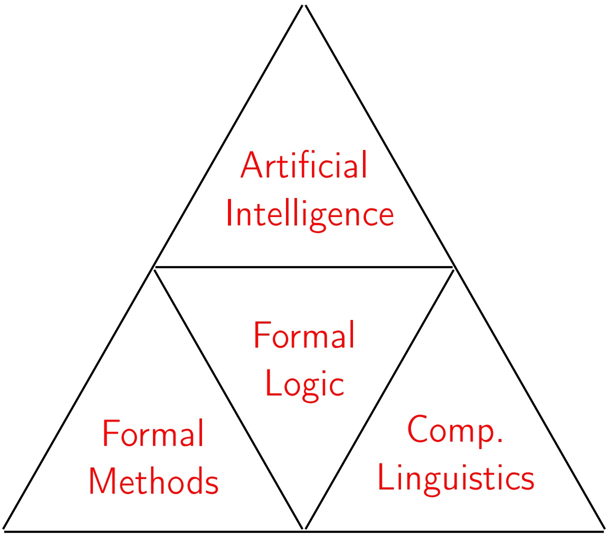
# Automating Expressive Non-classical Logics and their Combinations in Classical Higher Order Logic

Christoph Benzmüller

FU Berlin

Presentation at Potsdam University on November 15, 2011


Artificial Intelligence
Formal Logic
Formal Methods
Comp. Linguistics

# Research Background

Artificial Intelligence

Formal Logic

Computational Linguistics

Formal Methods

Artificial Intelligence

Formal Logic

Theory

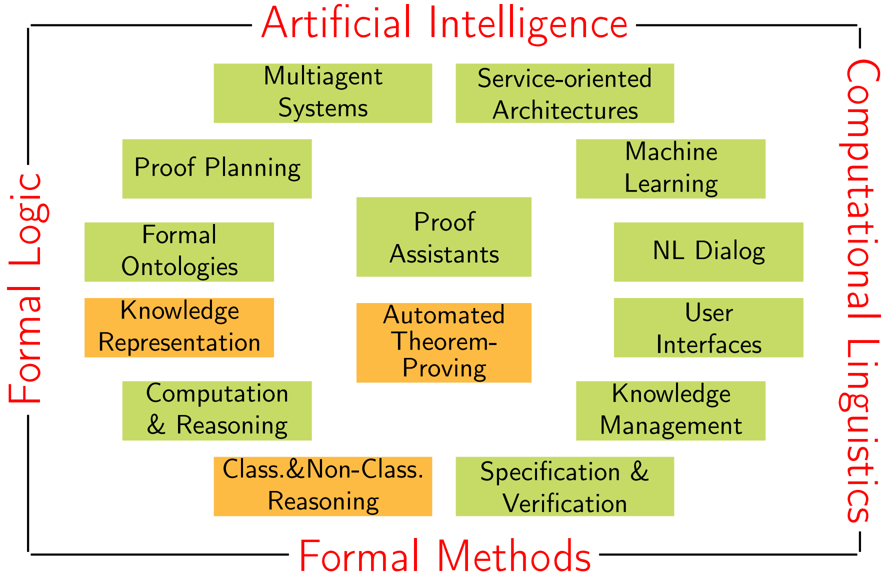
Modeling

Implementation

Application

Computational Linguistics

Formal Methods


Artificial Intelligence
Formal Logic
Computational Linguistics
Formal Methods
Multiagent Systems
Service-oriented Architectures
Proof Planning
Machine Learning
Proof Assistants
Formal Ontologies
NL Dialog
Knowledge Representation
Automated Theorem-Proving
User Interfaces
Computation & Reasoning
Knowledge Management
Class.&Non-Class. Reasoning
Specification & Verification

## Talk Overview

Core Questions:

1. Classical Higher Order Logic (HOL) as Universal Logic?
2. HOL Provers & Model Finders as Generic Reasoning Tools?
3. Combinations with Specialist Reasoners (if available)?

# Talk Overview

Core Questions:

1. Classical Higher Order Logic (HOL) as Universal Logic?
2. HOL Provers & Model Finders as Generic Reasoning Tools?
3. Combinations with Specialist Reasoners (if available)?

Outline:

- What is HOL?
- Mechanization & Automation of HOL
- Examples of Natural Fragments of HOL: Multimodal Logics & Others
- Automation of Logics and Logic Combinations in HOL
- Automation of Meta-Properties of Logics in HOL
- Conclusion

# What is HOL?

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

## What is HOL? (Church's Type Theory, Alonzo Church, 1940)

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

# What is HOL? (Church's Type Theory, Alonzo Church, 1940)

| 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 | − | ✓ | $reflexive_{(\iota\to\iota\to o)\to o} = \lambda R_{(\iota\to\iota\to o)} . \lambda X_\iota . R$ |

Simple Types: Prevent Paradoxes and Inconsistencies

# What is HOL? (Alonzo Church, 1940)

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

# What is HOL? (Alonzo Church, 1940)

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

Individuals

Booleans (True and False)

Functions

## What is HOL? (Alonzo Church, 1940)

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

Possible worlds

Individuals

Booleans (True and False)

Functions

- 
- HOL Syntax

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

Constant Symbols
Variable Symbols

- 
- HOL Syntax

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

Constant Symbols
Variable Symbols
Abstraction
Application

- 
- HOL Syntax

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

Constant Symbols
Variable Symbols
Abstraction
Application
Logical Connectives

- 
- HOL Syntax

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

# What is HOL? (Alonzo Church, 1940)

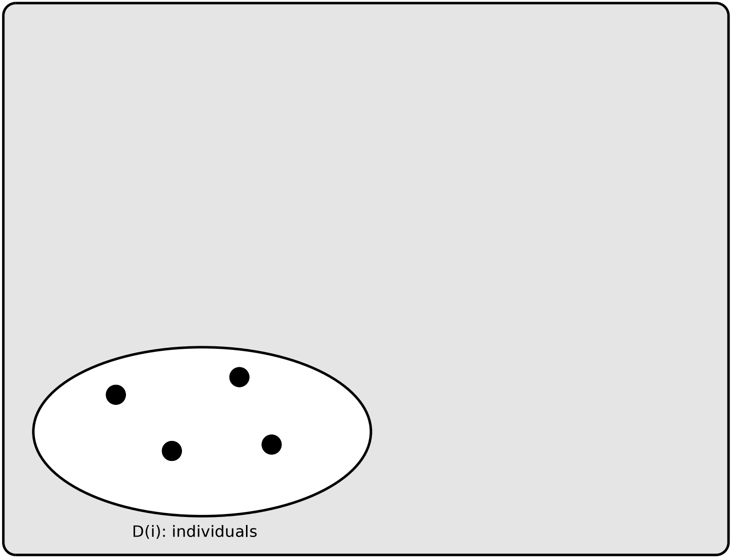
- 
- HOL Syntax

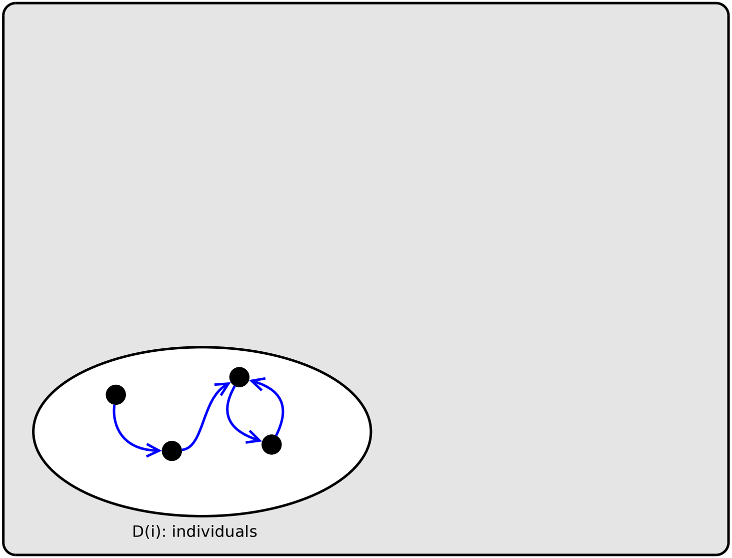
$$
\begin{aligned}
s, t \quad ::= \quad & p_\alpha \mid X_\alpha \\
& \mid (\lambda X_\alpha . s_\beta)_{\alpha\to\beta} \mid (s_{\alpha\to\beta}\, t_\alpha)_\beta \\
& \mid (\neg_{o\to o}\, s_o)_o \mid (s_o \vee_{o\to o\to o} t_o)_o \mid (\Pi_{(\alpha\to o)\to o}\, (\lambda X_\alpha . t_o))_o
\end{aligned}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

- 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. [Benzmüller et al., J.Symb.Log., 2004] [Muskens, J.Symb.Log., 2007]

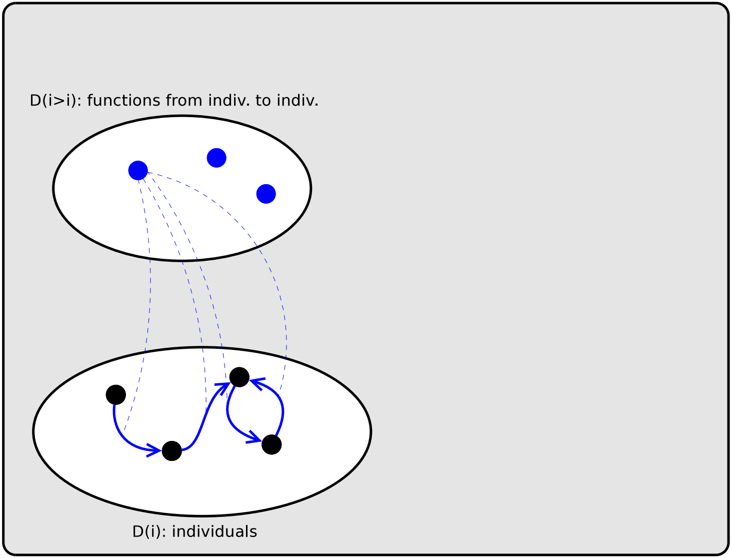
- HOL with Henkin-Semantics: semi-decidable & compact (like FOL)

