Automating Access Control Logics and Multimodal Logics in the Automatic Higher-Order Theorem Prover LEO-II[1]

Christoph E. Benzmüller

Pure and Applied Logic Seminar, CMU, Nov 19th, 2008

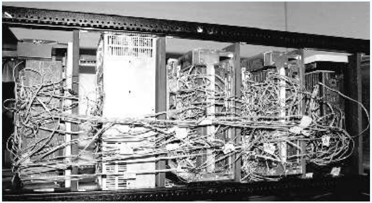
[1]Supported by EPSRC grant LEO-II at Cambridge University and EU grant THFTPTP.

Overview

- LEO-II
- (Normal) Multimodal Logic in LEO-II
- Access Control Logic in LEO-II

Church's Simple Type Theory (HOL)

Some folks say that Automation of HOL is like this:



I don't!

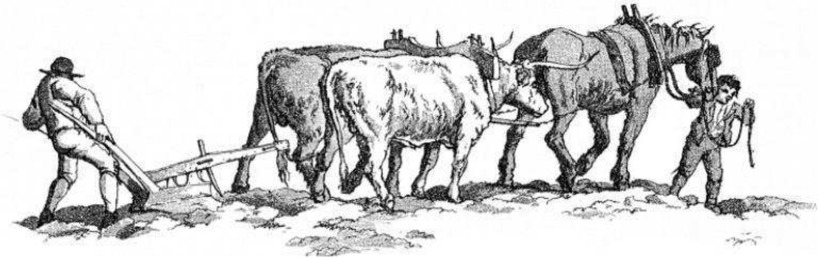
- Semantics (with C. Brown, M. Kohlhase) [JSL'04]
- Proof theory (with C. Brown) [IJCAR'06,LMCS'08]
- ATPs LEO and **LEO-II** [CADE'98,IJCAR'08a]
- HOL TPTP Infrastructure (with G. Sutcliffe) [IJCAR'08b]

LEO-II

An Effective Higher-Order Theorem Prover

UNIVERSITY OF CAMBRIDGE

UNIVERSITÄT DES SAARLANDES

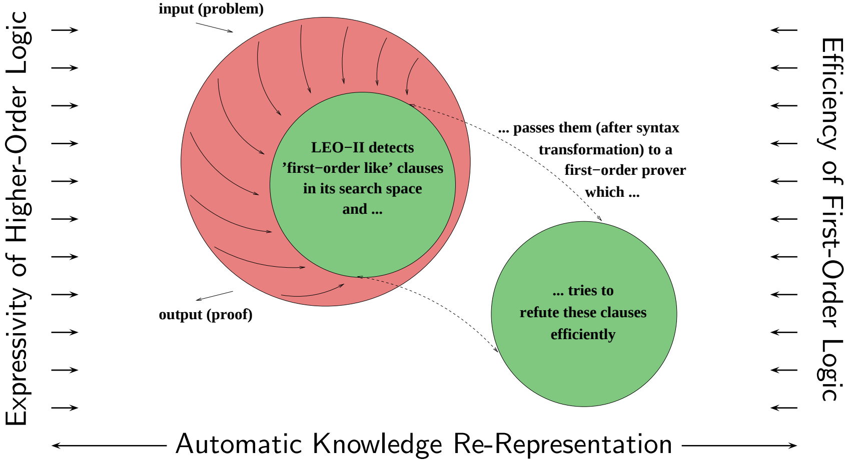


LEO-II employs FO-ATPs: E, Spass, Vampire

Roots of LEO-II

- Peter Andrews' work and TPS [various papers]
- Huet's Constrained Resolution [Huet'73]
- LEO hardwired to ΩMEGA (with M. Kohlhase) [CADE'98,PhD'99]
- Agent-based architecture O-ANTS (with V. Sorge) [AIMSA'98,EPIA'99,Calculemus'00]
- Collaboration of LEO with FO-ATP via O-ANTS (with V. Sorge) [KI'01,LPAR'05,JAL'07]
- EPSRC Project LEO-II at Cambridge University (with L. Paulson) [IJCAR'08a]
- EU Project THFTPTP: Infrastructure for ATP in HOL (with G. Sutcliffe) [IJCAR'08b]

Architecture of LEO-II



LEO-II's Calculus: A Sketch

Initialisation

Axioms: A $\qquad [A]^T$

Conjecture: C $\qquad [C]^F$

Clause Normalisation

$$\frac{C, [A \vee B]^F}{C, [A]^F \quad C, [B]^F} \quad \ldots$$

Resolution & Factorisation

$$\frac{C, [A]^F \quad D, [B]^F}{C, D, [A \neq^? B]} \quad \ldots$$

Rewriting (e.g. Definitions)

$\ldots$

Primitive Substitution

$$\frac{C, [X\, S^1 \ldots S^n]^T}{C, [(U\, S^1 \ldots S^n) \vee (V\, S^1 \ldots S^n)]^T} \quad \ldots$$

Eager Pre-Unification (depth limited!!!)

$$\frac{C, [\lambda X_\blacksquare T \neq^? \lambda Y_\blacksquare S]}{C, [T\, sk \neq^? S\, sk]} \quad \ldots$$

$$\frac{C, [X \neq^? T]}{\{T/X\} C} \quad \ldots$$

Extensional Pre-Unification

$$\frac{C, [A_o \neq^? B_o]}{C, [A_o \Leftrightarrow B_o]^F} \quad \ldots$$

LEO-II's Calculus: A Sketch

Initialisation

Axioms: A $\qquad [A]^T$

Conjecture: C $\qquad [C]^F$

Clause Normalisation

$$\frac{C, [A \vee B]^F}{C, [A]^F \quad C, [B]^F} \quad \ldots$$

Resolution & Factorisation

$$\frac{C, [A]^F \quad D, [B]^F}{C, D, [A \neq^? B]} \quad \ldots$$

Rewriting (e.g. Definitions)

...

