# Utilizing Higher-order Automated Theorem Provers as Universal Logic Engines[1]

Christoph Benzmüller, FU Berlin

Peter Andrews' Retirement Celebration, CMU, April 5, 2012

[1] This work has been funded by the DFG under grants BE 2501/6-1, BE 2501/8-1 and BE 2501/9-1

## Peter's influence and achievements (my favorites)

Theory

- Semantics of Church's Type Theory (HOL)
- Resolution in HOL
- Expansion proofs and matings in HOL

System Development

- TPS has been the strongest HOL-ATP for many years; presumably it still is the strongest HOL-ATP if cooperations with external reasoners are disallowed
- Integration of interaction and automation

Education

- Black (now white) book: "To Truth Through Proof"
- Use of ETPS in course

# Statement from Peter's website ...

*My research has focused on automated deduction and Church's type theory, which is a rich and expressive formulation of higher-order logic in which statements from many disciplines, particularly those involving mathematics, can readily be expressed.*

In this talk:

We utilize Church's Type Theory (HOL) for non-classical logics

# Can Peter retire happy?

- Chris thinks that Peter can retire happy, if he knows that HOL-ATP is fostered by someone

  $\Box_{knowledgeChris}($
  $\quad \Box_{knowledgePeter} \exists X.fosters(X, holatp) \supset canRetireHappy(peter))$

- Peter knows that Chris fosters HOL-ATP

  $\Box_{knowledgePeter} fosters(chris, holatp)$

- Peter knows that Chad fosters HOL-ATP

  $\Box_{knowledgePeter} fosters(chad, holatp)$

- Peter knows that other persons do foster HOL-ATP ...

  $\ldots$

- Chris thinks that Peter can retire happy

  $\Box_{knowledgeChris} canRetireHappy(peter)$

- Chris thinks that Peter can retire happy, if he knows that HOL-ATP is fostered by someone

$$\Box_{knowledgeChris}(\Box_{knowledgePeter}\exists X.fosters(X, holatp) \supset canRetireHappy(peter))$$

- Peter knows that Chris fosters HOL-ATP

$$\Box_{knowledgePeter} fosters(chris, holatp)$$

- Peter knows that Chad fosters HOL-ATP

$$\Box_{knowledgePeter} fosters(chad, holatp)$$

- Peter knows that other persons do foster HOL-ATP ...

$\ldots$

- Chris thinks that Peter can retire happy

$$\Box_{knowledgeChris} canRetireHappy(peter)$$

Peter's TPS prover for classical higher-order logic (HOL) can prove this!

- THFTPTP project: ensures that TPS is not retiring
- TPS and friends as universal logic engines
  - first-order modal logics (FML)
  - conditional logics (CL)
  - automated reasoning at object-level and at meta-level
  - Why am I interested in this?
- QMLTP project: HOL-ATPs perform well for FML

The THFTPTP Project:
ensures that TPS is not retiring

## Results of the EU Project THFTPTP

- Collaboration with Geoff Sutcliffe, Chad Brown and others
- Results
  - THF0 syntax for HOL
  - Online access to provers
  - Library with example problems (e.g. entire TPS library) and results
  - Ontology and syntax for proof results
  - International CASC competition for HOL-ATP
  - Various tools

Improved availability and robustness of HOL-ATPs: TPS, LEO-II, Isabelle, Satallax, Refute, Nitpick `http://www.tptp.org/cgi-bin/SystemOnTPTP`

[SutcliffeBenzmüller, J. Formalized Reasoning, 2010]

[BenzmüllerRabeSutcliffe, IJCAR, 2008]

# CASC Competitions in THF Category since 2009

2009

| THF | TPS 3.20080227G1d | LEO-II 1.0 | LEO-IIP 1.0 | IsabelleP 2009 |
|---|---|---|---|---|
| Attempted | 200 | 200 | 200 | 200 |
| Solved | 170 | 146 | 146 | 124 |
| Av. Time | 23.18 | 2.27 | 3.44 | 55.92 |
| Solutions | 0 | 0 | 146 | 124 |

2009

| THF | TPS 3.20080227G1d | LEO-II 1.0 | LEO-IIP 1.0 | IsabelleP 2009 |
|---|---|---|---|---|
| Attempted | 200 | 200 | 200 | 200 |
| Solved | 170 | 146 | 146 | 124 |
| Av. Time | 23.18 | 2.27 | 3.44 | 55.92 |
| Solutions | 0 | 0 | 146 | 124 |

2010

| THF | LEO-II 1.2 | Satallax 1.4 | IsabelleP 2009-2 | TPS 3.20080227G1d |
|---|---|---|---|---|
| Solved | 125/200 | 120/200 | 101/200 | 80/200 |
| Av. CPU Time | 16.65 | 55.24 | 100.75 | 36.15 |
| Solutions | 125/200 | 120/200 | 0/200 | 0/200 |

LEO-II 1.2 solved 56% more than previous winner

# CASC Competitions in THF Category since 2009

2009

| THF | TPS 3.20080227G1d | LEO-II 1.0 | LEO-IIP 1.0 | IsabelleP 2009 |
|---|---|---|---|---|
| Attempted | 200 | 200 | 200 | 200 |
| Solved | 170 | 146 | 146 | 124 |
| Av. Time | 23.18 | 2.27 | 3.44 | 55.92 |
| Solutions | 0 | 0 | 146 | 124 |

2010

| THF | LEO-II 1.2 | Satallax 1.4 | IsabelleP 2009-2 | TPS 3.20080227G1d |
|---|---|---|---|---|
| Solved | 125/200 | 120/200 | 101/200 | 80/200 |
| Av. CPU Time | 16.65 | 55.24 | 100.75 | 36.15 |
| Solutions | 125/200 | 120/200 | 0/200 | 0/200 |

LEO-II 1.2 solved 56% more than previous winner

2011

| THF/300 | Satallax 2.1 | LEO-II 1.2.8 | LEO-II 1.2 | Isabelle 2011 | TPS 3.110228S1a |
|---|---|---|---|---|---|
| Solved | 246/300 | 208/300 | 204/300 | 201/300 | 190/300 |
| Av. CPU Time | 12.04 | 8.97 | 4.95 | 36.55 | 18.69 |