# Semantics of HOL

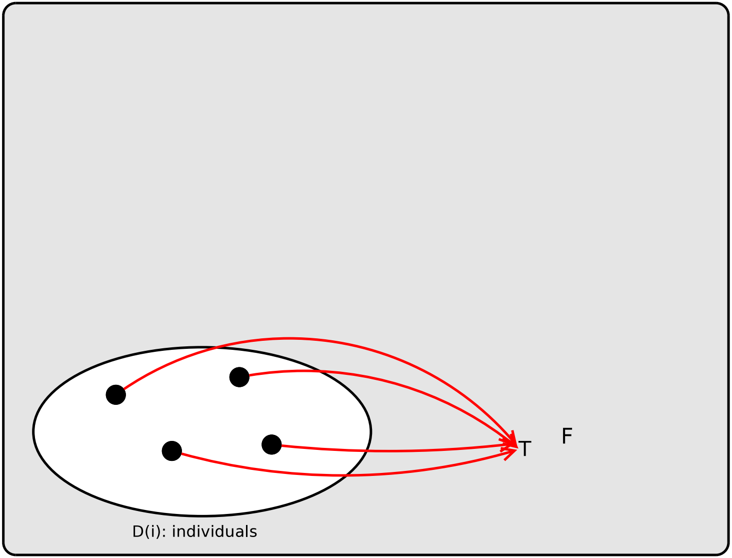



D(i): individuals

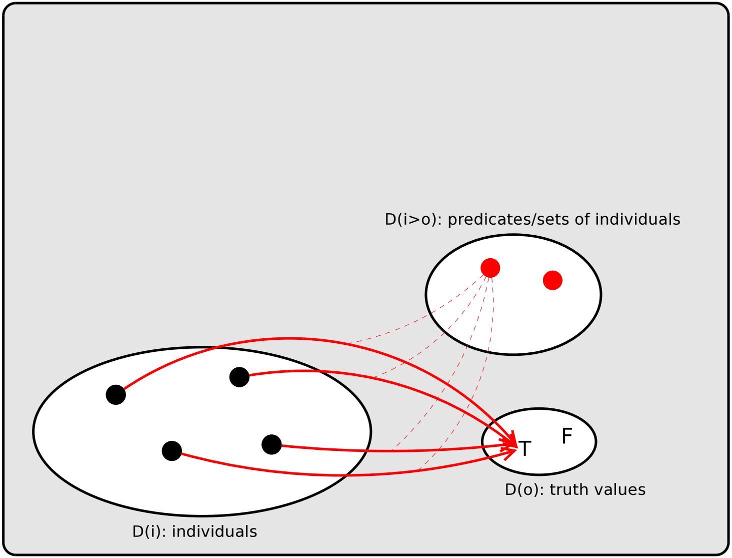
# Semantics of HOL



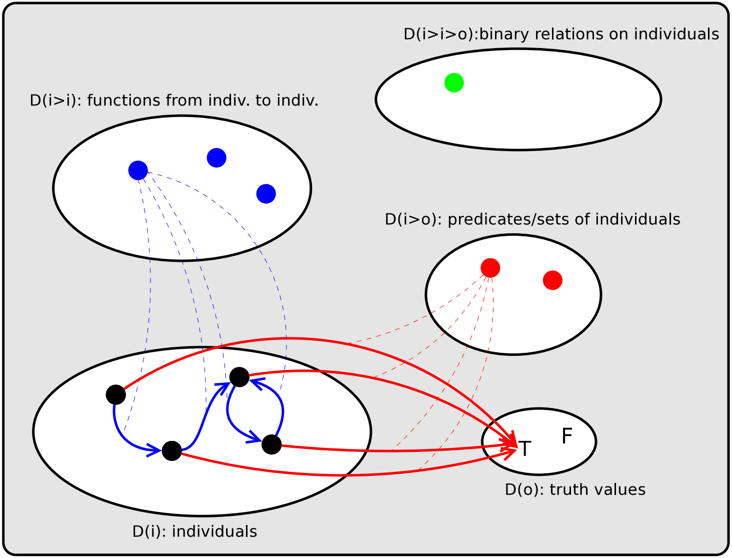
# Semantics of HOL



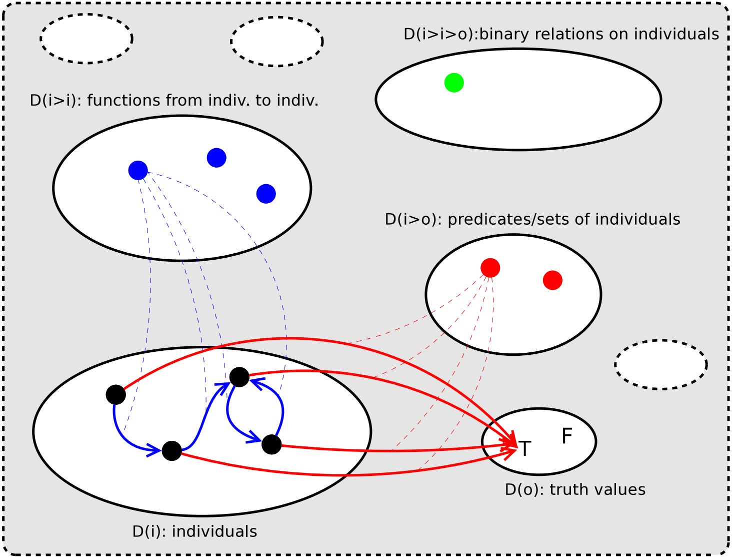
# Semantics of HOL



# Semantics of HOL



# Semantics of HOL



## Sets and Relations in HOL

| | | |
|---|---|---|
| $\in$ | $:=$ | $\lambda x. \lambda A. A(x)$ |
| $\emptyset$ | $:=$ | $\lambda x. \bot$ |
| $\cap$ | $:=$ | $\lambda A. \lambda B. (\lambda x. x \in A \wedge x \in B)$ $\quad \{x \mid x \in A or x \in B\}$ |
| $\cup$ | $:=$ | $\lambda A. \lambda B. (\lambda x. x \in A \vee x \in B)$ |
| $\setminus$ | $:=$ | $\lambda A. \lambda B. (\lambda x. x \in A \vee x \notin B)$ |
| ... | | |
| $\subseteq$ | $:=$ | $\lambda A. \lambda B. (\forall x. x \in A \Rightarrow x \in B)$ |
| $\mathcal{P}$ | $:=$ | $\lambda A. (\lambda B. B \subseteq A)$ |
| ... | | |
| `reflexive` | $:=$ | $\lambda R. (\forall x. R(x, x))$ |
| `transitive` | $:=$ | $\lambda R. (\forall x, y, z. (R(x, y) \wedge R(y, z)) \Rightarrow R(x, z))$ |
| ... | | |

[BenzmüllerEtAl., Journal of Applied Logic, 2008]

## Typed Sets and Relations in HOL

$$
\begin{array}{lll}
\in & := & \lambda x_\alpha.\, \lambda A_{\alpha\to o}.\, A(x) \\
\emptyset & := & \lambda x_\alpha.\, \bot \\
\cap & := & \lambda A_{\alpha\to o}.\, \lambda B_{\alpha\to o}.\, (\lambda x_\alpha.\, x \in A \wedge x \in B) \\
\cup & := & \lambda A_{\alpha\to o}.\, \lambda B_{\alpha\to o}.\, (\lambda x_\alpha.\, x \in A \vee x \in B) \\
\setminus & := & \lambda A_{\alpha\to o}.\, \lambda B_{\alpha\to o}.\, (\lambda x_\alpha.\, x \in A \vee x \notin B) \\
\ldots & &
\end{array}
$$

## Typed Sets and Relations in HOL

$$\begin{array}{lll}
\in & := & \lambda x_\alpha.\, \lambda A_{\alpha\to o}.\, A(x) \\
\emptyset & := & \lambda x_\alpha.\, \bot \\
\cap & := & \lambda A_{\alpha\to o}.\, \lambda B_{\alpha\to o}.\, (\lambda x_\alpha.\, x \in A \wedge x \in B) \\
\cup & := & \lambda A_{\alpha\to o}.\, \lambda B_{\alpha\to o}.\, (\lambda x_\alpha.\, x \in A \vee x \in B) \\
\setminus & := & \lambda A_{\alpha\to o}.\, \lambda B_{\alpha\to o}.\, (\lambda x_\alpha.\, x \in A \vee x \notin B) \\
\ldots & &
\end{array}$$

## Polymorphism is a Challenge for Automation

- One source of indeterminism / blind guessing

[TheissBenzmüller, IWIL-WS@LPAR, 2006]

Mechanization & Automation of HOL

## HOL Applications in Formal Methods

- Systems: Isabelle/HOL (Cambridge/München), HOL4 (Cambridge), HOL-Light (INTEL), PVS (SRI), Nuprl (Cornell), ..., OMEGA
- Project example (formal verification)
  - Flyspeck (Th. Hales, U Pittsburgh)
    - Goal: formal verification of his proof of Keppler's Conjecture (1611)
    - Application of HOL-Light & Isabelle/HOL & ...
    - 'may take up to 20 work-years' (Flyspeck website)

- Crucial resource: user interaction
- Countermeasure: improving the automation support

Knowledge Representation | Agent Systems | Formal Methods | Security

Knowledge | Obligations

Spatial Structures | Time

Counterfactuals | Rights

Norms

Extension/Intension | Constructivism

Beliefs

... | ...

Computational Linguistics | Philosophy | Mathematics

# Mechanization & Automation of HOL: Motivation 2

Knowledge Representation — Agent Systems — Formal Methods — Security

Knowledge — Obligations

Spatial Structures — Time

**Application areas for non-classical logics**

Norms

Extension/Intension — Constructivism

Beliefs

... — ...

Computational Linguistics — Philosophy — Mathematics

Knowledge Representation Agent Systems Formal Methods Security

Knowledge Obligations

- Many relevant non-classical logics
- Many relevant combinations of logics
- Few proof systems for individual logics
- Fewer proof systems for logic combinations
- Very few generic approaches

Beliefs

... ...

Computational Linguistics Philosophy Mathematics

Knowledge Representation | Agent Systems | Formal Methods | Security

Knowledge | Obligations

**Core Questions:**

1. Classical Higher Order Logic (HOL) as Universal Logic?
2. HOL Provers & Model Finders as Generic Reasoning Tools?
3. Combinations with Specialist Reasoners (if available)?

Extension/Intension | Constructivism

Beliefs

... | ...

Computational Linguistics | Philosophy | Mathematics

# Automation of HOL: A Nightmare?

## Undecidable and Infinitary Unification

$\exists F_{\iota\to\iota}. F(g(x)) = g(F(x))$

(1) $F \leftarrow \lambda y_i. y$
(2) $F \leftarrow \lambda y_i. g(y)$
(3) $F \leftarrow \lambda y_i. g(g(y))$
(4) $\ldots$

$\longrightarrow$ enforce decidability

# Automation of HOL: A Nightmare?

## Primitive Substitution

Example Theorem: $\exists S. \mathit{reflexive}(S)$

Negation and Expansion of Definitions:

$$\neg\exists S. (\forall x_\iota. S(x,x))$$

Clause Normalisation ($a(S)$ Skolem term):

$$\neg S(a(S), a(S))$$

**Guess** some suitable instances for $S$

$S \leftarrow \lambda y. \lambda z. \top$

$\rightsquigarrow \neg\top$

$S \leftarrow \lambda y. \lambda z. V(y,z) = W(y,z)$

$\rightsquigarrow V(a(S), a(S)) \neq W(a(S), a(S))$

$S \leftarrow \ldots$

# Automation of HOL: A Nightmare?

## Cut rule

$$\frac{A \Rightarrow \mathbf{C} \quad \mathbf{C} \Rightarrow B}{A \Rightarrow B}$$

considered as bad in ATP

## Cut rule

$$\frac{A \Rightarrow C \quad C \Rightarrow B}{A \Rightarrow B}$$

considered as bad in ATP

## [IJCAR-06]: Axioms that imply Cut

- Axiom of excluded middle
- Comprehension axioms
- Functional and Boolean extensionality
- Leibniz and other definitions of equality
- Axiom of induction
- Axiom of choice
- Axiom of description

[BenzmüllerEtAl., Logical Methods in Computer Science, 2009]

# Automation of HOL: A Nightmare?

## Cut rule

$$\frac{A \Rightarrow \mathbf{C} \quad \mathbf{C} \Rightarrow B}{A \Rightarrow B}$$

considered as bad in ATP

## Calculi that avoid axioms

- Axiom of excluded middle ✓
- Comprehension axioms ✓
- Functional and Boolean extensionality ✓[CADE-98,PhD-99]
- Leibniz and other definitions of equality ✓[CADE-99,PhD-99]
- Axiom of induction ?
- Axiom of choice ✓(see recent work of Brown)
- Axiom of description ✓(see recent work of Brown)

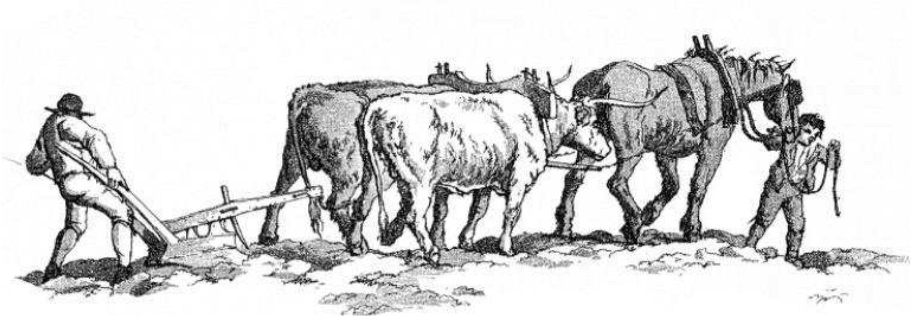
[BenzmüllerEtAl., Logical Methods in Computer Science, 2009]

LEO-II

An Effective Higher-Order Theorem Prover

UNIVERSITY OF CAMBRIDGE

UNIVERSITÄT DES SAARLANDES



**LEO-II employs FO-ATPs: E, Spass, Vampire**

Download and further Information: `www.leoprover.org`

[BenzmüllerEtAl., IJCAR, 2008]

## Motivation for LEO-II

- TPS system of Peter Andrews et al.

