Effiziente Automatisierung von Logik höherer Stufe – realisierbarer Traum oder ewiger Albtraum?

Christoph E. Benzmüller

6. Dezember 2007

Efficient Automation of Higher-Order Logic – viable Dream or perpetual Nightmare?

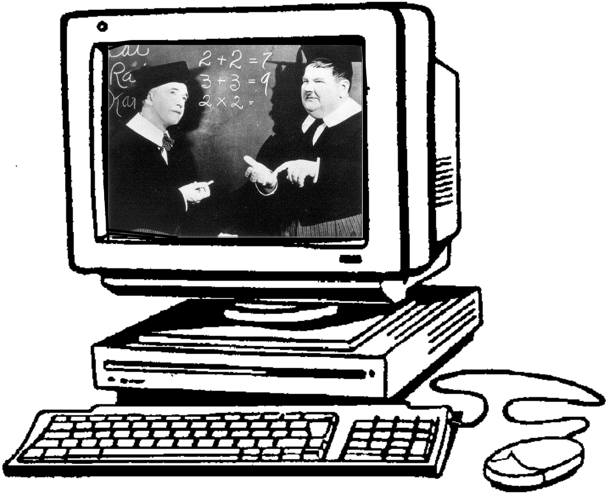
Christoph E. Benzmüller

December 6, 2007

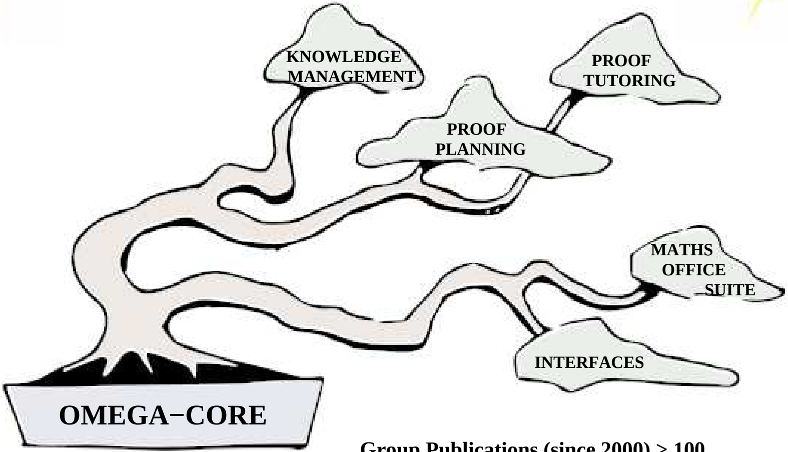
Overview

1. Research Context: The ΩMEGA Project
   (thanks to: J. Siekmann, S. Autexier, ΩMEGA group)
2. Higher-Order Logic (HOL)
   (thanks to: C. Brown and M. Kohlhase)
   - The Good Thing: Expressivitity
   - The Bad Thing: Automation is a Challenge
3. The LEO-II Prover
   (thanks to: L. Paulson, F. Theiss, A. Fietzke)
   - Motivation and Architecture
   - Solving Lightweight Problems
   - Solving Less Lightweight Problems
   - Ongoing and Future Work

Ωmega Project: Long Term Goal



Let it grow, let it grow,
Let it blossom, let it flow,
...



Group Publications (since 2000) > 100

The Ωmega Group

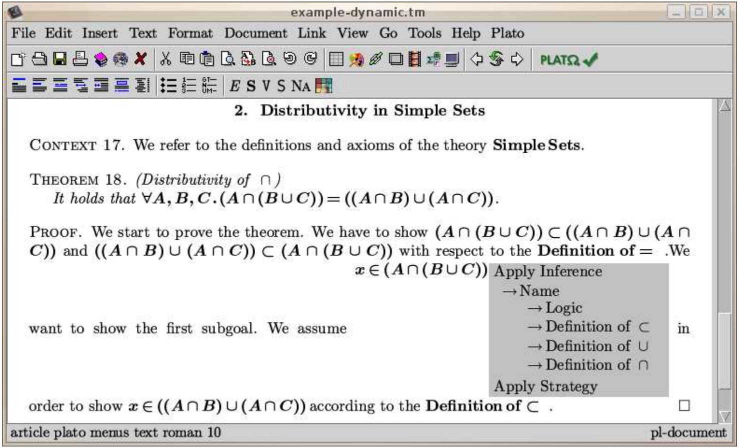
Current Team:

Dr. Serge Autexier
Dr. Christoph Benzmüller
Dominik Dietrich
Andreas Franke
PD Dr. Helmut Horacek
Dr. Henri Lesourd
Marvin Schiller
Ewaryst Schulz
Prof. Dr. Jörg Siekmann
Frank Theiss
Marc Wagner

Former Members Include:

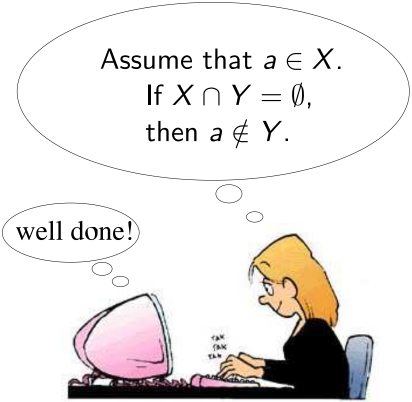
Dr. Chad Brown
Mark Buckley
Dr. Detlef Fehrer
Dr. Manfred Kerber
Prof. Dr. Michael Kohlhase
Dr. Karsten Konrad
Dr. Andreas Meier
PD Dr. Erica Melis
Dr. Markus Moschner
Martin Pollet
Dr. Volker Sorge
Dimitra Tsovaltzi
Dr. Claus-Peter Wirth
Jürgen Zimmer
...

Ωmega Research Direction I: Scientific Office Suite



with SFB 378 Project OMEGA and DFG Project VeriMathDoc

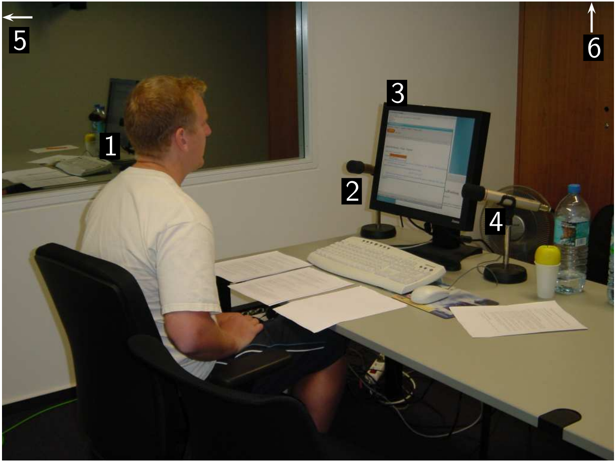
Ωmega Research Direction II: Tutor System for Proofs



with SFB-378 Project DIALOG

Wizard of Oz Experiments: Student Room

1 Subject
2 Audio Recording
3 Subject GUI
4 Audio Control
5 Dome Camera
6 Camera



1 Audio Recording
2 Video Recording
3 Experimenter
4 Overall Control
5 Wizard
6 Wizard GUI

Dialog Example

	correctness	granularity	relevance
Let R, S, T be binary relations.			
Show that $(R \cup S) \circ T = (R \circ T) \cup (S \circ T)$.			
Let $(a, b) \in (R \cup S) \circ T$	✓	✓	✓
Fine. Go ahead.			
...			
From $(a, z) \in S$ and $(z, b) \in T$ it follows that $(a, b) \in S \circ T$	✓	✓	✓
Very good. Please continue!			
Thus $(R \cup S) \circ T \subseteq (R \circ T) \cup (S \circ T)$	✓	–	✓
You cannot directly infer this.			
...			

