

Automating Access Control Logics in Simple Type Theory with LEO-II¹

Christoph Benzmüller

International University in Germany, Bruchsal, Germany
& Articulate Software, Angwin, CA, U.S.

IFIP/SEC-2009, Paphos, Cyprus, May 18-20, 2009

¹This work was supported by EU grant PIIF-GA-2008-219982 (THFTPTP)



The Story — on a single slide



Simple Type Theory / HOL – an Expressive Logic



Multimodal Logics as Fragments of HOL



Access Control Logics as Fragments of S4 and hence HOL



Mechanization and Automation in HOL (prover LEO-II)



Simple Type Theory / HOL

Simple Type Theory / HOL

- ▶ simple types $\alpha, \beta ::= \iota | o | \alpha \rightarrow \beta$ (additional base types μ_i)
- ▶ simple type theory / HOL defined by

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

- ▶ semantics well understood [Henkin50, Andrews72a/b, BenzmüllerEtAl04]
- Henkin semantics
- ▶ base logic of many (interactive) proof assistants:
Isabelle/HOL, HOL, HOL-light, PVS, OMEGA, ...
- ▶ (too) few ATPs so far \longrightarrow EU IIF Project THFTPTP

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	–	✓	<i>continuous</i> $(\lambda x. x)$
- predicates/sets/relations	–	✓	<i>reflexive</i> $(=)$



Multimodal Logics as Fragments of HOL

Multimodal Logics as Fragments of 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 $\iota \rightarrow o$: multimodal logic formulas

$$\begin{aligned}[\neg s] &= \lambda w_{i \cdot \bullet} \neg ([s] w) \\ [s \vee t] &= \lambda w_{i \cdot \bullet} [s] w \vee [t] w \\ [\Box_r s] &= \lambda w_{i \cdot \bullet} \forall y_{i \cdot \bullet} [r] w y \Rightarrow [s] y \\ [p] &= p_{\iota \rightarrow o}\end{aligned}$$

Related Work: [Gallin73], [Ohlbach88], [Carpenter98], [Merz99], [Brown05], [Hardt&Smolka07], [Kaminski&Smolka07]

Multimodal Logics as Fragments of 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 $\iota \rightarrow o$: multimodal logic formulas

$$\mid \neg \mid = \lambda s_{\iota \rightarrow o} \lambda w_{\iota} \neg (s w)$$

$$\mid \vee \mid = \lambda s_{\iota \rightarrow o} \lambda t_{\iota \rightarrow o} \lambda w_{\iota} s w \vee t w$$

$$\mid \Box \mid = \lambda r_{\iota \rightarrow \iota \rightarrow o} \lambda s_{\iota \rightarrow o} \lambda w_{\iota} \forall y_{\iota} r w y \Rightarrow s y$$

$$\mid p \mid = p_{\iota \rightarrow o}$$

$$\mid r \mid = r_{\iota \rightarrow \iota \rightarrow o}$$

Related Work: [Gallin73], [Ohlbach88], [Carpenter98], [Merz99], [Brown05], [Hardt&Smolka07], [Kaminski&Smolka07]

Encoding of Validity

$$\begin{aligned} |\mathbf{Mval} \ s_{l \rightarrow o}| &= \forall w_{l \cdot} s \ w \\ |\mathbf{Mval}| &= \lambda s_{l \rightarrow o} \cdot \forall w_{l \cdot} s \ w \end{aligned}$$

Local Definition Expansion

$$\begin{aligned} |\mathbf{Mval} \ \Box_r \ T| &= |\mathbf{Mval}| \ |\Box| \ |r| \ |T| \\ &=^{\beta\eta} \forall w_{l \cdot} \forall y_{l \cdot} r \ w \ y \Rightarrow T \end{aligned}$$

Encoding of Validity

$$\begin{aligned} |\mathbf{Mval} \ s_{l \rightarrow o}| &= \forall w_{l \cdot} s \ w \\ |\mathbf{Mval}| &= \lambda s_{l \rightarrow o} \cdot \forall w_{l \cdot} s \ w \end{aligned}$$