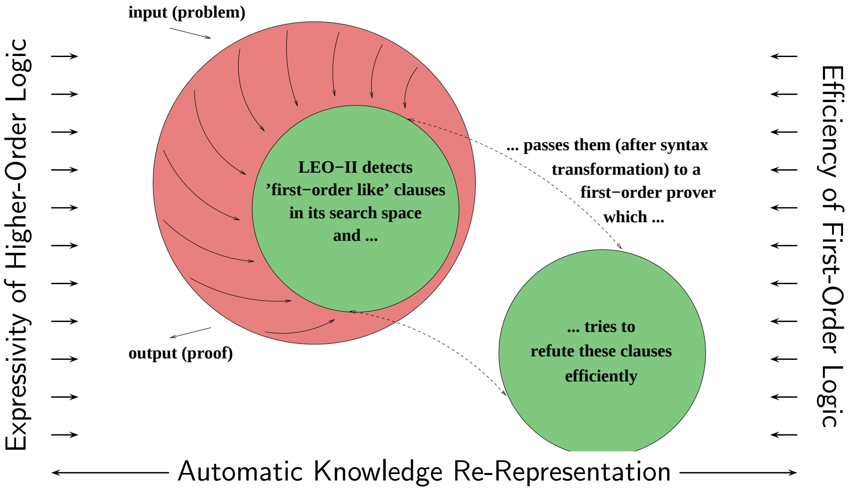
- LEO hardwired to $\Omega$mega (predecessor of LEO-II)
- Agent-based architecture $\Omega$-Ants
  (with V. Sorge) [AIMSA-98,EPIA-99,Calculemus-00]

- Collaboration of LEO with FO-ATP via $\Omega$-Ants
  (with V. Sorge) [KI-01,LPAR-05,JAL-08]

- Progress in Higher-Order Termindexing
  (with F. Theiss and A. Fietzke) [IWIL-06]

$\Longrightarrow$ Development of LEO-II with L. Paulson at Cambridge University

# Architecture of LEO-II



# Outline of the LEO-II Loop

Main Termination Criterion: generation of empty clause, then raise excepetion/stop

# Outline of the LEO-II Loop

Main Termination Criterion: generation of empty clause, then raise excepetion/stop

Pre-Processing

- abbreviation expansion, splitting, extensional normalisation and debth-bound extensional pre-unification, Skolemization, primitive substitution, simplification, etc.
- initialize clause sets: passive=emptyset, active=results from above
- call fo-atp with fo-like clauses; stop if refutation found

# Outline of the LEO-II Loop

Main Termination Criterion: generation of empty clause, then raise excepetion/stop

Pre-Processing

- abbreviation expansion, splitting, extensional normalisation and debth-bound extensional pre-unification, Skolemization, primitive substitution, simplification, etc.
- initialize clause sets: passive=emptyset, active=results from above
- call fo-atp with fo-like clauses; stop if refutation found

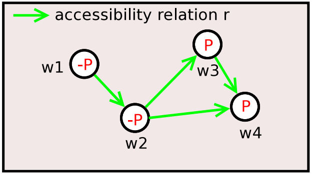
LEO-II Loop

- while 'Reasoning-Timeout' not yet reached do
  - increment loop counter (stop when maximal number of loops reached)
  - call fo-atp with fo-like clauses; stop if refutation found
  - choose new lightest clause form active and rename free vars
  - if lightest is-subsumed-by passive then nothing else
    - remove subsumed clauses from active and add lightest clause to passive
    - resolve all clauses in active against lightest clauses
    - (apply primitive substitution to ligtest clause)
    - (apply positive boolean extensionality to lightest clause)
    - apply restricted factorization to lightest clause
    - process resulting clauses with: extensional normalisation and debth-bound extensional pre-unification, simplification
    - add resulting clauses to active

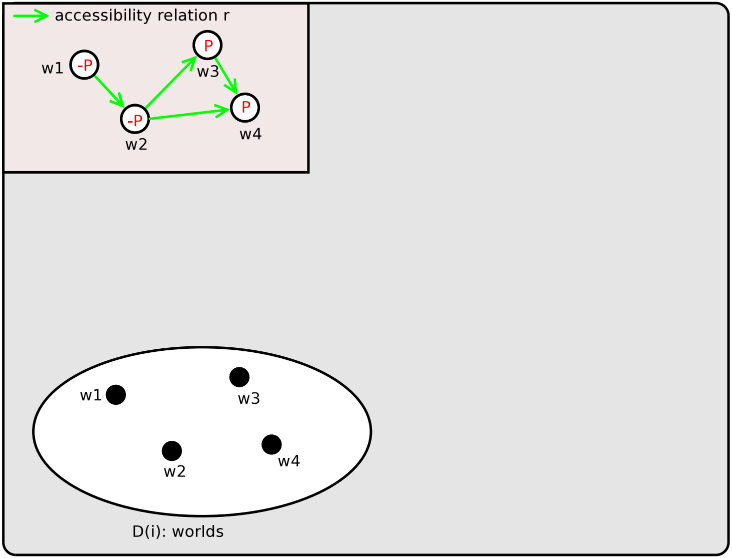
```
(** The Main Loop *)

let loop (st:state) =
    while (not (check_local_max_time st))
    do
     let lc = inc_loop_count st in
       if (st.flags.max_loop_count > 0 ) && (st.loop_count >= st.flags.max_loop_count)
       then raise (Failure "Max loops") else ();
       if (not (st.flags.atp_prover = "none"))
       then call_fo_atp_according_to_frequency_flag st st.flags.atp_prover
       else ();
       let lightest = choose_and_remove_lightest_from_active st in
       let lightest' = rename_free_variables lightest st in
         if is_subsumed_by lightest' (Clauseset.elements st.passive) st "fo-match" then ()
         else
           (
             set_passive st (list_to_set (delete_subsumed_clauses (Clauseset.elements st.passive)
                                                                   lightest' st "fo-match"));
             add_to_passive st lightest';
             let res_resolve = List.fold_right (fun cl cll -> (resolve lightest cl st)@cll)
                                              (Clauseset.elements st.passive) [] in
               let res_prim_subst = [] and res_pos_bool = []
               and res_fac_restr = (raise_to_list factorize_restricted) [lightest] st in
                 let res_processed =
                   compose [(raise_to_list unify_pre_ext);
                            exhaustive (raise_to_list cnf_normalize_step);
                            exhaustive (raise_to_list simplify)]
                           (res_resolve@res_prim_subst@res_pos_bool@res_fac_restr) st in
                   index_clauselist_with_role res_processed st;
                   set_active st (list_to_set (res_processed@(Clauseset.elements st.active)));
           )
    done
```

Examples of Natural Fragments of HOL:
Quantified Multimodal Logics & Others

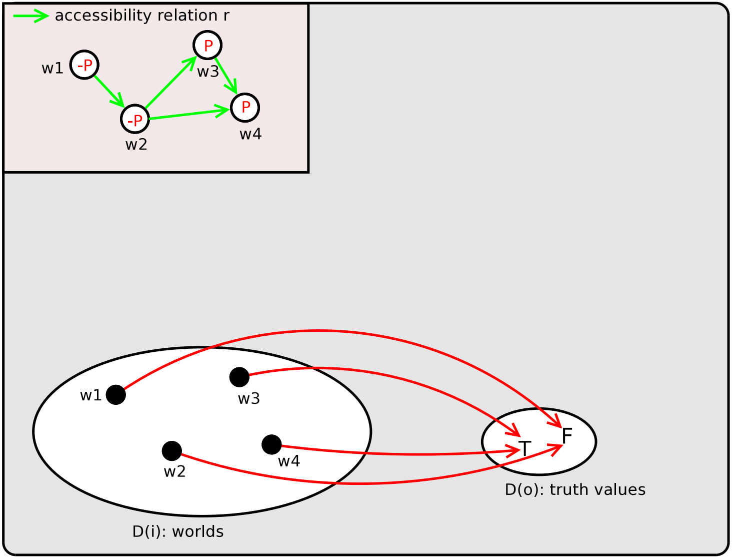
# Combining the Kripke View and the Tarski View on Logics



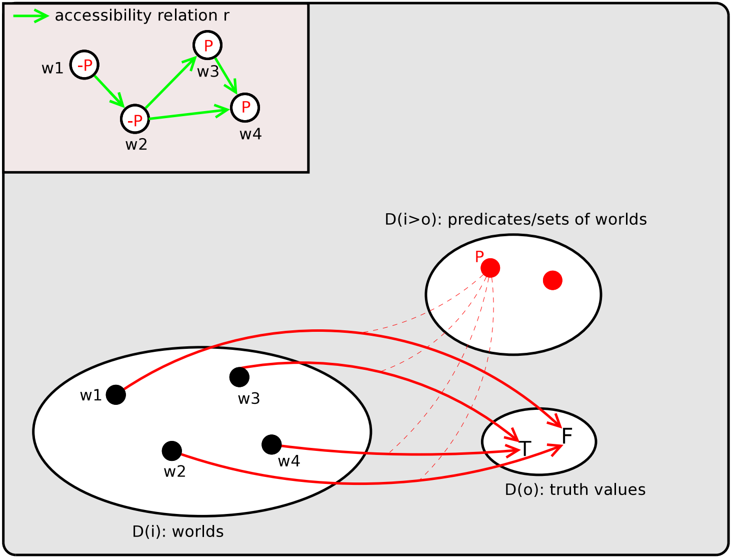
# Combining the Kripke View and the Tarski View on Logics

