Working with Automated Reasoning Tools
– LEO-II –

Christoph Benzmüller and Geoff Sutcliffe

SS08, Block Course at Saarland University, Germany

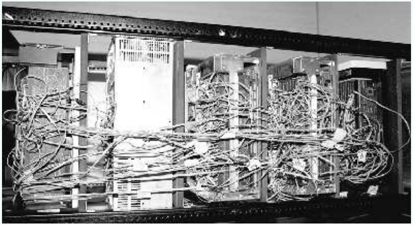
Overview

1. Higher-Order Logic (HOL)
   - The Good Thing: Expressivitity
   - The Bad Thing: Automation is a Challenge

2. The LEO-II Prover
   - Motivation and Architecture
   - Solving Lightweight Problems
   - Solving Less Lightweight Problems: Multimodal Logics
   - Ongoing and Future Work

Higher-Order Logic (HOL)

Some people say that HOL is like this:



I don't!

- Semantics (extensionality) [PhD-99,JSL-04]
- Proof theory [IJCAR-06]
- ATPs LEO and LEO-II [CADE-98,IJCAR-08]

Higher-Order Logic (HOL)
- on one slide -

Property	FOL	HOL	Example
Quantification over			
- individuals	✓	✓	$\forall x.\, P(F(x))$
- functions	–	✓	$\forall F.\, P(F(x))$
- predicates/sets/relations	–	✓	$\forall P.\, P(F(x))$
Unnamed			
- functions	–	✓	$(\lambda x.\, x)$
- predicates/sets/relations	–	✓	$(\lambda x.\, x \neq 2)$
Statements about			
- functions	–	✓	$\mathit{continuous}(\lambda x.\, x)$
- predicates/sets/relations	–	✓	$\mathit{reflexive}(=)$

Sets and Relations in HOL

$$A \cup B \;:=\; \{x \mid x \in A \vee x \in B\}$$

$$A \cup B \;:=\; (\lambda x_{\centerdot} x \in A \vee x \in B)$$

$$\cup \;:=\; \lambda A_{\centerdot} \lambda B_{\centerdot} (\lambda x_{\centerdot} x \in A \vee x \in B)$$

$$\mathtt{symmetric} \;:=\; \lambda F_{\centerdot} (\forall x, y_{\centerdot} F(x,y) = F(y,x))$$

$$\textit{Theorem}: \quad \mathtt{symmetric}(\cup)$$

Sets and Relations in HOL

$$A \cup B := \{x |\ x \in A \vee x \in B\}$$

$$A \cup B := (\lambda x.\, x \in A \vee x \in B)$$

$$\cup := \lambda A.\, \lambda B.\, (\lambda x.\, x \in A \vee x \in B)$$

$$\texttt{symmetric} := \lambda F.\, (\forall x, y.\, F(x, y) = F(y, x))$$

$$\mathit{Theorem}: \quad \texttt{symmetric}(\cup)$$

Sets and Relations in HOL

$$A \cup B := \{x \mid x \in A \vee x \in B\}$$

$$A \cup B := (\lambda x.\, x \in A \vee x \in B)$$

$$\cup := \lambda A.\, \lambda B.\, (\lambda x.\, x \in A \vee x \in B)$$

$$\mathtt{symmetric} := \lambda F.\, (\forall x, y.\, F(x, y) = F(y, x))$$

$$\mathit{Theorem}: \quad \mathtt{symmetric}(\cup)$$

Sets and Relations in HOL

$$A \cup B := \{x \mid x \in A \vee x \in B\}$$

$$A \cup B := (\lambda x_{\blacksquare} x \in A \vee x \in B)$$

$$\cup := \lambda A_{\blacksquare} \lambda B_{\blacksquare} (\lambda x_{\blacksquare} x \in A \vee x \in B)$$

$$\texttt{symmetric} := \lambda F_{\blacksquare} (\forall x, y_{\blacksquare} F(x, y) = F(y, x))$$

$$\textit{Theorem}: \quad \texttt{symmetric}(\cup)$$

Sets and Relations in HOL

$$A \cup B := \{x |\ x \in A \vee x \in B\}$$

$$A \cup B := (\lambda x. x \in A \vee x \in B)$$

$$\cup := \lambda A. \lambda B. (\lambda x. x \in A \vee x \in B)$$

$$\mathtt{symmetric} := \lambda F. (\forall x, y. F(x, y) = F(y, x))$$

$$\textit{Theorem} : \quad \mathtt{symmetric}(\cup)$$

Sets and Relations in HOL

$$
\begin{array}{lcl}
\in & := & \lambda x.\lambda A.A(x) \\
\emptyset & := & \lambda x.\bot \\
\cap & := & \lambda A.\lambda B.(\lambda x.x \in A \wedge x \in B) \\
\cup & := & \lambda A.\lambda B.(\lambda x.x \in A \vee x \in B) \\
\setminus & := & \lambda A.\lambda B.(\lambda x.x \in A \wedge x \notin B) \\
\ldots & & \\
\subseteq & := & \lambda A.\lambda B.(\forall x.x \in A \Rightarrow x \in B) \\
\mathcal{P} & := & \lambda A.(\lambda B.B \subseteq A) \\
\ldots & & \\
\texttt{reflexive} & := & \lambda R.(\forall x.R(x,x)) \\
\texttt{transitive} & := & \lambda R.(\forall x,y,z.(R(x,y) \wedge R(y,z)) \Rightarrow R(x,z)) \\
\ldots & &
\end{array}
$$

Types are Needed

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

- Another source of indeterminism / blind guessing

[TPHOLs-WP-07]

Automation of HOL: A Nightmare?

