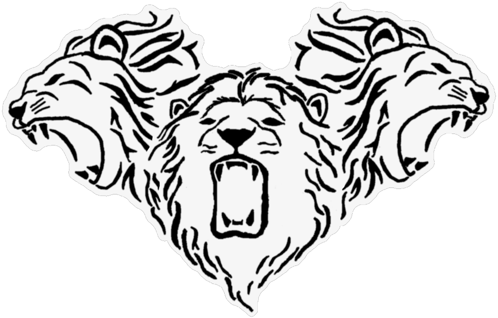


The Higher-Order Prover Leo-III

Alexander Steen[1,2], jww. C. Benzmüller and M. Wisniewski[1]
Freie Universität Berlin

Matryoshka Workshop 2018, Amsterdam

[1] This author has been supported by the DFG under grant BE 2501/11-1 (Leo-III).
[2] This author has been supported by the Volkswagenstiftung (project CRAP).

Talk outline

1. Higher-Order Logic (HOL)
2. The Leo-III Prover
3. Automation of Non-Classical Logics
4. Summary
5. Live Demo (optional)

Higher Order Logic (HOL)

Based on Church's "Simple type theory" (typed λ-calculus) [Church,1940]
More specifically: Extentional Type Theory (ExTT) [Henkin, JSL, 1950]

Syntax

- Simple types $\mathcal{T}$ generated by base types and $\longrightarrow$
- Typically, base types are o and i

Based on Church's "Simple type theory" (typed λ-calculus) [Church,1940]
More specifically: Extentional Type Theory (ExTT) [Henkin, JSL, 1950]

Syntax

- Simple types $\mathcal{T}$ generated by base types and $\rightarrow$
- Typically, base types are o and i
- Terms defined by ($\tau, \nu \in \mathcal{T}$)

$$s, t ::= c_\tau \in \Sigma \mid X_\tau \in \mathcal{V}$$

- Primitive logical connectives ($\tau \in \mathcal{T}$)

$$\{\neg_{o\rightarrow o}, \vee_{o\rightarrow o\rightarrow o}, \Pi^\tau_{(\tau\rightarrow o)\rightarrow o}, =^\tau_{\tau\rightarrow\tau\rightarrow o}\} \subseteq \Sigma$$

Higher Order Logic (HOL)

Based on Church's "Simple type theory" (typed λ-calculus) [Church,1940]
More specifically: Extentional Type Theory (ExTT) [Henkin, JSL, 1950]

Syntax

- Simple types $\mathcal{T}$ generated by base types and $\rightarrow$
- Typically, base types are o and i

Type of truth-values

- Terms defined by ($\tau, \nu \in \mathcal{T}$)

$$s, t ::= c_\tau \in \Sigma \mid X_\tau \in \mathcal{V}$$

- Primitive logical connectives ($\tau \in \mathcal{T}$)

$$\{\neg_{o\rightarrow o}, \vee_{o\rightarrow o\rightarrow o}, \Pi^{\tau}_{(\tau\rightarrow o)\rightarrow o}, =^{\tau}_{\tau\rightarrow\tau\rightarrow o}\} \subseteq \Sigma$$

Higher Order Logic (HOL)

Based on Church's "Simple type theory" (typed λ-calculus) [Church,1940]
More specifically: Extentional Type Theory (ExTT) [Henkin, JSL, 1950]

Syntax

- Simple types $\mathcal{T}$ generated by base types and $\rightarrow$
- Typically, base types are o and i

Type of individuals

- Terms defined by ($\tau, \nu \in \mathcal{T}$)

$$s, t ::= c_\tau \in \Sigma \mid X_\tau \in \mathcal{V}$$

- Primitive logical connectives ($\tau \in \mathcal{T}$)

$$\{\neg_{o\to o}, \vee_{o\to o\to o}, \Pi^\tau_{(\tau\to o)\to o}, =^\tau_{\tau\to\tau\to o}\} \subseteq \Sigma$$

Based on Church's "Simple type theory" (typed λ-calculus) [Church,1940]
More specifically: Extentional Type Theory (ExTT) [Henkin, JSL, 1950]

Syntax

- Simple types $\mathcal{T}$ generated by base types and $\rightarrow$
- Typically, base types are o and i

- Terms defined by ($\tau, \nu \in \mathcal{T}$)

$$s, t ::= c_\tau \in \Sigma \mid X_\tau \in \mathcal{V}$$

- Primitive logical connectives ($\tau \in \mathcal{T}$)

$$\{\neg_{o\rightarrow o}, \vee_{o\rightarrow o\rightarrow o}, \Pi^\tau_{(\tau\rightarrow o)\rightarrow o}, =^\tau_{\tau\rightarrow\tau\rightarrow o}\} \subseteq \Sigma$$

Higher Order Logic (HOL)

