LEO-II

A Higher-Order Theorem Prover

Christoph Benzmüller, Larry Paulson,
Frank Theiss, and Arnaud Fietzke

UNIVERSITY OF CAMBRIDGE Computer Laboratory

UNIVERSITÄT DES SAARLANDES

The LEO-II project at the University of Cambridge is supported by EPSRC Grant EP/D070511/1

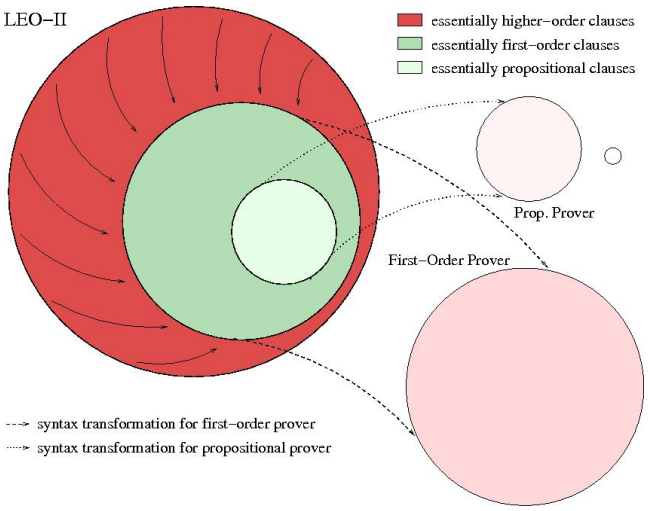
Overview on LEO-II

LEO-II is a standalone, resolution-based higher-order theorem prover that is designed for fruitful cooperation with specialist provers for first-order and propositional logic. The idea is to combine the strengths of the different systems. On the other hand, LEO-II itself, as an external reasoner, wants to support interactive proof assistants such as Isabelle/HOL, HOL, and OMEGA by efficiently automating subproblems and thereby reducing user effort.

LEO-II predominantly addresses higher-order aspects in its reasoning process with the aim to quickly remove higher-order clauses from the search space and to turn them into essentially first-order clauses which can then be refuted with a first-order prover. Furthermore, the project investigates whether techniques that have proved very successful in automated first-order theorem proving, such as shared data structures and term indexing, can be lifted to the higher-order setting.

LEO-II also provides an interactive mode in which user and system can interact to produce resolution proofs in simple type theory.

LEO-II is implemented in Objective Caml; it is the successor of LEO, which was implemented in LISP and hardwired to the OMEGA proof assistant.



Input Syntax: TPTP THF

```
thf(reflexiv,definition,
 (reflexive = (^[R:($i>($i>$o))]: (![X:$i]: ((R @ X) @ X))))).

thf(symmetric,definition,
 (symmetric =
  (^[R:($i>($i>$o))]: (![X:$i,Y:$i]:
   ((R @ X) @ Y) => ((R @ Y) @ X))))).

thf(transitive,definition,
  (transitive =
   (^[R:($i>($i>$o))]: (![X:$i,Y:$i,Z:$i]:
    ((((R @ X) @ Y) & ((R @ Y) @ Z)) => ((R @ X) @ Z)))))).

thf(equiv_rel,definition,
  (equiv_rel =
   (^[R:($i>($i>$o))]:
    (reflexive @ R) & (symmetric @ R) & (transitive @ R)))).

thf(test,conjecture,(?[R:($i>($i>$o))]: ~(equiv_rel @ R))).
```

First Experiments with LEO-II

We evaluate LEO-II's performance on simple problems about sets, relations, and functions. The example problems are taken from the TPTP library and for LEO/Vampire and LEO-II/E they have been reformalized in higher-order logic.

Some examples:

SET171+3	$\forall X_{o\alpha}, Y_{o\alpha}, Z_{o\alpha}.X \cup (Y \cap Z) = (X \cup Y) \cap (X \cup Z)$
SET611+3	$\forall X_{o\alpha}, Y_{o\alpha}.(X \cap Y = \emptyset) \Leftrightarrow (X \setminus Y = X)$
SET624+3	$\forall X_{o\alpha}, Y_{o\alpha}, Z_{o\alpha}.$ $\mathsf{Meets}(X, Y \cap Z) \Leftrightarrow \mathsf{Meets}(X, Y) \vee \mathsf{Meets}(X, Z)$
SET646+3	$\forall x_\alpha, y_\beta.\mathsf{Subrel}(\mathsf{Pair}(x, y), (\lambda u_\alpha.T) \times (\lambda v_\beta.T))$
SET670+3	$\forall Z_{o\alpha}, R_{o\beta\alpha}, X_{o\alpha}, Y_{o\beta}.\mathsf{IsRelOn}(R, X, Y) \Rightarrow$ $\mathsf{IsRelOn}(\mathsf{RestrictRDom}(R, Z), Z, Y)$