Satallax 2.1 solved 21% more than previous winner

# CASC Competitions in THF Category since 2009

2011

| THF/300 | Satallax 2.1 | LEO-II 1.2.8 | *LEO-II* *1.2* | Isabelle 2011 | TPS 3.110228S1a |
|---|---|---|---|---|---|
| Solved | 246/300 | 208/300 | 204/300 | 201/300 | 190/300 |
| Av. CPU Time | 12.04 | 8.97 | 4.95 | 36.55 | 18.69 |

2011

| THF/300 | Satallax 2.1 | LEO-II 1.2.8 | LEO-II 1.2 | Isabelle 2011 | TPS 3.110228S1a |
|---|---|---|---|---|---|
| Solved | 246/300 | 208/300 | 204/300 | 201/300 | 190/300 |
| Av. CPU Time | 12.04 | 8.97 | 4.95 | 36.55 | 18.69 |

LEO-II cooperates with FOL prover E

# CASC Competitions in THF Category since 2009

2011

| THF/300 | Satallax 2.1 | LEO-II 1.2.8 | *LEO-II* *1.2* | Isabelle 2011 | TPS 3.110228S1a |
|---|---|---|---|---|---|
| Solved | 246/300 | 208/300 | 204/300 | 201/300 | 190/300 |
| Av. CPU Time | 12.04 | 8.97 | 4.95 | 36.55 | 18.69 |

Isabelle cooperates with FOL provers (sledgehammer) and SMT solvers (smt)

# CASC Competitions in THF Category since 2009

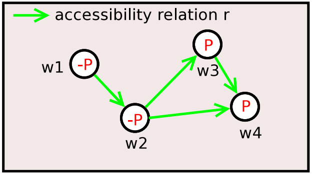
2011

| THF/300 | Satallax 2.1 | LEO-II 1.2.8 | *LEO-II* *1.2* | Isabelle 2011 | TPS 3.110228S1a |
|---|---|---|---|---|---|
| Solved | 246/300 | 208/300 | 204/300 | 201/300 | 190/300 |
| Av. CPU Time | 12.04 | 8.97 | 4.95 | 36.55 | 18.69 |

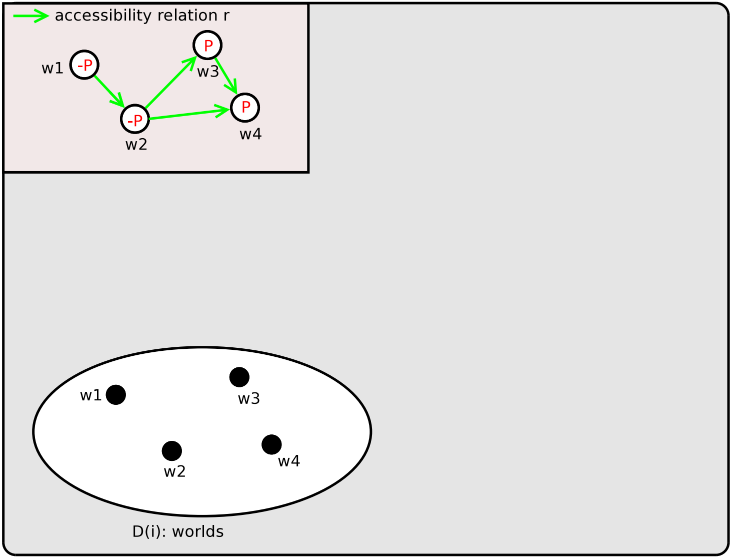
Satallax cooperates with SAT solver Minisat

TPS and friends as universal logic engines

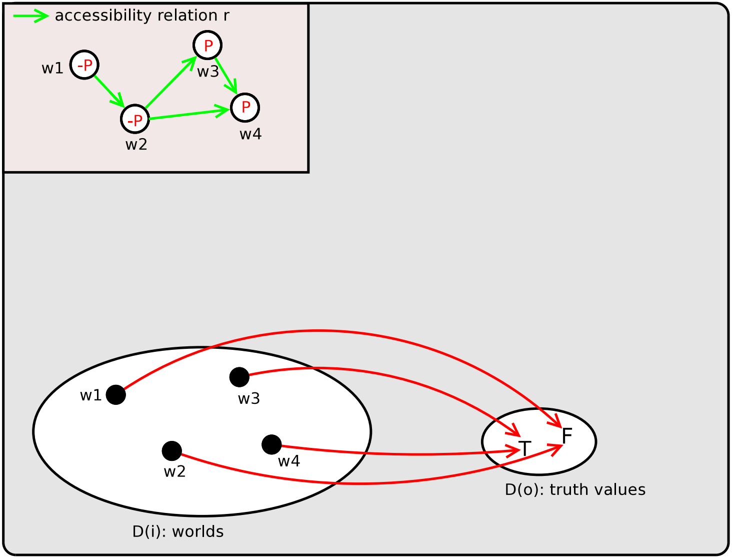
# Combining the Kripke View and the Tarski View on Logics



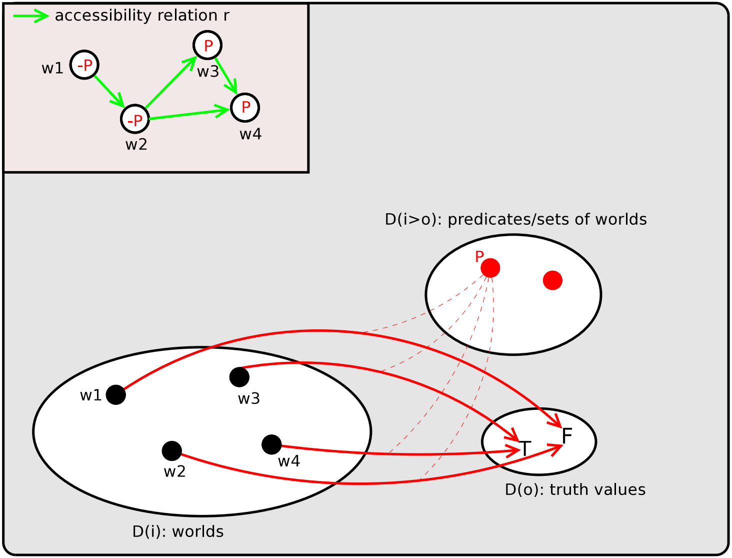
# Combining the Kripke View and the Tarski View on Logics