Local Definition Expansion

$$\begin{aligned} |\mathbf{Mval} \ \Box_r \ T| &= |\mathbf{Mval}| \ |\Box| \ |r| \ |T| \\ &=^{\beta\eta} \forall w_{l \cdot} \forall y_{l \cdot} r \ w \ y \Rightarrow T \end{aligned}$$

Encoding of Validity

$$\begin{aligned} |\mathbf{Mval} \ s_{l \rightarrow o}| &= \forall w_{l \bullet} s w \\ |\mathbf{Mval}| &= \lambda s_{l \rightarrow o} \bullet \forall w_{l \bullet} s w \end{aligned}$$

Local Definition Expansion

$$\begin{aligned} |\mathbf{Mval} \ \Box_r \ T| &= |\mathbf{Mval}| \ |\Box| \ |r| \ |T| \\ &=^{\beta\eta} \forall w_{l \bullet} \bullet \forall y_{l \bullet} r w y \Rightarrow T \end{aligned}$$

Even simpler: Reasoning within Multimodal Logics

Problem	LEO-II
$ \text{Mval } \Box_r \top $	0.025s
$ \text{Mval } \Box_r a \supset \Box_r a $	0.026s
$ \text{Mval } \Box_r a \supset \Box_s a $	–
$ \text{Mval } \Box_s (\Box_r a \supset \Box_r a) $	0.026s
$ \text{Mval } \Box_r (a \wedge b) \Leftrightarrow (\Box_r a \wedge \Box_r b) $	0.044s
$ \text{Mval } \Diamond_r (a \supset b) \supset \Box_r a \supset \Diamond_r b $	0.030s
$ \text{Mval } \neg \Diamond_r a \supset \Box_r (a \supset b) $	0.029s
$ \text{Mval } \Box_r b \supset \Box_r (a \supset b) $	0.026s
$ \text{Mval } (\Diamond_r a \supset \Box_r b) \supset \Box_r (a \supset b) $	0.027s
$ \text{Mval } (\Diamond_r a \supset \Box_r b) \supset (\Box_r a \supset \Box_r b) $	0.029s
$ \text{Mval } (\Diamond_r a \supset \Box_r b) \supset (\Diamond_r a \supset \Diamond_r b) $	0.030s

Example Proof: $\neg \text{Mval } \Box_s (\Box_r a \supset \Box_r a)$

Initialization of problem

$$\neg \text{Mval } \Box_s (\Box_r a \supset \Box_r a)$$

Definition expansion

$$\neg (\forall x_{l_1}. \forall y_{l_2}. \neg s x y \vee ((\neg (\forall u_{l_3}. \neg r y u \vee a u)) \vee (\forall v_{l_4}. \neg r y v \vee a v)))$$

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

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

Translation to FOL [Kerber94], [Hurd02], [MengPaulson04]

$$\begin{array}{ll} [\text{@}\cdots(\text{@}\cdots(s, x), y)]^T & [\text{@}\cdots(a, u)]^F \\ [\text{@}\cdots(\text{@}\cdots(r, y), u)]^T & [\text{@}\cdots(a, V)]^T \vee [\text{@}\cdots(\text{@}\cdots(r, y), V)]^F \end{array}$$

Example Proof: $\neg \text{Mval } \Box_s (\Box_r a \supset \Box_r a)$

Initialization of problem

$$\neg \text{Mval } \Box_s (\Box_r a \supset \Box_r a)$$

Definition expansion

$$\neg (\forall x_v. \forall y_v. \neg s x y \vee ((\neg (\forall u_v. \neg r y u \vee a u)) \vee (\forall v_v. \neg r y v \vee a v)))$$

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

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

Translation to FOL [Kerber94], [Hurd02], [MengPaulson04]

$$\begin{array}{ll} [\text{@}\cdots(\text{@}\cdots(s, x), y)]^T & [\text{@}\cdots(a, u)]^F \\ [\text{@}\cdots(\text{@}\cdots(r, y), u)]^T & [\text{@}\cdots(a, V)]^T \vee [\text{@}\cdots(\text{@}\cdots(r, y), V)]^F \end{array}$$

