} \end{cases}$$

Set representation

Object	$\longrightarrow$	*Representation in HOL*
Set M of objects of type α	$\longrightarrow$	$f_M : \alpha \to o$
e.g. $\{x \in T \mid P(x)\}$	$\longrightarrow$	$\lambda x_T.\, P_{T \to o}\, x$

Truth-degree mapping

Freie Universität Berlin

Sets in HOL

Sets M are represented by their characteristic function

$$f_M(x) = \begin{cases} T & \text{, iff } x \in M \\ F & \text{otherwise} \end{cases}$$

Set representation

Object	$\longrightarrow$	*Representation in HOL*
Set M of objects of type α	$\longrightarrow$	$f_M : \alpha \to o$
e.g. $\{x \in T \mid P(x)\}$	$\longrightarrow$	$\lambda x_T.\, P_{T \to o}\, x$

Truth-degrees as HOL sets

Truth-degrees of SIXTEEN are sets of sets of Booleans, thus of type

$$(o \to o) \to o$$

Truth-degree mapping

Sets in HOL

Sets M are represented by their characteristic function

$$f_M(x) = \begin{cases} T & \text{, iff } x \in M \\ F & \text{otherwise} \end{cases}$$

Set representation

Object	$\longrightarrow$	*Representation in HOL*
Set M of objects of type α	$\longrightarrow$	$f_M : \alpha \to o$
e.g. $\{x \in T \mid P(x)\}$	$\longrightarrow$	$\lambda x_T .\, P_{T \to o}\, x$

Truth-degrees as HOL sets

Truth-degrees of SIXTEEN are sets of sets of Booleans, thus of type

$$(o \to o) \to o$$

Truth-degree mapping

Sets in HOL

Sets M are represented by their characteristic function

$$f_M(x) = \begin{cases} T & \text{, iff } x \in M \\ F & \text{otherwise} \end{cases}$$

Set representation

Object	$\longrightarrow$	*Representation in HOL*
Set M of objects of type α	$\longrightarrow$	$f_M : \alpha \to o$
e.g. $\{x \in T \mid P(x)\}$	$\longrightarrow$	$\lambda x_T .\, P_{T \to o}\, x$

Truth-degrees as HOL sets

Truth-degrees of SIXTEEN are sets of sets of Booleans, thus of type

$$(o \to o) \to o$$

Truth-degree mapping

Sets in HOL

Sets M are represented by their characteristic function

$$f_M(x) = \begin{cases} T & \text{, iff } x \in M \\ F & \text{otherwise} \end{cases}$$

Set representation

Object	$\longrightarrow$	*Representation in HOL*
Set M of objects of type α	$\longrightarrow$	$f_M : \alpha \to o$
e.g. $\{x \in T \mid P(x)\}$	$\longrightarrow$	$\lambda x_T .\, P_{T\to o}\, x$

Truth-degrees as HOL sets

Truth-degrees of SIXTEEN are sets of sets of Booleans, thus of type

$$(o \to o) \to o$$

Example

Encoding of $\boldsymbol{F}$ (the singleton set $\{\mathbf{F}\}$):

$$\boldsymbol{F} \equiv \lambda n_{oo}. \underbrace{\qquad\qquad\qquad\qquad}_{\text{some Boolean valued term}}$$

Example

Encoding of $\boldsymbol{F}$ (the singleton set $\{\mathbf{F}\}$):

$$\boldsymbol{F} \stackrel{?}{\equiv} \lambda n_{oo}.\, n\, F$$

Example

Encoding of $\boldsymbol{F}$ (the singleton set $\{\mathbf{F}\}$):

$$\boldsymbol{F} \equiv \lambda n_{oo}.\, n\, F \wedge \neg n\, T$$

Truth-degree mapping

N	=	$\lambda n_{oo}.\, F$
$\boldsymbol{N}$	=	$\lambda n_{oo}.\, \neg n\, F \wedge \neg n\, T$
$\boldsymbol{T}$	=	$\lambda n_{oo}.\, \neg n\, F \wedge n\, T$
$\boldsymbol{F}$	=	$\lambda n_{oo}.\, n\, F \wedge \neg n\, T$
$\boldsymbol{B}$	=	$\lambda n_{oo}.\, n\, F \wedge n\, T$
NF	=	$\lambda n_{oo}.\, \neg n\, T$
NT	=	$\lambda n_{oo}.\, \neg n\, F$
NB	=	$\lambda n_{oo}.\, (\neg n\, F \wedge \neg n\, T) \vee (n\, F \wedge n\, T)$
FT	=	$\lambda n_{oo}.\, (n\, F \wedge \neg n\, T) \vee (\neg n\, F \wedge n\, T)$
FB	=	$\lambda n_{oo}.\, n\, F$
TB	=	$\lambda n_{oo}.\, n\, T$
NFT	=	$\lambda n_{oo}.\, \neg n\, F \vee \neg n\, T$
NFB	=	$\lambda n_{oo}.\, n\, F \vee \neg n\, T$
NTB	=	$\lambda n_{oo}.\, \neg n\, F \vee n\, T$
FTB	=	$\lambda n_{oo}.\, n\, F \vee n\, T$
A	:=	**NFTB** $= \lambda n_{oo}.\, T$

