Automating Access Control Logics in Simple Type Theory with LEO-II[1]

Christoph Benzmüller

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

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

[1] This 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 $\alpha, \beta ::= \iota | o | \alpha \to \beta$ (additional base types μ_i)
- simple type theory / HOL defined by

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

- 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

Simple Type Theory / HOL – Expressivity

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	–	✓	$\mathit{reflexive}(=)$

Multimodal Logics as Fragments of HOL

Multimodal Logics as Fragments of 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

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

Related Work: [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 $\iota \to o$: multimodal logic formulas

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

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

Encoding of Validity

$$\lfloor \texttt{Mval}\ s_{\iota\to o} \rfloor \quad = \quad \forall w_\iota \centerdot s\ w$$

$$\lfloor \texttt{Mval} \rfloor \quad = \quad \lambda s_{\iota\to o} \centerdot \forall w_\iota \centerdot s\ w$$

Local Definition Expansion

$$\lfloor \texttt{Mval}\ \Box_r\ \top \rfloor \quad = \quad \lfloor \texttt{Mval} \rfloor\, \lfloor \Box \rfloor\, \lfloor r \rfloor\, \lfloor \top \rfloor$$

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

(Normal) Multimodal Logic in HOL

Encoding of Validity

$$\begin{aligned} |\texttt{Mval}\ s_{\iota\to o}| &= \forall w_\iota\ldotp s\ w \\ |\texttt{Mval}| &= \lambda s_{\iota\to o}\ldotp \forall w_\iota\ldotp s\ w \end{aligned}$$

Local Definition Expansion

$$\begin{aligned} |\texttt{Mval}\ \Box_r\ \top| &= |\texttt{Mval}|\ |\Box|\ |r|\ |\top| \\ &=^{\beta\eta} \forall w_\iota\ldotp \forall y_\iota\ldotp r\ w\ y \Rightarrow \top \end{aligned}$$

(Normal) Multimodal Logic in HOL

Encoding of Validity

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

Local Definition Expansion

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

Even simpler: Reasoning within Multimodal Logics

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

Example Proof: $|\mathtt{Mval}\ \Box_s (\Box_r\, a \supset \Box_r\, a)|$

Initialization of problem

$$\neg|\mathtt{Mval}\ \Box_s (\Box_r\, a \supset \Box_r\, a)|$$

Definition expansion

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

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]

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

Example Proof: $|\mathtt{Mval}\ \Box_s (\Box_r a \supset \Box_r a)|$

Initialization of problem

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

Definition expansion

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

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]

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

Example Proof: $|\texttt{Mval}\ \Box_s (\Box_r a \supset \Box_r a)|$

Initialization of problem

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

Definition expansion

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

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]

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

Example Proof: $|\texttt{Mval}\ \Box_s\,(\Box_r\, a \supset \Box_r\, a)|$

Initialization of problem

$$\neg|\texttt{Mval}\ \Box_s\,(\Box_r\, a \supset \Box_r\, a)|$$

Definition expansion

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

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]

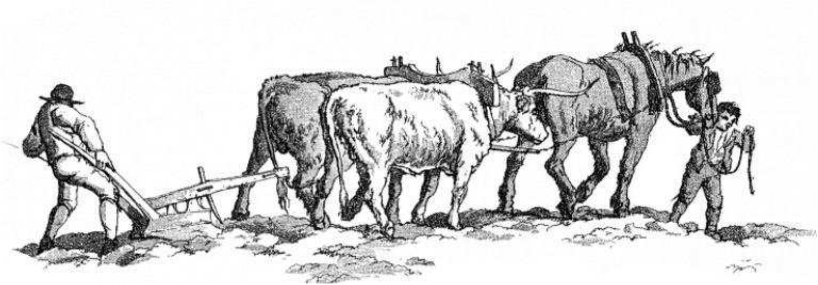
$$\begin{array}{ll} [@^{\cdots}(@^{\cdots}(s,x),y)]^T & [@^{\cdots}(a,u)]^F \\ [@^{\cdots}(@^{\cdots}(r,y),u)]^T & [@^{\cdots}(a,V)]^T \vee [@^{\cdots}(@^{\cdots}(r,y),V)]^F \end{array}$$

LEO-II

An Effective Higher-Order Theorem Prover

UNIVERSITY OF CAMBRIDGE

UNIVERSITÄT DES SAARLANDES



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"

$\texttt{(Admin says deletefile1)} \supset \texttt{deletefile1}$
If Admin says that file1 should be deleted, then this must be the case.

$\texttt{Admin says ((Bob says deletefile1)} \supset \texttt{deletefile1)}$
Admin trusts Bob to decide whether file1 should be deleted.

$\texttt{Bob says deletefile1}$
Bob wants to delete file1.

$\texttt{deletefile1}$
Is deletion permitted?

Example I

[GargAbadi08]: A Modal Deconstruction of Access Control Logics

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

`(Admin says deletefile1)` $\supset$ `deletefile1`
If Admin says that file1 should be deleted, then this must be the case.

`Admin says ((Bob says deletefile1)` $\supset$ `deletefile1)`
Admin trusts Bob to decide whether file1 should be deleted.