Example Proof: $\neg \text{Mval } \Box_s (\Box_r a \supset \Box_r a)$

Initialization of problem

$$\neg \text{Mval } \Box_s (\Box_r a \supset \Box_r a)$$

Definition expansion

$$\neg (\forall x_v. \forall y_v. \neg s x y \vee ((\neg (\forall u_v. \neg r y u \vee a u)) \vee (\forall v_v. \neg r y v \vee a v)))$$

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

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

Translation to FOL [Kerber94], [Hurd02], [MengPaulson04]

$$\begin{array}{ll} [\text{@}\cdots(\text{@}\cdots(s, x), y)]^T & [\text{@}\cdots(a, u)]^F \\ [\text{@}\cdots(\text{@}\cdots(r, y), u)]^T & [\text{@}\cdots(a, V)]^T \vee [\text{@}\cdots(\text{@}\cdots(r, y), V)]^F \end{array}$$

Example Proof: $\neg \text{Mval } \Box_s (\Box_r a \supset \Box_r a)$

Initialization of problem

$$\neg \text{Mval } \Box_s (\Box_r a \supset \Box_r a)$$

Definition expansion

$$\neg (\forall x_v. \forall y_v. \neg s x y \vee ((\neg (\forall u_v. \neg r y u \vee a u)) \vee (\forall v_v. \neg r y v \vee a v)))$$

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

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

Translation to FOL [Kerber94], [Hurd02], [MengPaulson04]

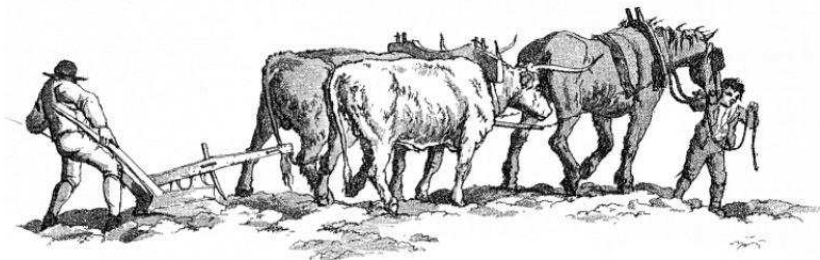
$$\begin{array}{ll} [\text{@}\cdots(\text{@}\cdots(s, x), y)]^T & [\text{@}\cdots(a, u)]^F \\ [\text{@}\cdots(\text{@}\cdots(r, y), u)]^T & [\text{@}\cdots(a, V)]^T \vee [\text{@}\cdots(\text{@}\cdots(r, y), V)]^F \end{array}$$

LEO-II

UNIVERSITY OF
CAMBRIDGE

UNIVERSITÄT
DES
SAARLANDES

An Effective Higher-Order Theorem Prover



LEO-II employs FO-ATPs:

E, Spass, Vampire

www.leoprover.org



Access Control Logics are
fragments of S4 and hence HOL

[GargAbadi08]:

A Modal Deconstruction of Access Control Logics

- ▶ ICL: Propositional Intuitionistic Logic + "says"

$(\text{Admin says deletefile1}) \supset \text{deletefile1}$

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

$\text{Admin says } ((\text{Bob says deletefile1}) \supset \text{deletefile1})$

Admin trusts Bob to decide whether file1 should be deleted.

$\text{Bob says deletefile1}$

Bob wants to delete file1.

deletefile1

Is deletion permitted?

Example I

[GargAbadi08]:

A Modal Deconstruction of Access Control Logics

- ▶ ICL: Propositional Intuitionistic Logic + "says"
- ▶ ICL^{\Rightarrow} : ICL + \Longrightarrow (speaks for)

$(\text{Admin says deletefile1}) \supset \text{deletefile1}$

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

$\text{Admin says } ((\text{Bob says deletefile1}) \supset \text{deletefile1})$

Admin trusts Bob to decide whether file1 should be deleted.

$\text{Bob says } (\text{Alice} \Longrightarrow \text{Bob})$

Bob delegates his authority to delete file1 to Alice

$\text{Alice says deletefile1}$

Alice wants to delete file1.

deletefile1

Is deletion permitted?