Based on Church's "Simple type theory" (typed λ-calculus) [Church,1940]
More specifically: Extentional Type Theory (ExTT) [Henkin, JSL, 1950]

Syntax

- Simple types $\mathcal{T}$ generated by base types and $\rightarrow$
- Typically, base types are o and i

- Terms defined by ($\tau, \nu \in \mathcal{T}$)

$$s, t ::= c_\tau \in \Sigma \mid X_\tau \in \mathcal{V} \\ \mid (\lambda X_\tau . s_\nu)_{\tau\rightarrow\nu} \mid (s_{\tau\rightarrow\nu}\, t_\tau)_\nu$$

- Primitive logical connectives ($\tau \in \mathcal{T}$)

$$\{\neg_{o\rightarrow o}, \vee_{o\rightarrow o\rightarrow o}, \Pi^\tau_{(\tau\rightarrow o)\rightarrow o}, =^\tau_{\tau\rightarrow\tau\rightarrow o}\} \subseteq \Sigma$$

Higher Order Logic (HOL)

Based on Church's "Simple type theory" (typed λ-calculus) [Church,1940]
More specifically: Extentional Type Theory (ExTT) [Henkin, JSL, 1950]

Syntax

- Simple types $\mathcal{T}$ generated by base types and $\rightarrow$
- Typically, base types are o and i

- Terms defined by ($\tau, \nu \in \mathcal{T}$)

$$s, t ::= c_\tau \in \Sigma \mid X_\tau \in \mathcal{V}$$
$$\mid (\lambda X_\tau .\, s_\nu)_{\tau\rightarrow\nu} \mid (s_{\tau\rightarrow\nu}\, t_\tau)_\nu$$

- Primitive logical connectives ($\tau \in \mathcal{T}$)

$$\left\{\neg_{o\rightarrow o}, \vee_{o\rightarrow o\rightarrow o}, \Pi^{\tau}_{(\tau\rightarrow o)\rightarrow o}, =^{\tau}_{\tau\rightarrow\tau\rightarrow o}\right\} \subseteq \Sigma$$

Freie Universität Berlin

Semantics

- Leo-III automates HOL with Henkin semantics
- Some valid axioms (axiom schemes):
  - Boolean Extensionality

$$\mathrm{EXT}^{o} := \forall P_o.\, \forall Q_o.\, (P \Leftrightarrow Q) \Rightarrow P =^{o} Q$$

  - Functional Extensionality

$$\mathrm{EXT}^{\nu\tau} := \forall F_{\nu\tau}.\, \forall G_{\nu\tau}.\, (\forall X_\tau.\, F\, X =^{\nu} G\, X) \Rightarrow F =^{\nu\tau} G$$

  - Type-restricted comprehension

$$\mathrm{COM}^{\tau,\nu} := \forall G_\nu.\, \exists F_{\nu\overline{\tau^n}}.\, \forall \overline{X^n}.\, F\, \overline{X^n} = G_\nu$$

- Further semantics exist:
  - Without Extensionality ⇝ Elementary Type Theory [Andrews, 1974]
  - Intermediate systems [Benzmüller et al.,2004]
  - Andrews' v-complexes [Andrews, 1971]
  - Intensional models [Muskens, 2007]

Higher Order Logic (HOL), cont.

Semantics

- Leo-III automates HOL with Henkin semantics
- Some valid axioms (axiom schemes):
  - Boolean Extensionality

$$\mathrm{EXT}^{o} := \forall P_o.\, \forall Q_o.\, (P \Leftrightarrow Q) \Rightarrow P =^{o} Q$$

  - Functional Extensionality

$$\mathrm{EXT}^{\nu\tau} := \forall F_{\nu\tau}.\, \forall G_{\nu\tau}.\, (\forall X_\tau.\, F\, X =^{\nu} G\, X) \Rightarrow F =^{\nu\tau} G$$

  - Type-restricted comprehension

$$\mathrm{COM}^{\tau,\nu} := \forall G_\nu.\, \exists F_{\nu\overline{\tau^n}}.\, \forall \overline{X^n}.\, F\, \overline{X^n} = G_\nu$$

- Further semantics exist:
  - Without Extensionality ⇝ Elementary Type Theory [Andrews, 1974]
  - Intermediate systems [Benzmüller et al.,2004]
  - Andrews' v-complexes [Andrews, 1971]
  - Intensional models [Muskens, 2007]

Higher Order Logic (HOL), cont.

