The Projects DIALOG and LEO-II (Computational Logic + AI = Great Fun)

PD Dr.-Ing. Christoph E. Benzmüller

International University in Germany

Bruchsal, June 19, 2008

Overview

- Who am I?
- Research Example I:
  Tutorial Natural Language Dialog on Proofs
- Research Example II:
  Cooperative Higher-Order Theorem Prover LEO-II

Who am I?

Enthusiastic Middle- and Long-Distance-Runner

- German champion 1990 (men's cross-country team)
- 3rd German championships 1989 over 5000m (Junioren)
- $>$ 25x Champion of the Rhineland/Rhineland-Palatine
- Try to beat my personal records:

1000m	2:25min	5000m	14:13min
1500m	3:49min	10000m	30:04min

(Two weeks ago: 2nd overall finisher in the Sacramento 10K race)

So, why have I never made it to the Olympic Games?

Prof. Jörg H. Siekmann
(Saarland University/DFKI)

OMEGA GROUP

Research Question That Caught Me!

Can machines think?

At the end of the century, the use of words and general educated opinion will have changed so much that one will be able to speak of "machines thinking" without expecting to be contradicted.

Alan Turing, 1950

The last match man vs machine?

Can machines play chess?

And how about mathematics?
Can we built tools that master non-trivial tasks in mathematics and other related disciplines?

Research Question That Caught Me!

Can machines think?

At the end of the century, the use of words and general educated opinion will have changed so much that one will be able to speak of "machines thinking" without expecting to be contradicted.

Alan Turing, 1950

The last match man vs machine?

Can machines play chess?

And how about mathematics?
Can we built tools that master non-trivial tasks in mathematics and other related disciplines?

Research Question That Caught Me!

Can machines think?

At the end of the century, the use of words and general educated opinion will have changed so much that one will be able to speak of "machines thinking" without expecting to be contradicted.

Alan Turing, 1950

The last match man vs machine?

Can machines play chess?

And how about mathematics?
Can we built tools that master non-trivial tasks in mathematics and other related disciplines?

Research Question That Caught Me!

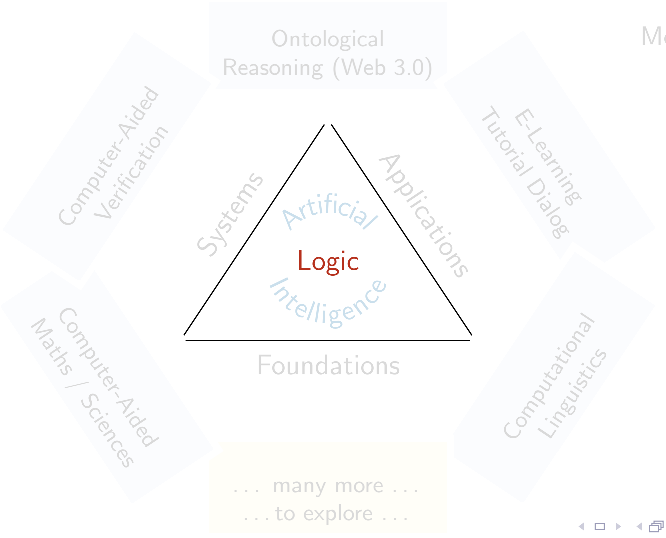
Can machines think?

> *At the end of the century, the use of words and general educated opinion will have changed so much that one will be able to speak of "machines thinking" without expecting to be contradicted.*
>
> *Alan Turing, 1950*

The last match man vs machine?

Can machines play chess?

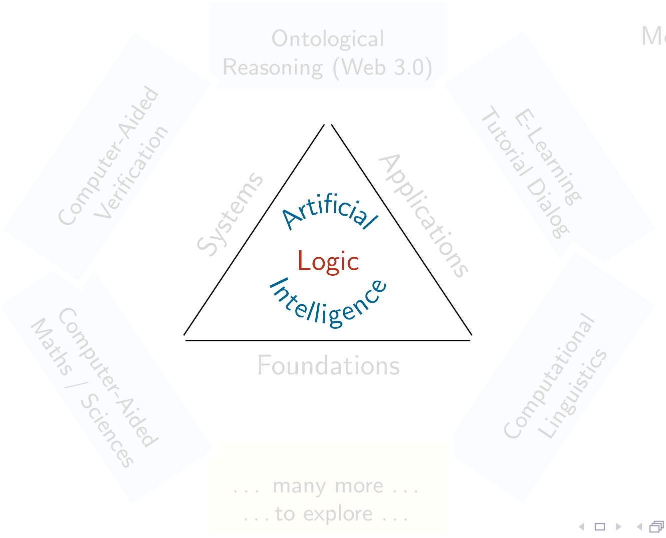
And how about mathematics?
Can we built tools that master non-trivial tasks in mathematics and other related disciplines?