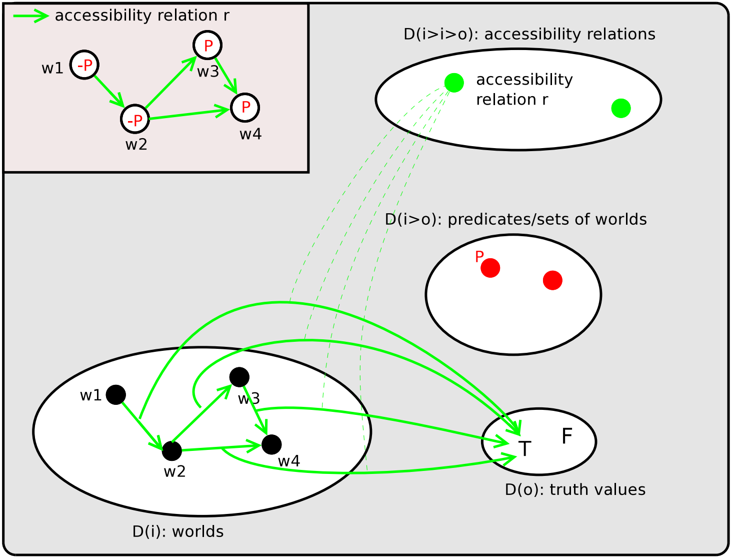


# Combining the Kripke View and the Tarski View on Logics

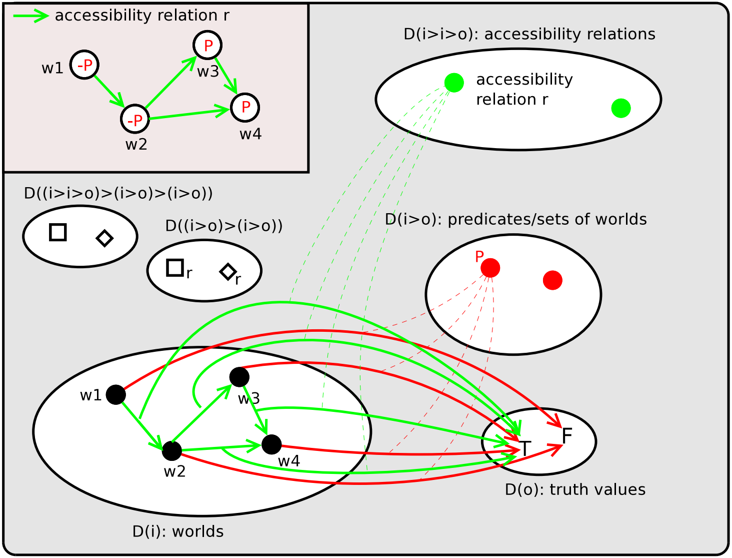


# Combining the Kripke View and the Tarski View on Logics

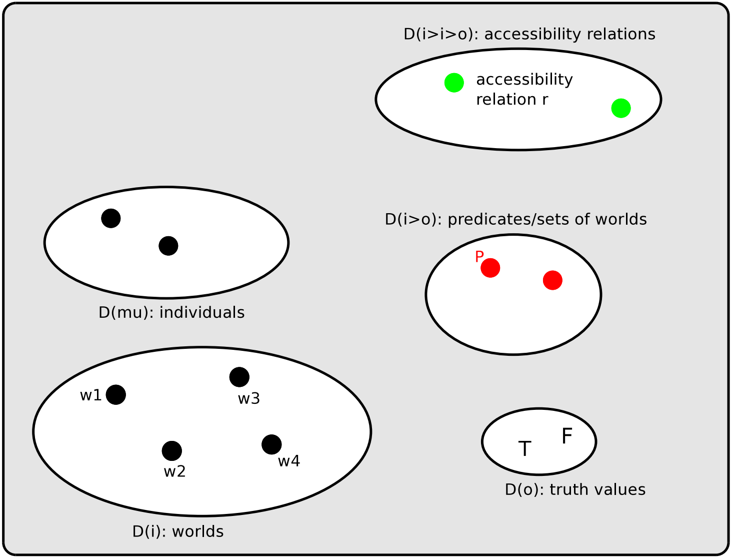



accessibility relation r
w1
-P
w2
-P
w3
P
w4
P
D(i>i>o): accessibility relations
accessibility
relation r
D(i>o): predicates/sets of worlds
P
w1
w2
w3
w4
T
F
D(o): truth values
D(i): worlds

# Combining the Kripke View and the Tarski View on Logics



# Combining the Kripke View and the Tarski View on Logics



# Multimodal Logics in HOL

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

- Syntax MML
  - formulas s

  Kripke Semantics
  - worlds w
  - accessibility relations r

  $\xrightarrow[\text{transformation}]{\text{explicit}}$ First Order Logic

  e.g. work of Ohlbach

Not Needed!

# Multimodal Logics in HOL

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

- HOL

  Syntax MML
  - formulas s

  Kripke Semantics
  - worlds w
  - accessibility relations r

  $\longrightarrow$ terms $s_{\iota \to o}$

  $\longrightarrow$ terms $w_\iota$

  $\longrightarrow$ terms $r_{\iota \to \iota \to o}$

# Multimodal Logics in HOL

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

- HOL

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

- MML Syntax as Abbreviations of HOL-Terms

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

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

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

HOL

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

- MML Syntax as Abbreviations of HOL-Terms

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

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

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

- HOL

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

- MML Syntax as Abbreviations of HOL-Terms

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

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

# Embedding Meta-Level Notions

- Validity

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

Also

- Satisfiability
- Countersatisfiability
- Unsatisfiability

# Embedding Multimodal Logics in HOL

- Kripke style semantics

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

- Semantic embedding: ML $\longrightarrow$ HOL terms of type $\iota \rightarrow o$
  Base type $\iota$ is identified with set of worlds $W \neq \emptyset$

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

- Kripke style semantics

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

- Semantic embedding: ML $\longrightarrow$ HOL terms of type $\iota \to o$
  Base type $\iota$ is identified with set of worlds $W \neq \emptyset$

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

# Embedding Multimodal Logics in HOL

- Kripke style semantics

| | | |
|---|---|---|
| $M, w \models P$ | arbitrary | |
| $M, w \models \neg s$ | iff | not $M, w \models s$ |
| $M, w \models s \vee t$ | iff | $M, w \models s$ or $M, w \models s$ |
| $M, w \models \Box_r\, s$ | iff | $M, u \models s$ for all $v$ mit $r(w, v)$ |

- Semantic embedding: ML $\longrightarrow$ HOL terms of type $\iota \rightarrow o$

  Base type $\iota$ is identified with set of worlds $W \neq \emptyset$

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

- Kripke style semantics

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

- Semantic embedding: ML $\longrightarrow$ HOL terms of type $\iota \rightarrow o$
  Base type $\iota$ is identified with set of worlds $W \neq \emptyset$

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

- Kripke style semantics

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

- Semantic embedding: ML $\longrightarrow$ HOL terms of type $\iota \to o$
  Base type $\iota$ is identified with set of worlds $W \neq \emptyset$

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

- Kripke style semantics

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

- Semantic embedding: ML $\longrightarrow$ HOL terms of type $\iota \to o$
  Base type $\iota$ is identified with set of worlds $W \neq \emptyset$

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

- Kripke style semantics

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

- Semantic embedding: ML $\longrightarrow$ HOL terms of type $\iota \rightarrow o$
  Base type $\iota$ is identified with set of worlds $W \neq \emptyset$

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

# Embedding Multimodal Logics in HOL

- Kripke style semantics

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

- Semantic embedding: ML $\longrightarrow$ HOL terms of type $\iota \to o$
  Base type $\iota$ is identified with set of worlds $W \neq \emptyset$

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

# Embedding ~~Multimodal~~ Conditional Logics in HOL

- Kripke style semantics — higher-order selection function!

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

- Semantic embedding: ML $\longrightarrow$ HOL terms of type $\iota \rightarrow o$

  Base type $\iota$ is identified with set of worlds $W \neq \emptyset$

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

# Embedding ~~Multimodal~~ Conditional Logics in HOL

- Kripke style semantics

higher-order selection function!

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

- Semantic embedding: ML $\longrightarrow$ HOL terms of type $\iota \to o$

  Base type $\iota$ is identified with set of worlds $W \neq \emptyset$

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

- Kripke style semantics — higher-order selection function!

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

- Semantic embedding: ML $\longrightarrow$ HOL terms of type $\iota \rightarrow o$

  Base type $\iota$ is identified with set of worlds $W \neq \emptyset$

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

# Trivial Examples

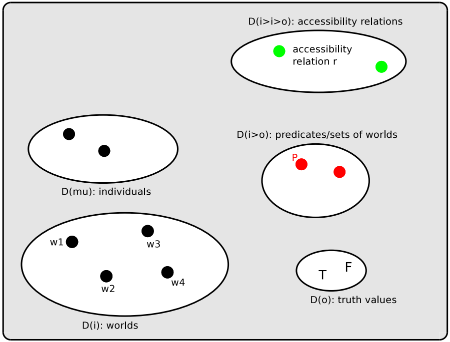
- Remember

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

Examples on blackboard

- valid $\forall^\mu X.\,(m\, X)$
- valid $\forall^\mu X.\,\Box_r\,(m\, X)$

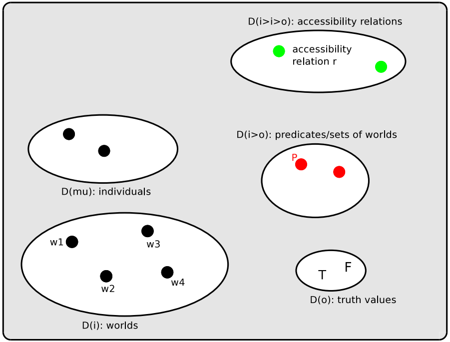
## Constant Domain




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

[BenzmüllerPaulson, Logica Universalis, to appear]

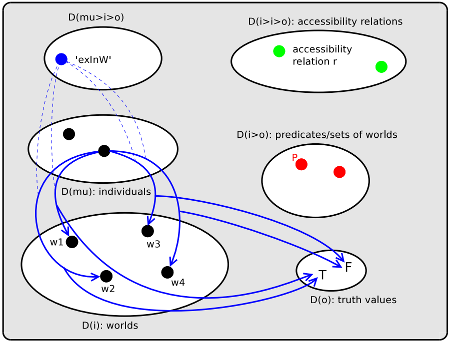
## Constant Domain




$$\textcolor{red}{\forall^{\mu}} = \lambda Q. \lambda W_{\iota}. \forall X_{\mu}. (Q\ X\ W)$$

[BenzmüllerPaulson, Logica Universalis, to appear]

## Cumulative Domain




$$\textcolor{red}{\forall^{\mu}} = \lambda Q. \lambda W_{\iota}. \forall X_{\mu}. (\textcolor{blue}{\text{exInW}}\ X\ W) \Rightarrow (Q\ X\ W)$$

1: $\forall X_{\mu}, V_{\iota}, W_{\iota}. (\textcolor{blue}{\text{exInW}}\ X\ V) \wedge (r\ V\ W) \Rightarrow (\textcolor{blue}{\text{exInW}}\ X\ W)$
2: $\forall W_{\iota}. \exists X_{\mu}. (\textcolor{blue}{\text{exInW}}\ X\ W)$
3($\textcolor{red}{c}$): $\forall W_{\iota}. (\textcolor{blue}{\text{exInW}}\ \textcolor{red}{c}\ W)$

[ongoing work with Otten and Raths]

# Region Connection Calculus (RCC) is a Fragment of HOL

Region Connection Calculus for spatial reasoning [RandellCuiCohn, 1992]

| | | |
|---|---|---|
| disconnected : | *dc* | $= \lambda X_\tau. \lambda Y_\tau. \neg(c\ X\ Y)$ |
| part of : | *p* | $= \lambda X_\tau. \lambda Y_\tau. \forall Z. ((c\ Z\ X) \Rightarrow (c\ Z\ Y))$ |
| identical with : | *eq* | $= \lambda X_\tau. \lambda Y_\tau. ((p\ X\ Y) \wedge (p\ Y\ X))$ |
| overlaps : | *o* | $= \lambda X_\tau. \lambda Y_\tau. \exists Z. ((p\ Z\ X) \wedge (p\ Z\ Y))$ |
| partially o : | *po* | $= \lambda X_\tau. \lambda Y_\tau. ((o\ X\ Y) \wedge \neg(p\ X\ Y) \wedge \neg(p\ Y\ X))$ |
| ext. connected : | *ec* | $= \lambda X_\tau. \lambda Y_\tau. ((c\ X\ Y) \wedge \neg(o\ X\ Y))$ |
| proper part : | *pp* | $= \lambda X_\tau. \lambda Y_\tau. ((p\ X\ Y) \wedge \neg(p\ Y\ X))$ |
| tangential pp : | *tpp* | $= \lambda X_\tau. \lambda Y_\tau. ((pp\ X\ Y) \wedge \exists Z. ((ec\ Z\ X) \wedge (ec\ Z\ Y)))$ |
| nontang. pp : | *ntpp* | $= \lambda X_\tau. \lambda Y_\tau. ((pp\ X\ Y) \wedge \neg\exists Z. ((ec\ Z\ X) \wedge (ec\ Z\ Y)))$ |

[Benzmüller, AMAI, 2011]

# Semantic Web Language OWL is a Fragment of HOL

- Class expressions become terms of type $\iota \to o$
- Class membership becomes class application $(C\ a)$
- Role expressions become terms of type $\iota \to \iota \to o$
- Role membership becomes role application $(R\ a\ b)$