Example II

[GargAbadi08]:

A Modal Deconstruction of Access Control Logics

- ▶ ICL: Propositional Intuitionistic Logic + "says"
- ▶ ICL^{\Rightarrow} : ICL + \implies (speaks for)
- ▶ ICL^B : ICL + Boolean combinations of principals

$(\text{Admin says } \perp) \supset \text{deletefile1}$

Admin is trusted on deletefile1 and its consequences.

$\text{Admin says } ((\text{Bob } \supset \text{Admin}) \text{ says deletefile1})$

Admin further delegates this authority to Bob.

$\text{Bob says deletefile1}$

Bob wants to delete file1.

deletefile1

Is deletion permitted?

Example III

[GargAbadi08]:

A Modal Deconstruction of Access Control Logics

- ▶ ICL: Propositional Intuitionistic Logic + "says"
- ▶ ICL^{\Rightarrow} : ICL + \Longrightarrow (speaks for)
- ▶ ICL^B : ICL + Boolean combinations of principals

[GargAbadi08]:

A Modal Deconstruction of Access Control Logics

- ▶ ICL: Propositional Intuitionistic Logic + "says"
- ▶ ICL^{\Rightarrow} : ICL + \Longrightarrow (speaks for)
- ▶ ICL^B : ICL + Boolean combinations of principals

Sound and Complete Translations to Modal Logic S4

[GargAbadi08]:

A Modal Deconstruction of Access Control Logics

- ▶ ICL: Propositional Intuitionistic Logic + "says"
- ▶ ICL^{\Rightarrow} : ICL + \Longrightarrow (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 Logics as Fragments of S4 and HOL

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

Translation $[\cdot]$ (of Garg and Abadi) into S4

$$\begin{aligned} [p] &= \Box p \\ [s \wedge t] &= [s] \wedge [t] \\ [s \vee t] &= [s] \vee [t] \\ [s \supset t] &= \Box([s] \supset [t]) \\ [\top] &= \top \\ [\perp] &= \perp \\ [A \text{ says } s] &= \Box(A \vee [s]) \end{aligned}$$

Access Control Logics as Fragments of S4 and HOL

$$s, t ::= p \mid s \wedge t \mid s \vee t \mid s \supset t \mid \perp \mid \top \mid A \text{ says } s \mid s \Longrightarrow t$$

Translation $[\cdot]$ (of Garg and Abadi) into S4

$$\begin{aligned} [p] &= \Box p \\ [s \wedge t] &= [s] \wedge [t] \\ [s \vee t] &= [s] \vee [t] \\ [s \supset t] &= \Box([s] \supset [t]) \\ [\top] &= \top \\ [\perp] &= \perp \\ [A \text{ says } s] &= \Box(A \vee [s]) \\ [s \Longrightarrow t] &= \Box([s] \supset [t]) \end{aligned}$$

Access Control Logics as Fragments of S4 and HOL

$$s, t ::= p \mid s \wedge t \mid s \vee t \mid s \supset t \mid \perp \mid \top \mid A \text{ says } s \mid s \Longrightarrow t$$

Translation $\|\cdot\|$ to HOL

$$\begin{aligned} & |r| \quad (\text{we fix one single } r!!!) \\ \|p\| &= |\Box_r p| \\ \|A\| &= |A| \\ \|\wedge\| &= \lambda s. \lambda t. |s \wedge t| \\ \|\vee\| &= \lambda s. \lambda t. |s \vee t| \\ \|\supset\| &= \lambda s. \lambda t. |\Box(s \supset t)| \\ \|\top\| &= |\top| \\ \|\perp\| &= |\perp| \\ \|\text{says}\| &= \lambda A. \lambda s. |\Box_r (A \vee s)| \\ \|\Longrightarrow\| &= \lambda s. \lambda t. |\Box_r (s \supset t)| \end{aligned}$$

Access Control Logics as Fragments of S4 and HOL

$$s, t ::= p \mid s \wedge t \mid s \vee t \mid s \supset t \mid \perp \mid \top \mid A \text{ says } s \mid s \Longrightarrow t$$

