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

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)
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 $\mu_i$)
- simple type theory / HOL defined by

  $$s, t ::= p_\alpha \mid X_\alpha \mid (\lambda X_\alpha \cdot s_\beta)_{\alpha \to \beta} \mid (s_{\alpha \to \beta} \cdot t_\alpha)_{\beta} \mid (\neg o \to o \cdot s_o)_{o} \mid (s_o \lor o \to o \cdot t_o)_{o} \mid (\Pi (\alpha \to o) \to o \cdot t_{\alpha \to o})_{o}$$

- 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 THF TPTP
<table>
<thead>
<tr>
<th>Property</th>
<th>FOL</th>
<th>HOL</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quantification over</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- individuals</td>
<td>✓</td>
<td>✓</td>
<td>$\forall x . P(F(x))$</td>
</tr>
<tr>
<td>- functions</td>
<td>-</td>
<td>✓</td>
<td>$\forall F . P(F(x))$</td>
</tr>
<tr>
<td>- predicates/sets/relations</td>
<td>-</td>
<td>✓</td>
<td>$\forall P . P(F(x))$</td>
</tr>
<tr>
<td>Unnamed</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- functions</td>
<td>-</td>
<td>✓</td>
<td>$(\lambda x . x)$</td>
</tr>
<tr>
<td>- predicates/sets/relations</td>
<td>-</td>
<td>✓</td>
<td>$(\lambda x . x \neq 2)$</td>
</tr>
<tr>
<td>Statements about</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>- functions</td>
<td>-</td>
<td>✓</td>
<td>$continuous(\lambda x . x)$</td>
</tr>
<tr>
<td>- predicates/sets/relations</td>
<td>-</td>
<td>✓</td>
<td>$reflexive(=)$</td>
</tr>
</tbody>
</table>
Multimodal Logics as Fragments of HOL
Multimodal Logics as Fragments of HOL

\[ s, t ::= p \mid \neg s \mid s \lor t \mid \Box_r s \]

Simple, Straightforward Encoding

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

\[
\begin{align*}
[\neg s] &= \lambda w. \neg([s]w) \\
[s \lor t] &= \lambda w.[s]w \lor [t]w \\
[\Box_r s] &= \lambda w. \forall y. [r]w y \Rightarrow [s]y \\
[p] &= p_{\iota \rightarrow o}
\end{align*}
\]

Related Work: [Gallin73], [Ohlbach88], [Carpenter98], [Merz99], [Brown05], [Hardt&Smolka07], [Kaminski&Smolka07]
Multimodal Logics as Fragments of HOL

\[ s, t ::= p | \neg s | s \lor t | \square_r s \]

Simple, Straightforward Encoding

- base type \( \iota \):
  - set of possible worlds

- (certain) terms of type \( \iota \rightarrow o \):
  - multimodal logic formulas

\[
\begin{align*}
\neg & \quad = \lambda s_{\iota \rightarrow o} \lambda w_{\iota} \neg (s \ w) \\
\lor & \quad = \lambda s_{\iota \rightarrow o} \lambda t_{\iota \rightarrow o} \lambda w_{\iota} s \ w \lor t \ w \\
\square & \quad = \lambda r_{\iota \rightarrow \iota \rightarrow o} \lambda s_{\iota \rightarrow o} \lambda w_{\iota} \forall y_{\iota} r \ w \ y \Rightarrow s \ y \\
p & \quad = p_{\iota \rightarrow o} \\
r & \quad = r_{\iota \rightarrow \iota \rightarrow o}
\end{align*}
\]

Related Work: [Gallin73], [Ohlbach88], [Carpenter98], [Merz99], [Brown05], [Hardt&Smolka07], [Kaminski&Smolka07]
(Normal) Multimodal Logic in HOL

Encoding of Validity

\[ |\text{Mval} s_{l \rightarrow o}| = \forall w_{l \cdot s} w \]
\[ |\text{Mval}| = \lambda s_{l \rightarrow o} \forall w_{l \cdot s} w \]