Motivation:

- Philosophical
- Technical
- Practical

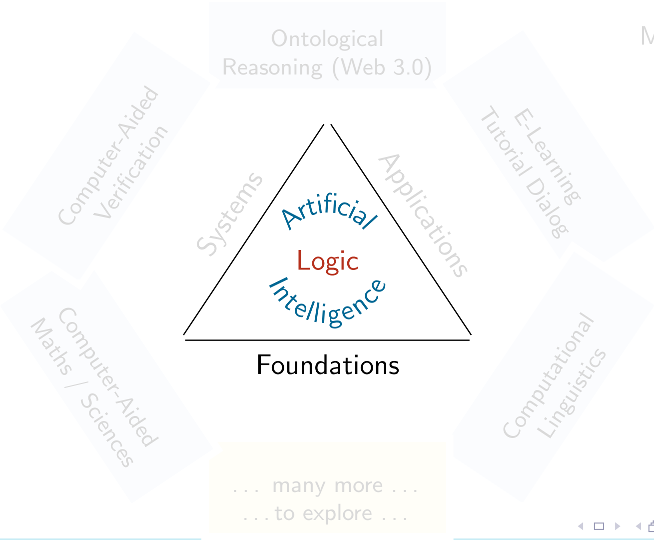
Computational Logic and Artificial Intelligence



Motivation:

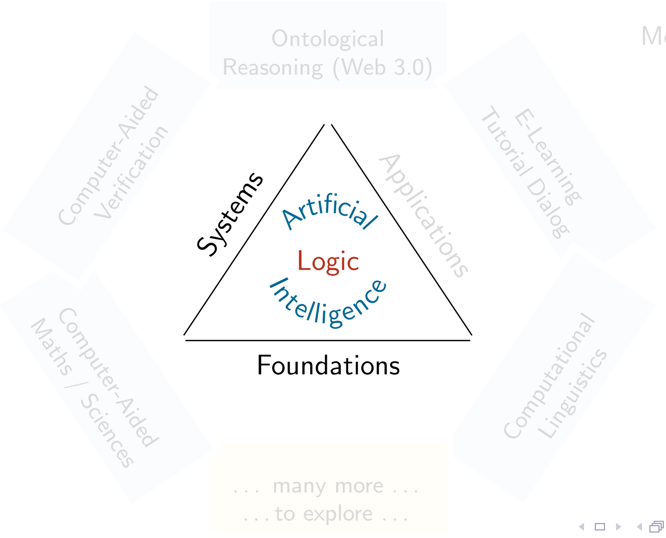
- Philosophical
- Technical
- Practical

Computational Logic and Artificial Intelligence



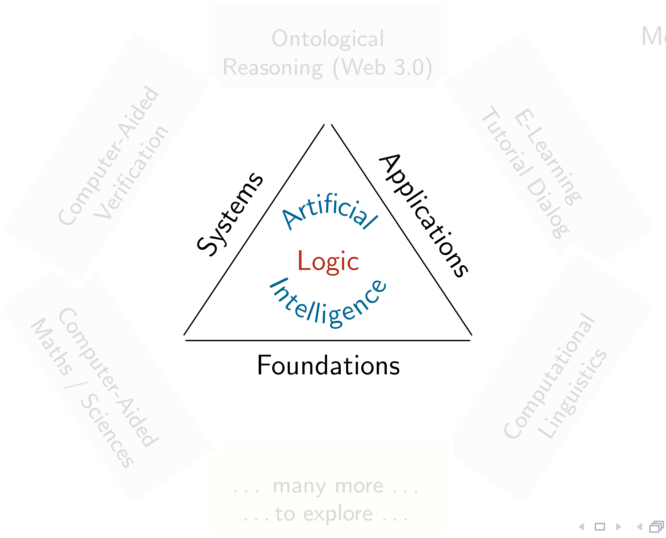
Motivation:

- Philosophical
- Technical
- Practical



Motivation:

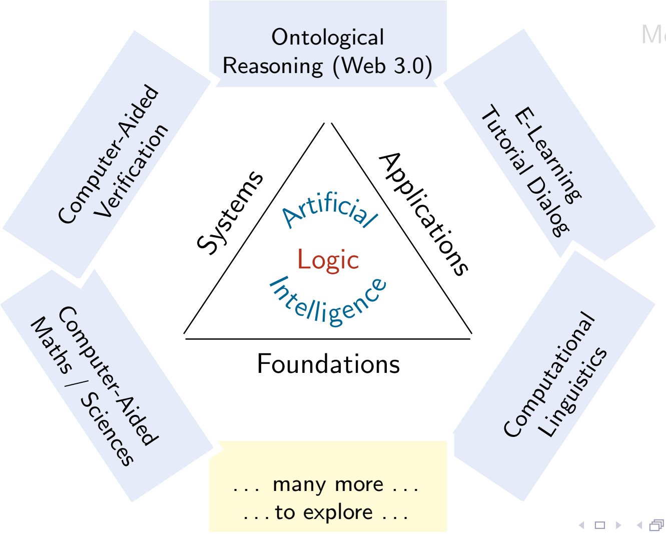
- Philosophical
- Technical
- Practical



Motivation:

- Philosophical
- Technical
- Practical

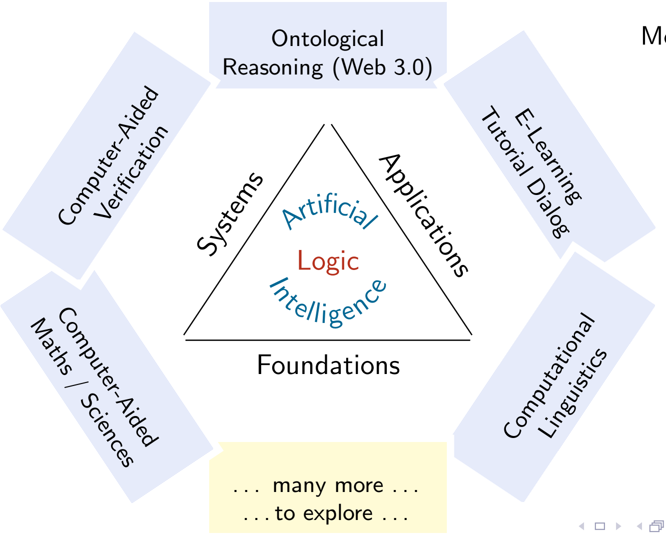
Computational Logic and Artificial Intelligence



Motivation:

- Philosophical
- Technical
- Practical

Computational Logic and Artificial Intelligence



Motivation:

- Philosphical
- Technical
- Practical

Research Example I: DIALOG
(SFB 378 at Saarland University):
Tutorial Natural Language Dialog on Proofs

The DIALOG Project

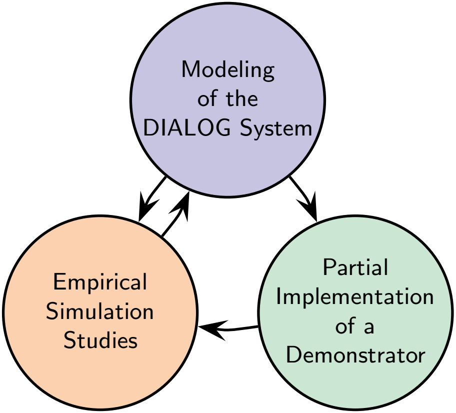
Tutorial NL Dialog for **Mathematical Proofs**.

- Natural language analysis
- **Mathematical domain reasoning**
- Dialog management
- Output generation and verbalization

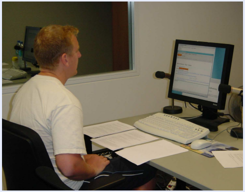
Assume that $a \in X$.
If $X \cap Y = \emptyset$,
then $a \notin Y$.