Semantics

- Leo-III automates HOL with Henkin semantics
- Some valid axioms (axiom schemes):
  - Boolean Extensionality

$$\text{EXT}^{o} := \forall P_o.\, \forall Q_o.\, (P \Leftrightarrow Q) \Rightarrow P =^{o} Q$$

  - Functional Extensionality

$$\text{EXT}^{\nu\tau} := \forall F_{\nu\tau}.\, \forall G_{\nu\tau}.\, (\forall X_\tau.\, F\, X =^{\nu} G\, X) \Rightarrow F =^{\nu\tau} G$$

  - Type-restricted comprehension

$$\text{COM}^{\tau,\nu} := \forall G_\nu.\, \exists F_{\nu\overline{\tau^n}}.\, \forall \overline{X^n}.\, F\, \overline{X^n} = G_\nu$$

- Further semantics exist:
  - Without Extensionality ⇝ Elementary Type Theory [Andrews, 1974]
  - Intermediate systems [Benzmüller et al.,2004]
  - Andrews' *v*-complexes [Andrews, 1971]
  - Intensional models [Muskens, 2007]

Higher Order Logic (HOL), cont.

Semantics

- Leo-III automates HOL with Henkin semantics
- Some valid axioms (axiom schemes):
  - Boolean Extensionality

$$\mathrm{EXT}^{o} := \forall P_o.\, \forall Q_o.\, (P \Leftrightarrow Q) \Rightarrow P =^{o} Q$$

  - Functional Extensionality

$$\mathrm{EXT}^{\nu\tau} := \forall F_{\nu\tau}.\, \forall G_{\nu\tau}.\, (\forall X_\tau.\, F\, X =^{\nu} G\, X) \Rightarrow F =^{\nu\tau} G$$

  - Type-restricted comprehension

$$\mathrm{COM}^{\tau,\nu} := \forall G_\nu.\, \exists F_{\nu\overline{\tau^n}}.\, \forall \overline{X^n}.\, F\, \overline{X^n} = G_\nu$$

- Further semantics exist:
  - Without Extensionality ⇝ Elementary Type Theory [Andrews, 1974]
  - Intermediate systems [Benzmüller et al.,2004]
  - Andrews' v-complexes [Andrews, 1971]
  - Intensional models [Muskens, 2007]

Higher Order Logic (HOL), cont.

Semantics

- Leo-III automates HOL with Henkin semantics
- Some valid axioms (axiom schemes):
  - Boolean Extensionality

$$\mathrm{EXT}^{o} \quad := \forall P_o .\, \forall Q_o .\, (P \Leftrightarrow Q) \Rightarrow P =^{o} Q$$

  - Functional Extensionality

$$\mathrm{EXT}^{\nu\tau} \quad := \forall F_{\nu\tau} .\, \forall G_{\nu\tau} .\, (\forall X_\tau .\, F\, X =^{\nu} G\, X) \Rightarrow F =^{\nu\tau} G$$

  - Type-restricted comprehension

$$\mathrm{COM}^{\tau,\nu} := \forall G_\nu .\, \exists F_{\nu\overline{\tau^n}} .\, \forall \overline{X^n} .\, F\, \overline{X^n} = G_\nu$$

- Further semantics exist:
  - Without Extensionality ⇝ Elementary Type Theory [Andrews, 1974]
  - Intermediate systems [Benzmüller et al.,2004]
  - Andrews' *v*-complexes [Andrews, 1971]
  - Intensional models [Muskens, 2007]

Talk outline

1. Higher-Order Logic (HOL)
2. **The Leo-III Prover**
3. Automation of Non-Classical Logics
4. Summary
5. Live Demo (optional)

Evolution of the Leo Provers

LEO-I [Benzmüller et al.,CADE,1998] **(1997–2006 at Saarbrücken/Birmingham)**

- Extensional higher-order RUE-resolution approach
- Pioneered higher-order—first-order cooperation (E prover)
- Hard-wired to the ΩMEGA proof assistant

LEO-II [Benzmüller et al.,JAR,2015] **(2006-2012 at Cambridge/Berlin)**

- Extensional higher-order RUE-resolution approach
- Primitive equality, first steps towards polymorphism and choice/description,
- Fostered & paralleled the development of TPTP THF (EU FP7 project)
- First CASC winner in THF category in 2010

Evolution of the Leo Provers

LEO-I [Benzmüller et al.,CADE,1998] **(1997–2006 at Saarbrücken/Birmingham)**

- Extensional higher-order RUE-resolution approach
- Pioneered higher-order—first-order cooperation (E prover)
- Hard-wired to the ΩMEGA proof assistant

LEO-II [Benzmüller et al.,JAR,2015] **(2006-2012 at Cambridge/Berlin)**

- Extensional higher-order RUE-resolution approach
- Primitive equality, first steps towards polymorphism and choice/description,
- Fostered & paralleled the development of TPTP THF (EU FP7 project)
- First CASC winner in THF category in 2010

Freie Universität Berlin

Leo-III **(since 2013 at FU Berlin)**

- Extensional higher-order paramodulation
- Primitive equality, choice/description and native polymorphism
- Supports all common TPTP formats: THF, TFF, FOF, CNF
- Strong focus on collaboration with external TFF ATP
- Support for non-classical logics
  - Every normal higher-order modal logic ($\geq$ 200 distinct logics)
  - **new**: Dynadic deontic logic (Carmo/Jones)