Primitive Substitution

$$\frac{C, [X\, S^1 \ldots S^n]^T}{C, [(U\, S^1 \ldots S^n) \vee (V\, S^1 \ldots S^n)]^T} \quad \ldots$$

Eager Pre-Unification (depth limited!!!)

$$\frac{C, [\lambda X_{\blacksquare} T \neq^? \lambda Y_{\blacksquare} S]}{C, [T\, sk \neq^? S\, sk]} \quad \ldots$$

$$\frac{C, [X \neq^? T]}{\{T/X\}C} \quad \ldots$$

Extensional Pre-Unification

$$\frac{C, [A_o \neq^? B_o]}{C, [A_o \Leftrightarrow B_o]^F} \quad \ldots$$

Cooperation with Specialist Provers for Fragments of HOL

Provers supported (so far only FOL)

E, SPASS, Vampire

Translations supported so far

$@_\alpha$-FO-translation [Kerber'94]:

$[X_{\alpha\to\beta\to o}\ a_\alpha\ b_\beta]^T \to$

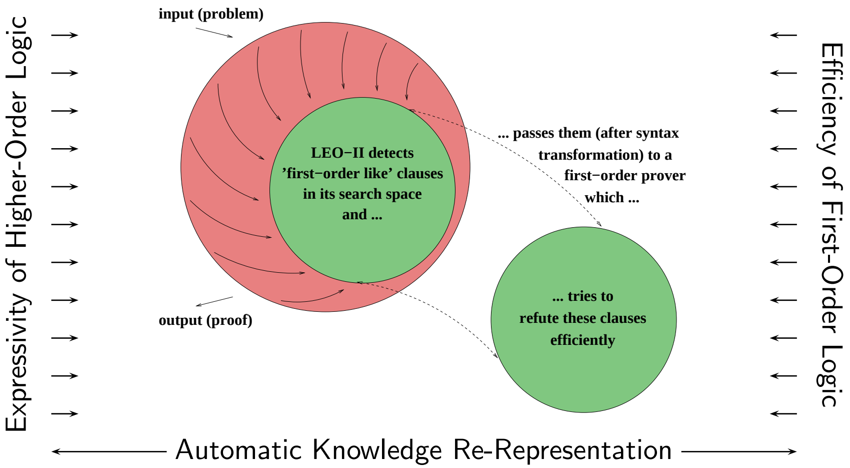
$$[@^{(\beta\to o)\to\alpha\to o}(@^{(\alpha\to\beta\to o)\to\alpha\to(\beta\to o)}(X,a),b)]^T$$

fully typed FO-translation [Hurd'02]:

$[X_{\alpha\to\beta\to o}\ a_\alpha\ b_\beta]^T \to$

$$[ti(@(ti(@(ti(X,\alpha\to\beta\to o),ti(a,\alpha)),\beta\to o),ti(b,\beta)),o)]^T$$

Architecture of LEO-II



LEO-II: Intermediate Summary

What it is special about LEO-II? The combination of

- extensional higher-order constrained resolution
- automatic reduction to first-order representations
- cooperation with first-order ATPs
- higher-order termsharing and termindexing techniques
- (automatic and interactive mode)

Try LEO-II (running under Ocaml 3.10)

- Website: `http://www.ags.uni-sb.de/~leo`
  - very easy to install; runs under Linux, MacOS, Cygwin
  - online demo
- Systems on TPTP:
  `http://www.cs.miami.edu/~tptp/cgi-bin/SystemOnTPTP`

Quick Online Demo: www.tptp.org

TPTP Problem SET171+3 in FOL

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---3.01
% Problem  : SET171+3
% SPASS beiseite: Ran out of time.

% E---1.0
% Problem  : SET171+3
% Failure: Ran out of time

% Vampire---10.0
% Problem  : SET171+3
% Result   : Theorem 102.2s
```

Performance in HOL: LEO-II + E

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

TPTP Problem SET171+3 in FOL

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---3.01
% Problem : SET171+3
% SPASS beiseite: Ran out of time.

% E---1.0
% Problem : SET171+3
% Failure: Ran out of time

% Vampire---10.0
% Problem : SET171+3
% Result : Theorem 102.2s
```

Performance in HOL: LEO-II + E

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

TPTP Problem SET171+3 in HOL

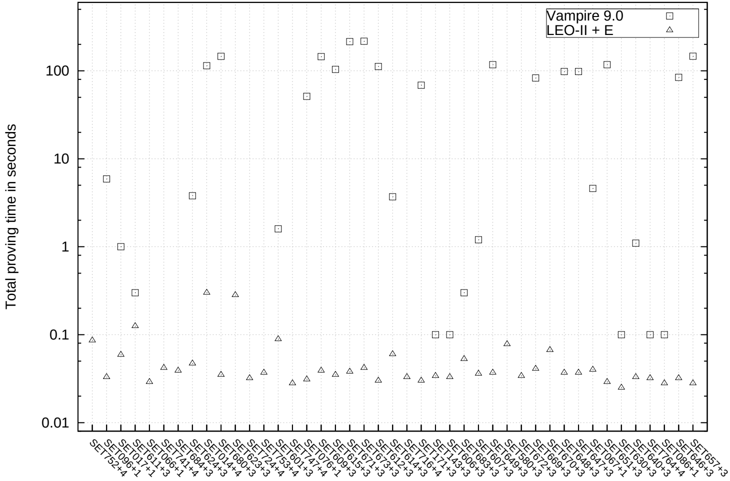
Sets in HOL

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

Proof Goal:

$$\forall A_{\alpha\to o}, B_{\alpha\to o}, C_{\alpha\to o} \blacksquare A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$$

Results



Vampire 9.0
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

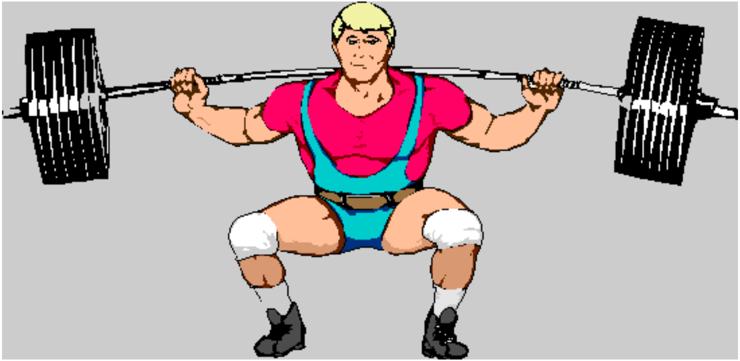
Results

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

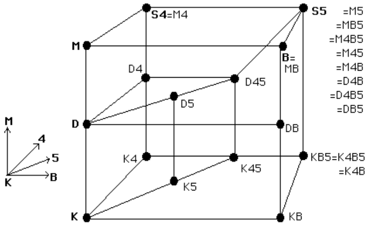
Less Lightweight Problems



Multimodal Logics

Logic Systems Interrelationships

Modal Logics Challenge



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

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

Example

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

Theorems:

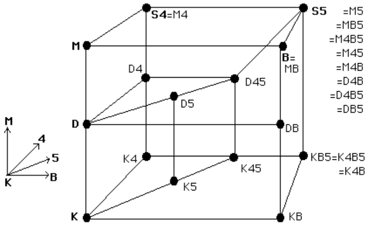
$$
\begin{array}{cclr}
S4 & \not\subseteq & K & (1) \\
(M \wedge 4) & \Leftrightarrow & (refl.\,R \wedge trans.\,R) & (2)
\end{array}
$$

Experiments

	FO-ATPs [SutcliffeEtal-08]	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

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

Theorems:

$$S4 \not\subseteq K \tag{1}$$

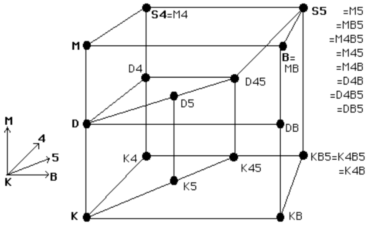
$$(M \wedge 4) \Leftrightarrow (refl.\ R \wedge trans.\ R) \tag{2}$$

Experiments

	FO-ATPs [SutcliffeEtal-08]	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

$$\begin{array}{rcll} S4 & = & K & \\ & + & M(T): & \Box a \Rightarrow a \\ & + & 4: & \Box a \Rightarrow \Box \Box a \end{array}$$

Theorems:

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

$$(M \wedge 4) \Leftrightarrow (refl.\ R \wedge trans.\ R) \quad (2)$$

Experiments

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

(Normal) Multimodal Logic in HOL

$$s, t ::= p \mid \neg\, s \mid s \vee t \mid \Box_r\, s$$

Simple, Straightforward Encoding

- base type ι: set of possible worlds
- (certain) terms of type $\iota \to o$: multimodal logic formulas

$$\begin{aligned}
\lfloor \neg\; s \rfloor &= \lambda w_\iota.\, \neg(\lfloor s \rfloor\, w) \\
\lfloor s \vee t \rfloor &= \lambda w_\iota.\, \lfloor s \rfloor\, w \vee \lfloor t \rfloor\, w \\
\lfloor \Box_r\; s \rfloor &= \lambda w_\iota.\, \forall y_\iota.\, \lfloor r \rfloor\, w\, y \Rightarrow \lfloor s \rfloor\, y \\
\lfloor p \rfloor &= p_{\iota \to o}
\end{aligned}$$

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

(Normal) Multimodal Logic in HOL

$$s, t ::= p | \neg\, s | s \vee t | \Box_r\, s$$

Simple, Straightforward Encoding

- base type ι: set of possible worlds
- (certain) terms of type $\iota \to o$: multimodal logic formulas

$$
\begin{aligned}
|\neg| &= \lambda s_{\iota\to o} \lambda w_\iota \neg(s\, w)\\
|\vee| &= \lambda s_{\iota\to o} \lambda t_{\iota\to o} \lambda w_\iota\, s\, w \vee t\, w\\
|\Box| &= \lambda r_{\iota\to\iota\to o} \lambda s_{\iota\to o} \lambda w_\iota \forall y_\iota\, r\, w\, y \Rightarrow s\, y\\
|p| &= p_{\iota\to o}\\
|r| &= r_{\iota\to\iota\to o}
\end{aligned}
$$

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

(Normal) Multimodal Logic in HOL

Encoding of Validity

$$\texttt{valid}\, s_{\iota\to o} = \forall w_\iota.\, s\, w$$
$$|\texttt{valid}| = \lambda s_{\iota\to o}.\, \forall w_\iota.\, s\, w$$

Local Definition Expansion

$$|\texttt{valid}\ \Box_r\ \top| = |\texttt{valid}|\, |\Box|\, |r|\, |\top|$$
$$=^{\beta\eta} \forall w_\iota.\, \forall y_\iota.\, \neg r\, w\, y \vee \top$$

(Normal) Multimodal Logic in HOL

Encoding of Validity

$$\mathtt{valid}\, s_{\iota\to o} \quad = \quad \forall w_\iota \centerdot s\, w$$

$$|\mathtt{valid}| \quad = \quad \lambda s_{\iota\to o} \centerdot \forall w_\iota \centerdot s\, w$$

Local Definition Expansion

$$|\mathtt{valid}\ \Box_r\ \top| \quad = \quad |\mathtt{valid}|\,|\Box|\,|r|\,|\top|$$

$$=^{\beta\eta} \quad \forall w_\iota \centerdot \forall y_\iota \centerdot \neg r\, w\, y \vee \top$$

(Normal) Multimodal Logic in HOL

Encoding of Validity

$$\begin{aligned} \mathtt{valid}\, s_{\iota\to o} &= \forall w_\iota \centerdot s\, w \\ |\mathtt{valid}| &= \lambda s_{\iota\to o} \centerdot \forall w_\iota \centerdot s\, w \end{aligned}$$

Local Definition Expansion

$$\begin{aligned} |\mathtt{valid}\ \Box_r\ \top| &= |\mathtt{valid}|\, |\Box\,|\, |r|\, |\top| \\ &=^{\beta\eta} \forall w_\iota \centerdot \forall y_\iota \centerdot \neg r\, w\, y \vee \top \end{aligned}$$

Even simpler: Reasoning within Multimodal Logics

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

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

Initialisation of problem

$$[|\texttt{valid}\ \Box_s (\Box_r\, a \Rightarrow \Box_r\, a)|]^F$$

Definition expansion

$$[\forall x_\iota \centerdot \forall y_\iota \centerdot \neg s\, x\, y \vee ((\neg(\forall u_\iota \centerdot \neg r\, y\, u \vee a\, u)) \vee (\forall v_\iota \centerdot \neg r\, y\, v \vee a\, v))]^F$$

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

$$[s\, x\, y]^T \qquad [a\, u]^F$$
$$[r\, y\, u]^T \qquad [a\, V]^T \vee [r\, y\, V]^F$$

Translation to first-order logic [Kerber-94], [Hurd-02], [MengPaulson-04]

$$[@^{\cdots}(@^{\cdots}(s, x), y)]^T \qquad [@^{\cdots}(a, u)]^F$$
$$[@^{\cdots}(@^{\cdots}(r, y), u)]^T \qquad [@^{\cdots}(a, V)]^T \vee [@^{\cdots}(@^{\cdots}(r, y), V)]^F$$