Undecidable and Infinitary Unification

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

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

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

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 –
- Axiom of description –

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

[IJCAR-06]: Axioms that imply Cut   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 –
- Axiom of description –

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

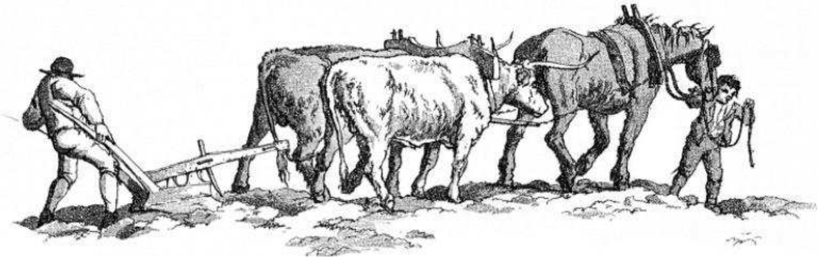
- Axiom of excluded middle ✓
- Comprehension axioms ✓
- Functional and Boolean extensionality ✓[CADE-98,PhD-99]
- Leibniz and other definitions of equality ✓[CADE-99,PhD-99]
- Axiom of induction ?
- Axiom of choice –
- Axiom of description –

LEO-II

An Effective Higher-Order Theorem Prover

UNIVERSITY OF CAMBRIDGE

UNIVERSITÄT DES SAARLANDES

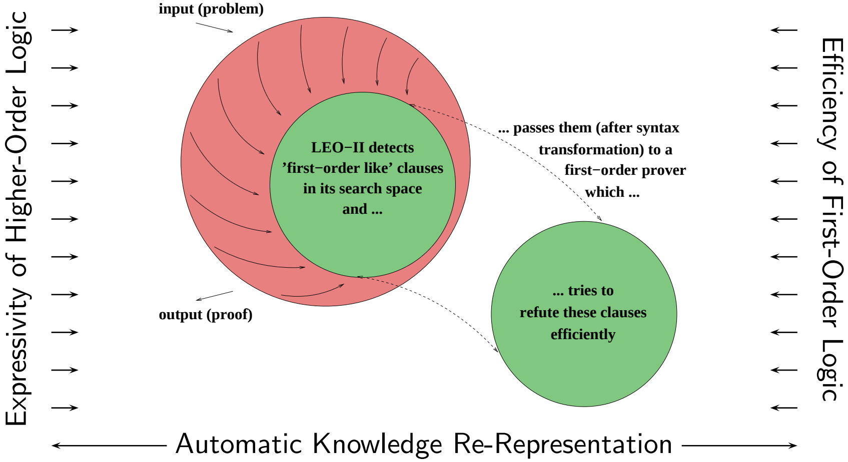


LEO-II employs FO-ATPs: E, Spass, Vampire

Motivation for LEO-II

- TPS system of Peter Andrews et al.
- LEO hardwired to ΩMEGA (predecessor of LEO-II)
- Agent-based architecture Ω-ANTS
  (with V. Sorge) [AIMSA-98,EPIA-99,Calculemus-00]
- Collaboration of LEO with FO-ATP via Ω-ANTS
  (with V. Sorge) [KI-01,LPAR-05,JAL-07]
- Progress in Higher-Order Termindexing
  (with F. Theiss and A. Fietzke) [IWIL-06]

Architecture of LEO-II



Solving Lightweight Problems

Example: TPTP Problem SET171+3

Axiomatization in FO Set Theory

Assumptions:

$\forall B, C, x.(x \in (B \cup C) \Leftrightarrow x \in B \vee x \in C)$

$\forall B, C, x.(x \in (B \cap C) \Leftrightarrow x \in B \wedge x \in C)$

$\forall B, C.(B \subseteq C \Leftrightarrow \forall x.x \in B \Rightarrow x \in C)$

$\forall B, C.(B \cup C = C \cup B)$

$\forall B, C.(B \cap C = C \cap B)$

$\forall B, C.(B = C \Leftrightarrow B \subseteq C \wedge C \subseteq B)$

$\forall B, C.(B = C \Leftrightarrow \forall x.x \in B \Leftrightarrow x \in C)$

Proof Goal:

$$\forall B, C, D.\ B \cup (C \cap D) = (B \cup C) \cap (B \cup D)$$

Performance: FO-ATPs

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

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

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

Performance: LEO-II + E

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

Example: TPTP Problem SET171+3

Axiomatization in FO Set Theory

Assumptions:

$$\forall B, C, x. (x \in (B \cup C) \Leftrightarrow x \in B \lor x \in C)$$

$$\forall B, C, x. (x \in (B \cap C) \Leftrightarrow x \in B \land x \in C)$$

$$\forall B, C. (B \subseteq C \Leftrightarrow \forall x. x \in B \Rightarrow x \in C)$$

$$\forall B, C. (B \cup C = C \cup B)$$

$$\forall B, C. (B \cap C = C \cap B)$$

$$\forall B, C. (B = C \Leftrightarrow B \subseteq C \land C \subseteq B)$$

$$\forall B, C. (B = C \Leftrightarrow \forall x. x \in B \Leftrightarrow x \in C)$$

Proof Goal:

$$\forall B, C, D. \quad B \cup (C \cap D) = (B \cup C) \cap (B \cup D)$$

Performance: FO-ATPs

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

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

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

Performance: LEO-II + E

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

Example: TPTP Problem SET171+3

Axiomatization in FO Set Theory

Assumptions:

$\forall B, C, x. (x \in (B \cup C) \Leftrightarrow x \in B \vee x \in C)$

$\forall B, C, x. (x \in (B \cap C) \Leftrightarrow x \in B \wedge x \in C)$

$\forall B, C. (B \subseteq C \Leftrightarrow \forall x. x \in B \Rightarrow x \in C)$

$\forall B, C. (B \cup C = C \cup B)$

$\forall B, C. (B \cap C = C \cap B)$

$\forall B, C. (B = C \Leftrightarrow B \subseteq C \wedge C \subseteq B)$

$\forall B, C. (B = C \Leftrightarrow \forall x. x \in B \Leftrightarrow x \in C)$

Proof Goal:

$\forall B, C, D.$

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

Performance: FO-ATPs

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

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

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

Performance: LEO-II + E

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

Example: TPTP Problem SET171+3

Axiomatization in FO Set Theory

Assumptions:

$\forall B, C, x. (x \in (B \cup C) \Leftrightarrow x \in B \vee x \in C)$

$\forall B, C, x. (x \in (B \cap C) \Leftrightarrow x \in B \wedge x \in C)$

$\forall B, C. (B \subseteq C \Leftrightarrow \forall x. x \in B \Rightarrow x \in C)$

$\forall B, C. (B \cup C = C \cup B)$

$\forall B, C. (B \cap C = C \cap B)$

$\forall B, C. (B = C \Leftrightarrow B \subseteq C \wedge C \subseteq B)$

$\forall B, C. (B = C \Leftrightarrow \forall x. x \in B \Leftrightarrow x \in C)$

Proof Goal:

$\forall B, C, D.$

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

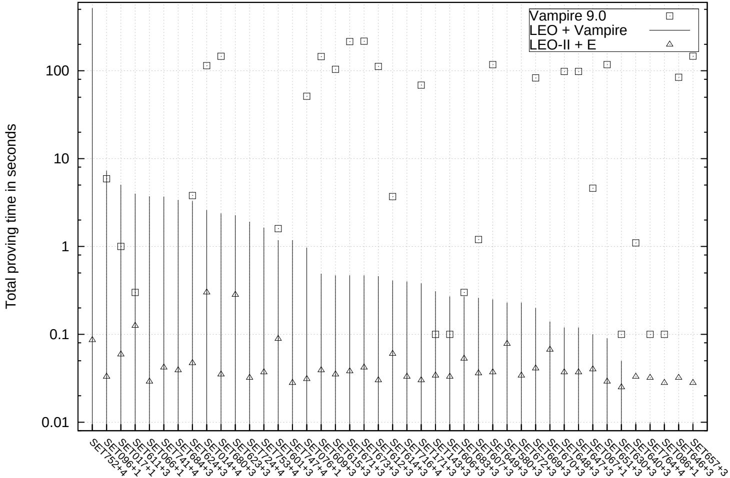
Performance: FO-ATPs

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

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

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

Performance: LEO-II + E

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

Results



Vampire 9.0
LEO + Vampire
LEO-II + E
Total proving time in seconds
100
10
1
0.1
0.01
SET752+4
SET096+1
SET017+1
SET611+3
SET066+1
SET741+4
SET684+3
SET624+3
SET014+4
SET680+3
SET623+3
SET724+4
SET753+4
SET601+3
SET747+4
SET076+1
SET609+3
SET615+3
SET671+3
SET673+3
SET612+3
SET614+3
SET716+4
SET171+3
SET143+3
SET606+3
SET683+3
SET607+3
SET649+3
SET580+3
SET672+3
SET669+3
SET670+3
SET648+3
SET647+3
SET067+1
SET651+3
SET630+3
SET640+3
SET764+4
SET086+1
SET646+3
SET657+3

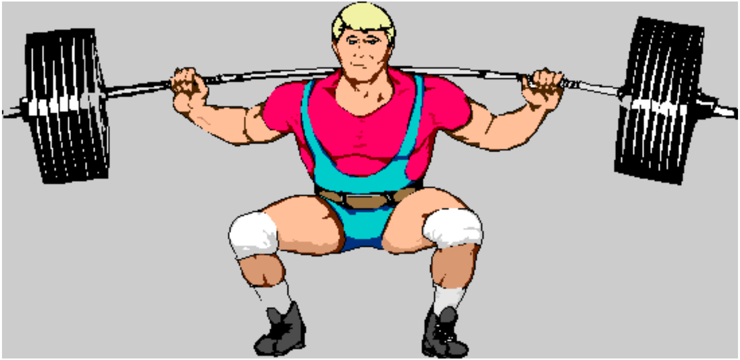
Results

Problem	Vamp. 9.0	LEO+Vamp.	LEO-II+E
014+4	114.5	2.60	0.300
017+1	1.0	5.05	0.059
066+1	–	3.73	0.029
067+1	4.6	0.10	0.040
076+1	51.3	0.97	0.031
086+1	0.1	0.01	0.028
096+1	5.9	7.29	0.033
143+3	0.1	0.31	0.034
171+3	68.6	0.38	0.030
580+3	0.0	0.23	0.078
601+3	1.6	1.18	0.089
606+3	0.1	0.27	0.033
607+3	1.2	0.26	0.036
609+3	145.2	0.49	0.039
611+3	0.3	4.00	0.125
612+3	111.9	0.46	0.030
614+3	3.7	0.41	0.060
615+3	103.9	0.47	0.035
623+3	–	2.27	0.282
624+3	3.8	3.29	0.047
630+3	0.1	0.05	0.025
640+3	1.1	0.01	0.033
646+3	84.4	0.01	0.032
647+3	98.2	0.12	0.037

Problem	Vamp. 9.0	LEO+Vamp.	LEO-II+E
648+3	98.2	0.12	0.037
649+3	117.5	0.25	0.037
651+3	117.5	0.09	0.029
657+3	146.6	0.01	0.028
669+3	83.1	0.20	0.041
670+3	–	0.14	0.067
671+3	214.9	0.47	0.038
672+3	–	0.23	0.034
673+3	217.1	0.47	0.042
680+3	146.3	2.38	0.035
683+3	0.3	0.27	0.053
684+3	–	3.39	0.039
716+4	–	0.40	0.033
724+4	–	1.91	0.032
741+4	–	3.70	0.042
747+4	–	1.18	0.028
752+4	–	516.00	0.086
753+4	–	1.64	0.037
764+4	0.1	0.01	0.032

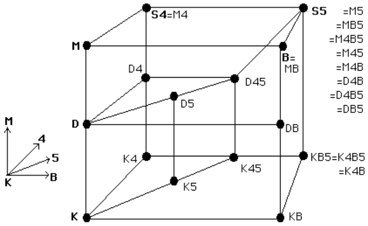
Vamp. 9.0: 2.80GHz, 1GB memory, 600s time limit
LEO+Vamp.: 2.40GHz, 4GB memory, 120s time limit
LEO-II+E: 1.60GHz, 1GB memory, 60s time limit