## Semantic Web Language OWL is a Fragment of HOL

- Class expressions become terms of type $\iota \rightarrow o$
- Class membership becomes class application $(C\ a)$
- Role expressions become terms of type $\iota \rightarrow \iota \rightarrow o$
- Role membership becomes role application $(R\ a\ b)$
- The class connectives $\bot, \top, \neg, \sqcup, \forall, \geq_n$ can be defined as

$$\begin{aligned}
\top &= \lambda X_\iota . \top \\
\bot &= \lambda X_\iota . \bot \\
\neg &= \lambda C_{\iota\rightarrow o} . \lambda X_\iota . \neg (C\ X) \\
\sqcup &= \lambda C_{\iota\rightarrow o} . \lambda D_{\iota\rightarrow o} . \lambda X_\iota . (C\ X) \vee (C\ X) \\
\forall &= \lambda R_{\iota\rightarrow\iota\rightarrow o} . \lambda C_{\iota\rightarrow o} . \lambda X_\iota . \forall Y_\iota . (R\ X\ Y) \Rightarrow (C\ Y) \\
\geq_n &= \lambda R_{\iota\rightarrow\iota\rightarrow o} . \lambda C_{\iota\rightarrow o} . \lambda X_\iota . ((\#\ \lambda Y_\iota . (R\ X\ Y) \wedge (C\ Y)) \geq n)
\end{aligned}$$

  (# is an appropriately defined set cardinality function)
- Role inverse $R^-$ is defined as $^- = \lambda R_{\iota\rightarrow\iota\rightarrow o} . \lambda X_\iota . \lambda Y_\iota . (R\ Y\ X)$

## Semantic Web Language OWL is a Fragment of HOL

- Class expressions become terms of type $\iota \to o$
- Class membership becomes class application $(C\ a)$
- Role expressions become terms of type $\iota \to \iota \to o$
- Role membership becomes role application $(R\ a\ b)$
- Definition of further connectives

$$\sqsubseteq^1 = \lambda C_{\iota\to o}.\lambda D_{\iota\to o}.\forall X_\iota.(C\ X) \Rightarrow (D\ X)$$

$$\sqsubseteq^2 = \lambda R_{\iota\to\iota\to o}.\lambda S_{\iota\to\iota\to o}.\forall X_\iota.\forall Y_\iota.(R\ X\ Y) \Rightarrow (S\ X\ Y)$$

$$\mathrm{Dis} = \lambda R_{\iota\to\iota\to o}.\lambda S_{\iota\to\iota\to o}.\neg\exists X_\iota.\exists Y_\iota.(R\ X\ Y) \wedge (S\ X\ Y)$$

$$\circ = \lambda R_{\iota\to\iota\to o}.\lambda S_{\iota\to\iota\to o}.\lambda X_\iota.\lambda Y_\iota.\exists Z_\iota.(R\ X\ Z) \wedge (S\ Z\ Y)$$

# Semantic Web Language OWL is a Fragment of HOL

- Class expressions become terms of type $\iota \rightarrow o$
- Class membership becomes class application $(C\ a)$
- Role expressions become terms of type $\iota \rightarrow \iota \rightarrow o$
- Role membership becomes role application $(R\ a\ b)$

As we have seen before:

OWL connectives are just abbreviations of HOL terms

[Benzmüller, Research Proposal]

## Soundness and Completeness

$$\models^{ML} s \quad \text{iff} \quad \models^{HOL} \text{valid}\ s_{\iota \to o}$$

- Propositional Multimodal Logics [BenzmüllerPaulson, Log.J.IGPL, 2010]
- Quantified Multimodal Logics [BenzmüllerPaulson, CLIMA XI, 2010] [Benzmüller, AMAI, 2011], [BenzmüllerPaulson, Logica Universal., to appear]
- Propositional & Quantified Conditional Logics [BenzmüllerGenovese, NCMPL, 2011], [BenzmüllerGenoveseGabbayRispoli, submitted, 2011]
- Temporal Logics: use semantic modeling of 'irreflexive'
- Intuitionistic Logics: [BenzmüllerPaulson, Log.J.IGPL, 2010]
- Access Control Logics: [Benzmüller, IFIP SEC, 2009]

Work in progress

- Spatial Reasoning 'RCC' [Benzmüller, AMAI, 2011]
- Semantic Web Language 'OWL' [Benzmüller, Research Proposal, 2010]
- ...

Automation of Logics and Logic Combinations in HOL

# A Simple Example

Problem in Multimodal Logic K $\neg(\Box_a \Box_b P) \vee \Box_a P$

Problem in HOL valid $\neg(\Box_a \Box_b P) \vee \Box_a P$

## A Simple Example

Problem in Multimodal Logic K $\neg(\Box_a \Box_b P) \vee \Box_a P$

Problem in HOL $\text{valid}\ \neg(\Box_a \Box_b P) \vee \Box_a P$

$\forall W_\iota . (\neg(\Box_a \Box_b P) \vee \Box_a P)\ W$

expanded abbreviation

## A Simple Example

Problem in Multimodal Logic K $\neg(\Box_a \Box_b P) \vee \Box_a P$

Problem in HOL valid $\neg(\Box_a \Box_b P) \vee \Box_a P$

$$\forall W_\iota.(\neg(\Box_a \Box_b P) \vee \Box_a P)\, W$$

expanded abbreviation

$$\forall W_\iota.(\neg(\Box_a \Box_b P)\, W) \vee (\Box_a P\, W)$$

## A Simple Example

Problem in Multimodal Logic K $\neg(\Box_a \Box_b P) \vee \Box_a P$

Problem in HOL valid $\neg(\Box_a \Box_b P) \vee \Box_a P$

$$\forall W_\iota. (\neg(\Box_a \Box_b P) \vee \Box_a P)\, W$$

expanded abbreviation

$$\forall W_\iota. (\neg(\Box_a \Box_b P)\, W) \vee (\Box_a P\, W)$$

$$\forall W_\iota. \neg(\Box_a \Box_b P\, W) \vee (\Box_a P\, W)$$

## A Simple Example

Problem in Multimodal Logic K $\neg(\Box_a \Box_b P) \vee \Box_a P$

Problem in HOL valid $\neg(\Box_a \Box_b P) \vee \Box_a P$

$$\forall W_\iota. (\neg(\Box_a \Box_b P) \vee \Box_a P)\, W$$

expanded abbreviation

$$\forall W_\iota. (\neg(\Box_a \Box_b P)\, W) \vee (\Box_a P\, W)$$

$$\forall W_\iota. \neg(\Box_a \Box_b P\, W) \vee (\Box_a P\, W)$$

$$\forall W_\iota. \neg(\Box_a \Box_b P\, W) \vee (\forall V_\iota. \neg(a\, W\, V) \vee (P\, W))$$

## A Simple Example

Problem in Multimodal Logic K $\neg(\Box_a \Box_b P) \vee \Box_a P$

Problem in HOL valid $\neg(\Box_a \Box_b P) \vee \Box_a P$

$$\forall W_\iota \centerdot (\neg(\Box_a \Box_b P) \vee \Box_a P)\, W$$

expanded abbreviation

$$\forall W_\iota \centerdot (\neg(\Box_a \Box_b P)\, W) \vee (\Box_a P\, W)$$

$$\forall W_\iota \centerdot \neg(\Box_a \Box_b P\, W) \vee (\Box_a P\, W)$$

$$\forall W_\iota \centerdot \neg(\Box_a \Box_b P\, W) \vee (\forall V_\iota \centerdot \neg(a\, W\, V) \vee (P\, W))$$

$$\forall W_\iota \centerdot \neg(\Box_a \Box_b P\, W) \vee (\forall V_\iota \centerdot \neg(a_{\iota\to\iota\to o}\, W\, V) \vee (P_{\iota\to o}\, W))$$

## A Simple Example

Problem in Multimodal Logic K $\neg(\Box_a \Box_b P) \vee \Box_a P$

Problem in HOL $\text{valid } \neg(\Box_a \Box_b P) \vee \Box_a P$

$$\forall W_\iota \centerdot (\neg(\Box_a \Box_b P) \vee \Box_a P)\, W$$

expanded abbreviation

$$\forall W_\iota \centerdot (\neg(\Box_a \Box_b P)\, W) \vee (\Box_a P\, W)$$

$$\forall W_\iota \centerdot \neg(\Box_a \Box_b P\, W) \vee (\Box_a P\, W)$$

$$\forall W_\iota \centerdot \neg(\Box_a \Box_b P\, W) \vee (\forall V_\iota \centerdot \neg(a\, W\, V) \vee (P\, W))$$

$$\forall W_\iota \centerdot \neg(\Box_a \Box_b P\, W) \vee (\forall V_\iota \centerdot \neg(a_{\iota\to\iota\to o}\, W\, V) \vee (P_{\iota\to o}\, W))$$

...

## A Simple Example

Problem in Multimodal Logic K $\neg(\Box_a \Box_b P) \vee \Box_a P$

Problem in HOL

$$\text{valid } \neg(\Box_a \Box_b P) \vee \Box_a P$$

$$\forall W_\iota.\, (\neg(\Box_a \Box_b P) \vee \Box_a P)\, W$$

$$\forall W_\iota.\, (\neg(\Box_a \Box_b P)\, W) \vee (\Box_a P\, W)$$

$$\forall W_\iota.\, \neg(\Box_a \Box_b P\, W) \vee (\Box_a P\, W)$$

$$\forall W_\iota.\, \neg(\Box_a \Box_b P\, W) \vee (\forall V_\iota.\, \neg(a\, W\, V) \vee (P\, W))$$

$$\forall W_\iota.\, \neg(\Box_a \Box_b P\, W) \vee (\forall V_\iota.\, \neg(a_{\iota\to\iota\to o}\, W\, V) \vee (P_{\iota\to o}\, W))$$

$$\ldots$$

$$\forall W_\iota.\, \neg(\ldots\ldots\ldots\ldots\, W) \vee (\forall V_\iota.\, \neg(a_{\iota\to\iota\to o}\, W\, V) \vee (P_{\iota\to o}\, W))$$

## A Simple Example

Problem in Multimodal Logic K $\neg(\Box_a \Box_b P) \vee \Box_a P$

Problem in HOL valid $\neg(\Box_a \Box_b P) \vee \Box_a P$

$$\forall W_\iota. (\neg(\Box_a \Box_b P) \vee \Box_a P)\, W$$

$$\forall W_\iota. (\neg(\Box_a \Box_b P)\, W) \vee (\Box_a P\, W)$$

$$\forall W_\iota. \neg(\Box_a \Box_b P\, W) \vee (\Box_a P\, W)$$

$$\forall W_\iota. \neg(\Box_a \Box_b P\, W) \vee (\forall V_\iota. \neg(a\, W\, V) \vee (P\, W))$$

$$\forall W_\iota. \neg(\Box_a \Box_b P\, W) \vee (\forall V_\iota. \neg(a_{\iota\to\iota\to o}\, W\, V) \vee (P_{\iota\to o}\, W))$$

$$\ldots$$

$$\forall W_\iota. \neg(\ldots\ldots\ldots\ldots\, W) \vee (\forall V_\iota. \neg(a_{\iota\to\iota\to o}\, W\, V) \vee (P_{\iota\to o}\, W))$$

HOL model finder Nitpick (IsabelleN) quickly finds a countermodel

# A Simple Example

Problem in Multimodal Logic K $\neg(\Box_a \Box_b P) \vee \Box_a P$

Problem in HOL valid $\neg(\Box_a \Box_b P) \vee \Box_a P$

$$\forall W_\iota \centerdot (\neg(\Box_a \Box_b P) \vee \Box_a P)\, W$$

$$\forall W_\iota \centerdot (\neg(\Box_a \Box_b P)\, W) \vee (\Box_a P\, W)$$

$$\forall W_\iota \centerdot \neg(\Box_a \Box_b P\, W) \vee (\Box_a P\, W)$$

$$\forall W_\iota \centerdot \neg(\Box_a \Box_b P\, W) \vee (\forall V_\iota \centerdot \neg(a\, W\, V) \vee (P\, W))$$

$$\forall W_\iota \centerdot \neg(\Box_a \Box_b P\, W) \vee (\forall V_\iota \centerdot \neg(a_{\iota\to\iota\to o}\, W\, V) \vee (P_{\iota\to o}\, W))$$

$$\dots$$

$$\forall W_\iota \centerdot \neg(\dots\dots\dots\dots\, W) \vee (\forall V_\iota \centerdot \neg(a_{\iota\to\iota\to o}\, W\, V) \vee (P_{\iota\to o}\, W))$$

HOL model finder Nitpick (IsabelleN) quickly finds a countermodel

Countermodel for $\neg(\Box_a \Box_b P) \vee \Box_a P$

## Exemplary study of combinations of logics

- Agent scenarios (e.g. Wise Men Puzzle)
  - common knowledge & knowledge of single agents & time

[Benzmüller, AMAI, 2011]

- Novel combinations
  - knowledge of agents & spatial reasoning

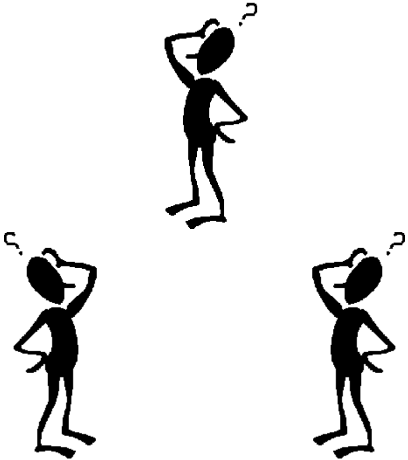
[Benzmüller, DagstuhlSeminar, 2010]

- Combinations that are relevant for expressive ontologies (SUMO)
  - knowledge & belief & time & spatial reasoning & ...