Typical Machine Proofs are Inadequate in Proof Tutoring

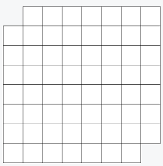
Proof of First-Order Automated Theorem Prover (FO-ATP)

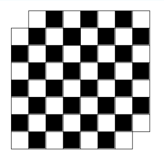
3 [] setequal(x,y)— -subset(x,y)— -subset(y,x).
5 [] subset(x,y)— -member($f1(x,y),y).
22 [] -setequal(intersection($c2,$c1),intersection($c2,$c1)).
23 [factor,3.2.3] setequal(x,x)— -subset(x,x).
27 [] subset(x,y)—member($f1(x,y),x).
29 [hyper,27,23] member($f1(x,x),x)—setequal(x,x).
32 [hyper,29,22] member($f1(intersection($c2,$c1),intersection($c2,$c1)),intersection($c2,$c1)).
41 [hyper,32,5] subset(intersection($c2,$c1),intersection($c2,$c1)).
53 [hyper,41,23] setequal(intersection($c2,$c1),intersection($c2,$c1)).
54 [binary,53.1,22.1] $F.

[KI-07]

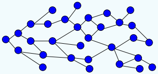
Representation Matters!

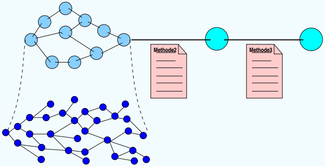
A general lesson in AI . . .



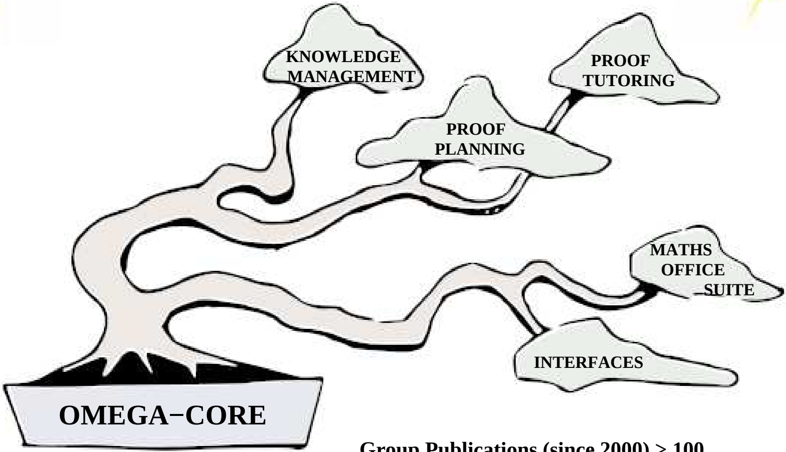


. . . and a specific lesson here





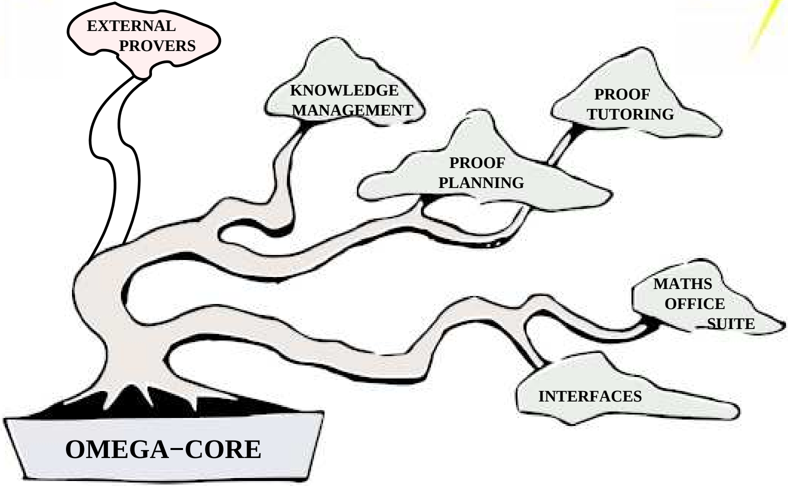
Let it grow, let it grow,
Let it blossom, let it flow,
...