Local Definition Expansion

\[ |\text{Mval} \Box r T| = |\text{Mval}| \Box |r| |T| \]
\[ = \beta \eta \forall w_{l \cdot} \forall y_{l \cdot} r w y \Rightarrow T \]
Encoding of Validity

\[ |M\text{val} \ xu \xrightarrow{\lambda} o| = \forall w. \xrightarrow{l} s \ w \]
\[ |M\text{val}| = \lambda s \xrightarrow{l} o. \forall w. \xrightarrow{l} s \ w \]

Local Definition Expansion

\[ |M\text{val} \boxdot r \top| = |M\text{val}| |\boxdot | r | \top| \]
\[ = \beta\eta \forall w. \forall y. r \ w \ y \Rightarrow \top \]
Encoding of Validity

\[ |Mval s \to o | = \forall w s w \]
\[ |Mval | = \lambda s \to o \cdot \forall w s w \]

Local Definition Expansion

\[ |Mval \Box r \top | = |Mval| \Box |r| \top | \]
\[ =_{\beta\eta} \forall w \forall y r w y \Rightarrow \top \]
<table>
<thead>
<tr>
<th>Problem</th>
<th>LEO-II</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\text{Mval } \Box_r \top)</td>
<td>0.025s</td>
</tr>
<tr>
<td>(\text{Mval } \Box_r a \supset \Box_r a)</td>
<td>0.026s</td>
</tr>
<tr>
<td>(\text{Mval } \Box_r a \supset \Box_s a)</td>
<td>–</td>
</tr>
<tr>
<td>(\text{Mval } \Box_s (\Box_r a \supset \Box_r a))</td>
<td>0.026s</td>
</tr>
<tr>
<td>(\text{Mval } \Box_r (a \land b) \iff (\Box_r a \land \Box_r b))</td>
<td>0.044s</td>
</tr>
<tr>
<td>(\text{Mval } \Diamond_r (a \supset b) \supset \Box_r a \supset \Diamond_r b)</td>
<td>0.030s</td>
</tr>
<tr>
<td>(\text{Mval } \neg \Diamond_r a \supset \Box_r (a \supset b))</td>
<td>0.029s</td>
</tr>
<tr>
<td>(\text{Mval } \Box_r b \supset \Box_r (a \supset b))</td>
<td>0.026s</td>
</tr>
<tr>
<td>(\text{Mval } (\Diamond_r a \supset \Box_r b) \supset \Box_r (a \supset b))</td>
<td>0.027s</td>
</tr>
<tr>
<td>(\text{Mval } (\Diamond_r a \supset \Box_r b) \supset (\Box_r a \supset \Box_r b))</td>
<td>0.029s</td>
</tr>
<tr>
<td>(\text{Mval } (\Diamond_r a \supset \Box_r b) \supset (\Diamond_r a \supset \Diamond_r b))</td>
<td>0.030s</td>
</tr>
</tbody>
</table>
Example Proof: $\neg \text{Mval } \square_s (\square_r a \supset \square_r a)$

Initialization of problem

$$\neg \text{Mval } \square_s (\square_r a \supset \square_r a)$$

Definition expansion

$$\neg (\forall x_i \forall y_i \neg s x y \lor ((\neg (\forall u_i \neg r y u \lor a u)) \lor (\forall v_i \neg r y v \lor a v)))$$

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

\[
\begin{align*}
  s x y & \quad \neg a u \\
  r y u & \quad a V \lor \neg r y V
\end{align*}
\]

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

\[
\begin{align*}
  [\circ\circ\circ(\circ\circ\circ(s, x), y)]^T & \quad [\circ\circ\circ(a, u)]^F \\
  [\circ\circ\circ(\circ\circ\circ(r, y), u)]^T & \quad [\circ\circ\circ(a, V)]^T \lor [\circ\circ\circ(\circ\circ\circ(r, y), V)]^F
\end{align*}
\]
Example Proof: $\neg Mval \Box_s (\Box_r a \supseteq \Box_r a)$