Encoding of all 16 truth-degrees

Operation mapping

Operations in HOL *(operations based on falsehood omitted)*

$$\leq_t := \lambda v_{o(oo)}.\lambda w_{o(oo)}.\forall n_{oo}.\ ((v^t\ n) \supset (w^t\ n)) \wedge ((w^{-t}\ n) \supset (v^{-t}\ n))$$

Operation mapping

Operations in HOL *(operations based on falsehood omitted)*

$$\leq_t := \lambda v_{o(oo)}.\lambda w_{o(oo)}.\forall n_{oo}.\ \underbrace{((v^t\ n) \supset (w^t\ n))}_{v^t \subseteq w^t} \wedge ((w^{-t}\ n) \supset (v^{-t}\ n))$$

Operation mapping

Operations in HOL *(operations based on falsehood omitted)*

$$\leq_t := \lambda v_{o(oo)}.\lambda w_{o(oo)}.\forall n_{oo}.\ ((v^t\ n) \supset (w^t\ n)) \land \underbrace{((w^{-t}\ n) \supset (v^{-t}\ n))}_{w^{-t} \subseteq v^{-t}}$$

Operation mapping

Operations in HOL *(operations based on falsehood omitted)*

$$\leq_t := \lambda v_{o(oo)}.\lambda w_{o(oo)}.\forall n_{oo}.\ ((v^t\ n) \supset (w^t\ n)) \wedge ((w^{-t}\ n) \supset (v^{-t}\ n))$$

$$\sqcup_t := \lambda v_{o(oo)}.\lambda w_{o(oo)}.v^t \cup w^t \cup \left(w^{-t} \cap v^{-t}\right)$$

$$\sqcap_t := \lambda v_{o(oo)}.\lambda w_{o(oo)}.v^{-t} \cup w^{-t} \cup \left(w^t \cap v^t\right)$$

Operation mapping

Operations in HOL *(operations based on falsehood omitted)*

$$\leq_t := \lambda v_{o(oo)}.\lambda w_{o(oo)}.\forall n_{oo}.\ ((v^t\ n) \supset (w^t\ n)) \wedge ((w^{-t}\ n) \supset (v^{-t}\ n))$$

$$\sqcup_t := \lambda v_{o(oo)}.\lambda w_{o(oo)}.v^t \cup w^t \cup \left(w^{-t} \cap v^{-t}\right)$$

$$\sqcap_t := \lambda v_{o(oo)}.\lambda w_{o(oo)}.v^{-t} \cup w^{-t} \cup \left(w^t \cap v^t\right)$$

with auxillary definitions :

$$(v)^t_{o(oo)} := \lambda n_{oo}.\ (v\ n) \wedge (n\ \mathrm{T}) \qquad (v)^{-t}_{o(oo)} := \lambda n_{oo}.\ (v\ n) \wedge \neg(n\ \mathrm{T})$$

Embedding a proof task

Input: $x \wedge_t y \models_t x$

Embedding a proof task

Input: $x \wedge_t y \models_t x$

Translation to a HOL proof task: $\lceil x \wedge_t y \rceil \lceil \leq_t \rceil \lceil x \rceil$

Proof task example

Embedding a proof task

Input: $x \wedge_t y \models_t x$

Translation to a HOL proof task: $\lceil x \wedge_t y \rceil \, \lceil \leq_t \rceil \, \lceil x \rceil$

$$\Downarrow$$

Expansion of definitions,
β-normalization: $\underbrace{x^{-t} \cup y^{-t} \cup \left(x^t \cap y^t\right)}_{=:A} \lceil \leq_t \rceil \, x$

Proof task example

Embedding a proof task

Input: $x \wedge_t y \models_t x$

Translation to a HOL proof task: $\lceil x \wedge_t y \rceil \, \lceil \leq_t \rceil \, \lceil x \rceil$

$$\Downarrow$$

Expansion of definitions,
β-normalization: $\underbrace{x^{-t} \cup y^{-t} \cup \left(x^t \cap y^t\right)}_{=:A} \; \lceil \leq_t \rceil \; x$

$$\Downarrow$$

$$\forall n_{oo}.\ ((\lceil A^t \rceil\, n) \supset (\lceil x^t \rceil\, n)) \wedge ((\lceil x^{-t} \rceil\, n) \supset (\lceil A^{-t} \rceil\, n))$$

Proof task example

Embedding a proof task

Input: $$x \wedge_t y \models_t x$$

Translation to a HOL proof task: $$\lceil x \wedge_t y \rceil \lceil \leq_t \rceil \lceil x \rceil$$
$$\Downarrow$$