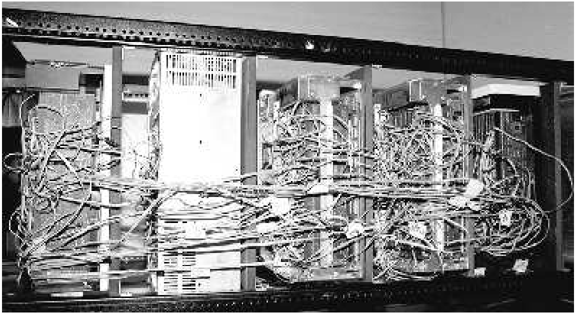
Group Publications (since 2000) > 100

Mein Freund der Baum ist tot,
er fiel im frühen Morgenrot
...

Let it grow, let it grow,
Let it blossom, let it flow,
...



Higher-Order Logic (HOL)



Higher-Order Logic (HOL)

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	–	✓	$\mathit{reflexive}(=)$

HOL is Expressive

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

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

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

$$commutative \quad := \quad \lambda R.\, (\forall x, y.\, R(x, y) \Rightarrow R(y, x))$$

$$Theorem: \quad commutative(\cup)$$

HOL is Expressive

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

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

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

$$commutative := \lambda R.\, (\forall x, y.\, R(x, y) \Rightarrow R(y, x))$$

$$Theorem: \quad commutative(\cup)$$

HOL is Expressive

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

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

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

$$commutative \;:=\; \lambda R.\, (\forall x, y.\, R(x, y) \Rightarrow R(y, x))$$

$$Theorem: \quad commutative(\cup)$$

HOL is Expressive

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

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

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

$$commutative \;:=\; \lambda R.\, (\forall x, y.\, R(x,y) \Rightarrow R(y,x))$$

$$Theorem: \quad commutative(\cup)$$

HOL is Expressive

$$
\begin{aligned}
A \cup B &:= \{x \mid x \in A \vee x \in B\} \\
A \cup B &:= (\lambda x.\, x \in A \vee x \in B) \\
\cup &:= \lambda A.\, \lambda B.\, (\lambda x.\, x \in A \vee x \in B)
\end{aligned}
$$

$$
\mathit{commutative} := \lambda R.\, (\forall x, y.\, R(x,y) \Rightarrow R(y,x))
$$

$$
\mathit{Theorem} : \quad \mathit{commutative}(\cup)
$$

HOL is Expressive

Sets and Relations in HOL

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

Without Types: HOL is too Expressive

Russel's Paradox

The set of all sets which do not contain themselves:

$$\{x|x \notin x\} \text{ resp. } (\lambda x_{\blacksquare} x \notin x) \text{ resp. } (\lambda x_{\blacksquare} \neg x(x))$$

Typed Sets and Relations in HOL

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

Without Types: HOL is too Expressive

Russel's Paradox

The set of all sets which do not contain themselves:

$$\{x | x \notin x\} \text{ resp. } (\lambda x_\blacksquare x \notin x) \text{ resp. } (\lambda x_\blacksquare \neg x(x))$$

Typed Sets and Relations in HOL

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

Increasing Interest in HOL

Research Areas

Formal Methods | Functional and Logical Programming
Artificial Intelligence | Computational Linguistics
Philosophy | ...

Industrial Context

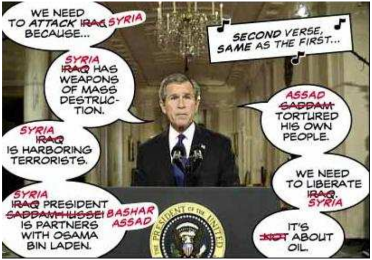
Verisoft, Cycorp, Intel, Articulate Software, Lemma 1, Microsoft ...

Automation of HOL: A Nightmare?

Undecidable and Infinitary Unification

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

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



Automation of HOL: A Nightmare?