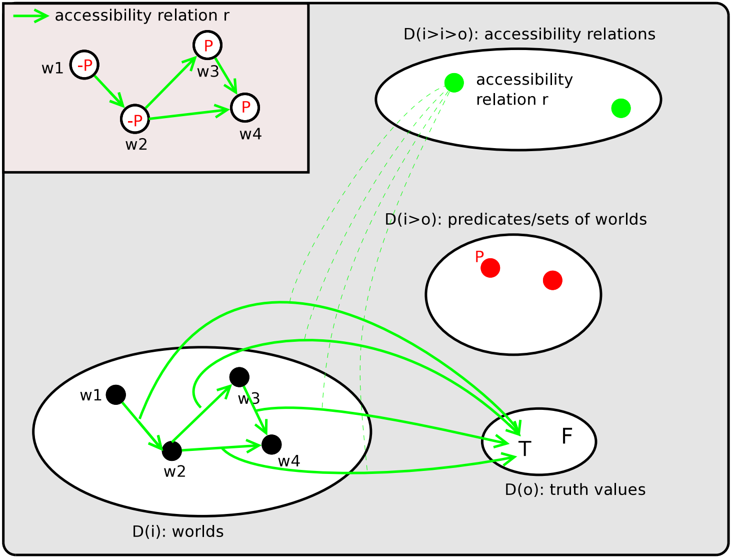
# Combining the Kripke View and the Tarski View on Logics



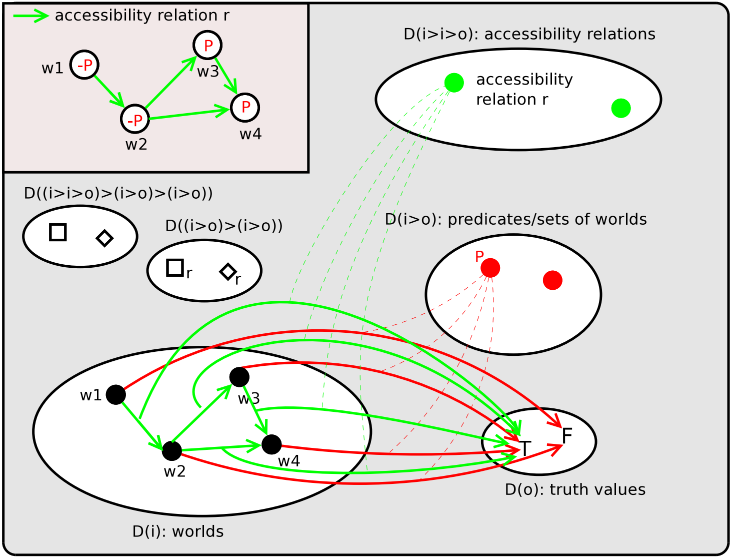
# Combining the Kripke View and the Tarski View on Logics



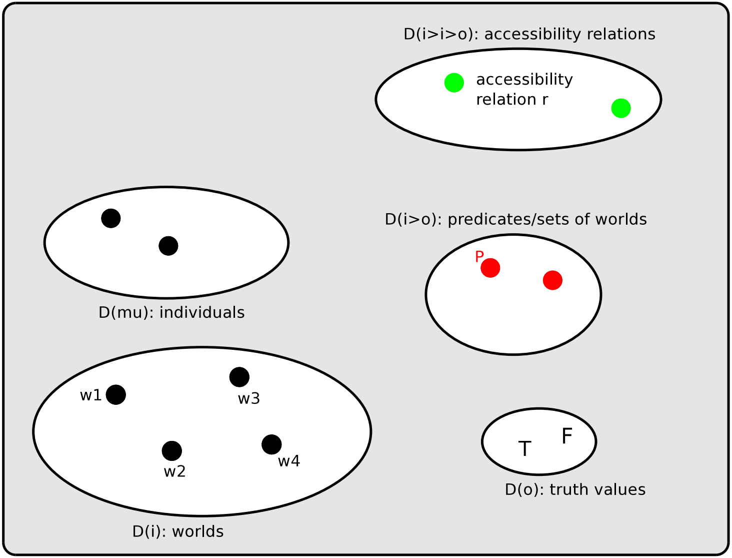
# Combining the Kripke View and the Tarski View on Logics



# Combining the Kripke View and the Tarski View on Logics



# Combining the Kripke View and the Tarski View on Logics



# Modal Logics in HOL

- Syntax: $s, t \quad ::= \quad P \mid \neg\, s \mid s \vee t \mid \Box_r\, s \mid$

HOL

- Syntax
  - formulas s $\longrightarrow$ terms $s_{\iota\to o}$

  Kripke Semantics
  - worlds w $\longrightarrow$ terms $w_{\iota}$
  - accessibility relations r $\longrightarrow$ terms $r_{\iota\to\iota\to o}$

# Modal Logics in HOL

- Syntax: $s, t \quad ::= \quad P \mid \neg\, s \mid s \vee t \mid \Box_r\, s \mid$

- HOL

| | | |
|---|---|---|
| Syntax<br>- formulas s | $\longrightarrow$ | terms $s_{\iota\rightarrow o}$ |
| Kripke Semantics<br>- worlds w | $\longrightarrow$ | terms $w_{\iota}$ |
| - accessibility relations r | $\longrightarrow$ | terms $r_{\iota\rightarrow\iota\rightarrow o}$ |

- Syntax of embedded logic as abbreviations of HOL-terms

$$\begin{array}{lcl}
P & = & P_{\iota\rightarrow o} \\
\neg & = & \lambda S_{\iota\rightarrow o}.\lambda W_{\iota}.\neg(S\,W) \\
\vee & = & \lambda S_{\iota\rightarrow o}.\lambda T_{\iota\rightarrow o}.\lambda W_{\iota}.(S\,W) \vee (T\,W) \\
\Box & = & \lambda R_{\iota\rightarrow\iota\rightarrow o}.\lambda S_{\iota\rightarrow o}.\lambda W_{\iota}.\forall V_{\iota}.\neg(R\,W\,V) \vee (S\,V)
\end{array}$$