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

Initialisation of problem

$$[|\mathtt{valid}\ \Box_s (\Box_r a \Rightarrow \Box_r a)|]^F$$

Definition expansion

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

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

$$[s\,x\,y]^T \qquad [a\,u]^F$$

$$[r\,y\,u]^T \qquad [a\,V]^T \vee [r\,y\,V]^F$$

Translation to first-order logic [Kerber-94], [Hurd-02], [MengPaulson-04]

$$[@^{\cdots}(@^{\cdots}(s,x),y)]^T \qquad [@^{\cdots}(a,u)]^F$$

$$[@^{\cdots}(@^{\cdots}(r,y),u)]^T \qquad [@^{\cdots}(a,V)]^T \vee [@^{\cdots}(@^{\cdots}(r,y),V)]^F$$

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

Initialisation of problem

$$[|\texttt{valid}\ \Box_s (\Box_r a \Rightarrow \Box_r a)|]^F$$

Definition expansion

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

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

$$[s\, x\, y]^T \qquad [a\, u]^F$$

$$[r\, y\, u]^T \qquad [a\, V]^T \vee [r\, y\, V]^F$$

Translation to first-order logic [Kerber-94], [Hurd-02], [MengPaulson-04]

$$[@^{\cdots}(@^{\cdots}(s, x), y)]^T \qquad [@^{\cdots}(a, u)]^F$$

$$[@^{\cdots}(@^{\cdots}(r, y), u)]^T \qquad [@^{\cdots}(a, V)]^T \vee [@^{\cdots}(@^{\cdots}(r, y), V)]^F$$

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

Initialisation of problem

$$[|\texttt{valid}\ \Box_s (\Box_r\, a \Rightarrow \Box_r\, a)|]^F$$

Definition expansion

$$[\forall x_\iota\text{.}\forall y_\iota\text{.}\neg s\, x\, y \vee ((\neg(\forall u_\iota\text{.}\neg r\, y\, u \vee a\, u)) \vee (\forall v_\iota\text{.}\neg r\, y\, v \vee a\, v))]^F$$

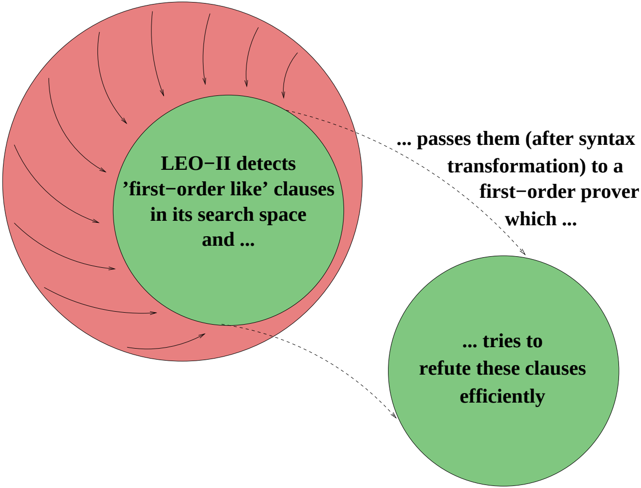
Normalisation (x, y, u are now Skolem constants, V is a free variable)

$$[s\, x\, y]^T \qquad [a\, u]^F$$
$$[r\, y\, u]^T \qquad [a\, V]^T \vee [r\, y\, V]^F$$

Translation to first-order logic [Kerber-94], [Hurd-02], [MengPaulson-04]

$$[@^{\cdots}(@^{\cdots}(s, x), y)]^T \qquad [@^{\cdots}(a, u)]^F$$
$$[@^{\cdots}(@^{\cdots}(r, y), u)]^T \qquad [@^{\cdots}(a, V)]^T \vee [@^{\cdots}(@^{\cdots}(r, y), V)]^F$$

Architecture of LEO-II



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 $\lambda u, v. \forall p_\blacksquare\, p\, u \Rightarrow p\, v$

- initialisation, definition expansion and normalisation:

$$[p\,(\lambda w_\iota. \forall y_{\iota\blacksquare} \neg r\, w\, y \vee (a\, y \vee b\, y))]^T$$
$$[p\,(\lambda w_\iota. \forall y_{\iota\blacksquare} \neg r\, w\, y \vee (b\, y \vee a\, y))]^F$$

More Examples . . .

A simple equation between modal logic formulas

$$\forall r_\blacksquare \forall a_\blacksquare \forall b_\blacksquare \lfloor \Box_r (a \vee b) \rfloor \doteq \lfloor \Box_r (b \vee a) \rfloor$$

where $\doteq$ is defined as $\lambda u, v. \forall p_\blacksquare p\, u \Rightarrow p\, v$

- resolution:

$$[p\,(\lambda w_\iota . \forall y_{\iota\blacksquare} \neg r\, w\, y \vee (a\, y \vee b\, y))$$
$$\neq^?$$
$$p\,(\lambda w_\iota . \forall y_{\iota\blacksquare} \neg r\, w\, y \vee (b\, y \vee a\, y))]$$

More Examples ...

A simple equation between modal logic formulas

$$\forall r.\forall a.\forall b.\, |\Box_r\,(a \vee b)| \doteq |\Box_r\,(b \vee a)|$$

where $\doteq$ is defined as $\lambda u, v.\forall p.\, p\, u \Rightarrow p\, v$

- decomposition:

$$\begin{array}{l} [\lambda w_\iota.\forall y_\iota.\, \neg r\, w\, y \vee (a\, y \vee b\, y) \\ \ \neq^? \\ \ \lambda w_\iota.\forall y_\iota.\, \neg r\, w\, y \vee (b\, y \vee a\, y)] \end{array}$$

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 $\lambda u, v. \forall p_{\blacksquare}\, p\, u \Rightarrow p\, v$

- functional extensionality (w is now Skolem constant):

$$\begin{array}{l}[\forall y_{\iota \blacksquare} \neg r\, w\, y \vee (a\, y \vee b\, y) \\ \quad \neq^{?} \\ \ \forall y_{\iota \blacksquare} \neg r\, w\, y \vee (b\, y \vee a\, y)]\end{array}$$

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 $\lambda u, v. \forall p. p\, u \Rightarrow p\, v$

- Boolean extensionality:

$$[\forall y_\iota. \neg r\, w\, y \vee (a\, y \vee b\, y)$$
$$\Leftrightarrow$$
$$\forall y_\iota. \neg r\, w\, y \vee (b\, y \vee a\, y)]^F$$

More Examples ...