Initialization of problem

$\neg Mval \Box_s (\Box_r a \supseteq \Box_r a)$

Definition expansion

$\neg (\forall x. \forall y. \neg s \land y \lor (\neg (\forall u. \neg r \land y \lor a u) \lor (\forall v. \neg r \land y \lor a v)))$

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

$s \land x \land y \quad \neg a \land u$

$r \land y \land u \quad a \lor V \lor \neg r \land y \land V$

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

$[\Box \Box (\Box \Box (s, x), y)]^T$ $[\Box \Box (a, u)]^F$

$[\Box \Box (\Box \Box (r, y), u)]^T$ $[\Box \Box (a, V)]^T \lor [\Box \Box (\Box \Box (r, y), V)]^F$
Example Proof: $\vdash \text{Mval } \Box_s (\Box_r a \supset \Box_r a)$

Initialization of problem

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

Definition expansion

$$\neg (\forall x. \forall y. \neg s x y \lor ((\neg (\forall u. \neg r y u \lor a u)) \lor (\forall v. \neg r y v \lor a v)))$$

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

$$s x y \quad \neg a u$$
$$r y u \quad a V \lor \neg r y V$$

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

$$[@\cdots (@\cdots (s, x), y)]^T \quad [@\cdots (a, u)]^F$$
$$[@\cdots (@\cdots (r, y), u)]^T \quad [@\cdots (a, V)]^T \lor [@\cdots (@\cdots (r, y), V)]^F$$
Example Proof: $\neg \text{Mval } □_s (□_r a \supset □_r a)$

Initialization of problem

$\neg \text{Mval } □_s (□_r a \supset □_r a)$

Definition expansion

$\neg (\forall x_t \cdot \forall y_t \cdot \neg s x y \lor ((\neg (\forall u_t \cdot \neg r y u \lor a u)) \lor (\forall v_t \cdot \neg r y v \lor a v)))$

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

$s x y \quad \neg a u$

$r y u \quad a V \lor \neg r y V$

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

$[@\cdots (@\cdots(s, x), y)]^T \quad [\cdot @\cdots (a, u)]^F$

$[@\cdots (@\cdots(r, y), u)]^T \quad [\cdot @\cdots (a, V)]^T \lor [\cdot @\cdots (@\cdots(r, y), V)]^F$
LEO-II employs FO-ATPs: E, Spass, Vampire

www.leoprover.org
Access Control Logics are fragments of S4 and hence HOL
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.

\[\text{deletefile1}\]
Is deletion permitted?

Example 1
A Modal Deconstruction of Access Control Logics

- ICL: Propositional Intuitionistic Logic + "says"
- ICL ⇒: ICL + → (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 ((Bob says deletefile1) \supset \text{deletefile1})}
Admin trusts Bob to decide whether file1 should be deleted.

\text{Bob says (Alice ⇒ Bob)}
Bob delegates his authority to delete file1 to Alice

\text{Alice says deletefile1}
Alics wants to delete file1.

\text{deletefile1}
Is deletion permitted?

Example II
[GargAbadi08]: A Modal Deconstruction of Access Control Logics

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

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

Is deletion permitted?

Example III
[GargAbadi08]: A Modal Deconstruction of Access Control Logics

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

Sound and Complete Translations to Modal Logic S4
Access Control Logics

[GargAbadi08]:
A Modal Deconstruction of Access Control Logics

- ICL: Propositional Intuitionistic Logic + "says"
- ICL⇒: ICL + \(\implies\) (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
\[ s, t ::= p | s \land t | s \lor t | s \supset t | \bot | \top | A \text{ says } s \]