[BenzmüllerPaulson, Log.J.IGPL, 2010], [BenzmüllerPaulson, Logica Universalis, to appear]

- Syntax: $s,t \quad ::= \quad P \mid \neg\, s \mid s \vee t \mid \Box_r\, s \mid$

HOL

| | | |
|---|---|---|
| Syntax<br>- formulas s | $\longrightarrow$ | terms $s_{\iota\to o}$ |
| Kripke Semantics<br>- worlds w | $\longrightarrow$ | terms $w_\iota$ |
| - accessibility relations r | $\longrightarrow$ | terms $r_{\iota\to\iota\to o}$ |

- Syntax of embedded logic as abbreviations of HOL-terms

$$
\begin{array}{rcl}
P & = & P_{\iota\to o} \\
\neg & = & \lambda S_{\iota\to o}.\lambda W_\iota.\neg(S\,W) \\
\vee & = & \lambda S_{\iota\to o}.\lambda T_{\iota\to o}.\lambda W_\iota.(S\,W)\vee(T\,W) \\
\Box & = & \lambda R_{\iota\to\iota\to o}.\lambda S_{\iota\to o}.\lambda W_\iota.\forall V_\iota.\neg(R\,W\,V)\vee(S\,V) \\
(\forall^p), \forall^\mu & = & \lambda Q_{\mu\to(\iota\to o)}.\lambda W_\iota.\forall X_\mu.(Q\,X\,W)
\end{array}
$$

[BenzmüllerPaulson, Log.J.IGPL, 2010], [BenzmüllerPaulson, Logica Universalis, to appear]

- Syntax: $s, t \quad ::= \quad P \mid \neg\, s \mid s \vee t \mid \Box_r\, s \mid$

HOL

- | | | |
  |---|---|---|
  | Syntax<br>- formulas s | $\longrightarrow$ | terms $s_{\iota\to o}$ |
  | Kripke Semantics<br>- worlds w | $\longrightarrow$ | terms $w_\iota$ |
  | - accessibility relations r | $\longrightarrow$ | terms $r_{\iota\to\iota\to o}$ |

- Syntax of embedded logic as abbreviations of HOL-terms

$$
\begin{aligned}
P &= P_{\iota\to o}\\
\neg &= \lambda S_{\iota\to o}.\lambda W_\iota.\neg(S\,W)\\
\vee &= \lambda S_{\iota\to o}.\lambda T_{\iota\to o}.\lambda W_\iota.(S\,W)\vee(T\,W)\\
\Box &= \lambda R_{\iota\to\iota\to o}.\lambda S_{\iota\to o}.\lambda W_\iota.\forall V_\iota.\neg(R\,W\,V)\vee(S\,V)\\
(\forall^p), \forall^\mu &= \lambda Q_{\mu\to(\iota\to o)}.\lambda W_\iota.\forall X_\mu.(Q\,X\,W)\\
\Rightarrow_f &= \lambda S_{\iota\to o}.\lambda T_{\iota\to o}.\lambda W_\iota.\forall V_\iota.\neg(f\,W\,S\,V)\vee(T\,V)
\end{aligned}
$$

[BenzmüllerGenovese, NCMPL, 2011], [BenzmüllerGabbayGenoveseRispoli, Logica Universalis, to appear]

# Embedding Meta-Level Notions

- Validity

$$\text{valid} \quad = \quad \lambda\varphi_{\iota\to o}.\,\forall W_\iota.\,\varphi\,W$$

Similar: Satisfiability, Countersatisfiability, Unsatisfiability

- Validity

$$\text{valid} \quad = \quad \lambda\varphi_{\iota\to o}.\,\forall W_\iota.\,\varphi\, W$$

Similar: Satisfiability, Countersatisfiability, Unsatisfiability

**Soundness and Completeness Theorem**

$$\models \varphi \quad \text{iff} \quad \models^{HOL} \text{valid}\, \varphi_{\iota\to o}$$

Consequence: Automation for free in classical HOL ATPs!

**Ax1** valid $\Box_{knowledgeChris}\ ($
$\Box_{knowledgePeter}\ \exists X.fosters(X, holatp) \supset canRetireHappy(peter))$

**Ax2** valid $\Box_{knowledgePeter}\ fosters(chris, holatp)$

**Ax3** valid $\Box_{knowledgePeter}\ fosters(chad, holatp)$

further axioms $\Box_{knowledgeChris}, \Box_{knowledgePeter}$ are S4 operators

**Conj** valid $\Box_{knowledgeChris}\ canRetireHappy(peter)$

**Ax1** valid $\Box_{knowledgeChris}$ (
$\Box_{knowledgePeter}$ $\exists X . fosters(X, holatp) \supset canRetireHappy(peter))$

**Ax2** valid $\Box_{knowledgePeter}$ $fosters(chris, holatp)$

**Ax3** valid $\Box_{knowledgePeter}$ $fosters(chad, holatp)$

further axioms $\Box_{knowledgeChris}$, $\Box_{knowledgePeter}$ are S4 operators

**Conj** valid $\Box_{knowledgeChris}$ $canRetireHappy(peter)$

$\forall W_\iota . \Box_{knowledgeChris}$ $canRetireHappy(peter)$ $W$

expanded abbreviation

## Automating Peter's Retirement Example

**Ax1** valid $\Box_{knowledgeChris}$ (
$\Box_{knowledgePeter}\ \exists X.fosters(X, holatp) \supset canRetireHappy(peter))$

**Ax2** valid $\Box_{knowledgePeter}\ fosters(chris, holatp)$

**Ax3** valid $\Box_{knowledgePeter}\ fosters(chad, holatp)$

further axioms $\Box_{knowledgeChris}, \Box_{knowledgePeter}$ are S4 operators

**Conj** valid $\Box_{knowledgeChris}\ canRetireHappy(peter)$

$$\forall W_\iota.\ \Box_{knowledgeChris}\ canRetireHappy(peter)\ W$$

$$\forall W_\iota.\ \forall V_\iota.\ \neg(knowledgeChris\ W\ V) \vee canRetireHappy(peter)\ W$$

expanded abbreviation

# Automating Peter's Retirement Example

**Ax1** valid $\Box_{knowledgeChris}$ (
$\Box_{knowledgePeter}\ \exists X. fosters(X, holatp) \supset canRetireHappy(peter))$