Solving Less Lightweight Problems



Logic Systems Interrelationships

Modal Logics Challenge



John Halleck (U Utah):
`http://www.cc.utah.edu/~nahaj/`

$100 Modal Logic Challenge:
`www.tptp.org`

Example

$$
\begin{array}{rcl}
S4 & = & K \\
 & + & M(T): \quad \Box_R A \Rightarrow A \\
 & + & 4: \quad \Box_R A \Rightarrow \Box_R \Box_R A
\end{array}
$$

Theorems:

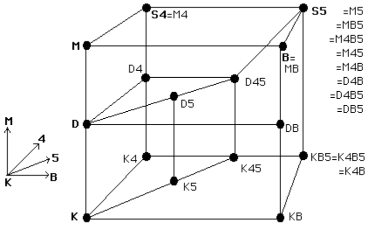
$$
\begin{array}{rclr}
S4 & \not\subseteq & K & (1) \\
(M \wedge 4) & \Leftrightarrow & (\mathit{refl}.(R) \wedge \mathit{trans}.(R)) & (2)
\end{array}
$$

Experiments

	FO-ATPs **[SutcliffeEtal-07]**	LEO-II + E **[BePa-08]**
(1)	16min + 2710s	17.3s
(2)	???	2.4s

Logic Systems Interrelationships

Modal Logics Challenge



John Halleck (U Utah):
`http://www.cc.utah.edu/~nahaj/`

$100 Modal Logic Challenge:
`www.tptp.org`

Example

$$
\begin{array}{rcl}
S4 & = & K \\
 & + & M(T):\quad \Box_R A \Rightarrow A \\
 & + & 4:\quad \Box_R A \Rightarrow \Box_R \Box_R A
\end{array}
$$

Theorems:

$$S4 \not\subseteq K \qquad (1)$$

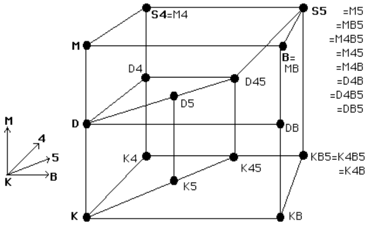
$$(M \wedge 4) \Leftrightarrow (\mathit{refl.}(R) \wedge \mathit{trans.}(R)) \qquad (2)$$

Experiments

	FO-ATPs [SutcliffeEtal-07]	LEO-II + E [BePa-08]
(1)	16min + 2710s	17.3s
(2)	???	2.4s

Logic Systems Interrelationships

Modal Logics Challenge



John Halleck (U Utah):
`http://www.cc.utah.edu/~nahaj/`

\$100 Modal Logic Challenge:
`www.tptp.org`

Example

$$
\begin{aligned}
S4 &= K \\
&+ \; M(T): \quad \Box_R A \Rightarrow A \\
&+ \; 4: \quad \Box_R A \Rightarrow \Box_R \Box_R A
\end{aligned}
$$

Theorems:

$$
\begin{array}{cccr}
S4 & \not\subseteq & K & (1) \\
(M \wedge 4) & \Leftrightarrow & (refl.(R) \wedge trans.(R)) & (2)
\end{array}
$$

Experiments

	FO-ATPs **[SutcliffeEtal-07]**	LEO-II + E **[BePa-08]**
(1)	16min + 2710s	17.3s
(2)	???	2.4s

Even simpler: Reasoning within Multimodal Logics

Problem	LEO-II + E
$\texttt{valid}(\Box_r \top)$	0.025s
$\texttt{valid}(\Box_r a \Rightarrow \Box_r a)$	0.026s
$\texttt{valid}(\Box_r a \Rightarrow \Box_s a)$	–
$\texttt{valid}(\Box_s (\Box_r a \Rightarrow \Box_r a))$	0.026s
$\texttt{valid}(\Box_r (a \land b) \Leftrightarrow (\Box_r a \land \Box_r b))$	0.044s
$\texttt{valid}(\Diamond_r (a \Rightarrow b) \Rightarrow \Box_r a \Rightarrow \Diamond_r b)$	0.030s
$\texttt{valid}(\neg \Diamond_r a \Rightarrow \Box_r (a \Rightarrow b))$	0.029s
$\texttt{valid}(\Box_r b \Rightarrow \Box_r (a \Rightarrow b))$	0.026s
$\texttt{valid}((\Diamond_r a \Rightarrow \Box_r b) \Rightarrow \Box_r (a \Rightarrow b))$	0.027s
$\texttt{valid}((\Diamond_r a \Rightarrow \Box_r b) \Rightarrow (\Box_r a \Rightarrow \Box_r b))$	0.029s
$\texttt{valid}((\Diamond_r a \Rightarrow \Box_r b) \Rightarrow (\Diamond_r a \Rightarrow \Diamond_r b))$	0.030s

(Normal) Multimodal Logic in HOL

Simple, Straightforward Encoding of Multimodal Logic

- base type ι: set of possible worlds
  certain terms of type $\iota \rightarrow o$: multimodal logic formulas
- multimodal logic operators:

$$
\begin{aligned}
\neg_{(\iota\rightarrow o)\rightarrow(\iota\rightarrow o)} &= \lambda A_{\iota\rightarrow o} . (\lambda x_\iota . \neg A(x)) \\
\vee_{(\iota\rightarrow o)\rightarrow(\iota\rightarrow o)\rightarrow(\iota\rightarrow o)} &= \lambda A_{\iota\rightarrow o}, B_{\iota\rightarrow o} . (\lambda x_\iota . A(x) \vee B(x)) \\
\Box_{R\,(\iota\rightarrow\iota\rightarrow o)\rightarrow(\iota\rightarrow o)\rightarrow(\iota\rightarrow o)} &= \lambda R_{\iota\rightarrow\iota\rightarrow o}, A_{\iota\rightarrow o} . \\
&\qquad (\lambda x_\iota . \forall y_\iota . R(x,y) \Rightarrow A(y))
\end{aligned}
$$

Related Work

[Gallin-73], [Carpenter-98], [Merz-99],
[Brown-05], [Hardt&Smolka-07], [Kaminski&Smolka-07]

(Normal) Multimodal Logic in HOL

Encoding of Validity

$$\texttt{valid} \quad := \quad \lambda A_{\iota\to o}.\,(\forall w_\iota.\,A(w))$$

