LEO-II version 1.5

Christoph Benzmüller[1] and Nik Sultana[2]

Freie Universität Berlin, Germany / Cambridge University, UK

Proof Exchange for Theorem Provers (PxTP)
Lake Placid, NY, USA, 2013

[1]Thanks to: DFG Heisenberg Fellowship BE-2501/9-1
[2]Thanks to: Grant from the German Academic Exchange Service (DAAD)

Talk Outline

A: Introduction

- Motivation for LEO prover(s)
- Logic HOL / TPTP THF0
- Reasoning principles of LEO provers
- LEO-II

B: New Stuff in LEO-II

- Support for different FOL translations
- Integration of proofs from EP
- Improved support for back-end provers
- Detection/removal of Leibniz- and Andrews-equality
- Support for choice in LEO-II
- Further improvements
- Experiments

C: Conclusion

A: Motivation for LEO prover(s)

OMEGA [BenzmüllerEtAl,CADE,1996][SiekmannEtAl,JApplLog,2006]:

- proof assistant with a focus on AI techniques
  - proof planning & agents
  - system integration: ATPs, computer algebra systems
  - knowledge management tools: MAYA
  - E-learning, tutorial NL dialog, user interfaces, ...
- foundation: classical higher-order logic (HOL) & ND calculus
- developed from early 90s until 'J. Siekmann's retirement'

LEO [BenzmüllerKohlhase,CADE,1998]

- **L**ogical **E**ngine of **O**MEGA
- traditional ATP for HOL; based on (RUE-)resolution
- originally implemented within the OMEGA framework
- early investigation of agent based cooperation with FO-ATPs in OMEGA

A: Motivation for LEO prover(s)

Freie Universität Berlin

OMEGA [BenzmüllerEtAl,CADE,1996][SiekmannEtAl,JApplLog,2006]:

- proof assistant with a focus on AI techniques
  - proof planning & agents
  - system integration: ATPs, computer algebra systems
  - knowledge management tools: MAYA
  - E-learning, tutorial NL dialog, user interfaces, ...
- foundation: classical higher-order logic (HOL) & ND calculus
- developed from early 90s until 'J. Siekmann's retirement'

LEO [BenzmüllerKohlhase,CADE,1998]

- **L**ogical **E**ngine of **O**MEGA
- traditional ATP for HOL; based on (RUE-)resolution
- originally implemented within the OMEGA framework
- early investigation of agent based cooperation with FO-ATPs in OMEGA

A: Logic HOL / TPTP THF0

- Simple Types $\alpha ::= \iota \mid \mu \mid o \mid \alpha_1 \rightarrow \alpha_2$
- HOL Syntax

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

- HOL is (meanwhile) well understood
  - Origin [Church,J.Symb.Log.,1940]
  - Henkin-Semantics [Henkin,J.Symb.Log.,1950]
    [Andrews, J.Symb.Log.,1971,1972]
  - Extens./Intens. [BenzmüllerEtAl.,J.Symb.Log.,2004]
    [Muskens,J.Symb.Log.,2007]
- TPTP THF0: HOL with Henkin-Semantics and Choice

A: Reasoning principles of LEO provers

- extensional higher-order RUE-resolution
- see [Benzmüller,Synthese,2002] or [SultanaBenzmüller,IWIL,2012] for more information

Here, I sketch the idea using a very simple example: `SET171^3`

$$\forall B_{\iota\to o}, C_{\iota\to o}, D_{\iota\to o}. (B \cup (C \cap D) = (B \cup C) \cap (B \cup D))$$

negation, def. expansion ($\cup := \lambda S. \lambda T. \lambda X. SX \vee TX \;\; / \;\; \cap := \ldots$)

$$\neg\forall B, C, D. (\lambda X_\alpha. BX \vee (CX \wedge DX)) = (\lambda X_\alpha. (BX \vee CX) \wedge (BX \vee DX))$$

normalisation, Skolemization (b, c, d new Skolem constant)

$$(\lambda X_\alpha. bX \vee (cX \wedge dX)) \neq (\lambda X_\alpha. (bX \vee cX) \wedge (bX \vee dX))$$

functional and Boolean extensionality (extensional pre-unification)

$$\exists X_\alpha. (bX \vee (cX \wedge dX)) \not\Leftrightarrow ((bX \vee cX) \wedge (bX \vee dX))$$

Skolemization (x new Skolem constant)

$$(bx \vee (cx \wedge dx)) \not\Leftrightarrow ((bx \vee cx) \wedge (bx \vee dx))$$