Relevant references:

- The Higher-Order Prover Leo-III, IJCAR, 2018 (to appear)
- Theorem Provers for Every Normal Modal Logic, LPAR, 2017
- Effective Normalization Techniques for HOL, IJCAR, 2016
- Agent-Based HOL Reasoning, ICMS, 2016
- LeoPARD – A Generic Platform for the Implementation of HO Reasoners, CICM, 2015

Overview of Leo-III

Leo-III **(since 2013 at FU Berlin)**

- Extensional higher-order paramodulation
- Primitive equality, choice/description and native polymorphism
- Supports all common TPTP formats: THF, TFF, FOF, CNF
- Strong focus on collaboration with external TFF ATP
- Support for non-classical logics
  - Every normal higher-order modal logic ($\geq$ 200 distinct logics)
  - **new**: Dynadic deontic logic (Carmo/Jones)

Relevant references:

- The Higher-Order Prover Leo-III, IJCAR, 2018 (to appear)
- Theorem Provers for Every Normal Modal Logic, LPAR, 2017
- Effective Normalization Techniques for HOL, IJCAR, 2016
- Agent-Based HOL Reasoning, ICMS, 2016
- LeoPARD – A Generic Platform for the Implementation of HO Reasoners, CICM, 2015

The Theory: Extensional Paramodulation

Clausification ... mostly standard ... (but see our IJCAR 2016 paper)

Primary inferences

Paramodulation

$$\frac{\mathcal{C} \vee [l \simeq r]^{\mathfrak{tt}} \qquad \mathcal{D} \vee [s \simeq t]^{\alpha}}{\mathcal{C} \vee \mathcal{D} \vee [s[r]_{\pi} \simeq t]^{\alpha} \vee [s|_{\pi} \simeq l]^{\mathfrak{ff}}} \text{(Para)}$$

The Theory: Extensional Paramodulation

Clausification ... mostly standard ... (but see our IJCAR 2016 paper)

Primary inferences

Paramodulation

$$\frac{\mathcal{C} \vee [l \simeq r]^{\mathfrak{tt}} \qquad \mathcal{D} \vee [s \simeq t]^{\alpha}}{\mathcal{C} \vee \mathcal{D} \vee \; [s[r]_\pi \simeq t]^{\alpha} \;\vee\; [s|_\pi \simeq l]^{\mathfrak{ff}}} \text{ (Para)}$$

Literal: Equation $s \simeq t$ with polarity $\alpha \in \{\mathfrak{tt}, \mathfrak{ff}\}$

The Theory: Extensional Paramodulation

Clausification ... mostly standard ... (but see our IJCAR 2016 paper)

Primary inferences

Paramodulation

$$\frac{\mathcal{C} \vee [l \simeq r]^{\mathfrak{tt}} \qquad \mathcal{D} \vee [s \simeq t]^{\alpha}}{\mathcal{C} \vee \mathcal{D} \vee [s[r]_{\pi} \simeq t]^{\alpha} \vee [s|_{\pi} \simeq l]^{\mathfrak{ff}}} \text{(Para)}$$

Replacement of subterm at position π

The Theory: Extensional Paramodulation

Clausification ... mostly standard ... (but see our IJCAR 2016 paper)

Primary inferences

Paramodulation

$$\frac{\mathcal{C} \vee [l \simeq r]^{\mathtt{tt}} \qquad \mathcal{D} \vee [s \simeq t]^{\alpha}}{\mathcal{C} \vee \mathcal{D} \vee [s[r]_{\pi} \simeq t]^{\alpha} \vee [s|_{\pi} \simeq l]^{\mathtt{ff}}}\ (\text{Para})$$

Unification constraint → $[s|_{\pi} \simeq l]^{\mathtt{ff}}$

The Theory: Extensional Paramodulation

Clausification ... mostly standard ... (but see our IJCAR 2016 paper)

Primary inferences

Paramodulation

$$\frac{\mathcal{C} \vee [l \simeq r]^{\mathtt{tt}} \qquad \mathcal{D} \vee [s \simeq t]^{\alpha}}{\mathcal{C} \vee \mathcal{D} \vee [s[r]_{\pi} \simeq t]^{\alpha} \vee [s|_{\pi} \simeq l]^{\mathtt{ff}}} \text{ (Para)}$$

Factorization

$$\frac{\mathcal{C} \vee [l \simeq r]^{\alpha} \vee [s \simeq t]^{\alpha}}{\mathcal{C} \vee [l \simeq r]^{\alpha} \vee [l \simeq s]^{\mathtt{ff}} \vee [r \simeq t]^{\mathtt{ff}}} \text{ (EqFac)}$$