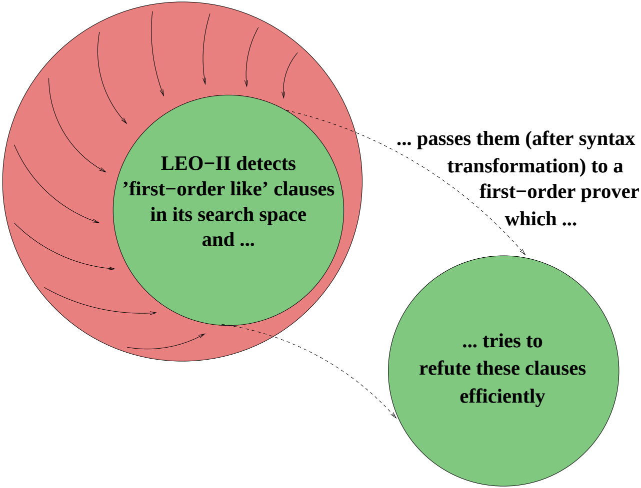
A simple equation between modal logic formulas

$$\forall r_\blacksquare \forall a_\blacksquare \forall b_\blacksquare \lfloor \Box_r (a \vee b) \rfloor \doteq \lfloor \Box_r (b \vee a) \rfloor$$

where $\doteq$ is defined as $\lambda u, v.\forall p_\blacksquare p\, u \Rightarrow p\, v$

- normalisation (v, z Skolem constants; V,Z Variables):
  - 40 : $[b\, V]^T \vee [a\, V]^T \vee [r\, w\, V]^F \vee [r\, w\, Z]^F \vee [b\, Z]^T \vee [a\, Z]^T$
  - 41 : $[r\, w\, z]^T \vee [r\, w\, v]^T$
  - 42 : $[a\, z]^F \vee [r\, w\, v]^T$
  - 43 : $[b\, z]^F \vee [r\, w\, v]^T$
  - 44 : $[r\, w\, z]^T \vee [a\, v]^F$
  - 45 : $[a\, z]^F \vee [a\, v]^F$
  - 46 : $[b\, z]^F \vee [a\, v]^F$
  - 47 : $[r\, w\, z]^T \vee [b\, v]^F$
  - 48 : $[a\, z]^F \vee [b\, v]^F$
  - 49 : $[b\, z]^F \vee [b\, v]^F$
- total proving time is $0.166s$

Architecture of LEO-II



More Examples ...

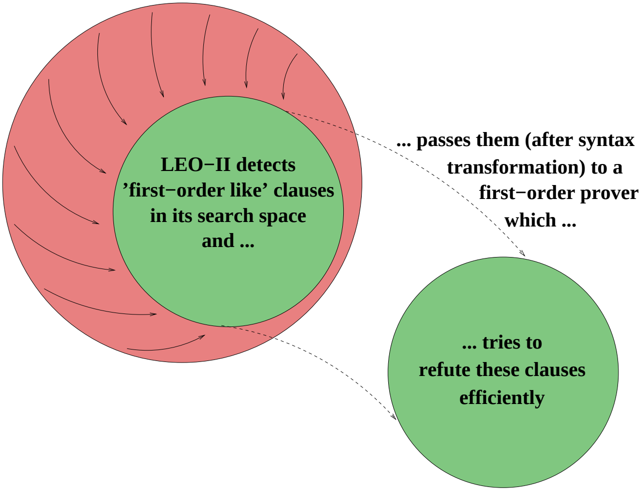
Axioms T and 4 are equivalent to reflexivity and transitivity of the accessibility relation r

$$\forall r_{\blacksquare}(\forall a_{\blacksquare}\lfloor \texttt{valid}\ \Box_r\, a \Rightarrow a\rfloor \wedge \lfloor \texttt{valid}\ \Box_r\, a \Rightarrow \Box_r\, \Box_r\, a\rfloor)$$
$$\Leftrightarrow (\texttt{reflexive}\, r \wedge \texttt{transitive}\, r)$$

$reflexive := \lambda r_{\blacksquare}\forall x_{\blacksquare}\, r\, x\, x$
$transitive := \lambda r_{\blacksquare}\forall x, y, z_{\blacksquare}\, r\, x\, y \wedge r\, y\, z \Rightarrow r\, x\, z$

- processing analogous to previous example
- now 70 clauses passed to E
- E generates **21769** clauses before finding the empty clause
- total proving time 2.4s

Architecture of LEO-II



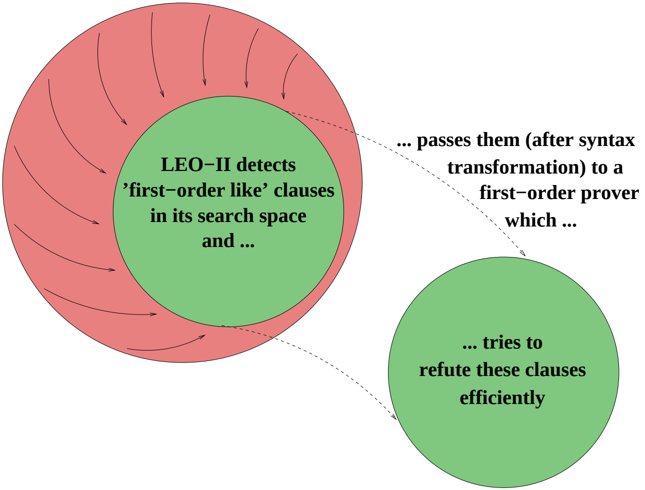
More Examples ...

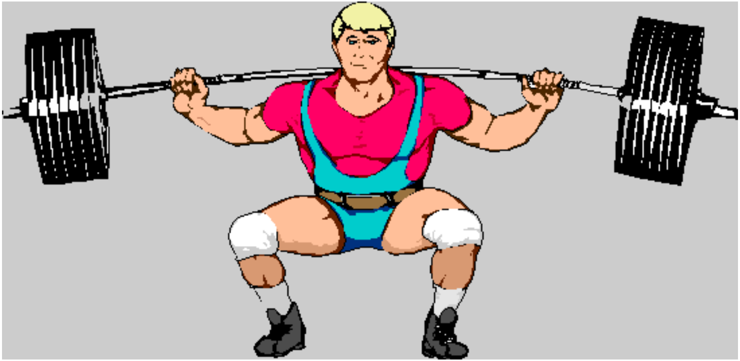
$\mathrm{S4} \not\subseteq \mathrm{K}$: Axioms T and 4 are not valid in modal logic $\mathbf{K}$

$$\neg\forall R.\forall A.\forall B.\lfloor \texttt{valid}\ \Box_R A \Rightarrow A\rfloor \wedge \lfloor \texttt{valid}\ \Box_R B \Rightarrow \Box_R \Box_R B\rfloor$$

- LEO-II shows that first axiom is not valid
- R is instantiated with $\lambda x.\lambda y.(H\,x\,y) \neq (H'\,x\,y)$ via primitive substitution
- total proving time 17.3s

Architecture of LEO-II





Access Control Logics

Access Control Logic

Example (from [GargAbadi08]): file-access scenario

- If `admin` says that `file1` should be deleted, then this must be the case.