with

$\_ \in \_$	$=$	$\lambda x_\alpha, A_{o\alpha}.[Ax]$
$\emptyset$	$=$	$[\lambda x_\alpha.F]$
$\_ \cap \_$	$=$	$\lambda A_{o\alpha}, B_{o\alpha}.[\lambda x_\alpha.x \in A \wedge x \in B]$
$\_ \cup \_$	$=$	$\lambda A_{o\alpha}, B_{o\alpha}.[\lambda x_\alpha.x \in A \vee x \in B]$
$\_ \setminus \_$	$=$	$\lambda A_{o\alpha}, B_{o\alpha}.[\lambda x_\alpha.x \in A \vee x \notin B]$
$\mathsf{Meets}(\_,\_)$	$=$	$\lambda A_{o\alpha}, B_{o\alpha}.[\exists x_\alpha.x \in A \wedge x \in B]$
$\mathsf{Pair}(\_,\_)$	$=$	$\lambda x_\alpha, y_\beta.[\lambda u_\alpha, v_\beta.u = x \wedge v = y]$
$\_ \times \_$	$=$	$\lambda A_{o\alpha}, B_{o\beta}.[\lambda u_\alpha, v_\beta.u \in A \wedge v \in B]$
$\mathsf{Subrel}(\_,\_)$	$=$	$\lambda R_{o\beta\alpha}, Q_{o\beta\alpha}.[\forall x_\alpha, y_\beta.Rxy \Rightarrow Qxy]$
$\mathsf{IsRelOn}(\_,\_,\_)$	$=$	$\lambda R_{o\beta\alpha}, A_{o\alpha}, B_{o\beta}.[\forall x_\alpha, y_\beta.Rxy$ $\Rightarrow x \in A \wedge y \in B]$
$\mathsf{RestrictRDom}(\_,\_)$	$=$	$\lambda R_{o\beta\alpha}, A_{o\alpha}.[\lambda x_\alpha, y_\beta.x \in A \wedge Rxy]$

Cooperation with Other Provers

Provers supported (so far): E, SPASS

Translations supported so far

[Kerber94] $(V^0_{\iota\to\iota\to o}\ V^1_\iota\ V^1_\iota)$ translates to
$@_{(\iota\to o)\to\iota\to o}(@_{(\iota\to\iota\to o)\to\iota\to(\iota\to o)}(V^0, V^1), V^1)$

[Hurd02] $(V^0_{\iota\to\iota\to o}\ V^1_\iota\ V^1_\iota)$ translates to
$ti(@(ti(@(ti(ti(V^0, \iota\to\iota\to o), ti(V^1, \iota)), \iota\to o), ti(V^1, \iota)), o)$

Results

Problem	Vampire 9.0[1]	LEO/Vamp.[2]	LEO-II/E[3]
SET014+4	114.5	2.60	0.300
SET017+1	1.0	5.05	0.059
SET066+1	–	3.73	0.029
SET067+1	4.6	0.10	0.040
SET076+1	51.3	0.97	0.031
SET086+1	0.1	0.01	0.028
SET096+1	5.9	7.29	0.033
SET143+3	0.1	0.31	0.034
SET171+3	68.6	0.38	0.030
SET580+3	0.0	0.23	0.078
SET601+3	1.6	1.18	0.089
SET606+3	0.1	0.27	0.033
SET607+3	1.2	0.26	0.036
SET609+3	145.2	0.49	0.039
SET611+3	0.3	4.00	0.125
SET612+3	111.9	0.46	0.030
SET614+3	3.7	0.41	0.060
SET615+3	103.9	0.47	0.035
SET623+3	–	2.27	0.282
SET624+3	3.8	3.29	0.047
SET630+3	0.1	0.05	0.025
SET640+3	1.1	0.01	0.033
SET646+3	84.4	0.01	0.032
SET647+3	98.2	0.12	0.037
SET648+3	98.2	0.12	0.037
SET649+3	117.5	0.25	0.037
SET651+3	117.5	0.09	0.029
SET657+3	146.6	0.01	0.028
SET669+3	83.1	0.20	0.041
SET670+3	–	0.14	0.067
SET671+3	214.9	0.47	0.038
SET672+3	–	0.23	0.034
SET673+3	217.1	0.47	0.042
SET680+3	146.3	2.38	0.035
SET683+3	0.3	0.27	0.053
SET684+3	–	3.39	0.039
SET716+4	–	0.40	0.033
SET724+4	–	1.91	0.032
SET741+4	–	3.70	0.042
SET747+4	–	1.18	0.028
SET752+4	–	516.00	0.086
SET753+4	–	1.64	0.037
SET764+4	0.1	0.01	0.032
SET770+4	145.0	–	–

[1] Intel(R) Pentium(R) 4 CPU 2.80GHz, 1GB, Linux, CPULimit 600s
[2] Intel(R) Xeon(TM) 4 CPU 2.40GHz, 4GB, Linux, CPULimit 120s
[3] Intel(R) Pentium(R) 1 CPU 1.60GHz, 1GB, Linux, CPULimit 60s