Primitive substitution

$$\frac{\mathcal{C} \vee [X_{\tau}\, \overline{s^i}]^{\alpha} \qquad g \in \mathcal{GB}_{\tau}^{\{\neg,\vee\} \cup \{\Pi^{\tau}, =^{\tau} | \tau \in \mathcal{T}\}}}{(\mathcal{C} \vee [X_{\tau}\, \overline{s^i}]^{\alpha})\{g/X\}} \text{ (PS)}$$

The Theory: Extensional Paramodulation

Clausification ... mostly standard ... (but see our IJCAR 2016 paper)

Primary inferences

Paramodulation

$$\frac{\mathcal{C} \vee [l \simeq r]^{\mathfrak{tt}} \qquad \mathcal{D} \vee [s \simeq t]^{\alpha}}{\mathcal{C} \vee \mathcal{D} \vee [s[r]_\pi \simeq t]^{\alpha} \vee [s|_\pi \simeq l]^{\mathfrak{ff}}} \text{ (Para)}$$

Factorization

$$\frac{\mathcal{C} \vee [l \simeq r]^{\alpha} \vee [s \simeq t]^{\alpha}}{\mathcal{C} \vee [l \simeq r]^{\alpha} \vee [l \simeq s]^{\mathfrak{ff}} \vee [r \simeq t]^{\mathfrak{ff}}} \text{ (EqFac)}$$

Primitive substitution

$$\frac{\mathcal{C} \vee [X_\tau \, \overline{s^i}]^{\alpha} \qquad g \in \mathcal{GB}_\tau^{\{\neg,\vee\} \cup \{\Pi^\tau, =^\tau | \tau \in \mathcal{T}\}}}{\left(\mathcal{C} \vee [X_\tau \, \overline{s^i}]^{\alpha}\right)\{g/X\}} \text{ (PS)}$$

Freie Universität Berlin

Extensionality rules

(1) Functional Extensionality

$$\frac{\mathcal{C} \vee [s_{\tau\to\nu} \simeq t_{\tau\to\nu}]^{\mathfrak{tt}}}{\mathcal{C} \vee [s\, X_\tau \simeq t\, X_\tau]^{\mathfrak{tt}}}\ \text{(PFE)}$$

where X_τ is a fresh variable

$$\frac{\mathcal{C} \vee [s_{\tau\to\nu} \simeq t_{\tau\to\nu}]^{\mathfrak{ff}}}{\mathcal{C} \vee [s\, \mathrm{sk}_\tau \simeq t\, \mathrm{sk}_\tau]^{\mathfrak{ff}}}\ \text{(NFE)}$$

where sk_τ is a fresh Skolem term

(2) Boolean Extensionality

$$\frac{\mathcal{C} \vee [s_o \simeq t_o]^{\alpha}}{\mathcal{C} \vee [s_o \Leftrightarrow t_o]^{\alpha}}\ \text{(BoolExt)}$$

Pre-unification

... based on Huet's procedure (not displayed here)

The Theory: Extensional Paramodulation (2)

Extensionality rules

(1) Functional Extensionality

$$\frac{\mathcal{C} \vee [s_{\tau\to\nu} \simeq t_{\tau\to\nu}]^{\mathfrak{tt}}}{\mathcal{C} \vee [s\, X_\tau \simeq t\, X_\tau]^{\mathfrak{tt}}}\ \text{(PFE)}$$

where X_τ is a fresh variable

$$\frac{\mathcal{C} \vee [s_{\tau\to\nu} \simeq t_{\tau\to\nu}]^{\mathfrak{ff}}}{\mathcal{C} \vee [s\, \mathrm{sk}_\tau \simeq t\, \mathrm{sk}_\tau]^{\mathfrak{ff}}}\ \text{(NFE)}$$

where sk_τ is a fresh Skolem term

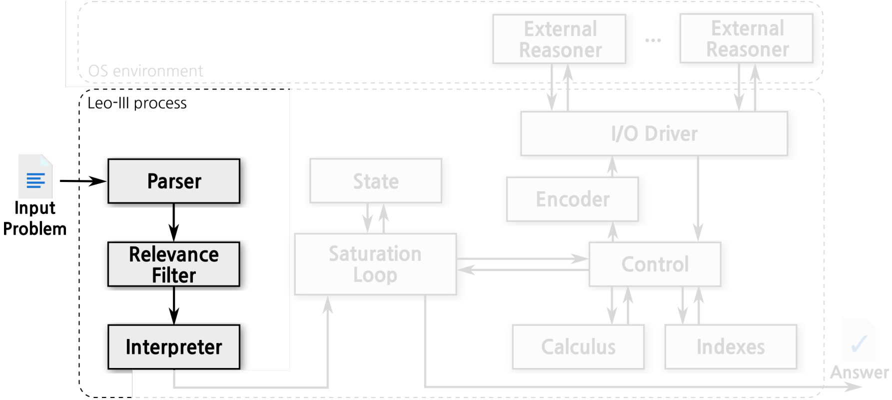
(2) Boolean Extensionality

$$\frac{\mathcal{C} \vee [s_o \simeq t_o]^{\alpha}}{\mathcal{C} \vee [s_o \Leftrightarrow t_o]^{\alpha}}\ \text{(BoolExt)}$$