Primitive Substitution

Example Theorem: $\exists S_{\bullet}\, reflexive(S)$

Negation and Expansion of Definitions:

$$\neg \exists S_{\bullet}\, (\forall x_{\iota \bullet}\, S(x, x))$$

Clause Normalisation ($a(S)$ Skolem term):

$$\neg S(a(S), a(S))$$

Guess some suitable instances for S

$S \leftarrow \lambda y_{\bullet}\, \lambda z_{\bullet}\, \top$

$$\rightsquigarrow \neg \top$$

$S \leftarrow \lambda y_{\bullet}\, \lambda z_{\bullet}\, V(y, z) = W(y, z)$

$$\rightsquigarrow V(a(S), a(S)) \neq W(a(S), a(S))$$

$S \leftarrow \ldots$

Automation of HOL: A Nightmare?

Cut rule

$$\frac{A \Rightarrow \mathbf{C} \quad \mathbf{C} \Rightarrow B}{A \Rightarrow B}$$

considered as bad in ATP

Calculi that avoid these axioms

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

Automation of HOL: A Nightmare?

Cut rule

$$\frac{A \Rightarrow \mathbf{C} \quad \mathbf{C} \Rightarrow B}{A \Rightarrow B}$$

considered as bad in ATP

[IJCAR-06]: Axioms that imply Cut Calculi that avoid these axioms

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

Automation of HOL: A Nightmare?

Cut rule

$$\frac{A \Rightarrow \mathbf{C} \quad \mathbf{C} \Rightarrow B}{A \Rightarrow B}$$

considered as bad in ATP

Calculi that avoid these axioms

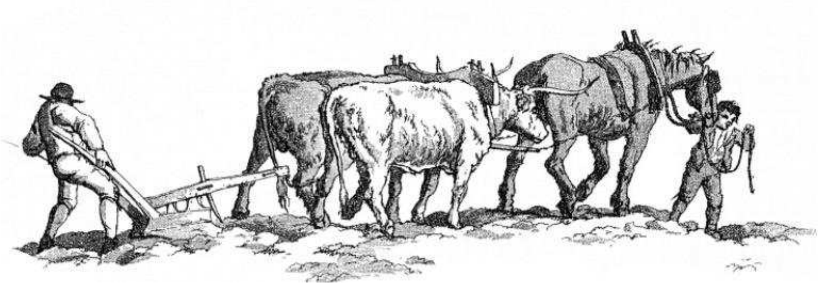
- Axiom of excluded middle ✓
- Comprehension axioms ✓
- Functional and Boolean extensionality ✓[CADE-98,PhD-99]
- Leibniz and other definitions of equality ✓[CADE-99,PhD-99]
- Axiom of induction –
- Axiom of choice –
- Axiom of description –

LEO-II

An Effective Higher-Order Theorem Prover

UNIVERSITY OF CAMBRIDGE

UNIVERSITÄT DES SAARLANDES



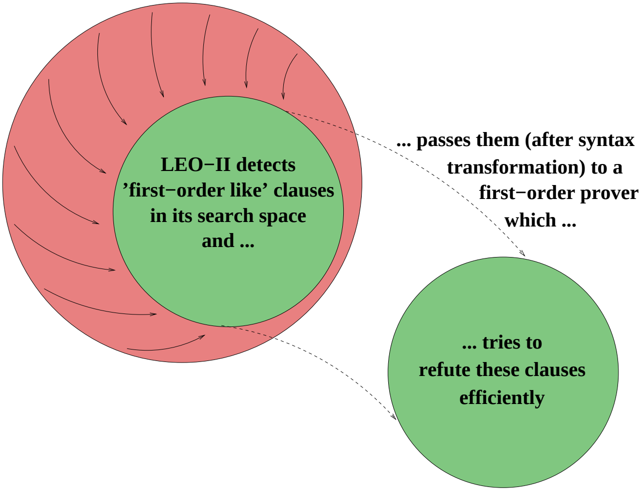
LEO-II employs FO-ATPs: E, Spass, Vampire

jww: L. Paulson (Cambridge), F. Theiss and A. Fietzke (Saarbr.)