**Ax2** valid $\Box_{knowledgePeter}\ fosters(chris, holatp)$

**Ax3** valid $\Box_{knowledgePeter}\ fosters(chad, holatp)$

further axioms $\Box_{knowledgeChris}$, $\Box_{knowledgePeter}$ are S4 operators

**Conj** valid $\Box_{knowledgeChris}\ canRetireHappy(peter)$

$$\forall W_\iota .\ \Box_{knowledgeChris}\ canRetireHappy(peter)\ W$$

$$\forall W_\iota .\ \forall V_\iota . \neg(knowledgeChris\ W\ V) \vee canRetireHappy(peter)\ W$$

$$\forall W_\iota .\ \forall V_\iota . \neg(knowledgeChris\ W\ V) \vee (canRetireHappy\ peter\ W)$$

expanded abbreviation

# Translating Kripke style semantics to HOL

- Kripke style semantics

$$
\begin{array}{lll}
M, w \models P & \text{arbitrary} & \\
M, w \models \neg s & \text{iff} & \text{not } M, w \models s \\
M, w \models s \vee t & \text{iff} & M, w \models s \text{ or } M, w \models t
\end{array}
$$

- Semantic embedding in HOL

$$
\begin{array}{lll}
P & = & P_{\iota \to o} \\
\neg & = & \lambda S_{\iota \to o} . \lambda W_{\iota} . \neg (S\, W) \\
\vee & = & \lambda S_{\iota \to o} . \lambda T_{\iota \to o} . \lambda W_{\iota} . (S\, W) \vee (T\, W)
\end{array}
$$

- Kripke style semantics

$$
\begin{array}{llll}
M, w \models P & \text{arbitrary} & \\
M, w \models \neg s & \text{iff} & \text{not } M, w \models s \\
M, w \models s \vee t & \text{iff} & M, w \models s \text{ or } M, w \models t \\
M, w \models \Box_r s & \text{iff} & M, v \models s \text{ for all v such that } r(w, v)
\end{array}
$$

- Semantic embedding in HOL

$$
\begin{array}{lll}
P & = & P_{\iota \to o} \\
\neg & = & \lambda S_{\iota \to o}. \lambda W_{\iota}. \neg (S\, W) \\
\vee & = & \lambda S_{\iota \to o}. \lambda T_{\iota \to o}. \lambda W_{\iota}. (S\, W) \vee (T\, W) \\
\Box & = & \lambda R_{\iota \to \iota \to o}. \lambda S_{\iota \to o}. \lambda W_{\iota}. \forall V_{\iota}. \neg (R\, W\, V) \vee (S\, V)
\end{array}
$$

# Translating Kripke style semantics to HOL

- Kripke style semantics

$$
\begin{array}{llll}
M, w \models P & \text{arbitrary} & & \\
M, w \models \neg s & \text{iff} & & \text{not } M, w \models s \\
M, w \models s \vee t & \text{iff} & & M, w \models s \text{ or } M, w \models t \\
M, w \models \Box_r\, s & \text{iff} & & M, v \models s \text{ for all v such that } r(w, v)
\end{array}
$$

- Semantic embedding in HOL

$$
\begin{array}{lll}
P & = & P_{\iota \to o} \\
\neg & = & \lambda S_{\iota \to o} . \lambda W_{\iota} . \neg (S\ W) \\
\vee & = & \lambda S_{\iota \to o} . \lambda T_{\iota \to o} . \lambda W_{\iota} . (S\ W) \vee (T\ W) \\
\Box & = & \lambda R_{\iota \to \iota \to o} . \lambda S_{\iota \to o} . \lambda W_{\iota} . \forall V_{\iota} . \neg (R\ W\ V) \vee (S\ V)
\end{array}
$$

# Translating Kripke style semantics to HOL

- Kripke style semantics

$$
\begin{array}{lll}
M, w \models P & \text{arbitrary} & \\
M, w \models \neg s & \text{iff} & \text{not } M, w \models s \\
M, w \models s \vee t & \text{iff} & M, w \models s \text{ or } M, w \models t \\
M, w \models \Box_r\, s & \text{iff} & M, v \models s \text{ for all v such that } r(w, v)
\end{array}
$$

- Semantic embedding in HOL

$$
\begin{array}{rcl}
P & = & P_{\iota\to o} \\
\neg & = & \lambda S_{\iota\to o}.\, \lambda W_\iota.\, \neg(S\; W) \\
\vee & = & \lambda S_{\iota\to o}.\, \lambda T_{\iota\to o}.\, \lambda W_\iota.\, (S\; W) \vee (T\; W) \\
\Box_r & = & \lambda S_{\iota\to o}.\, \lambda W_\iota.\, \forall V_\iota.\, \neg(r\; W\; V) \vee (S\; V)
\end{array}
$$

To model $\Box_r$ as T, S4 operator etc. add axioms like (*reflexive* $r$), etc.

# Translating Kripke style semantics to HOL

- Kripke style semantics

higher-order selection function!

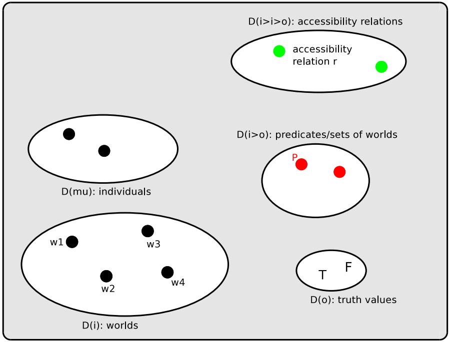
$$
\begin{array}{lll}
M, w \models P & \text{arbitrary} & \\
M, w \models \neg s & \text{iff} & \text{not } M, w \models s \\
M, w \models s \vee t & \text{iff} & M, w \models s \text{ or } M, w \models t \\
M, w \models \Box_r\, s & \text{iff} & M, v \models s \text{ for all v such that } r(w, v) \\
M, w \models s \Rightarrow_f t & \text{iff} & M, v \models t \text{ for all } v \in f(w, [s]) \\
 & & \quad \text{with } [s] = \{u \mid M, u \models s\}
\end{array}
$$