A: Working principles of LEO-II (SET171^3)

$$(bx \vee (cx \wedge dx)) \not\Leftrightarrow ((bx \vee cx) \wedge (bx \vee dx))$$

normalization

$$\neg bx \qquad bx \vee cx \qquad bx \vee dx \qquad \neg cx \vee \neg dx$$

passes clauses to FO-ATP

$$\neg @_{(\iota \to o) \to \iota \to o}(b, x) \qquad @_{(\iota \to o) \to \iota \to o}(b, x) \vee @_{(\iota \to o) \to \iota \to o}(c, x)$$

$$@_{(\iota \to o) \to \iota \to o}(b, x) \vee @_{(\iota \to o) \to \iota \to o}(d, x)$$

$$\neg @_{(\iota \to o) \to \iota \to o}(c, x) \vee \neg @_{(\iota \to o) \to \iota \to o}(d, x)$$

syntax transformation used here: [Kerber,PhD,1992]

Remark: SET171+3 is still a challenge for problem for FO-ATPs — Vampire-2.6, SPASS-3.7, EP-1.7, and Z3-4.0 (in standard mode) do not return proofs within 600s!!!

A: Working principles of LEO-II (SET171^3)

$$(bx \vee (cx \wedge dx)) \not\Leftrightarrow ((bx \vee cx) \wedge (bx \vee dx))$$

normalization

$$\neg bx \qquad bx \vee cx \qquad bx \vee dx \qquad \neg cx \vee \neg dx$$

passes clauses to FO-ATP

$$\neg @_{(\iota\to o)\to\iota\to o}(b,x) \qquad @_{(\iota\to o)\to\iota\to o}(b,x) \vee @_{(\iota\to o)\to\iota\to o}(c,x)$$

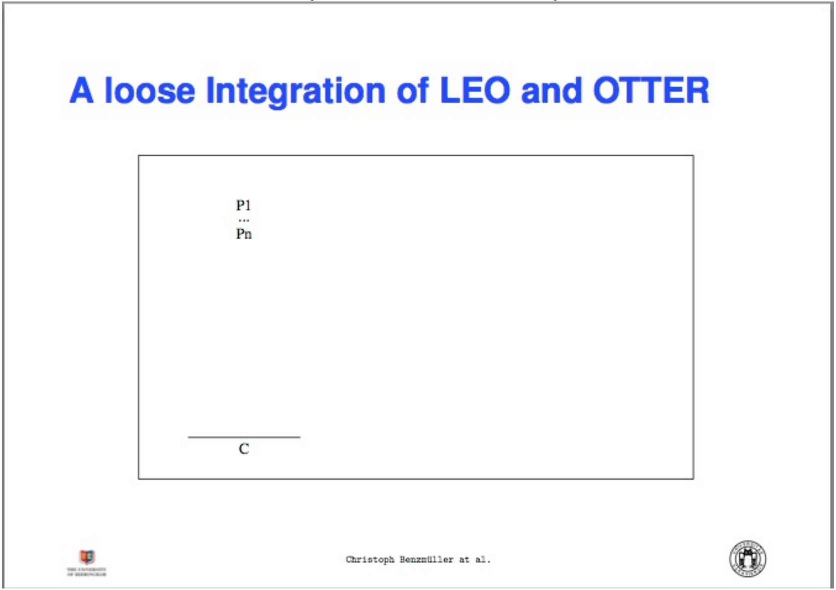
$$@_{(\iota\to o)\to\iota\to o}(b,x) \vee @_{(\iota\to o)\to\iota\to o}(d,x)$$

$$\neg @_{(\iota\to o)\to\iota\to o}(c,x) \vee \neg @_{(\iota\to o)\to\iota\to o}(d,x)$$

syntax transformation used here: [Kerber,PhD,1992]

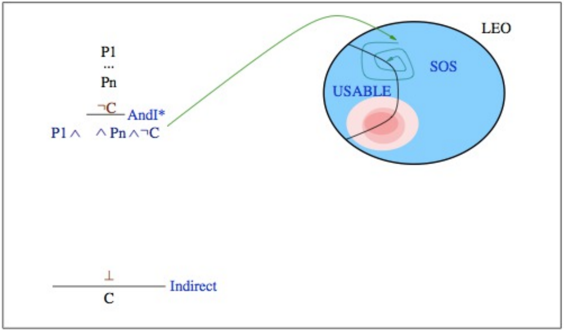
Remark: SET171+3 is still a challenge for problem for FO-ATPs — Vampire-2.6, SPASS-3.7, EP-1.7, and Z3-4.0 (in standard mode) do not return proofs within 600s!!!

A: Some slides from 2000 (Deduktionstreffen)



Freie Universität Berlin

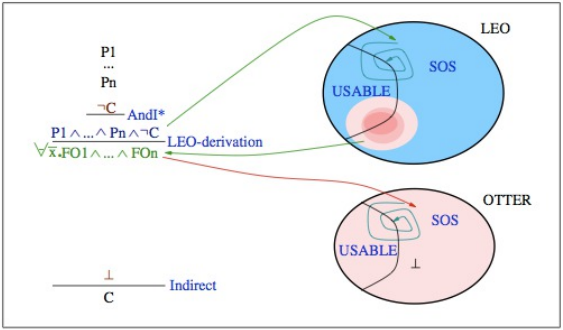
A loose Integration of LEO and OTTER



Christoph Benzmüller at al.