[BenzmüllerPease, ARCOE-WS@ECAI, 2010]
[BenzmüllerPease, PAAR-WS@IJCAR, 2010]
[BenzmüllerPease, AICom, to appear]
[BenzmüllerPease, JWS, in revision]

<u>*Wise Men Puzzle*</u>

*Once upon a time, a king wanted to find the wisest out of his three wisest men. He arranged them in a circle and told them that he would put a white or a black spot on their foreheads and that one of the three spots would certainly be white. The three wise men could see and hear each other but, of course, they could not see their faces reflected anywhere. The king, then, asked to each of them to find out the color of his own spot. After a while, the wisest correctly answered that his spot was white.*



[Benzmüller, AMAI, 2011]

<u>*Wise Men Puzzle*</u>

*Once upon a time, a king wanted to find the wisest out of his three wisest men. He arranged them in a circle and told them that he would put a white or a black spot on their foreheads and that one of the three spots would certainly be white. The three wise men could see and hear each other but, of course, they could not see their faces reflected anywhere. The king, then, asked to each of them to find out the color of his own spot. After a while, the wisest correctly answered that his spot was white.*

| | A | B | C | Answer |
|---|---|---|---|---|
| 1 | * | b | b | A w |
| 2 | b | * | b | B w |
| 3 | b | b | * | C w |
| 4 | * | w | b | A w (→ 2.) |
| 5 | * | b | w | A w (→ 3.) |
| 6 | w | * | b | B w (→ 1.) |
| 7 | b | * | w | B w (→ 3.) |
| 8 | w | b | * | C w (→ 1.) |
| 9 | b | w | * | C w (→ 2.) |
| 10 | * | w | w | A w (→ 9.) |
| 11 | w | * | w | B w (→ 8.) |
| 12 | w | w | * | C w (→ 4.) |

[Benzmüller, AMAI, 2011]

Wise Men Puzzle

*Once upon a time, a king wanted to find the wisest out of his three wisest men. He arranged them in a circle and told them that he would put a white or a black spot on their foreheads and that one of the three spots would certainly be white. The three wise men could see and hear each other but, of course, they could not see their faces reflected anywhere. The king, then, asked to each of them to find out the color of his own spot. After a while, the wisest correctly answered that his spot was white.*

(formalization adapted from: [Baldoni, PhD, 1998])

- epistemic modalities (knowledge):

  $\Box_a$, $\Box_b$, $\Box_c$ : individual knowledge of the men

  $\Box_{fool}$ : common knowledge

- predicate symbol:

  $ws$: 'has white spot'

- axioms for common knowledge (S4)

$$\text{valid } \forall^p \phi_\centerdot\ \Box_{\text{fool}}\, \phi \Rightarrow \phi \qquad \text{(M)}$$

$$\text{valid } \forall^p \phi_\centerdot\ \Box_{\text{fool}}\, \phi \Rightarrow \Box_{\text{fool}}\, \Box_{\text{fool}}\, \phi \qquad \text{(4)}$$

inclusion axioms

$$\text{valid } \forall^p \phi_\centerdot\ \Box_{\text{fool}}\, \phi \Rightarrow \Box_X\, \phi$$

$X \in \{a, b, c\}$

[Benzmüller, AMAI, 2011]

Wise Men Puzzle

*Once upon a time, a king wanted to find the wisest out of his three wisest men. He arranged them in a circle and told them that he would put a white or a black spot on their foreheads and that one of the three spots would certainly be white. The three wise men could see and hear each other but, of course, they could not see their faces reflected anywhere. The king, then, asked to each of them to find out the color of his own spot. After a while, the wisest correctly answered that his spot was white.*

(formalization adapted from: [Baldoni, PhD, 1998])

- common knowledge:

  at least one of the men has a white spot

  $$\text{valid } \Box_{\text{fool}} (ws\ a) \vee (ws\ b) \vee (ws\ c)$$

  if X has a white spot, then Y knows this

  $$\text{valid } \Box_{\text{fool}} (ws\ X) \Rightarrow \Box_Y (ws\ X)$$

  $$X \neq Y \in \{a, b, c\}$$

  if X does not have a white spot, then Y knows this

  $$\text{valid } \Box_{\text{fool}} \neg (ws\ X) \Rightarrow \Box_Y \neg (ws\ X)$$

  $$X \neq Y \in \{a, b, c\}$$

<u>*Wise Men Puzzle*</u>

*Once upon a time, a king wanted to find the wisest out of his three wisest men. He arranged them in a circle and told them that he would put a white or a black spot on their foreheads and that one of the three spots would certainly be white. The three wise men could see and hear each other but, of course, they could not see their faces reflected anywhere. The king, then, asked to each of them to find out the color of his own spot. After a while, the wisest correctly answered that his spot was white.*

(formalization adapted from: [Baldoni, PhD, 1998])

- if X knows $\phi$, then Y knows that X knows $\phi$

$$\text{valid } \forall^p \phi_{\blacksquare} \, (\Box_X \, \phi \Rightarrow \Box_Y \, \Box_X \, \phi)$$

$$X \neq Y \in \{a, b, c\}$$

- if X does not know $\phi$, then Y knows that X does not know $\phi$

$$\text{valid } \forall^p \phi_{\blacksquare} \, (\neg \, \Box_X \, \phi \Rightarrow \Box_Y \, \neg \, \Box_X \, \phi)$$

$$X \neq Y \in \{a, b, c\}$$

[Benzmüller, AMAI, 2011]

<u>*Wise Men Puzzle*</u>

*Once upon a time, a king wanted to find the wisest out of his three wisest men. He arranged them in a circle and told them that he would put a white or a black spot on their foreheads and that one of the three spots would certainly be white. The three wise men could see and hear each other but, of course, they could not see their faces reflected anywhere. The king, then, asked to each of them to find out the color of his own spot. After a while, the wisest correctly answered that his spot was white.*

(formalization adapted from: [Baldoni, PhD, 1998])

- a does not know his spot is white

$$\text{valid } \neg \Box_a (ws\ a)$$

- b does not know his spot is white

$$\text{valid } \neg \Box_b (ws\ b)$$

- to prove: c does know, that he has a white spot

$$\ldots \vdash^{HOL} \text{valid } \Box_c (ws\ c)$$

[Benzmüller, AMAI, 2011]

# Wise Men Puzzle

*Once upon a time, a king wanted to find the wisest out of his three wisest men. He arranged them in a circle and told them that he would put a white or a black spot on their foreheads and that one of the three spots would certainly be white. The three wise men could see and hear each other but, of course, they could not see their faces reflected anywhere. The king, then, asked to each of them to find out the color of his own spot. After a while, the wisest correctly answered that his spot was white.*

(formalization adapted from: [Baldoni, PhD, 1998])

- a does not know his spot is white

$$\text{valid } \neg \Box_a (ws\ a)$$

- b does not know his spot is white

$$\text{valid } \neg \Box_b (ws\ b)$$

- to prove: c does know, that he has a white spot

$$\ldots \vdash^{HOL} \text{valid } \Box_c (ws\ c)$$

LEO-II can do this effectively

# Wise Men Puzzle

Wise Men Puzzle

*Once upon a time, a king wanted to find the wisest out of his three wisest men. He arranged them in a circle and told them that he would put a white or a black spot on their foreheads and that one of the three spots would certainly be white. The three wise men could see and hear each other but, of course, they could not see their faces reflected anywhere. The king, then, asked to each of them to find out the color of his own spot. After a while, the wisest correctly answered that his spot was white.*

- temporal modality (time):
  $\Box_t$ : 'in the future it will be the case that'
- after waiting some time, two wise men still don't know the color of their spot

$$\text{valid } \Box_t \neg \Box_a (ws\ a)$$

$$\text{valid } \Box_t \Box_t \neg \Box_b (ws\ b)$$

- shortly later the third wise men then knows the color of his spot

$$\ldots \vdash^{HOL} \text{valid } \Box_t \Box_t \Box_t \Box_c (ws\ c)$$

[Benzmüller, AMAI, 2011]

# Further Examples  Wise Men Puzzle

<u>*Wise Men Puzzle*</u>

*Once upon a time, a king wanted to find the wisest out of his three wisest men. He arranged them in a circle and told them that he would put a white or a black spot on their foreheads and that one of the three spots would certainly be white. The three wise men could see and hear each other but, of course, they could not see their faces reflected anywhere. The king, then, asked to each of them to find out the color of his own spot. After a while, the wisest correctly answered that his spot was white.*

- wait a second: $\Box_t$ has not yet been characterized as temporal modality

  relation t is transitive: $\forall^p \phi.\ \Box_t \phi \Rightarrow \Box_t \Box_t \phi$

  relation t is irreflexive: $\boxed{???}$

[Benzmüller, AMAI, 2011]

<u>*Wise Men Puzzle*</u>

*Once upon a time, a king wanted to find the wisest out of his three wisest men. He arranged them in a circle and told them that he would put a white or a black spot on their foreheads and that one of the three spots would certainly be white. The three wise men could see and hear each other but, of course, they could not see their faces reflected anywhere. The king, then, asked to each of them to find out the color of his own spot. After a while, the wisest correctly answered that his spot was white.*

- wait a second: $\Box_t$ has not yet been characterized as temporal modality

relation t is transitive: $\forall^p \phi_{\blacksquare} \, \Box_t \, \phi \Rightarrow \Box_t \, \Box_t \, \phi$

relation t is irreflexive: (irreflexive t)

$$\text{irreflexive} = \lambda R_{\iota\rightarrow\iota\rightarrow o} . \forall W_{\iota} . \neg (R \, W \, W)$$

[Benzmüller, AMAI, 2011]

Wise Men Puzzle

*Once upon a time, a king wanted to find the wisest out of his three wisest men. He arranged them in a circle and told them that he would put a white or a black spot on their foreheads and that one of the three spots would certainly be white. The three wise men could see and hear each other but, of course, they could not see their faces reflected anywhere. The king, then, asked to each of them to find out the color of his own spot. After a while, the wisest correctly answered that his spot was white.*

- wait a second: $\Box_t$ has not yet been characterized as temporal modality

relation t is transitive: (transitive t)

relation t is irreflexive: (irreflexive t)

$$\text{irreflexive} = \lambda R_{\iota\to\iota\to o}.\forall W_\iota.\neg(R\,W\,W)$$