- Semantic embedding in HOL

$$
\begin{array}{lll}
P & = & P_{\iota \to o} \\
\neg & = & \lambda S_{\iota \to o} . \lambda W_{\iota} . \neg (S\, W) \\
\vee & = & \lambda S_{\iota \to o} . \lambda T_{\iota \to o} . \lambda W_{\iota} . (S\, W) \vee (T\, W) \\
\Box & = & \lambda R_{\iota \to \iota \to o} . \lambda S_{\iota \to o} . \lambda W_{\iota} . \forall V_{\iota} . \neg (R\, W\, V) \vee (S\, V) \\
\Rightarrow_f & = & \lambda S_{\iota \to o} . \lambda T_{\iota \to o} . \lambda W_{\iota} . \forall V_{\iota} . \neg (f\, W\, S\, V) \vee (T\, V)
\end{array}
$$

Add respective axioms for $f$

### Constant Domain

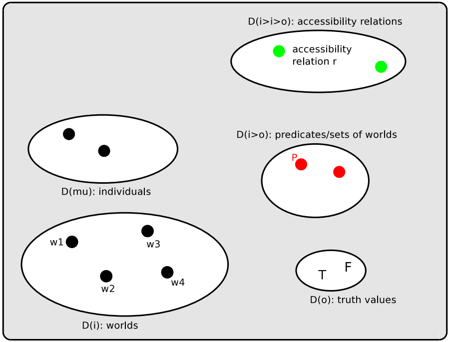



$$\Pi = \lambda Q. \lambda W_\iota. \forall X_\mu. (Q\, X\, W)$$

$$\forall Y. s = \Pi \lambda Y. s$$

# Quantified Modal Logics: Varying and Cumulative Domain

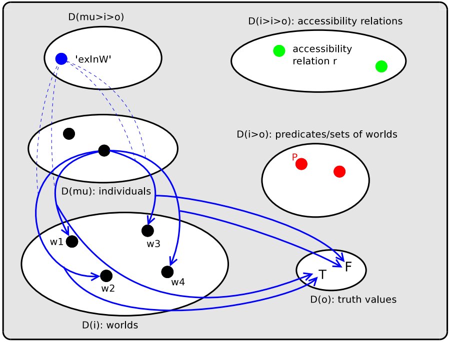
## Constant Domain




$$\Pi = \lambda Q.\lambda W_\iota.\forall X_\mu.(Q\,X\,W)$$

$$\forall Y.s = \Pi \lambda Y.s$$

## Varying and Cumulative Domain




$$\Pi_{var} = \lambda Q.\lambda W_\iota.\forall X_\mu.\neg(\text{exInW}\,X\,W) \vee (Q\,X\,W)$$

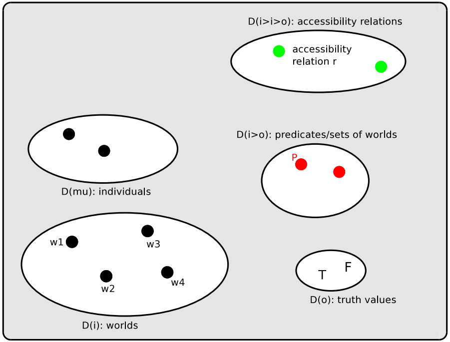
A: $\forall W_\iota.\exists X_\mu.(\text{exInW}\,X\,W)$

B($c$): $\forall W_\iota.(\text{exInW}\,c\,W)$

B($f$): $\forall W_\iota.(\text{exInW}\,t^1\,W) \wedge \ldots \wedge (\text{exInW}\,t^n\,W) \supset (\text{exInW}\,(f\,t^1, \ldots\, t^n)\,W)$

[ongoing work with Otten and Raths, submitted]

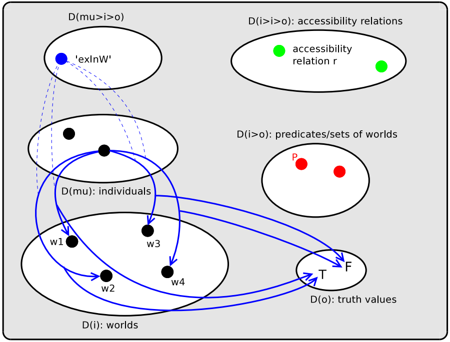
## Constant Domain




$$\Pi = \lambda Q_{\blacksquare} \lambda W_{\iota \blacksquare} \forall X_{\mu \blacksquare} (Q\, X\, W)$$

$$\forall Y_{\blacksquare} s = \Pi \lambda Y_{\blacksquare} s$$

## Varying and Cumulative Domain




$$\Pi_{var} = \lambda Q_{\blacksquare} \lambda W_{\iota \blacksquare} \forall X_{\mu \blacksquare} \neg(\text{exInW}\; X\; W) \vee (Q\, X\, W)$$

A: $\forall W_{\iota \blacksquare} \exists X_{\mu \blacksquare} (\text{exInW}\; X\; W)$

B(c): $\forall W_{\iota \blacksquare} (\text{exInW}\; c\; W)$

B(f): $\forall W_{\iota \blacksquare} (\text{exInW}\; t^1\; W) \wedge \ldots \wedge (\text{exInW}\; t^n\; W) \supset (\text{exInW}\; (f\; t^1, \ldots\; t^n)\; W)$

C: $\forall X_{\mu}, V_{\iota}, W_{\iota \blacksquare} (\text{exInW}\; X\; V) \wedge (r\; V\; W) \supset (\text{exInW}\; X\; W)$

[ongoing work with Otten and Raths, submitted]

## Soundness and Completeness Results

$$\models \varphi \quad \text{iff} \quad \models^{HOL} \text{valid}\, \varphi_{\iota\to o}$$

- Propositional Multimodal Logics [BenzmüllerPaulson, Log.J.IGPL, 2010]
- Quantified Multimodal Logics [BenzmüllerPaulson, Logica Universalis, to appear]
- Propositional Conditional Logics [BenzmüllerEtAl., AMAI, to appear]
- Quantified Conditional Logics [BenzmüllerGenovese, NCMPL, 2011]
- Intuitionistic Logics: [BenzmüllerPaulson, Log.J.IGPL, 2010]
- Access Control Logics: [Benzmüller, IFIP SEC, 2009]