Freie Universität Berlin

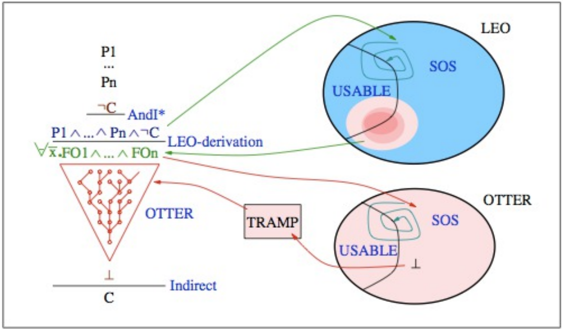
A loose Integration of LEO and OTTER



Christoph Benzmüller at al.

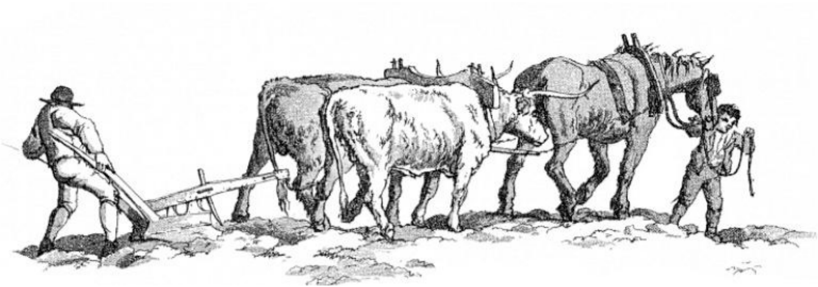
A: Some slides from 2000 (Deduktionstreffen)

A loose Integration of LEO and OTTER



Christoph Benzmüller at al.

A: Philosophy of LEO provers: tight collaboration



LEO resp. LEO-II **(Otter), EP, Spass, Vampire**

A: LEO-II

- website: http://leoprover.org
- developed since 2006/07
  (initial funding: project with Larry Paulson at Cambridge)
- independent implementation in OCaml
- direct collaboration with FO-ATPs: EP (Schulz) as first choice
- applications — THF0 provers as universal reasoners
  - HOL
  - quantified modal logics [ECAI,2012]
  - quantified conditional logics [IJCAI,2013]
  - ambitious logic puzzles [AnnMathArtifIntell,2012]
  - ontology reasoning (e.g. in SUMO) [JWebSemantics,2012]
  - access control logics [SEC,2009]
  - . . . more is on the way
- integrated with HETS, SigmaKEE, Isabelle

A: LEO-II – architecture and organization

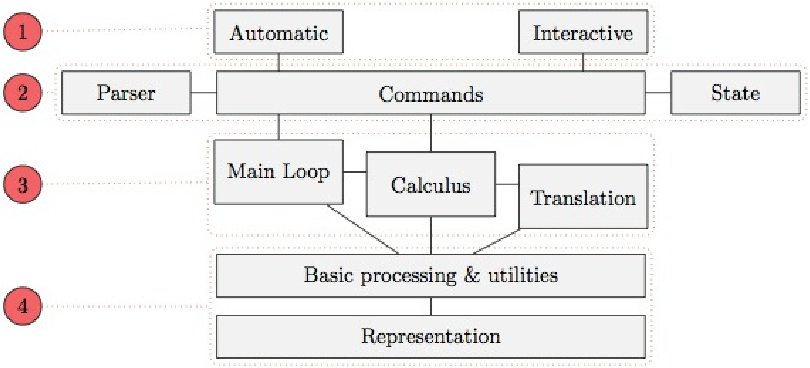


Figure 1: LEO-II's architecture

approx 30000 lines of Ocaml code

Talk Outline

A: Introduction

- Motivation for LEO prover(s)
- Logic HOL / TPTP THF0
- Reasoning principles of LEO provers
- LEO-II

B: New Stuff in LEO-II

- Support for different FOL translations
- Integration of proofs from EP
- Improved support for back-end provers
- Detection/removal of Leibniz- and Andrews-equality
- Support for choice in LEO-II
- Further improvements
- Experiments

C: Conclusion

B: Support for different FOL translations

FOL translations in LEO-II

- type-annotated @-operators [Kerber,PhD,1992]

$$\neg @_{(\iota \to o) \to \iota \to o}(b, x)$$

$$@_{(\iota \to o) \to \iota \to o}(b, x) \lor @_{(\iota \to o) \to \iota \to o}(c, x)$$