Translation $\|\cdot\|$ to HOL

$$\begin{aligned} & r_{\iota \rightarrow \iota \rightarrow o} \quad (\text{we fix one single } r!!!) \\ \|p\| &= \lambda x_{\iota}. \forall y_{\iota}. r_{\iota \rightarrow \iota \rightarrow o} x y \Rightarrow p_{\iota \rightarrow o} Y \\ \|A\| &= a_{\iota \rightarrow o} \quad (\text{distinct from the } p_{\iota \rightarrow o}) \\ \|\wedge\| &= \lambda s_{\iota \rightarrow o}. \lambda t_{\iota \rightarrow o}. \lambda w_{\iota}. s w \wedge t w \\ \|\vee\| &= \lambda s_{\iota \rightarrow o}. \lambda t_{\iota \rightarrow o}. \lambda w_{\iota}. s w \vee t w \\ \|\supset\| &= \lambda s_{\iota \rightarrow o}. \lambda t_{\iota \rightarrow o}. \lambda w_{\iota}. \forall y_{\iota}. r w y \Rightarrow (s y \Rightarrow t y) \\ \|\top\| &= \lambda s_{\iota \rightarrow o}. \top \\ \|\perp\| &= \lambda s_{\iota \rightarrow o}. \perp \\ \|\text{says}\| &= \lambda A_{\iota \rightarrow o}. \lambda s_{\iota \rightarrow o}. \lambda w_{\iota}. \forall y_{\iota}. r w y \Rightarrow (A y \vee s y) \\ \|\Longrightarrow\| &= \lambda s_{\iota \rightarrow o}. \lambda t_{\iota \rightarrow o}. \lambda w_{\iota}. \forall y_{\iota}. r w y \Rightarrow (s y \Rightarrow t y) \end{aligned}$$

Notion of Validity

$$\text{ICLval} = \text{Mval}$$

Addition of Modal Logic Axioms for S4

$$\begin{aligned} & \forall p_{t \rightarrow o}. |\text{Mval} \Box_r p \supset p| \\ & \forall p_{t \rightarrow o}. |\text{Mval} \Box_r p \supset \Box_r \Box_r p| \end{aligned}$$

Soundness and Completeness of Embedding

Proof: see paper; employs transformation from Kripke models into corresponding Henkin models and vice versa; combines this with results of [GargAbadi08]

Notion of Validity

$$\text{ICLval} = \text{Mval}$$

Addition of Modal Logic Axioms for S4

$$\begin{aligned} & \forall p_{\ell \rightarrow o}. |\text{Mval} \Box_r p \supset p| \\ & \forall p_{\ell \rightarrow o}. |\text{Mval} \Box_r p \supset \Box_r \Box_r p| \end{aligned}$$

Soundness and Completeness of Embedding

Proof: see paper; employs transformation from Kripke models into corresponding Henkin models and vice versa; combines this with results of [GargAbadi08]

Notion of Validity

$$\text{ICLval} = \text{Mval}$$

Addition of Modal Logic Axioms for S4

$$\begin{aligned} & \forall p_{\ell \rightarrow o}. |\text{Mval} \Box_r p \supset p| \\ & \forall p_{\ell \rightarrow o}. |\text{Mval} \Box_r p \supset \Box_r \Box_r p| \end{aligned}$$

Soundness and Completeness of Embedding

Proof: see paper; employs transformation from Kripke models into corresponding Henkin models and vice versa; combines this with results of [GargAbadi08]

Example 1 (from [GargAbadi08]):

ICLval (Admin says deletefile1) \supset deletefile1

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

ICLval Admin says ((Bob says deletefile1) \supset deletefile1)

Admin trusts Bob to decide whether file1 should be deleted.

ICLval Bob says deletefile1

Bob wants to delete file1.

ICLval deletefile1

Is deletion permitted?

Example 1 (from [GargAbadi08]):

$\| \text{ICLval (Admin says deletefile1)} \supset \text{deletefile1} \|$

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

$\| \text{ICLval Admin says ((Bob says deletefile1)} \supset \text{deletefile1}) \|$

Admin trusts Bob to decide whether file1 should be deleted.

$\| \text{ICLval Bob says deletefile1} \|$

Bob wants to delete file1.

$\| \text{ICLval deletefile1} \|$

Is deletion permitted?