## Why not throwing things together?

Terms: $m ::= c \mid X \mid (f\ m^1 \ldots m^n)$

Formulas:

$$s,t ::= P \mid (k\ m^1 \ldots m^n) \mid \neg s \mid s \vee t \mid \Box_r s \mid s \Rightarrow_f t \mid \forall X.s \mid \forall_{var} X.s \mid \forall P.s$$

Embedding in HOL:

$$
\begin{array}{rcl}
c & = & c_\mu \qquad X = X_\mu \qquad f = f_{\mu^n \to \mu} \\
P & = & P_{\iota\to o} \qquad k = k_{\mu^n \to \iota \to o} \\
r & = & k_{\iota\to\iota\to o} \quad (+axioms\ for\ r) \qquad f = f_{\iota\to\iota\to o} \quad (+axioms\ for\ f) \\
\neg & = & \lambda S_{\iota\to o}.\lambda W_\iota.\neg(S\,W) \\
\vee & = & \lambda S_{\iota\to o}.\lambda T_{\iota\to o}.\lambda W_\iota.(S\,W) \vee (T\,W) \\
\Box & = & \lambda R_{\iota\to\iota\to o}.\lambda S_{\iota\to o}.\lambda W_\iota.\forall V_\iota.\neg(R\,W\,V) \vee (S\,V) \\
\Rightarrow & = & \lambda F_{\iota\to(\iota\to o)\iota\to o}.\lambda S_{\iota\to o}.\lambda T_{\iota\to o}.\lambda W_\iota.\forall V_\iota.\neg(F\,W\,S\,V) \vee (T\,V) \\
\Pi & = & \lambda Q_{\mu\to(\iota\to o)}.\lambda W_\iota.\forall X_\mu.(Q\,X\,W) \\
\Pi_{var} & = & \lambda Q.\lambda W_\iota.\forall X_\mu.\neg(\mathrm{exInW}\,X\,W) \vee (Q\,X\,W) \\
\Pi_p & = & \lambda Q_{(\iota\to o)\to(\iota\to o)}.\lambda W_\iota.\forall P_{\iota\to o}.(Q\,P\,W) \\
\ldots & & 
\end{array}
$$

further non-classical connectives, quantification over higher types, predicate abstraction, definite description ...

### Automation of meta-level properties

[Benzmüller, Festschrift Walther, 2010]

- Correspondences between axioms and semantic properties

  valid $\forall \phi.\ \Box_r\, \phi \supset \Box_r\, \Box_r\, \phi$

  $\Leftrightarrow$ (transitive $r$)

- Dependence/independence of axioms

  base modal logic K $\not\models$ axiom 4?

- Inclusion/non-inclusion relations between logics

  Is logic K45 (K+M+5) included in logic S4 (K+M+4)?

- (Relative) Consistency of logics and logic combinations

  Is logic S4 (K+M+4) consistent?

### Experiments:

- Modal Logics — with Geoff Sutcliffe
- Conditional Logics — with Valerio Genovese, Dov Gabbay

Semantic Conditions for Conditional Logic Axioms

| | | | TPS |
|---|---|---|---|
| **ID** | Axiom | $A \Rightarrow_f A$ | |
| | Condition | $f(w,[A]) \subseteq [A]$ | ✓ |
| **MP** | Axiom | $(A \Rightarrow_f B) \supset (A \supset B)$ | |
| | Condition | $[A] \subseteq f(w,[A])$ | ✓ |
| **CS** | Axiom | $(A \wedge B) \supset (A \Rightarrow_f B)$ | |
| | Condition | $w \in [A] \supset f(w,[A]) \subseteq \{w\}$ | ✓ |
| **CEM** | Axiom | $(A \Rightarrow_f B) \vee (A \Rightarrow_f \neg B)$ | |
| | Condition | $\|f(w,[A])\| \leq 1$ | ✓ |
| **AC** | Axiom | $(A \Rightarrow_f B) \wedge (A \Rightarrow_f C) \supset (A \wedge C \Rightarrow_f B)$ | |
| | Condition | $f(w,[A]) \subseteq [B] \supset f(w,[A \wedge B]) \subseteq f(w,[A])$ | ✓ |
| **RT** | Axiom | $(A \wedge B \Rightarrow_f C) \supset ((A \Rightarrow_f B) \supset (A \Rightarrow_f C))$ | |
| | Condition | $f(w,[A]) \subseteq [B] \supset f(w,[A]) \subseteq f(w,[A \wedge B])$ | ✓ |
| **CV** | Axiom | $(A \Rightarrow_f B) \wedge \neg(A \Rightarrow_f \neg C) \supset (A \wedge C \Rightarrow_f B)$ | |
| | Condition | $(f(w,[A]) \subseteq [B]$ and $f(w,[A]) \cap [C] \neq \emptyset) \supset f(w,[A \wedge C]) \subseteq [B]$ | ✓ |
| **CA** | Axiom | $(A \Rightarrow_f B) \wedge (C \Rightarrow_f B) \supset (A \vee C \Rightarrow_f B)$ | |
| | Condition | $f(w,[A \vee B]) \subseteq f(w,[A]) \cup f(w,[B])$ | ✓ |

[BenzmüllerEtAl., AMAI, to appear]

# Proofs and Countermodels at Meta-Level

The correct interpretation of the proof task for MP is

$$[\forall A, B.(A \Rightarrow_f B) \supset (A \supset B)] \leftrightarrow [\forall A, W.A \subseteq (f\ W\ A)]$$

versus (incorrect statement for MP)

$$\forall A, B.[((A \Rightarrow_f B) \supset (A \supset B)) \leftrightarrow \forall W.A \subseteq (f\ W\ A)]$$

The former is provable.