B: Support for different FOL translations

FOL translations in LEO-II

- type-annotated @-operators [Kerber,PhD,1992]
- fully-typed [Hurd,CADE,2002]

$$\neg @_{(\iota \to o) \to \iota \to o}(b, x)$$

$$@_{(\iota \to o) \to \iota \to o}(b, x) \vee @_{(\iota \to o) \to \iota \to o}(c, x)$$

```
~(leoLit(leoTi(leoAt(leoTi(b,leoFt(i,o)),leoTi(x,i)),o)))

(leoLit(leoTi(leoAt(leoTi(c,leoFt(i,o)),leoTi(x,i)),o)) |
leoLit(leoTi(leoAt(leoTi(b,leoFt(i,o)),leoTi(x,i)),o)))
```

B: Support for different FOL translations

FOL translations in LEO-II

- type-annotated @-operators [Kerber,PhD,1992]
- fully-typed [Hurd,CADE,2002]

shortcomings in the implementation; see e.g. example:

$$(=) = (=)$$

negation, input processing

```
~leoLit(leoTi(true,o))
```

but: LEO-II didn't provide axioms such as

```
leoLit(leoTi(true,o))
```

B: Support for different FOL translations

FOL translations in LEO-II

- type-annotated @-operators [Kerber,PhD,1992]
- fully-typed [Hurd,CADE,2002]

Instead of

```
~leoLit(leoTi(true,o))
```

LEO-II now simply generates

```
~ $true
```

B: Support for different FOL translations

FOL translations in LEO-II

- type-annotated @-operators [Kerber,PhD,1992]
- fully-typed [Hurd,CADE,2002]
- new (Nik Sultana): fof_full

When proxy terms are needed LEO-II adds axioms like

```
$true <=> leoLit(leoTi(true,o))
```

B: Support for different FOL translations

FOL translations in LEO-II

- type-annotated @-operators [Kerber,PhD,1992]
- fully-typed [Hurd,CADE,2002]
- new (Nik Sultana): fof_full

In LEO-II's fully-typed translation lambda terms like $\lambda X_o. X$ were simply mapped to typed constants: `leoTi(abstrXX,leoFt(o,o))`

In the fof_full translation lambda-lifting is now employed.

B: Support for different FOL translations

FOL translations in LEO-II

- type-annotated @-operators [Kerber,PhD,1992]
- fully-typed [Hurd,CADE,2002]
- new (Nik Sultana): fof_full, fof_experiment

In the fof_experiment translation we are experimenting with lighter encodings of type information following [ClaessenEtal,CADE,2011].

Monotonicity analysis produces a SAT encoding; sent to MiniSat.

Interface for MiniSat has been adapted from Brown's Satallax.

B: Mapping of EP proofs

LEO-II supports different proof output modes

- no proof output (default, '`-po 0`' option)
- detailed proof by LEO-II / no EP proof ('`-po 1`' option)
- since v1.6.0 further options for LEO-II proof part available

```
% SZS status Theorem for SET171^3.p : (rf:0,axioms:0,...,translation:fully-typed)
%**** Beginning of derivation protocol ****
% SZS output start CNFRefutation
 ...
 thf(tp_intersection,type,(intersection: (($i>$o)>(($i>$o)>($i>$o))))).
 thf(tp_union,type,(union: (($i>$o)>(($i>$o)>($i>$o))))).
 ...
 thf(union,definition,(union = (^[X:($i>$o),Y:($i>$o),U:$i]: ((X@U) | (Y@U)))),
     file('SET171^3.p',union)).
 ...
 thf(1,conjecture,(![A:($i>$o),B:($i>$o),C:($i>$o)]:
     (((union@A)@((intersection@B)@C)) = ((intersection@((union@A)@B))@((union@A)@C)))),
     file('SET171^3.p',union_distributes_over_intersection)).
 ...
 thf(72,plain,((($false)=$true)),inference(fo_atp_e,[status(thm)],[11,71,70,69,68,61,60,59,58,
     54,53,52,51,14])).
 thf(73,plain,($false),inference(solved_all_splits,[solved_all_splits(join,[])],[72])).
% SZS output end CNFRefutation
```