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

\[
\begin{align*}
[p] &= \Box p \\
[s \land t] &= [s] \land [t] \\
[s \lor t] &= [s] \lor [t] \\
[s \supset t] &= \Box ([s] \supset [t]) \\
[\top] &= \top \\
[\bot] &= \bot \\
[A \text{ says } s] &= \Box (A \lor [s])
\end{align*}
\]
Access Control Logics as Fragments of S4 and HOL

\[ s, t ::= p \mid s \land t \mid s \lor t \mid s \supset t \mid \bot \mid \top \mid A \ says \ s \mid s \rightarrow t \]

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

\[
\begin{align*}
\llbracket p \rrbracket & = \Box p \\
\llbracket s \land t \rrbracket & = \llbracket s \rrbracket \land \llbracket t \rrbracket \\
\llbracket s \lor t \rrbracket & = \llbracket s \rrbracket \lor \llbracket t \rrbracket \\
\llbracket s \supset t \rrbracket & = \Box (\llbracket s \rrbracket \supset \llbracket t \rrbracket) \\
\llbracket \top \rrbracket & = \top \\
\llbracket \bot \rrbracket & = \bot \\
\llbracket A \ says \ s \rrbracket & = \Box (A \lor \llbracket s \rrbracket) \\
\llbracket s \rightarrow t \rrbracket & = \Box (\llbracket s \rrbracket \supset \llbracket t \rrbracket)
\end{align*}
\]
Access Control Logics as Fragments of S4 and HOL

\[ s, t ::= p \mid s \land t \mid s \lor t \mid s \supset t \mid \bot \mid T \mid A \text{ says } s \mid s \multimap t \]

Translation \[\| . \|\] to HOL

\[
\begin{align*}
\| r \| & \quad \text{(we fix one single } r!!!) \\
\| p \| &= \| \Box_r p \| \\
\| A \| &= \| A \| \\
\| \land \| &= \lambda s. \lambda t. \| s \land t \| \\
\| \lor \| &= \lambda s. \lambda t. \| s \lor t \| \\
\| \supset \| &= \lambda s. \lambda t. \| \Box (s \supset t) \| \\
\| T \| &= \| T \| \\
\| \bot \| &= \| \bot \| \\
\| \text{says} \| &= \lambda A. \lambda s. \| \Box_r (A \lor s) \| \\
\| \multimap \| &= \lambda s. \lambda t. \| \Box_r (s \supset t) \|
\end{align*}
\]
$s, t ::= p \mid s \land t \mid s \lor t \mid s \supset t \mid \bot \mid T \mid A \text{ says } s \mid s \implies t$

Translation $\parallel . \parallel$ to HOL

| $\parallel p \parallel$ | $= \lambda x_l \cdot \forall y_l . r_{l \rightarrow o} x y \Rightarrow p_{l \rightarrow o} Y$ |
| $\parallel A \parallel$ | $= a_{l \rightarrow o}$ (distinct from the $p_{l \rightarrow o}$) |
| $\parallel \land \parallel$ | $= \lambda s_{l \rightarrow o} \cdot \lambda t_{l \rightarrow o} \cdot \lambda w_{l \rightarrow o} . s w \land t w$ |
| $\parallel \lor \parallel$ | $= \lambda s_{l \rightarrow o} \cdot \lambda t_{l \rightarrow o} \cdot \lambda w_{l \rightarrow o} . s w \lor t w$ |
| $\parallel \supset \parallel$ | $= \lambda s_{l \rightarrow o} \cdot \lambda t_{l \rightarrow o} \cdot \lambda w_{l \rightarrow o} . \forall y_l . r w y \Rightarrow (s y \Rightarrow t y)$ |
| $\parallel T \parallel$ | $= \lambda s_{l \rightarrow o} . \top$ |
| $\parallel \bot \parallel$ | $= \lambda s_{l \rightarrow o} . \bot$ |
| $\parallel \text{ says} \parallel$ | $= \lambda A_{l \rightarrow o} \cdot \lambda s_{l \rightarrow o} \cdot \lambda w_{l \rightarrow o} . \forall y_l . r w y \Rightarrow (A y \lor s y)$ |
| $\parallel \implies \parallel$ | $= \lambda s_{l \rightarrow o} \cdot \lambda t_{l \rightarrow o} \cdot \lambda w_{l \rightarrow o} . \forall y_l . r w y \Rightarrow (s y \Rightarrow t y)$ |
Notion of Validity