Pre-unification

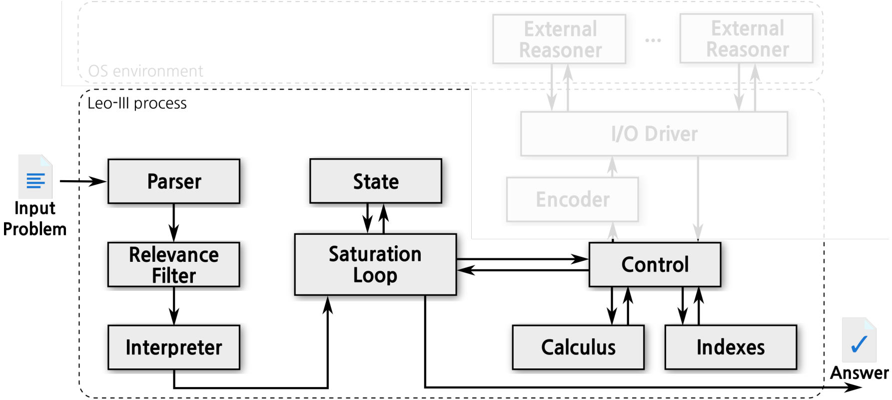
... based on Huet's procedure (not displayed here)

System Architecture



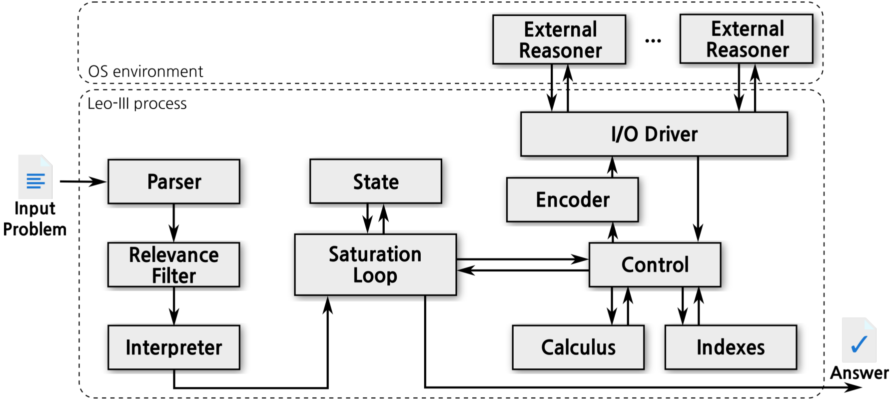
- Saturation based on given-clause algorithm [Schulz,LPAR-19, 2013]
- Asynchronous external cooperation (E, CVC4, iProver, Vampire, ...)

System Architecture



- Saturation based on given-clause algorithm [Schulz,LPAR-19, 2013]
- Asynchronous external cooperation (E, CVC4, iProver, Vampire, ...)

System Architecture



- Saturation based on given-clause algorithm [Schulz,LPAR-19, 2013]
- Asynchronous external cooperation (E, CVC4, iProver, Vampire, ...)

Saturation in practice

Further inference rules include

- Equational simplifications
- Reasoning with choice
- Replacement of defined equalities (Leibniz, Andrews)
- Function synthesis

Inference restrictions

- Depth-limited unification, fixed number of unifiers
- Under-approximation of inference partners
- Heuristic ordering using higher-order term ordering CPO

Proof search

- Selection heuristics for given-clause algorithm
- Eager unification (pattern unification, if possible)
- Restrict to FO-like terms
- Invocation of external reasoners

Saturation in practice

Further inference rules include

- Equational simplifications
- Reasoning with choice
- Replacement of defined equalities (Leibniz, Andrews)
- Function synthesis

Inference restrictions

- Depth-limited unification, fixed number of unifiers
- Under-approximation of inference partners
- Heuristic ordering using higher-order term ordering CPO

Proof search

- Selection heuristics for given-clause algorithm
- Eager unification (pattern unification, if possible)
- Restrict to FO-like terms
- Invocation of external reasoners

Saturation in practice

Further inference rules include

- Equational simplifications
- Reasoning with choice
- Replacement of defined equalities (Leibniz, Andrews)
- Function synthesis

Inference restrictions

- Depth-limited unification, fixed number of unifiers
- Under-approximation of inference partners
- Heuristic ordering using higher-order term ordering CPO

Proof search

- Selection heuristics for given-clause algorithm
- Eager unification (pattern unification, if possible)
- Restrict to FO-like terms
- Invocation of external reasoners