B: Mapping of EP proofs

LEO-II supports different proof output modes

- no proof output (default, '-po 0' option)
- detailed proof by LEO-II / no EP proof ('-po 1' option)
- since v1.6.0 further options for LEO-II proof part available

```
% SZS status Theorem for SET171^3.p : (rf:0,axioms:0,...,translation:fully-typed)
%**** Beginning of derivation protocol ****
% SZS output start CNFRefutation
 ...
 thf(tp_intersection,type,(intersection: (($i>$o)>(($i>$o)>($i>$o))))).
 thf(tp_union,type,(union: (($i>$o)>(($i>$o)>($i>$o))))).
 ...
 thf(union,definition,(union = (^[X:($i>$o),Y:($i>$o),U:$i]: ((X@U) | (Y@U)))),
     file('SET171^3.p',union)).
 ...
 thf(1,conjecture,(![A:($i>$o),B:($i>$o),C:($i>$o)]:
     (((union@A)@((intersection@B)@C)) = ((intersection@((union@A)@B))@((union@A)@C)))),
     file('SET171^3.p',union_distributes_over_intersection)).
 ...
 thf(72,plain,((($false)=$true)),inference(fo_atp_e,[status(thm)],[11,71,70,69,68,61,60,59,58,
     54,53,52,51,14])).
 thf(73,plain,($false),inference(solved_all_splits,[solved_all_splits(join,[])],[72])).
% SZS output end CNFRefutation
```

B: Mapping of EP proofs

Since version 1.4.0; see also [SultanaBenzmüller,IWIL,2012]:

- mapping of EP proofs into LEO-II proofs ('-po 2' option)

```
% SZS status Theorem for SET171^3.p : (rf:0,axioms:0,...,translation:fully-typed)
%**** Beginning of derivation protocol ****
% SZS output start CNFRefutation
 ...
 thf(tp_intersection,type,(intersection: (($i>$o)>(($i>$o)>($i>$o))))).
 thf(tp_union,type,(union: (($i>$o)>(($i>$o)>($i>$o))))).
 ...
 thf(union,definition,(union = (^[X:($i>$o),Y:($i>$o),U:$i]: ((X@U) |
% (Y@U)))),
    file('SET171^3.p',union)).
 ...
 thf(1,conjecture,(![A:($i>$o),B:($i>$o),C:($i>$o)]:
    (((union@A)@((intersection@B)@C)) = ((intersection@((union@A)@B))@((union@A)@C)))),
    file('SET171^3.p',union_distributes_over_intersection)).
 ...
 fof(74, axiom, ((leoLit(leoTi(leoAt( ... , inference(fof_translation, [status(thm)],[51])).
 fof(77, axiom, ((leoLit(leoTi(leoAt( ... , inference(fof_translation, [status(thm)],[54])).
 fof(78, axiom, ((~(leoLit(leoTi(leoAt( ... , inference(fof_translation, [status(thm)],[58])).
 fof(85, axiom, ((~(leoLit(leoTi(leoAt( ... , inference(fof_translation, [status(thm)],[71])).
 ...
 cnf(128,plain,($false),inference(rw, [status(thm)],[114,115,theory(equality)])).
 cnf(129,plain,($false),inference(cn,[status(thm)],[128, theory(equality,[symmetry])])).
 thf(130,plain,((($false)=$true)),inference(fo_atp_e,[status(thm)],[129])).
 thf(131,plain,($false),inference(solved_all_splits,[solved_all_splits(join,[])],[130])).
% SZS output end CNFRefutation
```

- very brittle for various reasons → PxTP Discussion?

B: Mapping of EP proofs

Since version 1.4.0; see also [SultanaBenzmüller,IWIL,2012]:

- mapping of EP proofs into LEO-II proofs ('-po 2' option)

```
% SZS status Theorem for SET171^3.p : (rf:0,axioms:0,...,translation:fully-typed)
%**** Beginning of derivation protocol ****
% SZS output start CNFRefutation
 ...
 thf(tp_intersection,type,(intersection: (($i>$o)>(($i>$o)>($i>$o))))).
 thf(tp_union,type,(union: (($i>$o)>(($i>$o)>($i>$o))))).
 ...
 thf(union,definition,(union = (^[X:($i>$o),Y:($i>$o),U:$i]: ((X@U) |
% (Y@U)))),
     file('SET171^3.p',union)).
 ...
 thf(1,conjecture,(![A:($i>$o),B:($i>$o),C:($i>$o)]:
     (((union@A)@((intersection@B)@C)) = ((intersection@((union@A)@B))@((union@A)@C)))),
     file('SET171^3.p',union_distributes_over_intersection)).
 ...
 fof(74, axiom, ((leoLit(leoTi(leoAt( ... , inference(fof_translation, [status(thm)],[51])).
 fof(77, axiom, ((leoLit(leoTi(leoAt( ... , inference(fof_translation, [status(thm)],[54])).
 fof(78, axiom, ((~(leoLit(leoTi(leoAt( ... , inference(fof_translation, [status(thm)],[58])).
 fof(85, axiom, ((~(leoLit(leoTi(leoAt( ... , inference(fof_translation, [status(thm)],[71])).
 ...
 cnf(128,plain,($false),inference(rw, [status(thm)],[114,115,theory(equality)])).
 cnf(129,plain,($false),inference(cn,[status(thm)],[128, theory(equality,[symmetry])])).
 thf(130,plain,((($false)=$true)),inference(fo_atp_e,[status(thm)],[129])).
 thf(131,plain,($false),inference(solved_all_splits,[solved_all_splits(join,[])],[130])).
% SZS output end CNFRefutation
```

- very brittle for various reasons → PxTP Discussion?