Motivation for LEO-II

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

Architecture of LEO-II



Solving Lightweight Problems

Example: TPTP Problem SET171+3

Axiomatization in FO Set Theory

Assumptions:

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

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

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

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

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

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

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

Proof Goal:

$\forall B, C, D.$

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

Performance: FO-ATPs

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

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

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

Performance: LEO-II + E

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

Example: TPTP Problem SET171+3

Axiomatization in FO Set Theory

Assumptions:

$\forall B, C, x_{\blacksquare}(x \in (B \cup C) \Leftrightarrow x \in B \vee x \in C)$

$\forall B, C, x_{\blacksquare}(x \in (B \cap C) \Leftrightarrow x \in B \wedge x \in C)$

$\forall B, C_{\blacksquare}(B \subseteq C \Leftrightarrow \forall x_{\blacksquare} x \in B \Rightarrow x \in C)$

$\forall B, C_{\blacksquare}(B \cup C = C \cup B)$

$\forall B, C_{\blacksquare}(B \cap C = C \cap B)$

$\forall B, C_{\blacksquare}(B = C \Leftrightarrow B \subseteq C \wedge C \subseteq B)$

$\forall B, C_{\blacksquare}(B = C \Leftrightarrow \forall x_{\blacksquare} x \in B \Leftrightarrow x \in C)$

Proof Goal:

$\forall B, C, D_{\blacksquare}$

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

Performance: FO-ATPs

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

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

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

Performance: LEO-II + E

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

Example: TPTP Problem SET171+3

Axiomatization in FO Set Theory

Assumptions:

$\forall B, C, x_{\blacksquare}(x \in (B \cup C) \Leftrightarrow x \in B \vee x \in C)$

$\forall B, C, x_{\blacksquare}(x \in (B \cap C) \Leftrightarrow x \in B \wedge x \in C)$

$\forall B, C_{\blacksquare}(B \subseteq C \Leftrightarrow \forall x_{\blacksquare} x \in B \Rightarrow x \in C)$

$\forall B, C_{\blacksquare}(B \cup C = C \cup B)$

$\forall B, C_{\blacksquare}(B \cap C = C \cap B)$

$\forall B, C_{\blacksquare}(B = C \Leftrightarrow B \subseteq C \wedge C \subseteq B)$

$\forall B, C_{\blacksquare}(B = C \Leftrightarrow \forall x_{\blacksquare} x \in B \Leftrightarrow x \in C)$

Proof Goal:

$\forall B, C, D_{\blacksquare}$

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

Performance: FO-ATPs

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

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

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

Performance: LEO-II + E

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

Example: TPTP Problem SET171+3

Axiomatization in FO Set Theory

Assumptions:

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

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

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

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

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

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

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

Proof Goal:

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

Performance: FO-ATPs

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

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

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

Performance: LEO-II + E

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

Results

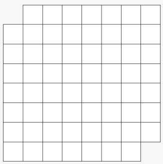
Problem	Vamp. 9.0	LEO+Vamp.	LEO-II+E
014+4	114.5	2.60	0.300
017+1	1.0	5.05	0.059
066+1	–	3.73	0.029
067+1	4.6	0.10	0.040
076+1	51.3	0.97	0.031
086+1	0.1	0.01	0.028
096+1	5.9	7.29	0.033
143+3	0.1	0.31	0.034
171+3	68.6	0.38	0.030
580+3	0.0	0.23	0.078
601+3	1.6	1.18	0.089
606+3	0.1	0.27	0.033
607+3	1.2	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

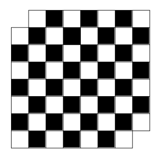
Problem	Vamp. 9.0	LEO+Vamp.	LEO-II+E
648+3	98.2	0.12	0.037
649+3	117.5	0.25	0.037
651+3	117.5	0.09	0.029
657+3	146.6	0.01	0.028
669+3	83.1	0.20	0.041
670+3	–	0.14	0.067
671+3	214.9	0.47	0.038
672+3	–	0.23	0.034
673+3	217.1	0.47	0.042
680+3	146.3	2.38	0.035
683+3	0.3	0.27	0.053
684+3	–	3.39	0.039
716+4	–	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

Representation (and the right System Architecture) Matters!