Expansion of definitions,
β-normalization: $$\underbrace{x^{-t} \cup y^{-t} \cup \left(x^t \cap y^t\right)}_{=:A} \lceil \leq_t \rceil x$$
$$\Downarrow$$
$$\forall n_{oo}.\ ((\lceil A^t \rceil\, n) \supset (\lceil x^t \rceil\, n)) \wedge ((\lceil x^{-t} \rceil\, n) \supset (\lceil A^{-t} \rceil\, n))$$
$$\Downarrow$$
$$\forall n_{oo}.\ ((\lambda n_{oo}.\ (\lceil A \rceil\, n) \wedge (n\, \top)\, n) \supset (\lambda n_{oo}.\ (x\, n) \wedge (n\, \top)\, n))$$
$$\wedge((\lambda n_{oo}.\ (x\, n) \wedge \neg(n\, \top)\, n) \supset (\lambda n_{oo}.\ (\lceil A \rceil\, n) \wedge \neg(n\, \top)\, n))$$
$$\Downarrow$$
$$\ldots$$

Proof task example

Embedding a proof task

Input: $x \wedge_t y \models_t x$

Translation to a HOL proof task: $\lceil x \wedge_t y \rceil \lceil \leq_t \rceil \lceil x \rceil$

$$\Downarrow$$

Expansion of definitions,
β-normalization:

$$\underbrace{x^{-t} \cup y^{-t} \cup \left(x^t \cap y^t\right)}_{=:A} \lceil \leq_t \rceil\, x$$

$$\Downarrow$$

$$\forall n_{oo}.\ ((\lceil A^t \rceil\, n) \supset (\lceil x^t \rceil\, n)) \wedge ((\lceil x^{-t} \rceil\, n) \supset (\lceil A^{-t} \rceil\, n))$$

$$\Downarrow$$

$$\forall n_{oo}.\ ((\lambda n_{oo}.\ (\lceil A \rceil\, n) \wedge (n\, \top)\, n) \supset (\lambda n_{oo}.\ (x\, n) \wedge (n\, \top)\, n))$$
$$\wedge((\lambda n_{oo}.\ (x\, n) \wedge \neg(n\, \top)\, n) \supset (\lambda n_{oo}.\ (\lceil A \rceil\, n) \wedge \neg(n\, \top)\, n))$$

$$\Downarrow$$

Can now be passed to standard HOL ATP ...

Experiments

- Embedding successfully employed
- Encoded in TPTP THF and Isabelle/HOL
- Some meta-logical properties verified

Example: Verfication of
Prop. 3.2 (from Shramko and Wansing, Truth and Falsehood):

$\mathbf{B} \in S \wedge \mathbf{B} \in T \Leftrightarrow \mathbf{B} \in S \sqcup T$	9 ms
$\mathbf{F} \in S \vee \mathbf{F} \in T \Leftrightarrow \mathbf{F} \in S \sqcup T$	8 ms
$\vdots$	$\vdots$
(16 overall)	Σ 131 ms

Experiments

- Embedding successfully employed
- Encoded in TPTP THF and Isabelle/HOL
- Some meta-logical properties verified

Example: Verfication of
Prop. 3.4 (from Shramko and Wansing, Truth and Falsehood):

$\mathbf{B} \in -_t S \Leftrightarrow \mathbf{F} \in S$	8 ms
$\mathbf{F} \in -_t S \Leftrightarrow \mathbf{B} \in S$	9 ms
$\vdots$	$\vdots$
(8 overall)	Σ 69 ms

Experiments

- Embedding successfully employed
- Encoded in TPTP THF and Isabelle/HOL
- Some meta-logical properties verified

Example: Verfication of
Def. 3.6 (from Shramko and Wansing, Truth and Falsehood):

$S \leq_t T \Rightarrow -_t T \leq_t -_t S$	421 ms
$S \leq_f T \Rightarrow -_t S \leq_f -_t T$	422 ms
$\vdots$	$\vdots$
(8 overall)	Σ 1734 ms

Freie Universität Berlin

Conclusion

- embedding of SIXTEEN into HOL
- currently, no automated deduction systems available
- embedding allows automation using common HOL ATPs
- **and** using interactive proof assistants (e.g. Coq)
- ... including reasoning *about* SIXTEEN

Conclusion and Outlook

Conclusion

- embedding of SIXTEEN into HOL
- currently, no automated deduction systems available
- embedding allows automation using common HOL ATPs
- **and** using interactive proof assistants (e.g. Coq)
- ... including reasoning *about* SIXTEEN

Outlook: Towards HOL based universal reasoning

- augment SIXTEEN embedding with quantification (of arbitrary order?)
- many quantified non-classical logics are fragments of HOL
- logic combinations: e.g. SIXTEEN with modalities
- out-of-the-box automation via HOL ATPs for free