B: Mapping of EP proofs

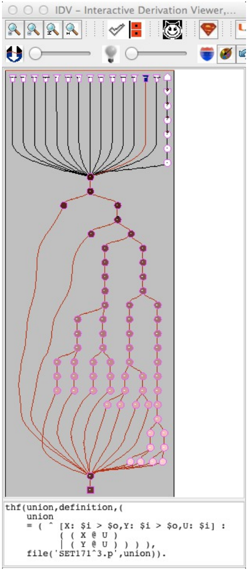
Since version 1.4.0; see also [SultanaBenzmüller,IWIL,2012]:

- mapping of EP proofs into LEO-II proofs ('-po 2' option)

```
% SZS status Theorem for SET171^3.p : (rf:0,axioms:0,...,translation:fully-typed)
%**** Beginning of derivation protocol ****
% SZS output start CNFRefutation
 ...
 thf(tp_intersection,type,(intersection: (($i>$o)>(($i>$o)>($i>$o))))).
 thf(tp_union,type,(union: (($i>$o)>(($i>$o)>($i>$o))))).
 ...
 thf(union,definition,(union = (^[X:($i>$o),Y:($i>$o),U:$i]: ((X@U) |
% (Y@U)))),
    file('SET171^3.p',union)).
 ...
 thf(1,conjecture,(![A:($i>$o),B:($i>$o),C:($i>$o)]:
    (((union@A)@((intersection@B)@C)) = ((intersection@((union@A)@B))@((union@A)@C)))),
    file('SET171^3.p',union_distributes_over_intersection)).
 ...
 fof(74, axiom, ((leoLit(leoTi(leoAt( ... , inference(fof_translation, [status(thm)],[51])).
 fof(77, axiom, ((leoLit(leoTi(leoAt( ... , inference(fof_translation, [status(thm)],[54])).
 fof(78, axiom, ((~(leoLit(leoTi(leoAt( ... , inference(fof_translation, [status(thm)],[58])).
 fof(85, axiom, ((~(leoLit(leoTi(leoAt( ... , inference(fof_translation, [status(thm)],[71])).
 ...
 cnf(128,plain,($false),inference(rw, [status(thm)],[114,115,theory(equality)])).
 cnf(129,plain,($false),inference(cn,[status(thm)],[128, theory(equality,[symmetry])])).
 thf(130,plain,((($false)=$true)),inference(fo_atp_e,[status(thm)],[129])).
 thf(131,plain,($false),inference(solved_all_splits,[solved_all_splits(join,[])],[130])).
% SZS output end CNFRefutation
```

- very brittle for various reasons → PxTP Discussion?

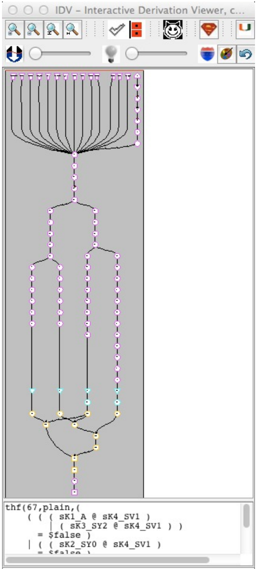
B: Mapping of EP proofs



'-po 1'

TSTP tools are applicable

IDV [TracPuzisSutcliffe,ENTCS,2007] visualization of (SET171^3.p)



'-po 2'

B: Improved support for back-end provers

Freie Universität Berlin

Back-end provers in LEO-II

- first choice: EP
- new: better support for SPASS, Vampire and others
- new: support for remote provers on SystemOnTPTP
- ongoing: parallelization of EP, SPASS, Vampire
- ongoing: incremental Z3

Experiment — TPTP v5.4.0; LEO-II timout 60s; FO-ATP timeout 30s

- no. of problems exclusively proved
  LEO-II(E): 37 LEO-II(SPASS): 5 LEO-II(Vampire): 20
- no. of missed problems which one of the others could solve
  LEO-II(E): 31 LEO-II(SPASS): 95 LEO-II(Vampire): 98

B: Improved support for back-end provers

Back-end provers in LEO-II

- first choice: EP
- new: better support for SPASS, Vampire and others
- new: support for remote provers on SystemOnTPTP
- ongoing: parallelization of EP, SPASS, Vampire
- ongoing: incremental Z3