Interdisciplinary Research in SFB 378

Collaboration: Computational Linguistics and Computer Science



Empirical Investigations
(Wizard-of-Oz)



Let R, S and T be relations in an arbitrary set M. It holds: $(R \cup S) \circ T = (R \circ T) \cup (S \circ T)$. Do the proof interactively with the system!

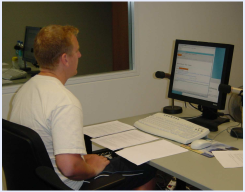
Let $(x, y) \in (R \cup S) \circ T$

Correct! Good start!

Then $\exists z$ such that (x, z) in $(R \cup S)$ and (z, y) in T

Correct!

Empirical Investigations (Wizard-of-Oz)



Let R, S and T be relations in an arbitrary set M. It holds: $(R \cup S) \circ T = (R \circ T) \cup (S \circ T)$. Do the proof interactively with the system!

Let $(x, y) \in (R \cup S) \circ T$

Correct! Good start!

Then $\exists z$ such that (x, z) in $(R \cup S)$ and (z, y) in T

Correct!

Wizard Room

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

Proof Step Evaluation: Correctness, Granularity and Relevance

student:	$(x, y) \in (R \circ S)^{-1}$
tutor:	Now try to draw inferences from that!
student:	$(x, y) \in S^{-1} \circ R^{-1}$
tutor:	One cannot directly deduce that. You need some intermediate steps!

correct	too coarse-grained	relevant

student:	$(x, y) \in (R \circ S)^{-1}$ if according to the inverse relation it holds that $(y, x) \in (R \circ S)$
tutor:	That is correct, but try to use $(x, y) \in (R \circ S)^{-1}$ as a precondition.

correct	appropriate	limited relevance

Automatic Proof Step Evaluation

- Automatic Resolution of Ambiguities and Underspecification
- Automatic Proof Step Evaluation:
  - Correctness:

    any theorem prover
  - Granularity:

    cognitively adequate theorem prover
    + teacher- and student-modeling
    + machine learning
  - Relevance:

    cognitively adequate theorem prover
    + teacher- and student-modeling
    + ???

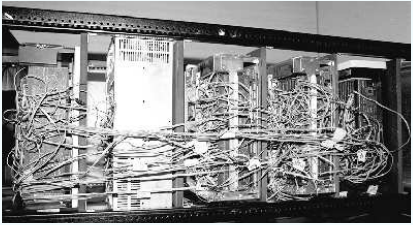
Research Example II: LEO-II

(Project at Cambridge University):

Cooperative Automatic Higher-Order Theorem Prover

Higher-Order Logic (HOL)

Some people say that HOL is like this:



I don't!

- Semantics (extensionality) [PhD-99,JSL-04]
- Proof theory [IJCAR-06]
- **ATPs LEO and LEO-II** [CADE-98,IJCAR-08]

Higher-Order Logic
– An Introduction on one Slide –

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	–	✓	$\mathit{reflexive}(=)$

Sets and Relations in HOL

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

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

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

$$\texttt{symmetric} \;:=\; \lambda F.\, (\forall x, y.\, F(x, y) = F(y, x))$$

$$\mathit{Theorem}: \quad \texttt{symmetric}(\cup)$$

Sets and Relations in HOL

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

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

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

$$\mathtt{symmetric} := \lambda F. (\forall x, y. F(x, y) = F(y, x))$$

$$\mathit{Theorem}: \quad \mathtt{symmetric}(\cup)$$

Sets and Relations in HOL

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

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

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

$$\texttt{symmetric} \;:=\; \lambda F.\, (\forall x, y.\, F(x, y) = F(y, x))$$

$$\textit{Theorem}: \quad \texttt{symmetric}(\cup)$$

Sets and Relations in HOL

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

$$
\texttt{symmetric} := \lambda F.\, (\forall x, y.\, F(x, y) = F(y, x))
$$

$$
\mathit{Theorem}: \quad \texttt{symmetric}(\cup)
$$

Sets and Relations in HOL

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

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

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

$$\mathtt{symmetric} \quad := \quad \lambda F. (\forall x, y. F(x, y) = F(y, x))$$

$$\textit{Theorem}: \quad \mathtt{symmetric}(\cup)$$

Sets and Relations in HOL

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

Types avoid Paradoxes

Russel's Paradox

"The set of all sets which do not contain themselves"

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

Types avoid Paradoxes

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

Types avoid Paradoxes

Russel's Paradox

"The set of all sets which do not contain themselves"

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

Types avoid Paradoxes

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

Automation of HOL: A Challenge!