ICLval = Mval

Addition of Modal Logic Axioms for S4

\[ \forall p \rightarrow_o . |Mval \Box_r p \supset p| \]

\[ \forall p \rightarrow_o . |Mval \Box_r p \supset \Box_r \Box_r p| \]

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

\[ \text{ICLval} = \text{Mval} \]

Addition of Modal Logic Axioms for S4

\[ \forall p \rightarrow o. [\text{Mval} \ \Box_r p \supset p] \]

\[ \forall p \rightarrow o. [\text{Mval} \ \Box_r p \supset \Box_r \Box_r p] \]

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

\[ \text{ICLval} = \text{Mval} \]

Addition of Modal Logic Axioms for S4

\[ \forall p \rightarrow o. | \text{Mval} \Box_r p \supset p | \]

\[ \forall p \rightarrow o. | \text{Mval} \Box_r p \supset \Box_r \Box_r p | \]

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]
Example 1 (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]):

\[\text{ICLval (Admin says deletefile1) } \supset \text{ deletefile1}\]  
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]):

\[\text{\textbf{ICLval (Admin says deletefile1)} } \supset \text{ deletefile1} \]
If Admin says that file1 should be deleted, then this must be the case.

\[\text{\textbf{ICLval Admin says ((Bob says deletefile1) } \supset \text{ deletefile1)}\]
Admin trusts Bob to decide whether file1 should be deleted.

\[\text{\textbf{Mval } \Box_r (Bob \lor \Box_r, \text{deletefile1})}\]
Bob wants to delete file1.

\[\text{\textbf{ICLval deletefile1}}\]
Is deletion permitted?
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.

\[\forall w. \forall y. r \ w \ y \Rightarrow (Bob y \lor \forall u. r \ w \ u \Rightarrow deletefile1 u)\]
Bob wants to delete file1.

\[ICLval\ deletefile1\]
Is deletion permitted?

LEO-II: 0.301 seconds
More Examples from [GargAbadi08]

- Example I: 0.301 seconds
- Example II (ICL⇒): 0.503 seconds
- Example III (ICLB): 0.077 seconds

Also possible: reasoning about meta-properties
  - ICL⇒ can be expressed in ICL^B: 0.073 seconds
ICL:

<table>
<thead>
<tr>
<th>Name</th>
<th>Problem</th>
<th>LEO (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>unit</td>
<td>({R,T} \models^{HOL} \parallel \text{ICLval } s \supset (A \text{ says } s)\parallel)</td>
<td>0.053</td>
</tr>
<tr>
<td>cuc</td>
<td>({R,T} \models^{HOL} \parallel \text{ICLval})</td>
<td></td>
</tr>
<tr>
<td></td>
<td>((A \text{ says } (s \supset t)) \supset (A \text{ says } s) \supset (A \text{ says } t)\parallel)</td>
<td>0.167</td>
</tr>
<tr>
<td>idem</td>
<td>({R,T} \models^{HOL} \parallel \text{ICLval } (A \text{ says } A \text{ says } s) \supset (A \text{ says } s)\parallel)</td>
<td>0.058</td>
</tr>
<tr>
<td>unit(^K)</td>
<td>(\models^{HOL} \parallel \text{ICLval } s \supset (A \text{ says } s)\parallel)</td>
<td>–</td>
</tr>
<tr>
<td>cuc(^K)</td>
<td>(\models^{HOL} \parallel \text{ICLval } (A \text{ says } (s \supset t)) \supset (A \text{ says } s) \supset (A \text{ says } t)\parallel)</td>
<td>–</td>
</tr>
<tr>
<td>idem(^K)</td>
<td>(\models^{HOL} \parallel \text{ICLval } (A \text{ says } A \text{ says } s) \supset (A \text{ says } s)\parallel)</td>
<td>–</td>
</tr>
</tbody>
</table>