Experiment — TPTP v5.4.0; LEO-II timout 60s; FO-ATP timeout 30s

- no. of problems exclusively proved
  LEO-II(E): 37 LEO-II(SPASS): 5 LEO-II(Vampire): 20
- no. of missed problems which one of the others could solve
  LEO-II(E): 31 LEO-II(SPASS): 95 LEO-II(Vampire): 98

B: Detection/removal of Leibniz- and Andrews-equality

$$\lambda X_\alpha \lambda Y_\alpha \forall P_{\alpha\to o}.\ P\ X \Rightarrow P\ Y$$

$$\lambda X_\alpha \lambda Y_\alpha \forall Q_{\alpha\to\alpha\to o}.\ \forall Z_\alpha (Q\ Z\ Z) \Rightarrow Q\ X\ Y$$

They support cut-simulation due to their impredicative nature.

We added two new rules to the calculus

$$\frac{\mathbf{C}\ \vee\ [P\ \mathbf{A}]^{\mathrm{ff}}\ \vee\ [P\ \mathbf{B}]^{\mathrm{tt}}}{\mathbf{C}\{\lambda X.\ \mathbf{A} = X/P\} \vee [\mathbf{A} = \mathbf{B}]^{\mathrm{tt}}}\ \text{LeibEQ} \qquad \frac{\mathbf{C}\ \vee\ [P\ \mathbf{A}\ \mathbf{A}]^{\mathrm{ff}}}{\mathbf{C}\{\lambda X \lambda Y.\ X = Y/P\}}\ \text{AndrEQ}$$

These rules are obviously sound.

Some TPTP problems with rating 1.0 can now be solved:
SYO246^5.p, SYO244^5.p, NUM817^5.p, NUM816^5.p, NUM814^5.p.

Use of primitive substitution (blind guessing) can often be avoided.

B: Detection/removal of Leibniz- and Andrews-equality

$$\lambda X_\alpha \lambda Y_\alpha \forall P_{\alpha\to o}.\ P\ X \Rightarrow P\ Y$$

$$\lambda X_\alpha \lambda Y_\alpha \forall Q_{\alpha\to\alpha\to o}.\ \forall Z_\alpha (Q\ Z\ Z) \Rightarrow Q\ X\ Y$$

They support cut-simulation due to their impredicative nature.

We added two new rules to the calculus

$$\frac{\mathbf{C}\ \vee\ [P\ \mathbf{A}]^{ff}\ \vee\ [P\ \mathbf{B}]^{tt}}{\mathbf{C}\{\lambda X.\ \mathbf{A} = X/P\} \vee [\mathbf{A} = \mathbf{B}]^{tt}}\ \text{LeibEQ} \qquad \frac{\mathbf{C}\ \vee\ [P\ \mathbf{A}\ \mathbf{A}]^{ff}}{\mathbf{C}\{\lambda X \lambda Y.\ X = Y/P\}}\ \text{AndrEQ}$$

These rules are obviously sound.

Some TPTP problems with rating 1.0 can now be solved:
SYO246^5.p, SYO244^5.p, NUM817^5.p, NUM816^5.p, NUM814^5.p.

Use of primitive substitution (blind guessing) can often be avoided.

B: Detection/removal of Leibniz- and Andrews-equality

$$\lambda X_\alpha \lambda Y_\alpha \forall P_{\alpha \to o}.\ P\ X \Rightarrow P\ Y$$

$$\lambda X_\alpha \lambda Y_\alpha \forall Q_{\alpha \to \alpha \to o}.\ \forall Z_\alpha (Q\ Z\ Z) \Rightarrow Q\ X\ Y$$

They support cut-simulation due to their impredicative nature.

We added two new rules to the calculus

$$\frac{\mathbf{C} \lor [P\ \mathbf{A}]^{ff} \lor [P\ \mathbf{B}]^{tt}}{\mathbf{C}\{\lambda X.\ \mathbf{A} = X / P\} \lor [\mathbf{A} = \mathbf{B}]^{tt}}\ \text{LeibEQ} \qquad \frac{\mathbf{C} \lor [P\ \mathbf{A}\ \mathbf{A}]^{ff}}{\mathbf{C}\{\lambda X \lambda Y.\ X = Y / P\}}\ \text{AndrEQ}$$

These rules are obviously sound.

Some TPTP problems with rating 1.0 can now be solved:
SYO246^5.p, SYO244^5.p, NUM817^5.p, NUM816^5.p, NUM814^5.p.

Use of primitive substitution (blind guessing) can often be avoided.

B: Support for choice in LEO-II

$$\exists E_{(\alpha\to o)\to\alpha} \forall P_{(\alpha\to o)}.\ \exists X_\alpha (P\ X) \Rightarrow P\ (E\ P)$$

Partial support for choice before LEO-II 1.5 (naïve Skolemization).