A general lesson in AI . . .

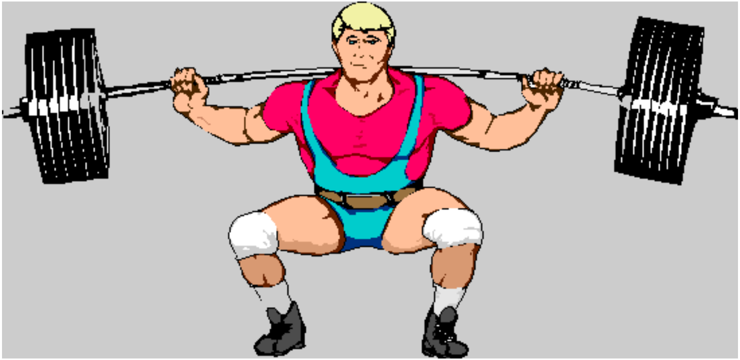




. . . and a specific lesson here

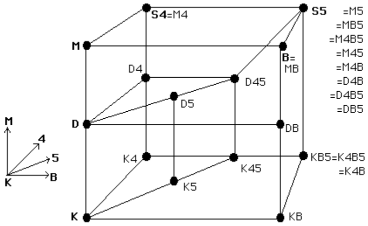
FOL	HOL
+	+
FO-ATP	LEO-II + FO-ATP

Solving Less Lightweight Problems



Logic Systems Interrelationships

Modal Logics Challenge



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

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

Example

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

Theorems:

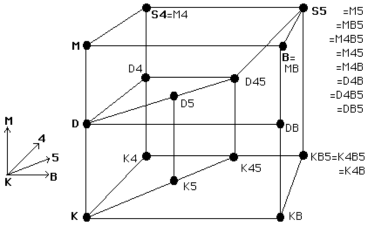
$$
\begin{array}{rcll}
S4 & \not\subseteq & K & (1) \\
(M \wedge 4) & \Leftrightarrow & (refl(R) \wedge trans(R)) & (2)
\end{array}
$$

Experiments

	FO-ATPs [SutcliffeEtal-07]	LEO-II + E [BePa-08]
(1)	16min + 2710s	17.3s
(2)	???	2.3s

Logic Systems Interrelationships

Modal Logics Challenge



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

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

Example

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

Theorems:

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

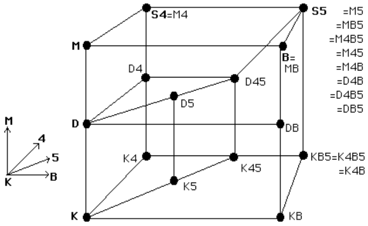
$$(M \wedge 4) \Leftrightarrow (refl(R) \wedge trans(R)) \qquad (2)$$

Experiments

	FO-ATPs [SutcliffeEtal-07]	LEO-II + E [BePa-08]
(1)	16min + 2710s	17.3s
(2)	???	2.3s

Logic Systems Interrelationships

Modal Logics Challenge



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

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

Example

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

Theorems:

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

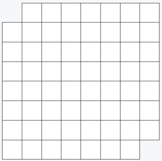
Experiments

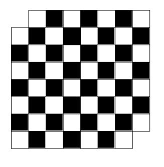
	FO-ATPs **[SutcliffeEtal-07]**	LEO-II + E **[BePa-08]**
(1)	16min + 2710s	17.3s
(2)	???	2.3s

Much simpler: Reasoning within Multimodal Logics

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

A general lesson in AI . . .





. . . and a specific lesson here

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

... there is much left to be done!