Three main Challenges:

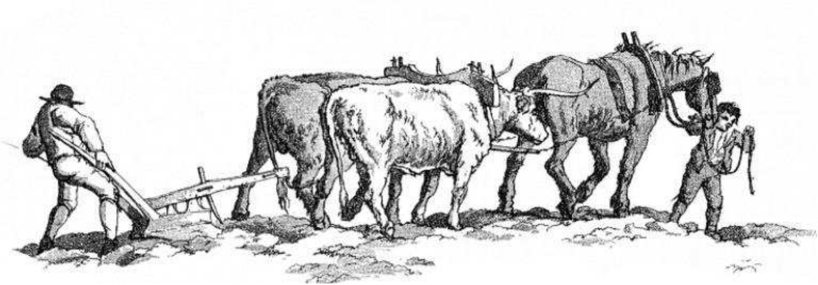
- Undecidable and Infinitary Unification
- Indeterminism and Blind Guessing (Set Variables)
- Cut-Simulation Effect

Interested in more Details?

- . . . ask me later . . .
- . . . and you will hardly get me to stop talking again . . .

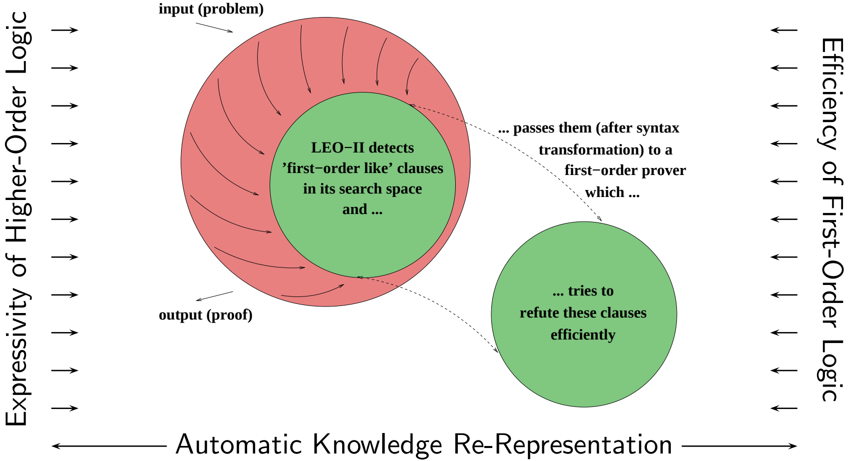
LEO-II
An Effective Higher-Order Theorem Prover

UNIVERSITY OF CAMBRIDGE
UNIVERSITÄT DES SAARLANDES



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

Architecture of LEO-II



Solving Lightweight Problems

Solving Lightweight Problems: TPTP Problem SET171+3

Problem Encoding in HOL

$$
\begin{array}{lcl}
\in & := & \lambda x. \lambda A. A(x) \\
\emptyset & := & \lambda x. \bot \\
\cap & := & \lambda A. \lambda B. (\lambda x. x \in A \wedge x \in B) \\
\cup & := & \lambda A. \lambda B. (\lambda x. x \in A \vee x \in B) \\
\setminus & := & \lambda A. \lambda B. (\lambda x. x \in A \vee x \notin B) \\
\ldots & &
\end{array}
$$

Theorem:

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

Solving Lightweight Problems: TPTP Problem SET171+3

Axiomatization in FO Set Theory

Assumptions:

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

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

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

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

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

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

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

Theorem:

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

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!)
```

Solving Lightweight Problems: TPTP Problem SET171+3

Axiomatization in FO Set Theory

Assumptions:

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

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

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

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

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

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

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

Theorem:

$\forall B, C, D.$

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

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!)
```

Solving Lightweight Problems: TPTP Problem SET171+3

Axiomatization in FO Set Theory

Assumptions:

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

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

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

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

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

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

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

Theorem:

$\forall B, C, D.$

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

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!)
```

Solving Lightweight Problems: TPTP Problem SET171+3

Axiomatization in FO Set Theory

Assumptions:

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

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

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

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

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

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

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

Theorem:

$\forall B, C, D.$

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

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

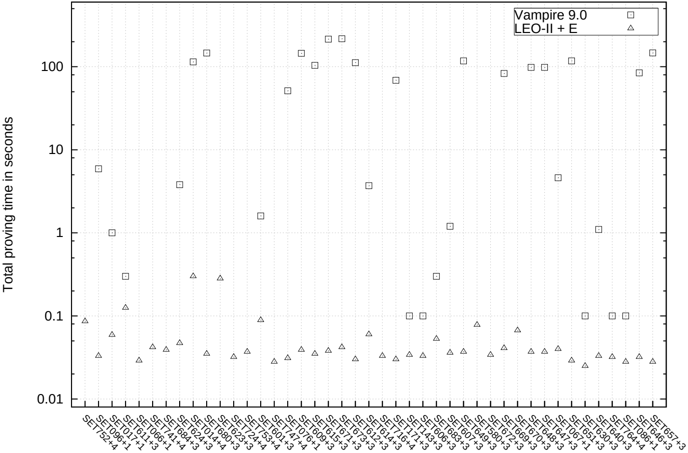
```
Eureka --- Thanks to Corina!
Total Reasoning Time: 0.03s
LEO-II (Proof Found!)
```

So, let's beat the Champion ...

LEO-II vs. Vampire 9.0

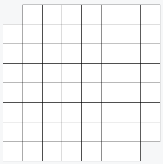
(on Problems about Sets, Relations and Functions)

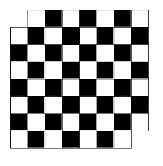
So, let's beat the Champion ...



Representation (and the right System Architecture) Matters!

A general lesson in AI . . .

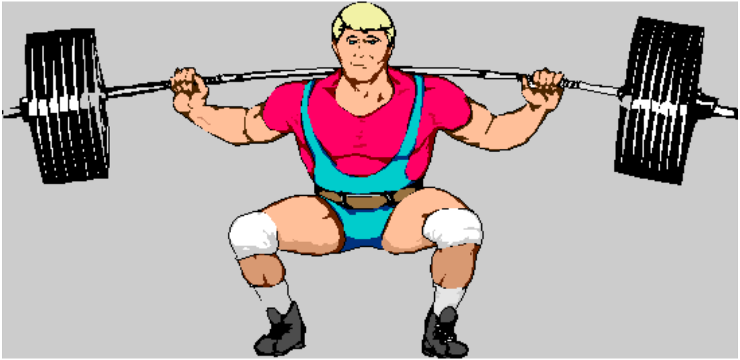




. . . and a specific lesson here

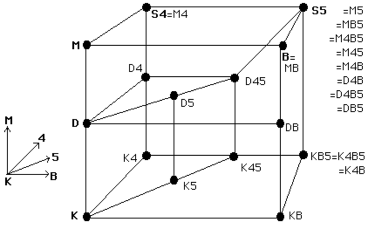
FOL	HOL
+	+
FO-ATP	LEO-II & FO-ATP

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

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

Theorems:

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

Experiments

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

Outlook

- Increasing interest in computational logic and formal methods in industry (examples: Microsoft and Intel)
- Increasing range of applications:
  Software- and Hardware Verification, Security, Semantic Web, E-Learning, Bio-Informatics, Finance, ...

- Academia needs to produce more
  - well-trained students
  - intelligent tools
    (combining techniques from computational logic and artificial intelligence/computational intelligence)

- This is what I want to support!

International Collaborators and Friends . . .

- With Joint Papers and/or Joint Projects (in CS)
  - Carnegie Mellon University, Pittsburgh, PA, U.S.
  - Miami University, FL, U.S.
  - Articulate Software, Napa Valley, CA, U.S.
  - Cambridge University, England
  - University of Birmingham, England
  - University of Edinburgh & Heriot-Watt Univ., Scotland
  - Université Paris-Sud, France
  - EU RTN Calculemus (2000-2004), 9 European partners
  - . . .
- Many Academic Friends (in CS, Economy and other areas)
  - Mahidol Univ. Bangkok & Thinkergy Ltd., Thailand
  - P. Univ. Catolica Madre y Maestra, S. Domingo, Dom. Rep.
  - Microsoft Research, Cambridge, England
  - . . .