Example Proof: $\textbf{valid}(\Box_s (\Box_r a \Rightarrow \Box_r a))$

Initialisation of problem

$$\neg\textbf{valid}(\Box_s (\Box_r a \Rightarrow \Box_r a))$$

Definition expansion

$$\neg(\forall x_\iota . \forall y_\iota . \neg s(x, y) \vee ((\neg(\forall u_\iota . \neg r(y, u) \vee a(u))) \vee (\forall v_\iota . \neg r(y, v) \vee a(v)))$$

Normalisation (x, y, u are now Skolem constants, V is a variable)

$$\begin{array}{ll} s(x, y) & \neg a(u) \\ r(y, u) & a(V) \vee \neg r(y, V) \end{array}$$

Translation to first-order logic

$$\begin{array}{ll} @_{(io)\_i}(@_{(i(io))\_i}(s, x), y) & \neg @_{(\iota o)\_\iota}(a, u) \\ @_{(io)\_i}(@_{(i(io))\_i}(r, y), u) & @_{(\iota o)\_\iota}(a, V) \vee \neg @_{(io)\_i}(@_{(i(io))\_i}(r, y), V) \end{array}$$

Example Proof: $\textbf{valid}(\Box_s (\Box_r a \Rightarrow \Box_r a))$

Initialisation of problem

$$\neg\textbf{valid}(\Box_s (\Box_r a \Rightarrow \Box_r a))$$

Definition expansion

$$\neg(\forall x_\iota \centerdot \forall y_\iota \centerdot \neg s(x, y) \lor ((\neg(\forall u_\iota \centerdot \neg r(y, u) \lor a(u))) \lor (\forall v_\iota \centerdot \neg r(y, v) \lor a(v)))$$

Normalisation (x, y, u are now Skolem constants, V is a variable)

$$\begin{array}{ll} s(x, y) & \neg a(u) \\ r(y, u) & a(V) \lor \neg r(y, V) \end{array}$$

Translation to first-order logic

$$\begin{array}{ll} @_{(io)\_i}(@_{(i(io))\_i}(s, x), y) & \neg @_{(\iota o)\_\iota}(a, u) \\ @_{(io)\_i}(@_{(i(io))\_i}(r, y), u) & @_{(\iota o)\_\iota}(a, V) \lor \neg @_{(io)\_i}(@_{(i(io))\_i}(r, y), V) \end{array}$$

Example Proof: valid($\Box_s$ ($\Box_r$ $a \Rightarrow \Box_r$ a))

Initialisation of problem

$$\neg\textbf{valid}(\Box_s\,(\Box_r\, a \Rightarrow \Box_r\, a))$$

Definition expansion

$$\neg(\forall x_\iota\blacksquare \forall y_\iota\blacksquare \neg s(x,y) \vee ((\neg(\forall u_\iota\blacksquare \neg r(y,u) \vee a(u))) \vee (\forall v_\iota\blacksquare \neg r(y,v) \vee a(v)))$$

Normalisation (x, y, u are now Skolem constants, V is a variable)

$$\begin{array}{ll} s(x,y) & \neg a(u) \\ r(y,u) & a(V) \vee \neg r(y,V) \end{array}$$

Translation to first-order logic

$$\begin{array}{ll} @_{(io)\_i}(@_{(i(io))\_i}(s,x),y) & \neg @_{(\iota o)\_\iota}(a,u) \\ @_{(io)\_i}(@_{(i(io))\_i}(r,y),u) & @_{(\iota o)\_\iota}(a,V) \vee \neg @_{(io)\_i}(@_{(i(io))\_i}(r,y),V) \end{array}$$

Example Proof: $\mathbf{valid}(\Box_s\,(\Box_r\, a \Rightarrow \Box_r\, a))$

Initialisation of problem

$$\neg\mathbf{valid}(\Box_s\,(\Box_r\, a \Rightarrow \Box_r\, a))$$

Definition expansion

$$\neg(\forall x_\iota\blacksquare\forall y_\iota\blacksquare\neg s(x,y) \vee ((\neg(\forall u_\iota\blacksquare\neg r(y,u) \vee a(u))) \vee (\forall v_\iota\blacksquare\neg r(y,v) \vee a(v)))$$

Normalisation (x, y, u **are now Skolem constants,** V **is a variable)**

$$\begin{array}{ll} s(x,y) & \neg a(u) \\ r(y,u) & a(V) \vee \neg r(y,V) \end{array}$$

Translation to first-order logic

$$\begin{array}{ll} @_{(io)\_i}(@_{(i(io))\_i}(s,x),y) & \neg @_{(\iota o)\_\iota}(a,u) \\ @_{(io)\_i}(@_{(i(io))\_i}(r,y),u) & @_{(\iota o)\_\iota}(a,V) \vee \neg @_{(io)\_i}(@_{(i(io))\_i}(r,y),V) \end{array}$$

A simple equation between modal logic formulas

$$\forall R_{\blacksquare}\forall A_{\blacksquare}\forall B_{\blacksquare}(\Box_R (A \vee B)) = (\Box_R (B \vee A))$$

- initialisation, definition expansion and normalisation:

$$(\lambda X_\iota.\forall Y_{\iota\blacksquare}\neg((r\, X)\, Y) \vee (a\, Y) \vee (b\, Y))$$
$$\neq$$
$$(\lambda X_\iota.\forall Y_{\iota\blacksquare}\neg((r\, X)\, Y) \vee (b\, Y) \vee (a\, Y))$$

More Examples ...