  $(\texttt{admin}\,\texttt{says}\,\texttt{deletefile1}) \supset \texttt{deletefile1}$

- `admin` trusts Bob to decide whether `file1` should be deleted.

  $\texttt{admin}\,\texttt{says}\,((\texttt{Bob}\,\texttt{says}\,\texttt{deletefile1}) \supset \texttt{deletefile1})$

- Bob wants to delete `file1`.

  $\texttt{Bob}\,\texttt{says}\,\texttt{deletefile1}$

- Is deletion permitted?

  $\texttt{deletefile1}$

Access Control Logic

Deepak Garg, Martín Abadi [FoSSaCS'08]:

A Modal Deconstruction of Access Control Logics

- Study of Prominent Access Control Logics:
  - ICL: Propositional Intuitionistic Logic + "says"
  - $\text{ICL}^{\Rightarrow}$: ICL + "speaks for"
  - ICL^{B}: ICL + Boolean combinations of principals
- Sound and Complete Translations to Modal Logic S4

So, let's combine this with our previous work . . . and apply LEO-II

Access Control Logic

Deepak Garg, Martín Abadi [FoSSaCS'08]:
A Modal Deconstruction of Access Control Logics

- Study of Prominent Access Control Logics:
  - ICL: Propositional Intuitionistic Logic + "says"
  - $\mathrm{ICL}^{\Rightarrow}$: ICL + "speaks for"
  - ICL^{B}: ICL + Boolean combinations of principals
- Sound and Complete Translations to Modal Logic S4

So, let's combine this with our previous work . . . and apply LEO-II

Access Control Logic translated to Modal Logic and HOL

$$s, t ::= p \mid s \wedge t \mid s \vee t \mid s \supset t \mid \bot \mid \top \mid A \,\texttt{says}\, s$$

Translation $\lceil . \rceil$ (of Garg and Abadi) into S4

$$\begin{aligned}
\lceil p \rceil &= \Box p \\
\lceil s \wedge t \rceil &= \lceil s \rceil \wedge \lceil t \rceil \\
\lceil s \vee t \rceil &= \lceil s \rceil \vee \lceil t \rceil \\
\lceil s \supset t \rceil &= \Box(\lceil s \rceil \Rightarrow \lceil t \rceil) \\
\lceil \top \rceil &= \top \\
\lceil \bot \rceil &= \bot \\
\lceil A \,\texttt{says}\, s \rceil &= \Box(A \vee \lceil s \rceil)
\end{aligned}$$

Access Control Logic translated to Modal Logic and HOL

$$s, t ::= p \mid s \wedge t \mid s \vee t \mid s \supset t \mid \bot \mid \top \mid A \,\texttt{says}\, s$$

Translation $\|.\|$ to HOL

$$
\begin{array}{lcl}
 & & |r| \quad \text{(we fix one single } r\text{!!!)} \\
\|p\| & = & |\Box_r\, p| \\
\|A\| & = & |A| \\
\|\wedge\| & = & \lambda s. \lambda t. |s \wedge t| \\
\|\vee\| & = & \lambda s. \lambda t. |s \vee t| \\
\|\supset\| & = & \lambda s. \lambda t. |\Box(s \Rightarrow t)| \\
\|\top\| & = & |\top| \\
\|\bot\| & = & |\bot| \\
\|\texttt{says}\| & = & \lambda A. \lambda s. |\Box_r\, (A \vee s)|
\end{array}
$$

Access Control Logic translated to Modal Logic and HOL

$$s, t ::= p \mid s \wedge t \mid s \vee t \mid s \supset t \mid \bot \mid \top \mid A \,\texttt{says}\, s$$

Translation $\|.\|$ to HOL

$$
\begin{array}{lcl}
 & & r_{\iota\to\iota\to o} \quad \text{(we fix one single } r\text{!!!)} \\
\|p\| & = & \lambda x_\iota \centerdot \forall y_\iota \centerdot r_{\iota\to\iota\to o}\, x\, y \Rightarrow p_{\iota\to o}\, Y \\
\|A\| & = & a_{\iota\to o} \;\; \texttt{(distinct from the } p_{\iota\to o}\texttt{)} \\
\|\wedge\| & = & \lambda s_{\iota\to o} \centerdot \lambda t_{\iota\to o} \centerdot \lambda w_\iota \centerdot s\, w \wedge t\, w \\
\|\vee\| & = & \lambda s_{\iota\to o} \centerdot \lambda t_{\iota\to o} \centerdot \lambda w_\iota \centerdot s\, w \vee t\, w \\
\|\supset\| & = & \lambda s_{\iota\to o} \centerdot \lambda t_{\iota\to o} \centerdot \lambda w_\iota \centerdot \forall y_\iota \centerdot r_{\iota\to\iota\to o}\, w\, y \Rightarrow (s\, y \Rightarrow t\, y) \\
\|\top\| & = & \lambda s_{\iota\to o} \centerdot \top \\
\|\bot\| & = & \lambda s_{\iota\to o} \centerdot \bot \\
\|\texttt{says}\| & = & \lambda A_{\iota\to o} \centerdot \lambda s_{\iota\to o} \centerdot \lambda w_\iota \centerdot \forall y_\iota \centerdot r_{\iota\to\iota\to o}\, w\, y \Rightarrow (A\, y \vee s\, y)
\end{array}
$$

Access Control Logic translated to Modal Logic and HOL

Notion of Validity

$$\texttt{iclval} = \texttt{valid}$$

Addition of Modal Logic Axioms for S4

$$\forall p_{\iota \to o}. \lfloor \texttt{valid}\ \Box_r\, p \Rightarrow p \rfloor$$

$$\forall p_{\iota \to o}. \lfloor \texttt{valid}\ \Box_r\, p \Rightarrow \Box_r\, \Box_r\, p \rfloor$$