[Benzmüller, AMAI, 2011]

# Wise Men Puzzle

<u>*Wise Men Puzzle*</u>

*Once upon a time, a king wanted to find the wisest out of his three wisest men. He arranged them in a circle and told them that he would put a white or a black spot on their foreheads and that one of the three spots would certainly be white. The three wise men could see and hear each other but, of course, they could not see their faces reflected anywhere. The king, then, asked to each of them to find out the color of his own spot. After a while, the wisest correctly answered that his spot was white.*

- wait a second: $\Box_t$ has not yet been characterized as temporal modality

relation t is transitive: (transitive t)

relation t is irreflexive: (irreflexive t)

$$\text{irreflexive} = \lambda R_{\iota\to\iota\to o}.\forall W_\iota.\neg(R\,W\,W)$$

LEO-II can solve the modified puzzle effectively

[Benzmüller, AMAI 2011]

Region Connection Calculus (RCC) as fragment of HOL: [RandellCuiCohn, 1992]

$$
\begin{aligned}
\text{disconnected}: \quad & dc && = \lambda X_\tau. \lambda Y_\tau. \neg(c\ X\ Y) \\
\text{part of}: \quad & p && = \lambda X_\tau. \lambda Y_\tau. \forall Z. ((c\ Z\ X) \Rightarrow (c\ Z\ Y)) \\
\text{identical with}: \quad & eq && = \lambda X_\tau. \lambda Y_\tau. ((p\ X\ Y) \wedge (p\ Y\ X)) \\
\text{overlaps}: \quad & o && = \lambda X_\tau. \lambda Y_\tau. \exists Z. ((p\ Z\ X) \wedge (p\ Z\ Y)) \\
\text{partially o}: \quad & po && = \lambda X_\tau. \lambda Y_\tau. ((o\ X\ Y) \wedge \neg(p\ X\ Y) \wedge \neg(p\ Y\ X)) \\
\text{ext. connected}: \quad & ec && = \lambda X_\tau. \lambda Y_\tau. ((c\ X\ Y) \wedge \neg(o\ X\ Y)) \\
\text{proper part}: \quad & pp && = \lambda X_\tau. \lambda Y_\tau. ((p\ X\ Y) \wedge \neg(p\ Y\ X)) \\
\text{tangential pp}: \quad & tpp && = \lambda X_\tau. \lambda Y_\tau. ((pp\ X\ Y) \wedge \exists Z. ((ec\ Z\ X) \wedge (ec\ Z\ Y))) \\
\text{nontang. pp}: \quad & ntpp && = \lambda X_\tau. \lambda Y_\tau. ((pp\ X\ Y) \wedge \neg\exists Z. ((ec\ Z\ X) \wedge (ec\ Z\ Y)))
\end{aligned}
$$

A trivial problem for RCC:

| | |
|---|---|
| Catalunya is a border region of Spain | (*tpp* catalunya spain), |
| Spain and France share a border | (*ec* spain france), |
| Paris is a region inside France | (*ntpp* paris france) |
| | $\vdash^{\mathsf{HOL}}$ |
| Catalunya and Paris are disconnected | (*dc* catalunya paris) |
| | $\wedge$ |
| Spain and Paris are disconnected | (*dc* spain paris) |

[Benzmüller, AMAI, 2011]

A trivial problem for RCC:

| | |
|---|---|
| Catalunya is a border region of Spain | (*tpp* catalunya spain), |
| Spain and France share a border | (*ec* spain france), |
| Paris is a region inside France | (*ntpp* paris france) |
| | $\vdash^{\mathsf{HOL}}_{2.3s}$ |
| Catalunya and Paris are disconnected | (*dc* catalunya paris) |
| | $\wedge$ |
| Spain and Paris are disconnected | (*dc* spain paris) |

[Benzmüller, AMAI, 2011]

$$
\begin{array}{ll}
 & \text{valid } \forall \phi.\ \Box_{\text{fool}}\, \phi \supset \Box_{\text{bob}}\, \phi, \\
 & \text{valid } \Box_{\text{fool}}\, (\lambda W.\, (\mathit{ec} \text{ spain france})), \\
 & \text{valid } \Box_{\text{bob}}\, (\lambda W.\, (\mathit{tpp} \text{ catalunya spain})), \\
 & \text{valid } \Box_{\text{bob}}\, (\lambda W.\, (\mathit{ntpp} \text{ paris france})) \\
\vdash^{HOL} & \text{valid } \Box_{\text{bob}}\, (\lambda W.\, ((\mathit{dc} \text{ catalunya paris}) \wedge (\mathit{dc} \text{ spain paris})))
\end{array}
$$

$$
\begin{array}{ll}
 & \text{valid } \forall\phi\mathbf{.}\ \Box_{\text{fool}}\,\phi \supset \Box_{\text{bob}}\,\phi, \\
 & \text{valid } \Box_{\text{fool}}\,(\lambda W\mathbf{.}\,(\mathit{ec}\ \text{spain france})), \\
 & \text{valid } \Box_{\text{bob}}\,(\lambda W\mathbf{.}\,(\mathit{tpp}\ \text{catalunya spain})), \\
 & \text{valid } \Box_{\text{bob}}\,(\lambda W\mathbf{.}\,(\mathit{ntpp}\ \text{paris france})) \\
\vdash^{\mathit{HOL}}_{20.4s} & \text{valid } \Box_{\text{bob}}\,(\lambda W\mathbf{.}\,((\mathit{dc}\ \text{catalunya paris}) \wedge (\mathit{dc}\ \text{spain paris})))
\end{array}
$$

$$
\begin{array}{ll}
 & \text{valid } \forall \phi.\ \Box_{\text{fool}}\, \phi \supset \Box_{\text{bob}}\, \phi, \\
 & \text{valid } \Box_{\text{fool}}\, (\lambda W.(\textit{ec} \text{ spain france})), \\
 & \text{valid } \Box_{\text{bob}}\, (\lambda W.(\textit{tpp} \text{ catalunya spain})), \\
 & \text{valid } \Box_{\text{bob}}\, (\lambda W.(\textit{ntpp} \text{ paris france})) \\
\vdash^{\textit{HOL}}_{20.4s} & \text{valid } \Box_{\text{bob}}\, (\lambda W.((\textit{dc} \text{ catalunya paris}) \wedge (\textit{dc} \text{ spain paris}))) \\
\not\vdash^{\textit{HOL}} & \text{valid } \Box_{\text{fool}}\, (\lambda W.((\textit{dc} \text{ catalunya paris}) \wedge (\textit{dc} \text{ spain paris})))
\end{array}
$$

$$
\begin{array}{ll}
 & \text{valid } \forall\phi.\ \Box_{\text{fool}}\, \phi \supset \Box_{\text{bob}}\, \phi, \\
 & \text{valid } \Box_{\text{fool}}\, (\lambda W.\, (\mathit{ec}\ \text{spain france})), \\
 & \text{valid } \Box_{\text{bob}}\, (\lambda W.\, (\mathit{tpp}\ \text{catalunya spain})), \\
 & \text{valid } \Box_{\text{bob}}\, (\lambda W.\, (\mathit{ntpp}\ \text{paris france})) \\
\vdash^{HOL}_{20.4s} & \text{valid } \Box_{\text{bob}}\, (\lambda W.\, ((\mathit{dc}\ \text{catalunya paris}) \wedge (\mathit{dc}\ \text{spain paris}))) \\
\nvdash^{HOL}_{39.7s} & \text{valid } \Box_{\text{fool}}\, (\lambda W.\, ((\mathit{dc}\ \text{catalunya paris}) \wedge (\mathit{dc}\ \text{spain paris})))
\end{array}
$$

$$
\begin{array}{rl}
 & \text{valid } \forall\phi\text{.}\ \Box_{\text{fool}}\,\phi \supset \Box_{\text{bob}}\,\phi, \\
 & \text{valid } \Box_{\text{fool}}\,(\lambda W\text{.}(\mathit{ec}\text{ spain france})), \\
 & \text{valid } \Box_{\text{bob}}\,(\lambda W\text{.}(\mathit{tpp}\text{ catalunya spain})), \\
 & \text{valid } \Box_{\text{bob}}\,(\lambda W\text{.}(\mathit{ntpp}\text{ paris france})) \\
\vdash^{\mathit{HOL}}_{20.4s} & \text{valid } \Box_{\text{bob}}\,(\lambda W\text{.}((\mathit{dc}\text{ catalunya paris}) \wedge (\mathit{dc}\text{ spain paris}))) \\
\nvdash^{\mathit{HOL}}_{39.7s} & \text{valid } \Box_{\text{fool}}\,(\lambda W\text{.}((\mathit{dc}\text{ catalunya paris}) \wedge (\mathit{dc}\text{ spain paris})))
\end{array}
$$

Key idea is "Lifting" of RCC propositions to modal predicates:

$$
\underbrace{(\mathit{tpp}\text{ catalunya spain})}_{\text{type } o} \longrightarrow \underbrace{(\lambda W\text{.}(\mathit{tpp}\text{ catalunya spain}))}_{\text{type } \iota\to o}
$$

Automation of Meta-Properties of Logics in HOL

Correspondences between properties of accessibility relations like

$$\begin{aligned} \text{symmetric} &= \lambda R.\forall S, T.\, R\,S\,T \Rightarrow R\,T\,S \\ \text{serial} &= \lambda R.\forall S.\exists T.\, R\,S\,T \end{aligned}$$

and corresponding axioms

$$\begin{aligned} \forall R.\,\text{symmetric}\,R &\Leftarrow \\ &\Rightarrow \quad \text{valid}\ \forall^p \phi.\, \phi \supset \Box_R \Diamond_R \phi \qquad (B) \end{aligned}$$

$$\begin{aligned} \forall R.\,\text{serial}\,R &\Leftarrow \\ &\Rightarrow \quad \text{valid}\ \forall^p \phi.\, \Box_R \phi \supset \Diamond_R \phi \qquad (D) \end{aligned}$$

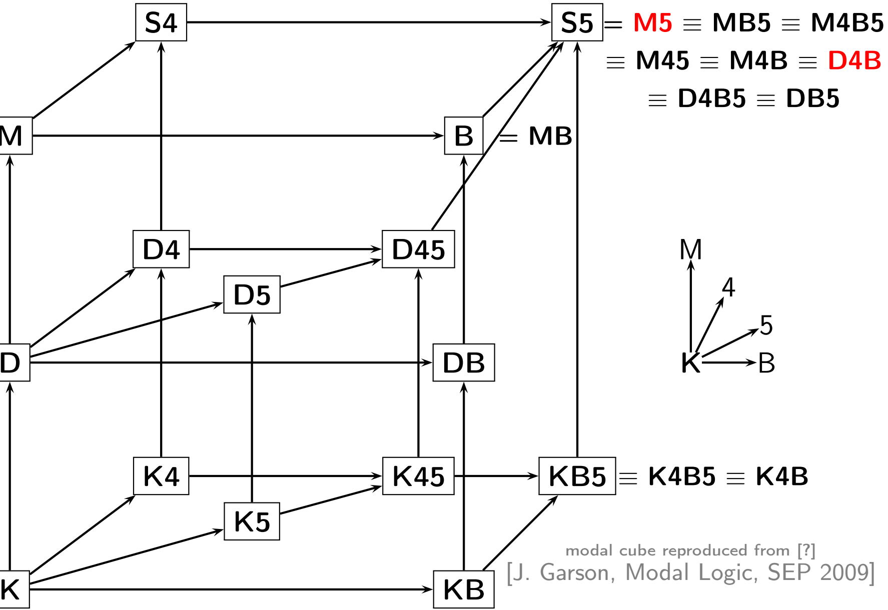
Correspondences between properties of accessibility relations like

$$\begin{array}{rcl} \text{symmetric} & = & \lambda R_{\bullet} \forall S, T_{\bullet} R\, S\, T \Rightarrow R\, T\, S \\ \text{serial} & = & \lambda R_{\bullet} \forall S_{\bullet} \exists T_{\bullet} R\, S\, T \end{array}$$

and corresponding axioms

$$\begin{array}{rcll} \forall R_{\bullet} \text{symmetric}\, R & \overset{0.0s}{\Leftarrow} & & \\ & \overset{0.0s}{\Rightarrow} & \text{valid}\ \forall^p \phi_{\bullet} \phi \supset \Box_R \Diamond_R \phi & (B) \\ \forall R_{\bullet} \text{serial}\, R & \overset{0.0s}{\Leftarrow} & & \\ & \overset{0.0s}{\Rightarrow} & \text{valid}\ \forall^p \phi_{\bullet} \Box_R \phi \supset \Diamond_R \phi & (D) \end{array}$$