Example 1 (from [GargAbadi08]):

$\| \text{ICLval (Admin says deletefile1)} \supset \text{deletefile1} \|$

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

$\| \text{ICLval Admin says ((Bob says deletefile1)} \supset \text{deletefile1}) \|$

Admin trusts Bob to decide whether file1 should be deleted.

$\| \text{Mval } \Box_r (\text{Bob} \vee \Box_r \text{deletefile1}) \|$

Bob wants to delete file1.

$\| \text{ICLval deletefile1} \|$

Is deletion permitted?

Example I (from [GargAbadi08]):

$\| \text{ICLval (Admin says deletefile1)} \supset \text{deletefile1} \|$

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

$\| \text{ICLval Admin says ((Bob says deletefile1)} \supset \text{deletefile1}) \|$

Admin trusts Bob to decide whether file1 should be deleted.

$\forall w_i. \forall y_i. r w y \Rightarrow (\text{Bob } y \vee \forall u_i. r w u \Rightarrow \text{deletefile1 } u)$

Bob wants to delete file1.

$\| \text{ICLval deletefile1} \|$

Is deletion permitted?

LEO-II: 0.301 seconds

- ▶ Example I: 0.301 seconds
- ▶ Example II (ICL^{\Rightarrow}): 0.503 seconds
- ▶ Example III ($ICLB$): 0.077 seconds

Also possible: reasoning about meta-properties

- ▶ ICL^{\Rightarrow} can be expressed in ICL^B : 0.073 seconds

Exp.: Access Control Logic in HOL

ICL:

Name	Problem	LEO (s)
unit	$\{R, T\} \models^{HOL} \ \text{ICLval } s \supset (A \text{ says } s)\ $	0.053
cuc	$\{R, T\} \models^{HOL} \ \text{ICLval } (A \text{ says } (s \supset t)) \supset (A \text{ says } s) \supset (A \text{ says } t)\ $	0.167
idem	$\{R, T\} \models^{HOL} \ \text{ICLval } (A \text{ says } A \text{ says } s) \supset (A \text{ says } s)\ $	0.058
unit ^K	$\models^{HOL} \ \text{ICLval } s \supset (A \text{ says } s)\ $	–
cuc ^K	$\models^{HOL} \ \text{ICLval } (A \text{ says } (s \supset t)) \supset (A \text{ says } s) \supset (A \text{ says } t)\ $	–
idem ^K	$\models^{HOL} \ \text{ICLval } (A \text{ says } A \text{ says } s) \supset (A \text{ says } s)\ $	–

R, T : reflexivity and transitivity axioms for S4 as seen before

Exp.: Access Control Logic in HOL

ICL \Rightarrow :

Name	Problem	LEO (s)
refl	$\{R, T\} \models^{HOL} \ \text{ICLval } A \Rightarrow A\ $	0.059
trans	$\{R, T\} \models^{HOL} \ \text{ICLval } (A \Rightarrow B) \supset (B \Rightarrow C) \supset (A \Rightarrow C)\ $	0.083
sp.-for	$\{R, T\} \models^{HOL} \ \text{ICLval } (A \Rightarrow B) \supset (A \text{ says } s) \supset (B \text{ says } s)\ $	0.107
handoff	$\{R, T\} \models^{HOL} \ \text{ICLval } (B \text{ says } (A \Rightarrow B)) \supset (A \Rightarrow B)\ $	0.075
refl ^K	$\models^{HOL} \ \text{ICLval } A \Rightarrow A\ $	0.034
trans ^K	$\models^{HOL} \ \text{ICLval } (A \Rightarrow B) \supset (B \Rightarrow C) \supset (A \Rightarrow C)\ $	-
sp.-for ^K	$\models^{HOL} \ \text{ICLval } (A \Rightarrow B) \supset (A \text{ says } s) \supset (B \text{ says } s)\ $	-
handoff ^K	$\models^{HOL} \ \text{ICLval } (B \text{ says } (A \Rightarrow B)) \supset (A \Rightarrow B)\ $	-

R, T : reflexivity and transitivity axioms as for S4 seen before

Exp.: Access Control Logic in HOL

ICL^B:

Name	Problem	LEO (s)
trust	$\{R, T\} \models^{HOL} \ \text{ICLval } (\perp \text{ says } s) \supset s\ $	0.058
untrust	$\{R, T, \ \text{ICLval } A \equiv \top\ \} \models^{HOL} \ \text{ICLval } A \text{ says } \perp\ $	0.046
cuc'	$\{R, T\} \models^{HOL} \ \text{ICLval } ((A \supset B) \text{ says } s) \supset (A \text{ says } s) \supset (B \text{ says } s)\ $	0.200
trust ^K	$\models^{HOL} \ \text{ICLval } (\perp \text{ says } s) \supset s\ $	-
untrust ^K	$\{\ \text{ICLval } A \equiv \top\ \} \models^{HOL} \ \text{ICLval } A \text{ says } \perp\ $	0.055
cuc' ^K	$\models^{HOL} \ \text{ICLval } ((A \supset B) \text{ says } s) \supset (A \text{ says } s) \supset (B \text{ says } s)\ $	-

R, T : reflexivity and transitivity axioms for S4 as seen before

- ▶ Prominent Access Control Logics are fragments of HOL
- ▶ Interactive and automated HOL provers can generally be applied for reasoning in and **about** these logics
- ▶ Challenge: How good does approach scale?
- ▶ Examples submitted to THFTPTP

Ongoing and Future Research

- ▶ THFTPTP infrastructure
- ▶ Improvement of LEO-II – make it scale for larger examples
- ▶ Combination of different logics
- ▶ Formal verification of approach e.g. in Isabelle/HOL



THFTPTP

(EU grant THFTPTP – PIIF-GA-2008-219982)

Thanks to hard working Geoff Sutcliffe

- ▶ THF syntax for HOL
- ▶ library for HOL (> 2700 problems)
- ▶ tools for HOL
(parser, type checker, pretty printer, ...)
- ▶ integrated HOL ATPs: IsabelleP, TPS, LEO-II
- ▶ integrated HOL model generator: IsabelleM
- ▶ SystemOnTPTP online interface

THFTPTP – Progress in ATP for HOL

- ALG** higher-order abstract syntax
- GRA** Ramsey numbers (several open)
- LCL** modal logic
- NUM** Landau's Grundlagen
- PUZ** puzzles
- SET/SEU** set theory, dependently typed set theory, binary relations
- SWV** security, access control logic
- SYN/SYO** simple test problems

	ALG	GRA	LCL	NUM	PUZ	SE?	SWV	SY?	Total	Unique
Problems	50	93	61	221	5	749	37	59	1275	
THM/UNS	50	25	51	221	5	746	25	47	1170	
CSA/SAT	0	0	10	0	0	3	5	11	29	
LEO-II 0.99a	34	0	48	181	3	401	19	42	725	127
IsabelleP 2008	0	0	0	197	5	361	1	30	594	74
TPS 3.0	10	0	40	150	3	285	9	35	532	6
Any	32	0	50	203	5	490	20	52	843	207
All	0	0	0	134	2	214	0	22	372	
None	18	93	12	18	0	259	17	15	432	
IsabelleM 2008	0	0	1	0	0	0	0	8	9	



LEO-II

(EPSRC grant EP/D070511/1 at Cambridge University)

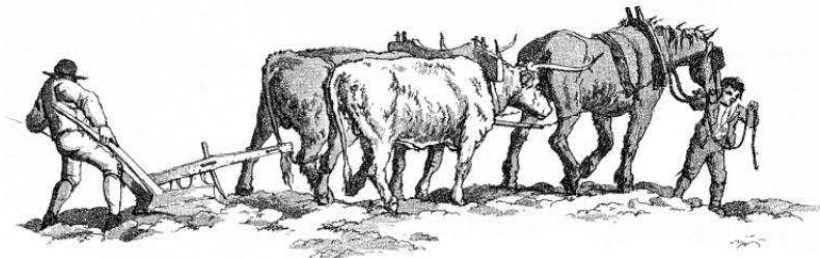
Thanks to Larry Paulson

LEO-II

UNIVERSITY OF
CAMBRIDGE

UNIVERSITÄT
DES
SAARLANDES

An Effective Higher-Order Theorem Prover



LEO-II employs FO-ATPs:

E, Spass, Vampire

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