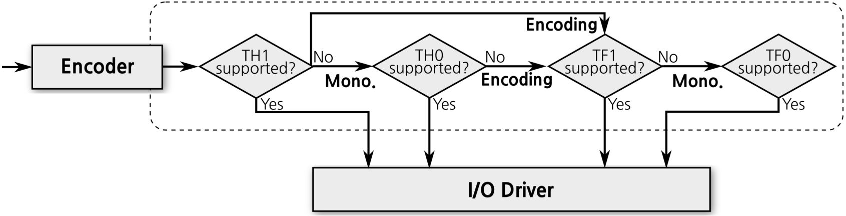
External cooperation

Cooperation with external reasoners

- External cooperation invoked during saturation
- Translate processed clauses to target logic of system
- If unsatisfiable, HO clauses are unsatisfiable as well

Cooperation with external reasoners

- External cooperation invoked during saturation
- Translate processed clauses to target logic of system
- If unsatisfiable, HO clauses are unsatisfiable as well



- Asynchronous communication
- Currently with all TPTP/TSTP-compatible provers
- Focus on typed first-order cooperation (TF1, TF0)

Leo-III Version 1.2

- Reasonably stable ATP system with extensible implementation
- Performance of Leo-III is on a par with established HO ATP systems
- Flexible external cooperation mechanism
- Verifiable proof certificates*

Leo-III Version 1.2

- Reasonably stable ATP system with extensible implementation
- Performance of Leo-III is on a par with established HO ATP systems
- Flexible external cooperation mechanism
- Verifiable proof certificates*

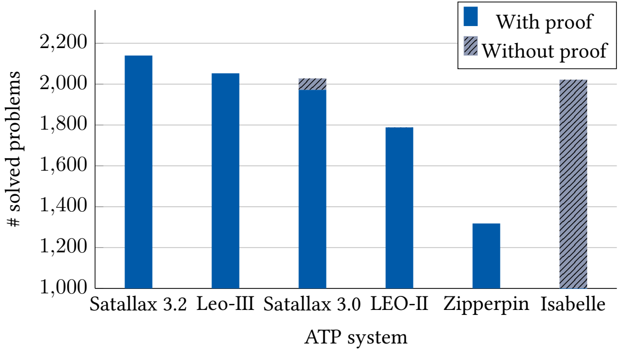


Figure: Benchmark over all TPTP TH0 problems (2463 problems)

Talk outline

1. Higher-Order Logic (HOL)
2. The Leo-III Prover
3. **Automation of Non-Classical Logics**
4. Summary
5. Live Demo (optional)

Reasoning in Non-Classical Logics

- Increasing relevance in various fields
  - Artificial Intelligence (e.g. Agents, Knowledge, Ethics)
  - Computer Linguistics (e.g. Semantics)
  - Mathematics (e.g. Geometry, Category theory)
  - Theoretical Philsophy (e.g. Metaphysics)
- Most powerful ATP/ITP: Classical logic only

Previous focus: Modal logics

- Prover for (propositional) modal logics exist
  - ModLeanTAP, Molle, Bliksem, FaCT++,
  - MOLTAP, KtSeqC, STeP, TRP
  - ...
- Only few for quantified variants
  - MleanTAP, MleanCoP, MleanSeP (J. Otten)
  - f2p+MSPASS
- Enabled by shallow semantical embedding

Reasoning in Non-Classical Logics

- Increasing relevance in various fields
  - Artificial Intelligence (e.g. **Agents, Knowledge**, Ethics)
  - Computer Linguistics (e.g. **Semantics)**
  - Mathematics (e.g. Geometry, Category theory)
  - Theoretical Philsophy (e.g. **Metaphysics)**
- Most powerful ATP/ITP: Classical logic only

Previous focus: Modal logics

- Prover for (propositional) modal logics exist
  - ModLeanTAP, Molle, Bliksem, FaCT++,
  - MOLTAP, KtSeqC, STeP, TRP
  - ...
- Only few for quantified variants
  - MleanTAP, MleanCoP, MleanSeP (J. Otten)
  - f2p+MSPASS
- Enabled by shallow semantical embedding

Reasoning in Non-Classical Logics

- Increasing relevance in various fields
  - Artificial Intelligence (e.g. Agents, Knowledge, Ethics)
  - Computer Linguistics (e.g. Semantics)
  - Mathematics (e.g. Geometry, Category theory)
  - Theoretical Philsophy (e.g. Metaphysics)
- Most powerful ATP/ITP: Classical logic only

Previous focus: Modal logics

- Prover for (propositional) modal logics exist
  - ModLeanTAP, Molle, Bliksem, FaCT++,
  - MOLTAP, KtSeqC, STeP, TRP
  - ...
- Only few for quantified variants
  - MleanTAP, MleanCoP, MleanSeP (J. Otten)
  - f2p+MSPASS
- Enabled by shallow semantical embedding