Such proofs — including axioms D, M, 4, B, 5 — can be automated with LEO-II in no-time!




modal cube reproduced from [?]
[J. Garson, Modal Logic, SEP 2009]

$$
\forall R.\quad
\left.
\begin{array}{ll}
 & \text{valid}\ \forall^p \phi.\ \Box_R\, \phi \supset \phi \\
\wedge & \text{valid}\ \forall^p \phi.\ \Diamond_R\, \phi \supset \Box_R \Diamond_R\, \phi
\end{array}
\right\} M5
$$

$$
\Leftrightarrow
$$

$$
\left.
\begin{array}{ll}
 & \text{valid}\ \forall^p \phi.\ \Box_R\, \phi \supset \Diamond_R\, \phi \\
\wedge & \text{valid}\ \forall^p \phi.\ \Box_R\, \phi \supset \Box_R \Box_R\, \phi \\
\wedge & \text{valid}\ \forall^p \phi.\ \phi \supset \Box_R \Diamond_R\, \phi
\end{array}
\right\} D4B
$$

[Benzmüller, Festschrift Walther, 2010]

$$\forall R.\ \left.\begin{array}{ll} & \text{valid } \forall^p \phi.\ \Box_R\, \phi \supset \phi \\ \wedge & \text{valid } \forall^p \phi.\ \Diamond_R\, \phi \supset \Box_R \Diamond_R\, \phi \end{array}\right\} M5$$

$$\Leftrightarrow$$

$$\left.\begin{array}{ll} & \text{serial } R \\ \wedge & \text{valid } \forall^p \phi.\ \Box_R\, \phi \supset \Box_R \Box_R\, \phi \\ \wedge & \text{symmetric } R \end{array}\right\} D4B$$

$$
\forall R.\quad
\left.\begin{array}{rl}
 & \text{reflexive } R \\
\wedge & \text{euclidean } R
\end{array}\right\} M5
\quad \Leftrightarrow \quad
\left.\begin{array}{rl}
 & \text{serial } R \\
\wedge & \text{transitive } R \\
\wedge & \text{symmetric } R
\end{array}\right\} D4B
$$

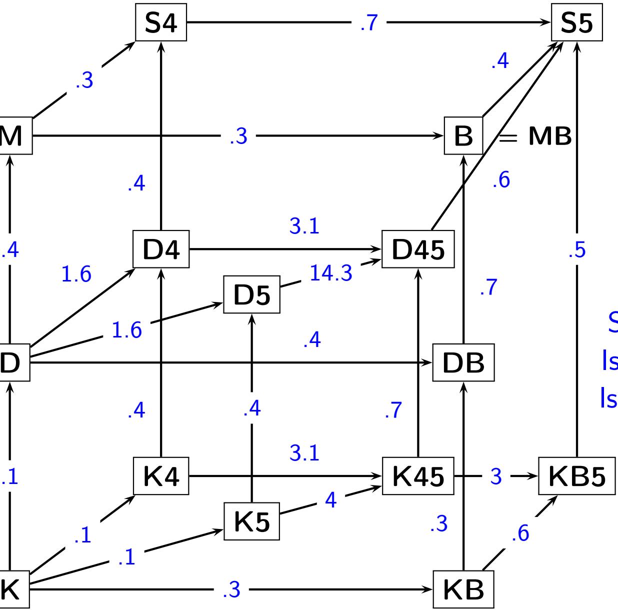
$$\forall R.\quad \left.\begin{array}{ll} & \text{reflexive } R \\ \wedge & \text{euclidean } R \end{array}\right\} M5 \quad \overset{0.1s}{\Leftrightarrow} \quad \left.\begin{array}{ll} & \text{serial } R \\ \wedge & \text{transitive } R \\ \wedge & \text{symmetric } R \end{array}\right\} D4B$$

Proof with LEO-II in 0.1s

[Benzmüller, Festschrift Walther, 2010]

$$
\forall R.\quad
\left.\begin{array}{ll}
 & \text{reflexive } R \\
\wedge & \text{euclidean } R
\end{array}\right\} M5
\quad \Leftrightarrow \quad
\left.\begin{array}{ll}
 & \text{serial } R \\
\wedge & \text{transitive } R \\
\wedge & \text{symmetric } R
\end{array}\right\} D4B
$$

[Benzmüller, Festschrift Walther, 2010]




$= \mathbf{M5} \overset{.1}{\equiv} \mathbf{MB5} \overset{.2}{\equiv} \mathbf{M4B5} \overset{.2}{\equiv} \mathbf{M45} \overset{.1}{\equiv} \mathbf{M4B} \overset{.1}{\equiv} \mathbf{D4B} \overset{.2}{\equiv} \mathbf{D4B5} \overset{.1}{\equiv} \mathbf{DB5}$

fastest results by:

LEO-II (prover)
Satallax (prover)
TPS (prover)
Satallax (mod.find.)
IsabelleN (mod.find.)
IsabelleM (mod.find.)

$\overset{.2}{\equiv} \mathbf{K4B5} \overset{.1}{\equiv} \mathbf{K4B}$

$\sum < 40\text{sec.}$

Automation of meta-level properties [Benzmüller, Festschrift Walther, 2010]

- Correspondences between axioms and semantic properties
  valid $\forall\phi.\ \Box_r \phi \supset \Box_r \Box_r \phi$ (axiom 4)
  $\Leftrightarrow$ (transitive $r$)
- Dependence/independence of axioms
  base modal logic K $\not\models$ axiom 4?
- Consistency of logics and logic combinations
  Is logic S4 (K+M+4) consistent?
- Inclusion/non-inclusion relations between logics
  Is logic K45 (K+M+5) included in logic S4 (K+M+4)? Why not?

Running experiments (thousands of problems): exploration of

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

[Segerberg, 1973]

- [Segerberg, 1973] discusses a 2-dimensional logic providing two S5 modalities $\Box_a$ and $\Box_b$.
- He adds further axioms stating that these modalities are commutative and orthogonal.
- It actually turns out that orthogonality is already implied in this context.

$$
\begin{array}{ll}
 & \text{reflexive } a, \text{transitive } a, \text{euclid. } a, \\
 & \text{reflexive } b, \text{transitive } b, \text{euclid. } b, \\
 & \text{valid } \forall \phi.\ \Box_a \Box_b \phi \Leftrightarrow \Box_b \Box_a \phi \\
\vdash^{HOL} & \\
 & \text{valid } \forall \phi, \psi.\ \Box_a (\Box_a \phi \vee \Box_b \psi) \supset (\Box_a \phi \vee \Box_a \psi) \\
 & \wedge \text{ valid } \forall \phi, \psi.\ \Box_b (\Box_a \phi \vee \Box_b \psi) \supset (\Box_b \phi \vee \Box_b \psi)
\end{array}
$$

[Benzmüller, AMAI, 2011]

[Segerberg, 1973]

- [Segerberg, 1973] discusses a 2-dimensional logic providing two S5 modalities $\Box_a$ and $\Box_b$.
- He adds further axioms stating that these modalities are commutative and orthogonal.
- It actually turns out that orthogonality is already implied in this context.

$$
\begin{array}{ll}
 & \text{reflexive } a, \text{transitive } a, \text{euclid. } a, \\
 & \text{reflexive } b, \text{transitive } b, \text{euclid. } b, \\
 & \text{valid } \forall\phi\text{.}\ \Box_a \Box_b \phi \Leftrightarrow \Box_b \Box_a \phi \\
\vdash^{HOL} & \\
 & \text{valid } \forall\phi,\psi\text{.}\ \Box_a (\Box_a \phi \vee \Box_b \psi) \supset (\Box_a \phi \vee \Box_a \psi) \\
 & \wedge\, \text{valid } \forall\phi,\psi\text{.}\ \Box_b (\Box_a \phi \vee \Box_b \psi) \supset (\Box_b \phi \vee \Box_b \psi)
\end{array}
$$

[Benzmüller, AMAI, 2011]

[Segerberg, 1973]

- [Segerberg, 1973] discusses a 2-dimensional logic providing two S5 modalities $\Box_a$ and $\Box_b$.
- He adds further axioms stating that these modalities are commutative and orthogonal.
- It actually turns out that orthogonality is already implied in this context.

$$
\begin{array}{ll}
 & \text{reflexive } a, \text{transitive } a, \text{euclid. } a, \\
 & \text{reflexive } b, \text{transitive } b, \text{euclid. } b, \\
 & \text{valid } \forall \phi.\ \Box_a \Box_b \phi \Leftrightarrow \Box_b \Box_a \phi \\
\vdash^{HOL} & \\
 & \text{valid } \forall \phi, \psi.\ \Box_a (\Box_a \phi \vee \Box_b \psi) \supset (\Box_a \phi \vee \Box_a \psi) \\
 & \wedge \text{ valid } \forall \phi, \psi.\ \Box_b (\Box_a \phi \vee \Box_b \psi) \supset (\Box_b \phi \vee \Box_b \psi)
\end{array}
$$

[Benzmüller, AMAI, 2011]

[Segerberg, 1973]

- [Segerberg, 1973] discusses a 2-dimensional logic providing two S5 modalities $\Box_a$ and $\Box_b$.
- He adds further axioms stating that these modalities are commutative and orthogonal.
- It actually turns out that orthogonality is already implied in this context.

$$\begin{array}{l} \text{reflexive } a, \text{transitive } a, \text{euclid. } a, \\ \text{reflexive } b, \text{transitive } b, \text{euclid. } b, \\ \text{valid } \forall \phi.\ \Box_a \Box_b \phi \Leftrightarrow \Box_b \Box_a \phi \end{array}$$

$\vdash^{HOL}$ proof by LEO-II in 0.2s

$$\begin{array}{l} \text{valid } \forall \phi, \psi.\ \Box_a (\Box_a \phi \vee \Box_b \psi) \supset (\Box_a \phi \vee \Box_a \psi) \\ \wedge \text{valid } \forall \phi, \psi.\ \Box_b (\Box_a \phi \vee \Box_b \psi) \supset (\Box_b \phi \vee \Box_b \psi) \end{array}$$

[Benzmüller, AMAI, 2011]

## Many experiments can be foudn in my recent papers

Essential: TPTP infrastructure for HOL (tptp.org) (with G. Sutcliffe)

- project result of: EU FP7 IIF grant THFTPTP
- standardized THF syntax for HOL (& more)
- problem library
- prover competition
- online access $\geq$ 6 ATPs/model finders
- tools for proof verification, ...

`www.tptp.org` $\longrightarrow$ PUZ087^1/2.p (Wise Men Puzzle)

[SutcliffeBenzmüller, Journal of Formalized Reasoning, 2010]

# Conclusion

Core Questions:

1. Classical Higher Order Logic (HOL) as Universal Logic?
2. HOL Provers & Model Finders as Generic Reasoning Tools?
3. Combinations with Specialist Reasoners (if available)?

- (1)&(2) are interesting and relevant: Evidence given in talk!?
- (3) not further discussed: – ongoing and future work

  My vision is an automated (& interactive) generic logic engine with HOL theorem provers and model finders as backbone, and with integrated, more effective specialist reasoners (if available) as collaborating agents.

- Don't forget: There are many reasons for the automation of HOL!