A simple equation between modal logic formulas

$$\forall R_{\blacksquare} \forall A_{\blacksquare} \forall B_{\blacksquare} (\Box_R (A \vee B)) = (\Box_R (B \vee A))$$

- functional and Boolean extensionality:

$$\neg((\forall Y_{\iota} \blacksquare \neg((r\,w)\,Y) \vee (a\,Y) \vee (b\,Y))$$
$$\Leftrightarrow$$
$$(\forall Y_{\iota} \blacksquare \neg((r\,w)\,Y) \vee (b\,Y) \vee (a\,Y)))$$

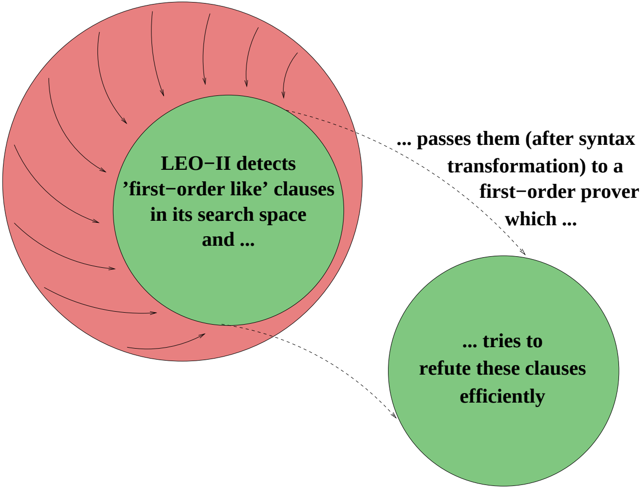
More Examples . . .

A simple equation between modal logic formulas

$$\forall R. \forall A. \forall B. (\Box_R (A \vee B)) = (\Box_R (B \vee A))$$

- normalisation:
  $40: (b\,V) \vee (a\,V) \vee \neg((r\,w)\,V) \vee \neg((r\,w)\,W) \vee (b\,W) \vee (a\,W)$
  $41: ((r\,w)\,z) \vee ((r\,w)\,v)$
  $42: \neg(a\,z) \vee ((r\,w)\,v)$
  $43: \neg(b\,z) \vee ((r\,w)\,v)$
  $44: ((r\,w)\,z) \vee \neg(a\,v)$
  $45: \neg(a\,z) \vee \neg(a\,v)$
  $46: \neg(b\,z) \vee \neg(a\,v)$
  $47: ((r\,w)\,z) \vee \neg(b\,v)$
  $48: \neg(a\,z) \vee \neg(b\,v)$
  $49: \neg(b\,z) \vee \neg(b\,v)$
- total proving time (notebook with 1.60GHz, 1GB): 0.071s

Architecture of LEO-II



More Examples ...

A simple equation between modal logic formulas

$$\forall R. \forall A. \forall B. (\Box_R (A \vee B)) \doteq (\Box_R (B \vee A))$$

where $\doteq$ is defined as $\lambda X, Y. \forall P. (P\, X) \Rightarrow (P\, Y)$

- initialisation, definition expansion and normalisation:

$$(p\,(\lambda X_\iota . \forall Y_\iota . \neg((r\,X)\,Y) \vee (a\,Y) \vee (b\,Y)))$$
$$\neg(p\,(\lambda X_\iota . \forall Y_\iota . \neg((r\,X)\,Y) \vee (b\,Y) \vee (a\,Y)))$$

More Examples ...

A simple equation between modal logic formulas

$$\forall R_{\blacksquare} \forall A_{\blacksquare} \forall B_{\blacksquare} (\Box_R (A \vee B)) \doteq (\Box_R (B \vee A))$$

where $\doteq$ is defined as $\lambda X, Y. \forall P_{\blacksquare} (P\,X) \Rightarrow (P\,Y)$

- resolution:

$$(p\,(\lambda X_\iota . \forall Y_{\iota\blacksquare} \neg((r\,X)\,Y) \vee (a\,Y) \vee (b\,Y)))$$
$$\neq$$
$$(p\,(\lambda X_\iota . \forall Y_{\iota\blacksquare} \neg((r\,X)\,Y) \vee (b\,Y) \vee (a\,Y)))$$

More Examples . . .

A simple equation between modal logic formulas

$$\forall R_\blacksquare \forall A_\blacksquare \forall B_\blacksquare (\Box_R (A \vee B)) \doteq (\Box_R (B \vee A))$$

where $\doteq$ is defined as $\lambda X, Y.\forall P_\blacksquare (P\,X) \Rightarrow (P\,Y)$

- decomposition:

$$(\lambda X_\iota.\forall Y_{\iota\blacksquare} \neg((r\,X)\,Y) \vee (a\,Y) \vee (b\,Y))$$
$$\neq$$
$$(\lambda X_\iota.\forall Y_{\iota\blacksquare} \neg((r\,X)\,Y) \vee (b\,Y) \vee (a\,Y))$$

More Examples . . .

A simple equation between modal logic formulas

$$\forall R_\blacksquare \forall A_\blacksquare \forall B_\blacksquare (\Box_R (A \vee B)) \doteq (\Box_R (B \vee A))$$

where $\doteq$ is defined as $\lambda X, Y.\forall P_\blacksquare (P\,X) \Rightarrow (P\,Y)$

- functional and Boolean extensionality:

$$\neg((\forall Y_{\iota\blacksquare} \neg((r\,w)\,Y) \vee (a\,Y) \vee (b\,Y)) \\ \Leftrightarrow \\ (\forall Y_{\iota\blacksquare} \neg((r\,w)\,Y) \vee (b\,Y) \vee (a\,Y)))$$

More Examples ...

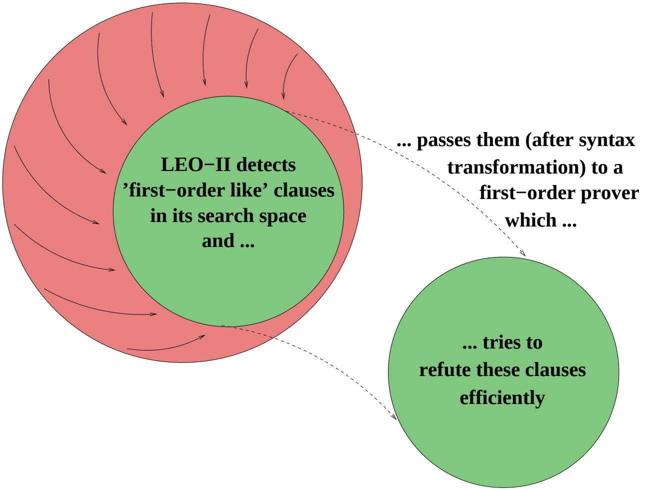
A simple equation between modal logic formulas

$$\forall R \centerdot \forall A \centerdot \forall B \centerdot (\Box_R (A \vee B)) \doteq (\Box_R (B \vee A))$$

where $\doteq$ is defined as $\lambda X, Y . \forall P \centerdot (P\, X) \Rightarrow (P\, Y)$