Soundness and Completeness of Embedding

see [SR-2008-01]: employs transformation from Kripke models into corresponding Henkin models and vice versa

Access Control Logic translated to Modal Logic and HOL

Notion of Validity

$$\texttt{iclval} = \texttt{valid}$$

Addition of Modal Logic Axioms for S4

$$\forall p_{\iota\to o}. \lfloor \texttt{valid}\ \Box_r\ p \Rightarrow p \rfloor$$

$$\forall p_{\iota\to o}. \lfloor \texttt{valid}\ \Box_r\ p \Rightarrow \Box_r\ \Box_r\ p \rfloor$$

Soundness and Completeness of Embedding

see [SR-2008-01]: employs transformation from Kripke models into corresponding Henkin models and vice versa

Access Control Logic translated to Modal Logic and HOL

Notion of Validity

$$\texttt{iclval} = \texttt{valid}$$

Addition of Modal Logic Axioms for S4

$$\forall p_{\iota \to o}. \lfloor \texttt{valid} \; \Box_r \, p \Rightarrow p \rfloor$$

$$\forall p_{\iota \to o}. \lfloor \texttt{valid} \; \Box_r \, p \Rightarrow \Box_r \, \Box_r \, p \rfloor$$

Soundness and Completeness of Embedding

see [SR-2008-01]: employs transformation from Kripke models into corresponding Henkin models and vice versa

Access Control Logic translated to Modal Logic and HOL

Example (from [GargAbadi08]): file-access scenario

- If `admin` says that `file1` should be deleted, then this must be the case.

  $$\|\mathtt{iclval}\ (\mathtt{admin}\,\mathtt{says}\,\mathtt{deletefile1}) \supset \mathtt{deletefile1}\|$$

- `admin` trusts Bob to decide whether `file1` should be deleted.

  $$\|\mathtt{iclval}\ \mathtt{admin}\,\mathtt{says}\,((\mathtt{Bob}\,\mathtt{says}\,\mathtt{deletefile1}) \supset \mathtt{deletefile1})\|$$

- Bob wants to delete `file1`.

  $$\|\mathtt{iclval}\ \mathtt{Bob}\,\mathtt{says}\,\mathtt{deletefile1}\|$$

- Is deletion permittted?

  $$\|\mathtt{iclval}\ \mathtt{deletefile1}\|$$

Access Control Logic translated to Modal Logic and HOL

Example (from [GargAbadi08]): file-access scenario

- If admin says that file1 should be deleted, then this must be the case.

  $$\|\texttt{iclval}\ (\texttt{admin}\,\texttt{says}\,\texttt{deletefile1}) \supset \texttt{deletefile1}\|$$

- admin trusts Bob to decide whether file1 should be deleted.

  $$\|\texttt{iclval}\ \texttt{admin}\,\texttt{says}\,((\texttt{Bob}\,\texttt{says}\,\texttt{deletefile1}) \supset \texttt{deletefile1})\|$$

- Bob wants to delete file1.

  $$|\texttt{valid}\ \Box_r\,(\texttt{Bob} \vee \Box_r\,\texttt{deletefile1})|$$

- Is deletion permittted?

  $$\|\texttt{iclval}\ \texttt{deletefile1}\|$$

Access Control Logic translated to Modal Logic and HOL

Example (from [GargAbadi08]): file-access scenario

- If admin says that file1 should be deleted, then this must be the case.

$$\|\texttt{iclval (admin says deletefile1)} \supset \texttt{deletefile1}\|$$

- admin trusts Bob to decide whether file1 should be deleted.

$$\|\texttt{iclval admin says ((Bob says deletefile1)} \supset \texttt{deletefile1)}\|$$

- Bob wants to delete file1.

$$\forall w_\iota\text{.}\forall y_\iota\text{.}\, r\; w\; y \Rightarrow (Bob\; y \vee \forall u_\iota\text{.}\, r\; w\; u \Rightarrow deletefile1\;\; u)$$

- Is deletion permittted?

$$\|\texttt{iclval deletefile1}\|$$

LEO-II: 3.494 seconds

More Examples from [GargAbadi08]

- Example I: 3.494 seconds
- Example II ($\mathrm{ICL}^{\Rightarrow}$): 0.698 seconds
- Example III ($\mathrm{ICL}B$): 0.076 seconds
- Validity of various axioms for "says" ("speaks-for", etc.): $<$ 0.2 seconds
- $\mathrm{ICL}^{\Rightarrow}$ can be expressed in $\mathrm{ICL}B$: 0.068 seconds

Conclusion

- Promising initial results for LEO-II (and TPS!)
  - sets
  - normal multimodal logics
  - access control logics (and intuitionistic logics)
  - ...?
- Does approach scale well? If yes, then there are many applications!

- What is special about LEO-II?
  - cooperation with specialist provers
  - termsharing and termindexing
  - extensional constrained resolution
  - lean system

... 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, ...

Challenges for HOL ATPs

- Cut-Simulation [IJCAR'06]
- Extensional Pre-Unification
- Primitive Substitution
- Primitive Equality [CADE'99]
- Definitions [BishopAndrews-CADE'98]

Exp.: Access Control Logic in HOL

Logic ICL:

Name	Problem	LEO (s)
unit	$\{\mathtt{R},\mathtt{T}\} \models \|\mathtt{ICLval}\ s \supset (A\,\mathsf{says}\,s)\|$	0.031
cuc	$\{\mathtt{R},\mathtt{T}\} \models \|\mathtt{ICLval}\ (A\,\mathsf{says}\,(s \supset t)) \supset (A\,\mathsf{says}\,s) \supset (A\,\mathsf{says}\,t)\|$	0.083
idem	$\{\mathtt{R},\mathtt{T}\} \models \|\mathtt{ICLval}\ (A\,\mathsf{says}\,A\,\mathsf{says}\,s) \supset (A\,\mathsf{says}\,s)\|$	0.037
Ex1	$\{\mathtt{R},\mathtt{T},\|\mathtt{ICLval}\ (1.1)\|,\ldots,\|\mathtt{ICLval}\ (1.3)\|\} \models \|\mathtt{ICLval}\ (1.4)\|$	3.494
unit^K	$\models \|\mathtt{ICLval}\ s \supset (A\,\mathsf{says}\,s)\|$	–
cuc^K	$\models \|\mathtt{ICLval}\ (A\,\mathsf{says}\,(s \supset t)) \supset (A\,\mathsf{says}\,s) \supset (A\,\mathsf{says}\,t)\|$	–
idem^K	$\models \|\mathtt{ICLval}\ (A\,\mathsf{says}\,A\,\mathsf{says}\,s) \supset (A\,\mathsf{says}\,s)\|$	–
Ex1^K	$\{\|\mathtt{ICLval}\ (1.1)\|,\ldots,\|\mathtt{ICLval}\ (1.3)\|\} \models \|\mathtt{ICLval}\ (1.4)\|$	–