The latter is countersatisfiable; the countermodel reported by Nitpick is:

choose $D_i = \{i1\}$, $A = \{i1\}$, $B = \{i1\}$, $W = i1$,
and

$$f = \left\{ \; i1 \quad \longrightarrow \left\{ \begin{array}{lcl} \emptyset & & \longrightarrow \emptyset \\ \{i1\} & & \longrightarrow \emptyset \end{array} \right. \right.$$

QMLTP project: HOL-ATPs perform well for FML
(thanks to Jens Otten and Thomas Raths)

# QMLTP project: HOL-ATPs perform well for FML

The QMLTP project: see `http://www.iltp.de/qmltp/`

- Jens Otten and Thomas Raths, University of Potsdam
- infrastructure and benchmark library for testing and evaluating ATP systems for first-order modal logic
- collaborators: myself, Geoff Sutcliffe's TPTP project
- standardized extended TPTP syntax (called 'fml')
- 600 problems in 11 problem domains
- 20 problems in first-order multimodal logic

# QMLTP project: HOL-ATPs perform well for FML

| | | ATP system | | | | | |
|---|---|---|---|---|---|---|---|
| Logic | Domain | SeP | TAP | LEO-II | Satallax | MSPASS | CoP |
| K | varying | - | - | 73 | 104 | - | - |
| | cumulative | 121 | - | 89 | 122 | 70 | - |
| | constant | 124 | - | 120 | 146 | 67 | - |
| D | varying | - | 100 | 81 | 113 | - | 179 |
| | cumulative | 130 | 120 | 100 | 133 | 79 | 200 |
| | constant | 134 | 135 | 135 | 160 | 76 | 217 |
| T | varying | - | 138 | 120 | 170 | - | 224 |
| | cumulative | 163 | 160 | 139 | 192 | 105 | 249 |
| | constant | 166 | 175 | 173 | 213 | 95 | 269 |
| S4 | varying | - | 169 | 140 | 207 | - | 274 |
| | cumulative | 197 | 205 | 166 | 238 | 121 | 338 |
| | constant | 197 | 220 | 200 | 261 | 111 | 352 |
| S5 | varying | - | 219 | 169 | 248 | - | 359 |
| | cumulative | - | 272 | 215 | 297 | 140 | 438 |
| | constant | - | 272 | 237 | 305 | 131 | 438 |
| | | | | ... | ... | | |
| | | | | ... | ... | | |

The prover GQML (by Cerrito and Thion) has turned out unsound as part of our experiments.

# Demo: HOL-ATPs as universal reasoners

Analyzing example formula

$$(\Diamond(\exists X.pX) \wedge \Box\forall Y.\Diamond pY \supset qY) \supset \Diamond\exists Z.qZ$$

with HOL-ATPs:

| | constant | varying | cumulative |
|---|---|---|---|
| K | CSA | CSA | ??? |
| D | CSA | CSA | ??? |
| T | THM | THM | THM |
| S4 | THM | THM | THM |
| S5 | THM | THM | THM |

CSA means Countersatisfiable, THM means Theorem

Remark: The flexible transformation from FML into THF syntax is provided by Thomas Raths.

Terminal — bash — 98×56

```
z8b8b:2012-CMU-Andrews christophbenzmueller$ more demo2.fml

qmf(con,conjecture,
    ( ( (#dia:  ? [X] : p(X))
        &
        (#box: ! [Y]: ((#dia: p(Y)) => q(Y))) )
      => #dia: ? [X] : q(X) )).
z8b8b:2012-CMU-Andrews christophbenzmueller$
z8b8b:2012-CMU-Andrews christophbenzmueller$
z8b8b:2012-CMU-Andrews christophbenzmueller$ ./universal-reasoner demo2.fml s4 vary
--- Running version 0.1 of the HOL-ATP based universal logic engine ---

INPUT: fml MODALLOGIC: s4 DOMAIN: vary

Converting from fml to thf (thanks to Thomas Raths)

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

 TPS---3.110228S1a (20 sec timeout)
   RESULT: SOT_JG2Nm_ - TPS---3.110228S1a says Theorem - CPU = 5.62 WC = 7.50 Mode = MODE-X5202
 LEO-II---1.3.1 (20 sec timeout)
   RESULT: SOT_HyrsXa - LEO-II---1.3.1 says Theorem - CPU = 0.04 WC = 0.12
 Satallax---2.2 (20 sec timeout)
   RESULT: SOT_6YiBu4 - Satallax---2.2 says Theorem - CPU = 0.04 WC = 0.09
 Isabelle---2011 (20 sec timeout)
   RESULT: SOT_AiyQ6a - Isabelle---2011 says Theorem - CPU = 3.87 WC = 3.93 SolvedBy = smt
 Refute---2011 (20 sec timeout)
   RESULT: SOT_kBCMO7 - Refute---2011 says Timeout - CPU = 21.75 WC = 22.21
 Nitpick---2011 (20 sec timeout)
   RESULT: SOT_vfrCvq - Nitpick---2011 says Timeout - CPU = 20.58 WC = 22.20

---
z8b8b:2012-CMU-Andrews christophbenzmueller$ ./universal-reasoner demo2.fml k const
--- Running version 0.1 of the HOL-ATP based universal logic engine ---

INPUT: fml MODALLOGIC: k DOMAIN: const

Converting from fml to thf (thanks to Thomas Raths)

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

 TPS---3.110228S1a (20 sec timeout)
   RESULT: SOT_18LNWW - TPS---3.110228S1a says Unknown - CPU = 12.70 WC = 13.06
 LEO-II---1.3.1 (20 sec timeout)
   RESULT: SOT_G7CJ5d - LEO-II---1.3.1 says Unknown - CPU = 4.95 WC = 5.03
 Satallax---2.2 (20 sec timeout)
   RESULT: SOT_3cM_9y - Satallax---2.2 says CounterSatisfiable - CPU = 0.00 WC = 0.04
 Isabelle---2011 (20 sec timeout)
   RESULT: SOT_rR3DwX - Isabelle---2011 says Unknown - CPU = 17.87 WC = 17.76
 Refute---2011 (20 sec timeout)
   RESULT: SOT_Bl1n1S - Refute---2011 says CounterSatisfiable - CPU = 3.57 WC = 3.37
 Nitpick---2011 (20 sec timeout)
   RESULT: SOT_C41Wci - Nitpick---2011 says CounterSatisfiable - CPU = 4.76 WC = 4.19

---
z8b8b:2012-CMU-Andrews christophbenzmueller$
```

# Conclusion

Thank you Peter!