`Bob says (Alice` $\Longrightarrow$ `Bob)`
Bob delegates his authority to delete file1 to Alice

`Alice says deletefile1`
Alics wants to delete file1.

`deletefile1`
Is deletion permitted?

Example II

[GargAbadi08]: A Modal Deconstruction of Access Control Logics

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

$(\texttt{Admin says } \bot) \supset \texttt{deletefile1}$
Admin is trusted on deletefile1 and its consequences.

$\texttt{Admin says } ((\texttt{Bob} \supset \texttt{Admin}) \texttt{ says deletefile1})$
Admin further delegates this authority to Bob.

$\texttt{Bob says deletefile1}$
Bob wants to delete file1.

$\texttt{deletefile1}$
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"
- $\mathrm{ICL}^{\Rightarrow}$: 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

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

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

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

Access Control Logics as Fragments of S4 and HOL

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

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

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

Access Control Logics as Fragments of S4 and HOL

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

Translation $\|.\|$ to HOL

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

Access Control Logics as Fragments of S4 and HOL

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

Translation $\|.\|$ to HOL

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

Notion of Validity

$$\texttt{ICLval} = \texttt{Mval}$$

Addition of Modal Logic Axioms for S4

$$\forall p_{\iota\to o}.|\texttt{Mval}\ \Box_r\, p \supset p|$$

$$\forall p_{\iota\to o}.|\texttt{Mval}\ \Box_r\, p \supset \Box_r\, \Box_r\, p|$$

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

$$\texttt{ICLval} = \texttt{Mval}$$

Addition of Modal Logic Axioms for S4

$$\forall p_{\iota \to o}. \left| \texttt{Mval}\ \Box_r\ p \supset p \right|$$

$$\forall p_{\iota \to o}. \left| \texttt{Mval}\ \Box_r\ p \supset\ \Box_r\ \Box_r\ p \right|$$

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]

Access Control Logics as Fragments of S4 and HOL

Notion of Validity

$$\texttt{ICLval} = \texttt{Mval}$$

Addition of Modal Logic Axioms for S4

$$\forall p_{\iota \to o}. \lfloor \texttt{Mval}\ \Box_r\ p \supset p \rfloor$$

$$\forall p_{\iota \to o}. \lfloor \texttt{Mval}\ \Box_r\ p \supset\ \Box_r\ \Box_r\ p \rfloor$$

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 I (from [GargAbadi08]):

`ICLval (Admin says deletefile1) ⊃ deletefile1`
If Admin says that file1 should be deleted, then this must be the case.

`ICLval Admin says ((Bob says deletefile1) ⊃ 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 I (from [GargAbadi08]):

$\|$`ICLval (Admin says deletefile1) ⊃ deletefile1`$\|$
If Admin says that file1 should be deleted, then this must be the case.

$\|$`ICLval Admin says ((Bob says deletefile1) ⊃ 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 I (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.

$|$Mval $\Box_r$ (Bob $\vee$ $\Box_r$ deletefile1)$|$
Bob wants to delete file1.

$\|$ICLval deletefile1$\|$
Is deletion permitted?

Example I (from [GargAbadi08]):

$\|\texttt{ICLval (Admin says deletefile1)} \supset \texttt{deletefile1}\|$
If Admin says that file1 should be deleted, then this must be the case.

$\|\texttt{ICLval Admin says ((Bob says deletefile1)} \supset \texttt{deletefile1)}\|$
Admin trusts Bob to decide whether file1 should be deleted.

$\forall w_\iota \blacksquare \forall y_\iota \blacksquare r\, w\, y \Rightarrow (Bob\, y \vee \forall u_\iota \blacksquare r\, w\, u \Rightarrow deletefile1\ u)$
Bob wants to delete file1.

$\|\texttt{ICLval deletefile1}\|$
Is deletion permitted?

LEO-II: 0.301 seconds

More Examples from [GargAbadi08]

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

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

Exp.: Access Control Logic in HOL

$\mathrm{ICL}^{\Rightarrow}$:

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

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

Exp.: Access Control Logic in HOL

ICL^B:

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

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

Conclusion

- 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

THFTPTP – Progress in ATP for HOL

- 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

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

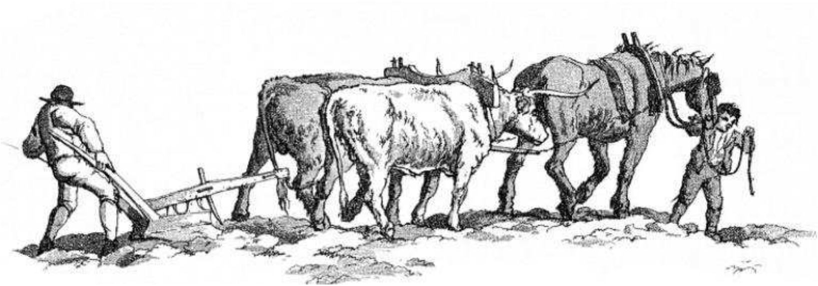
Thanks to Larry Paulson

LEO-II

An Effective Higher-Order Theorem Prover

UNIVERSITY OF CAMBRIDGE

UNIVERSITÄT DES SAARLANDES



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

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