- normalisation: ... see previous example ...
- total proving time is $0.166s$

Architecture of LEO-II



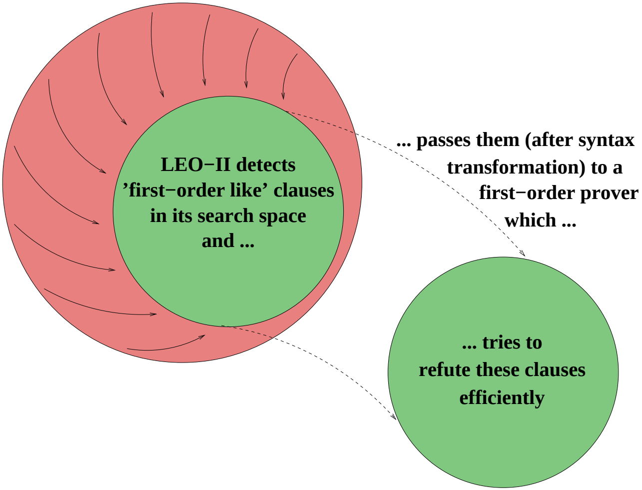
More Examples ...

In modal logic **K**, the axioms T and 4 are equivalent to reflexivity and transitivity of the accessibility relation R

$$\forall R. (\forall A. \texttt{valid}(\Box_R\, A \Rightarrow A) \land \texttt{valid}(\Box_R\, A \Rightarrow \Box_R\, \Box_R\, A))$$
$$\Leftrightarrow (\texttt{reflexive}(R) \land \texttt{transitive}(R))$$

- processing in LEO-II analogous to previous example
- now 70 clauses are passed to E
- E generates **21769** clauses before finding the empty clause
- total proving time 2.4s
- this proof cannot be found in LEO-II alone

Architecture of LEO-II



More Examples . . .

$\mathtt{S4} \not\subseteq \mathtt{K}$: Axioms T and 4 are not valid in modal logic $\mathbf{K}$

$$\neg\forall R. \forall A. \forall B. (\text{valid}(\Box_R\, A \Rightarrow A)) \wedge (\text{valid}(\Box_R\, B \Rightarrow \Box_R\, \Box_R\, B))$$

- LEO-II shows that axiom T is not valid
- R is instantiated with $\neq$ via primitive substitution
- total proving time 17.3s

More Examples . . .

$\mathtt{T} \not\subseteq \mathtt{K}$: Axiom T is not valid in modal logic K

$$\neg\forall R.\forall A.(\text{valid}(\Box_R A \Rightarrow A))$$

- initialisation, definition expansion and normalization generates:

$$((R\,W)\,s^{A,W,R}) \vee (A\,W)$$
$$\neg(A\,s^{A,W,R}) \vee (A\,W)$$

where $s^{A,W,R} = (((s\,A)\,W)\,R)$ is a new Skolem term

More Examples . . .

$\mathtt{T} \not\subseteq \mathtt{K}$: Axiom T is not valid in modal logic **K**

$$\neg\forall R_\blacksquare \forall A_\blacksquare (\mathrm{valid}(\Box_R A \Rightarrow A))$$

- the refutation employs only the former clause

$$((R\,W)\, s^{A,W,R}) \vee (A\,W)$$

More Examples ...

$\mathtt{T} \not\subseteq \mathtt{K}$: Axiom T is not valid in modal logic K

$$\neg\forall R. \forall A. (\text{valid}(\Box_R A \Rightarrow A))$$

- $((R\,W)\,s^{A,W,R}) \vee (A\,W)$
- LEO-II 'guesses' the instantiations

$$R \leftarrow \lambda X, Y.((M\,X)\,Y) \neq ((N\,X)\,Y)$$
$$A \leftarrow \lambda X.(O\,X) \neq (P\,X)$$

with primitive substitution rule (M, N, O, P are new free variables) ...

More Examples ...

$\mathtt{T} \not\subseteq \mathtt{K}$: Axiom T is not valid in modal logic **K**

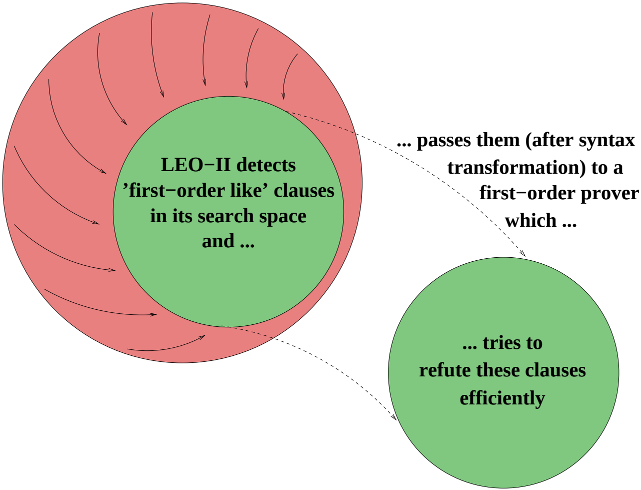
$$\neg\forall R.\forall A.(\text{valid}(\Box_R A \Rightarrow A))$$

- ...and applies them

$$((M\,(R\,W))\,s^{A,W,R}) \neq ((N\,(R\,W))\,s^{A,W,R})$$
$$\vee$$
$$(O\,W) \neq (P\,W)$$

- such flex-flex unification constraints are always solvable!
- total proving time 9.0s

Architecture of LEO-II



Limitations

LEO-II cannot prove the following example:

Modal logic $K4$ (which adds only axiom 4 to K) is not entailed in K:

$$\neg\forall R_\blacksquare \forall B_\blacksquare (\text{valid}(\Box_R\, B \Rightarrow \Box_R\, \Box_R\, B))$$

Limitations

LEO-II also cannot prove this related example:

$$\neg\forall R. \mathrm{trans}(R)$$

- reason: not a theorem; domain of possible worlds may well just consist of a single world w.
- LEO-II can in fact prove the latter example under the additional assumption

$$\neg\forall X. \forall Y. X = Y$$

Limitations

LEO-II also cannot prove this related example:

$$\neg \forall R. \mathrm{trans}(R)$$

- reason: not a theorem; domain of possible worlds may well just consist of a single world w.
- LEO-II can in fact prove the latter example under the additional assumption

$$\neg \forall X. \forall Y. X = Y$$

Limitations

LEO-II also cannot prove this related example:

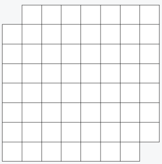
$$\neg \forall R. \text{trans}(R)$$

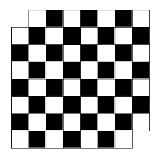
- reason: not a theorem; domain of possible worlds may well just consist of a single world w.
- LEO-II can in fact prove the latter example under the additional assumption

$$\neg \forall X. \forall Y. X = Y$$

Representation (and the right System Architecture) Matters!

A general lesson in AI . . .





. . . and a specific lesson here

FOL	HOL
+	+
FO-ATP	LEO-II + FO-ATP

... there is much left to be done!

LEO-II

- Equational Reasoning
- Termindexing
- Handling of Definitions

Cooperat. with Specialist Reasoners

- Monadic Second-Order Logic, Prop. Logic, Arithmetic, ...
- Logic Translations
- Feedback for LEO-II
- Proof Transf./Verification
- Agent-based Architecture

Integration into Proof Assistants

- Relevance of Axioms
- Proof Transf./Verification

International Infrastructure

- TPTP Language(s) for HOL
- Repository of Proof Problems
- HOL Prover Contest

Applications

Logic System Interrelationships, Ontology Reasoning (SUMO, CYC), Formal Methods, CL, ...

... there is much left to be done!

LEO-II

- Equational Reasoning
- Termindexing
- Handling of Definitions

Cooperat. with Specialist Reasoners

- Monadic Second-Order Logic, Prop. Logic, Arithmetic, ...
- Logic Translations
- Feedback for LEO-II
- Proof Transf./Verification
- Agent-based Architecture

Integration into Proof Assistants

- Relevance of Axioms
- Proof Transf./Verification

International Infrastructure

- TPTP Language(s) for HOL
- Repository of Proof Problems
- HOL Prover Contest

Applications

Logic System Interrelationships, Ontology Reasoning (SUMO, CYC), Formal Methods, CL, ...

... there is much left to be done!

LEO-II

- Equational Reasoning
- Termindexing
- Handling of Definitions

Cooperat. with Specialist Reasoners

- Monadic Second-Order Logic, Prop. Logic, Arithmetic, . . .
- Logic Translations
- Feedback for LEO-II
- Proof Transf./Verification
- Agent-based Architecture

Integration into Proof Assistants

- Relevance of Axioms
- Proof Transf./Verification

International Infrastructure

- TPTP Language(s) for HOL
- Repository of Proof Problems
- HOL Prover Contest

Applications

Logic System Interrelationships, Ontology Reasoning (SUMO, CYC), Formal Methods, CL, . . .

… there is much left to be done!

LEO-II

- Equational Reasoning
- Termindexing
- Handling of Definitions

Cooperat. with Specialist Reasoners

- Monadic Second-Order Logic, Prop. Logic, Arithmetic, . . .
- Logic Translations
- Feedback for LEO-II
- Proof Transf./Verification
- Agent-based Architecture

Integration into Proof Assistants

- Relevance of Axioms
- Proof Transf./Verification

International Infrastructure

- TPTP Language(s) for HOL
- Repository of Proof Problems
- HOL Prover Contest

Applications

Logic System Interrelationships, Ontology Reasoning (SUMO, CYC), Formal Methods, CL, . . .

... there is much left to be done!

LEO-II

- Equational Reasoning
- Termindexing
- Handling of Definitions

Cooperat. with Specialist Reasoners

- Monadic Second-Order Logic, Prop. Logic, Arithmetic, ...
- Logic Translations
- Feedback for LEO-II
- Proof Transf./Verification
- Agent-based Architecture

Integration into Proof Assistants

- Relevance of Axioms
- Proof Transf./Verification

International Infrastructure

- TPTP Language(s) for HOL
- Repository of Proof Problems
- HOL Prover Contest

Applications

Logic System Interrelationships, Ontology Reasoning (SUMO, CYC), Formal Methods, CL, ...

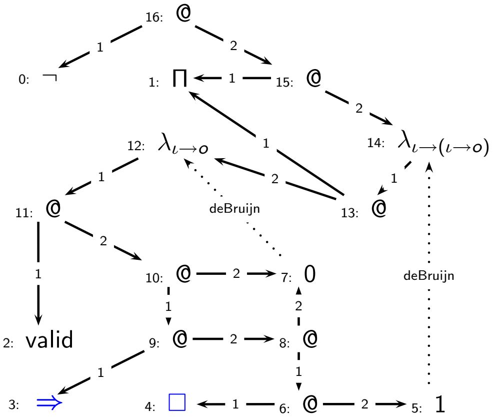
More Information on LEO-II

- Website with online version of LEO-II:

  `http://www.ags.uni-sb.de/~leo`

- System description [IJCAR-08]
- TPTP THF input syntax [IJCAR-THF-08]
- Reasoning in and about multimodal logic [Festschrift-Andrews-08]

Term Graph for: $\neg\forall R. \forall A. (\mathbf{valid}(\Box_R A \Rightarrow A))$



Term graph videos: `http://www.ags.uni-sb.de/~leo/art`

Latest Application of LEO-II: Dancefloor Animation

Grooving to an animation of LEO-II's dynamically growing termgraph (while LEO-II is proving Cantor's theorem)