\(R, T\): reflexivity and transitivity axioms for S4 as seen before
ICL⇒:

<table>
<thead>
<tr>
<th>Name</th>
<th>Problem</th>
<th>LEO (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>refl</td>
<td>{R,T} \models_{HOL} \parallel ICLval A \implies A∥</td>
<td>0.059</td>
</tr>
<tr>
<td>trans</td>
<td>{R,T} \models_{HOL} \parallel ICLval (A \implies B) \supset (B \implies C) \supset (A \implies C)∥</td>
<td>0.083</td>
</tr>
<tr>
<td>sp.-for</td>
<td>{R,T} \models_{HOL} \parallel ICLval (A \implies B) \supset (A says s) \supset (B says s)∥</td>
<td>0.107</td>
</tr>
<tr>
<td>handoff</td>
<td>{R,T} \models_{HOL} \parallel ICLval (B says (A \implies B)) \supset (A \implies B)∥</td>
<td>0.075</td>
</tr>
</tbody>
</table>

| refl^K   | \models_{HOL} \parallel ICLval A \implies A∥                          | 0.034   |
| trans^K  | \models_{HOL} \parallel ICLval (A \implies B) \supset (B \implies C) \supset (A \implies C)∥ | –       |
| sp.-for^K| \models_{HOL} \parallel ICLval (A \implies B) \supset (A says s) \supset (B says s)∥ | –       |
| handoff^K| \models_{HOL} \parallel ICLval (B says (A \implies B)) \supset (A \implies B)∥ | –       |

R, T: reflexivity and transitivity axioms as for S4 seen before
Exp.: Access Control Logic in HOL

$ICL^B$: 

<table>
<thead>
<tr>
<th>Name</th>
<th>Problem</th>
<th>LEO (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>trust</td>
<td>${R,T} \models^{HOL} ICLval (\bot \text{ says } s) \supset s$</td>
<td>0.058</td>
</tr>
<tr>
<td>untrust</td>
<td>${R,T, ICLval ~A \equiv \top} \models^{HOL} ICLval ~A \text{ says } \bot$</td>
<td>0.046</td>
</tr>
<tr>
<td>cuc'</td>
<td>${R,T} \models^{HOL} ICLval ((A \supset B) \text{ says } s) \supset (A \text{ says } s) \supset (B \text{ says } s)$</td>
<td>0.200</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Problem</th>
<th>LEO (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>trust$^K$</td>
<td>$\models^{HOL} ICLval (\bot \text{ says } s) \supset s$</td>
<td>–</td>
</tr>
<tr>
<td>untrust$^K$</td>
<td>${ ICLval ~A \equiv \top} \models^{HOL} ICLval ~A \text{ says } \bot$</td>
<td>0.055</td>
</tr>
<tr>
<td>cuc'$^K$</td>
<td>$\models^{HOL} ICLval ((A \supset B) \text{ says } s) \supset (A \text{ says } s) \supset (B \text{ says } s)$</td>
<td>–</td>
</tr>
</tbody>
</table>

$R, T$: 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
THF syntax for HOL
library for HOL (\(>\) 2700 problems)
tools for HOL
  (parser, type checker, pretty printer, \ldots)
integrated HOL ATPs: IsabelleP, TPS, LEO-II
integrated HOL model generator: IsabelleM
SystemOnTPTP online interface
Christoph Benzmüller

Automating Access Control Logics in STT with LEO-II
LEO-II
(EPRSC grant EP/D070511/1 at Cambridge University)

Thanks to Larry Paulson
LEO-II employs FO-ATPs: E, Spass, Vampire

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