LEO-II

- Equational Reasoning
- Termindexing
- Handling of Definitions

Cooperat. with Specialist Reasoners

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

Integration into Proof Assistants

- Relevance of Axioms
- Proof Transf./Verification

International Infrastructure

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

Applications

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

... there is much left to be done!

LEO-II

- Equational Reasoning
- Termindexing
- Handling of Definitions

Cooperat. with Specialist Reasoners

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

Integration into Proof Assistants

- Relevance of Axioms
- Proof Transf./Verification

International Infrastructure

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

Applications

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

. . . there is much left to be done!

LEO-II

- Equational Reasoning
- Termindexing
- Handling of Definitions

Cooperat. with Specialist Reasoners

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

Integration into Proof Assistants

- Relevance of Axioms
- Proof Transf./Verification

International Infrastructure

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

Applications

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

... there is much left to be done!

LEO-II

- Equational Reasoning
- Termindexing
- Handling of Definitions

Cooperat. with Specialist Reasoners

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

Integration into Proof Assistants

- Relevance of Axioms
- Proof Transf./Verification

International Infrastructure

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

Applications

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

... there is much left to be done!

LEO-II

- Equational Reasoning
- Termindexing
- Handling of Definitions

Cooperat. with Specialist Reasoners

- Monadic Second-Order Logic, Prop. Logic, Arithmetic, ...
- Logic Translations
- Feedback for LEΩ-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, ...

Dream or Nightmare?

(Normal) Multimodal Logic in HOL

Simple, Straightforward Encoding of Multimodal Logic

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

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

Related Work

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

(Normal) Multimodal Logic in HOL

Encoding of Validity

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

Encoding of the Theorems

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

(2.3s, LEO-II passes **70** clauses to E, E generates **21769** clauses before finding the empty clause)

(2) $\exists R.\exists A.\exists B.(\neg \mathrm{valid}(\Box_R A \Rightarrow A)) \vee (\neg \mathrm{valid}(\Box_R B \Rightarrow \Box_R \Box_R B))$

(17.3s, LEO-II instantiates R with $\neq$ via primitive substitution)

(Normal) Multimodal Logic in HOL

Encoding of Validity

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

Encoding of the Theorems

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

(2.3s, LEO-II passes **70** clauses to E, E generates **21769** clauses before finding the empty clause)

(2) $\exists R. \exists A. \exists B. (\neg \mathrm{valid}(\Box_R A \Rightarrow A)) \vee (\neg \mathrm{valid}(\Box_R B \Rightarrow \Box_R \Box_R B))$

(17.3s, LEO-II instantiates R with $\neq$ via primitive substitution)

(Normal) Multimodal Logic in HOL

Encoding of Validity

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

Encoding of the Theorems

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

(2.3s, LEO-II passes **70** clauses to E, E generates **21769** clauses before finding the empty clause)

(2) $\exists R. \exists A. \exists B. (\neg\text{valid}(\Box_R A \Rightarrow A)) \vee (\neg\text{valid}(\Box_R B \Rightarrow \Box_R \Box_R B))$

(17.3s, LEO-II instantiates R with $\neq$ via primitive substitution)

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

Initialisation of problem

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

Definition expansion

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

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

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

Translation to first-order logic

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

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

Initialisation of problem

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

Definition expansion

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

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

$$s(x, y) \qquad \neg a(u)$$
$$r(y, u) \qquad a(V) \vee \neg r(y, V)$$

Translation to first-order logic

$$@_{(io)\_i}(@_{(i(io))\_i}(s, x), y) \qquad \neg @_{(\iota o)\_\iota}(a, u)$$
$$@_{(io)\_i}(@_{(i(io))\_i}(r, y), u) \qquad @_{(\iota o)\_\iota}(a, V) \vee \neg @_{(io)\_i}(@_{(i(io))\_i}(r, y), V)$$

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

Initialisation of problem

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

Definition expansion

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

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

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

Translation to first-order logic

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

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

Initialisation of problem

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

Definition expansion

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

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

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

Translation to first-order logic

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