B: Support for choice in LEO-II

$$\exists E_{(\alpha \to o) \to \alpha} \forall P_{(\alpha \to o)}. \exists X_\alpha (P\, X) \Rightarrow P\, (E\, P)$$

Partial support for choice before LEO-II 1.5 (naïve Skolemization).

Instances of AC axiom scheme could be added:

$$\exists E_{(\iota \to o) \to \iota} \forall P_{(\iota \to o)}. \exists X_\iota (P\, X) \Rightarrow P\, (E\, P)$$

However, such impredicative axioms support cut-simulation.

B: Support for choice in LEO-II

$$\exists E_{(\alpha\to o)\to\alpha} \forall P_{(\alpha\to o)}.\ \exists X_\alpha (P\ X) \Rightarrow P\ (E\ P)$$

We added two new rules (the set CFs maintains choice functions and is initialized with one choice function for each type).

$$\frac{[PX]^{\mathrm{ff}} \vee [P(f_{(\alpha\to o)\to\alpha} P)]^{\mathrm{tt}}}{\mathrm{CFs} \longleftarrow \mathrm{CFs} \cup \{f_{(\alpha\to o)\to\alpha}\}}\ \mathrm{detectChoiceFn}$$

$$\frac{C := \mathbf{C}' \vee [\mathbf{A}[E_{(\alpha\to o)\to\alpha}\mathbf{B}]]^p \qquad \begin{array}{c}\epsilon \in \mathrm{CFs},\ E = \epsilon \textit{ or } E \in \textit{freeVars}(C), \\ \textit{freeVars}(\mathbf{B}) \subseteq \textit{freeVars}(C),\ Y \textit{ fresh}\end{array}}{[\mathbf{B}\ Y]^{\mathrm{ff}} \vee [\mathbf{B}\ (\epsilon_{(\alpha\to o)\to\alpha}\mathbf{B})]^{\mathrm{tt}}}\ \mathrm{choice}$$

Rule choice is related to [Mints,JSL,1999].
Both rules are obviously sound.

B: Further improvements

- detection of satisfiable resp. countersatisfiable problems
  (supporting choice was essential for achieving this)
- improved support for flexible strategy scheduling
  (but: we still do not have good schedules!)
- reimplementation of depth-bounded extensional pre-unification
  (extensionality can now be disabled)
- parser, status reporting, avoiding redundant computations, factorisation, subsumption, clause selection, ...

B: Experiments

SZS Status	fully-typed	fof_full	fof_experiment
Thm	64.8	64.9	65.3
All	60.9	61	61.3

Table : Comparing FOL encodings in LEO-II version 1.5 (30s timeout). Table shows the percentage of matches between LEO-II's SZS output and the 'Status' field of problems.

Timeout (s)	v1.2		v1.4.3		v1.5	
	Thm	*All*	*Thm*	*All*	*Thm*	*All*
30	58.4	51.1	62.1	54.4	64.3	61.3
60	58.7	51.3	65	56.9	67.1	62.9

Table : Percentage match between different versions of LEO-II and the Status field of TPTP problems. LEO-II v1.2 was the winner of the CASC competition in 2010, and v1.4.3 was the last public release. Version 1.5 was run with the `fof_experiment` encoding.

B: Experiments

SZS Status	`fully-typed`	`fof_full`	`fof_experiment`
Thm	64.8	64.9	65.3
All	60.9	61	61.3

Table : Comparing FOL encodings in LEO-II version 1.5 (30s timeout). Table shows the percentage of matches between LEO-II's SZS output and the 'Status' field of problems.

Timeout (s)	**v1.2**		**v1.4.3**		**v1.5**	
	Thm	*All*	*Thm*	*All*	*Thm*	*All*
30	58.4	51.1	62.1	54.4	64.3	61.3
60	58.7	51.3	65	56.9	67.1	62.9

Table : Percentage match between different versions of LEO-II and the Status field of TPTP problems. LEO-II v1.2 was the winner of the CASC competition in 2010, and v1.4.3 was the last public release. Version 1.5 was run with the `fof_experiment` encoding.

LEO-II

- strongly collaborates with FO-ATPs
- proof exchange/verification is thus an important issue
- version 1.5 of LEO-II has several new, interesting features, some performance gain on TPTP (but not overwhelming yet)

Btw, did you know that LEO-II

- paralleled and strongly influenced the development of THF0 (EU project with Geoff Sutcliffe)
- has been the first prover to accept THF0, FOF and CNF
- is the **only** THF0 prover that has been running at CASC in proof producing mode!

Conclusion: Ongoing and future work

- parallelization of E, Vampire, SPASS
- exploitation of incremental provers (Z3)
- exploitations of term orderings (towards superposition for HOL)
- exploitation of term sharing information
- improvements for choice
- induction
- scheduling / parameter selection
- premise selection