R, T: reflexivity and transitivity axioms as seen before

Exp.: Access Control Logic in HOL

Logic ICL$^{\Rightarrow}$:

Name	Problem	LEO (s)
refl	$\{\mathtt{R},\mathtt{T}\} \models \lVert\mathtt{ICLval}\ A \Rightarrow A\rVert$	0.052
trans	$\{\mathtt{R},\mathtt{T}\} \models \lVert\mathtt{ICLval}\ (A \Rightarrow B) \supset (B \Rightarrow C) \supset (A \Rightarrow C)\rVert$	0.105
sp.-for	$\{\mathtt{R},\mathtt{T}\} \models \lVert\mathtt{ICLval}\ (A \Rightarrow B) \supset (A\,\mathsf{says}\,s) \supset (B\,\mathsf{says}\,s)\rVert$	0.062
handoff	$\{\mathtt{R},\mathtt{T}\} \models \lVert\mathtt{ICLval}\ (B\,\mathsf{says}\,(A \Rightarrow B)) \supset (A \Rightarrow B)\rVert$	0.036
Ex2	$\{\mathtt{R},\mathtt{T},\lVert\mathtt{ICLval}\ (2.1)\rVert,\ldots,\lVert\mathtt{ICLval}\ (2.4)\rVert\} \models \lVert\mathtt{ICLval}\ (2.5)\rVert$	0.698
refl^K	$\models \lVert\mathtt{ICLval}\ A \Rightarrow A\rVert$	0.031
trans^K	$\models \lVert\mathtt{ICLval}\ (A \Rightarrow B) \supset (B \Rightarrow C) \supset (A \Rightarrow C)\rVert$	–
sp.-for^K	$\models \lVert\mathtt{ICLval}\ (A \Rightarrow B) \supset (A\,\mathsf{says}\,s) \supset (B\,\mathsf{says}\,s)\rVert$	–
handoff^K	$\models \lVert\mathtt{ICLval}\ (B\,\mathsf{says}\,(A \Rightarrow B)) \supset (A \Rightarrow B)\rVert$	–
Ex2^K	$\{\lVert\mathtt{ICLval}\ (2.1)\rVert,\ldots,\lVert\mathtt{ICLval}\ (2.4)\rVert\} \models \lVert\mathtt{ICLval}\ (2.5)\rVert$	–

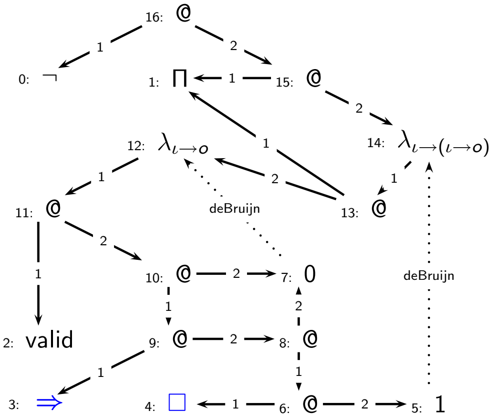
R, T: reflexivity and transitivity axioms as seen before

Logic ICL^B:

Name	Problem	LEO (s)
trust	$\{\mathtt{R},\mathtt{T}\} \models \|\mathtt{ICLval}\ (\bot\ \mathsf{says}\ s) \supset s\|$	0.049
untrust	$\{\mathtt{R},\mathtt{T}, \|\mathtt{ICLval}\ A \equiv \top\|\} \models \|\mathtt{ICLval}\ A\,\mathsf{says}\ \bot\|$	0.053
cuc'	$\{\mathtt{R},\mathtt{T}\} \models \|\mathtt{ICLval}\ ((A \supset B)\ \mathsf{says}\ s) \supset (A\,\mathsf{says}\ s) \supset (B\,\mathsf{says}\ s)\|$	0.131
Ex3	$\{\mathtt{R},\mathtt{T}, \|\mathtt{ICLval}\ (3.1)\|, \ldots, \|\mathtt{ICLval}\ (3.3)\|\} \models \|\mathtt{ICLval}\ (3.4)\|$	0.076
trust^K	$\models \|\mathtt{ICLval}\ (\bot\ \mathsf{says}\ s) \supset s\|$	–
$\mathrm{untrust}^K$	$\{\|\mathtt{ICLval}\ A \equiv \top\|\} \models \|\mathtt{ICLval}\ A\,\mathsf{says}\ \bot\|$	0.041
$\mathrm{cuc'}^K$	$\models \|\mathtt{ICLval}\ ((A \supset B)\ \mathsf{says}\ s) \supset (A\,\mathsf{says}\ s) \supset (B\,\mathsf{says}\ s)\|$	–
$\mathrm{Ex3}^K$	$\{\|\mathtt{ICLval}\ (3.1)\|, \ldots, \|\mathtt{ICLval}\ (3.3)\|\} \models \|\mathtt{ICLval}\ (3.4)\|$	–

R, T: reflexivity and transitivity axioms as seen before

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

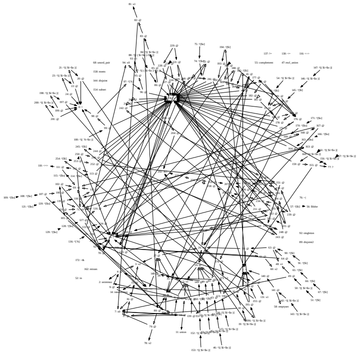


Term graph videos: `http://www.ags.uni-sb.de/~leo/art`

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)

Termsharing



In LEO-II:

- Terms as unique instances
- Perfect Term Sharing
- Shallow data structures

Features:

- β-η-normalization
- DeBruijn indices
- local contexts for polymorphic type variables

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:

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

- 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

$$\neg\forall X\text{.}\forall Y\text{.}X = Y$$

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:

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

- 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

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