Problem representation

Current work: Extension of TPTP THF syntax for modal logic

(1) **Formula syntax**

```
thf( classical, axiom, ! [X:$i]: (p @ X)).
```

↓ Extend syntax with modalities

```
thf( modal, axiom, ! [X:$i]: ($box @ (p @ X))).
```

(2) **Semantics configuration**

Add "logic"-annotated statements to the problem:

```
thf( s5_spec , logic , ( $modal := [
  $constants := $rigid,
  $quantification := $cumulative,
  $consequence := $local,
  $modalities := $modal_system_S5 ] ) ).

...(problem statement)...
```

- Intended semantics is attached to the problem
- User can flexibly adjust semantical setting

Problem representation

Current work: Extension of TPTP THF syntax for modal logic

(1) **Formula syntax**

```
thf( classical, axiom, ! [X:$i]: (p @ X)).
```

↓ Extend syntax with modalities

```
thf( modal, axiom, ! [X:$i]: ($box @ (p @ X))).
```

(2) **Semantics configuration**

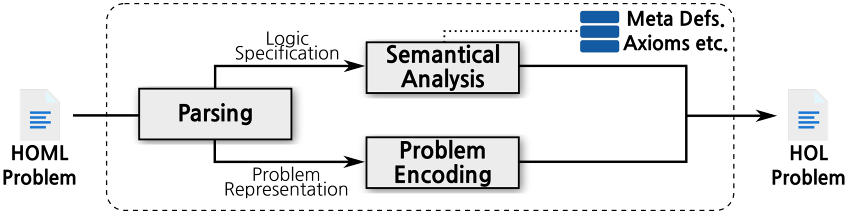
Add "logic"-annotated statements to the problem:

```
thf( s5_spec , logic , ( $modal := [
  $constants := $rigid,
  $quantification := $cumulative,
  $consequence := $local,
  $modalities := $modal_system_S5 ] ) ).

...(problem statement)...
```

- Intended semantics is attached to the problem
- User can flexibly adjust semantical setting

Embedding procedure directly included into Leo-III



- Technical details are hidden from the user
  - Semantic specification is analyzed first
  - Definitions of logical and meta-logical notions are included
  - The problem itself is translated
  - Output format: Plain (classical) THF
- Also available as external pre-processing tool

Performance of Leo-III

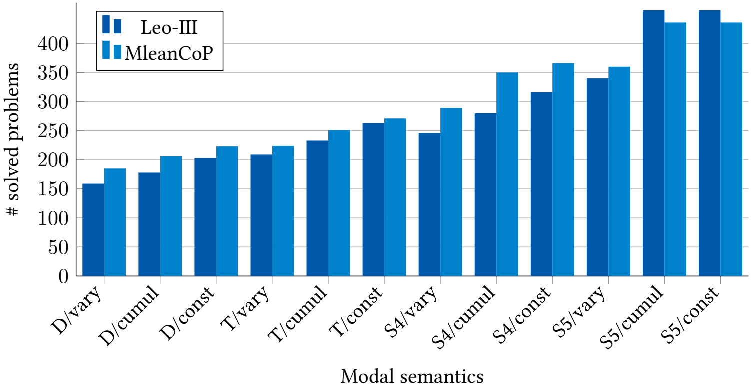


Figure: Benchmark over all monomonodal QMLTP problems (580 problems)

- Modal logic reasoning is competitive with special purpose reasoners
- More supported semantical settings (not shown here)

Most recent work

Brand new: Support for Dyadic Deontic Logic (Carmo/Jones)

- Based on another embedding [Benzüller,2018]
- Enhance propositional TPTP fragment with
  1. Dyadic deontic obligation $O(p/q)
  2. Actual/Primary deontic obligations $O_a(p), $O_p(p)
  3. Box operators $box(p), $box_a(p),$box_p(p)
- Integrated into Leo-III (stand-alone tool available)

ASCII	Syntax	Meaning
~	$\neg$	Negation
|	$\vee$	Disjunction
&	$\wedge$	Conjunction
=>	$\Rightarrow$	Material implication
<=>	$\Leftrightarrow$	Equivalence
$O(p/q)	$O(p/q)$	Dyadic deontic obligation (It ought to be *p* given that *q*)
$box(p)	$\Box(p)$	In all worlds *p*

Input statements: ddl(<name>, <role>, <formula>).

Most recent work

Brand new: Support for Dyadic Deontic Logic (Carmo/Jones)

- Based on another embedding [Benzüller,2018]
- Enhance propositional TPTP fragment with
  1. Dyadic deontic obligation `$O(p/q)`
  2. Actual/Primary deontic obligations `$O_a(p)`, `$O_p(p)`
  3. Box operators `$box(p)`, `$box_a(p)`,`$box_p(p)`
- Integrated into Leo-III (stand-alone tool available)

ASCII	Syntax	Meaning
`~`	$\neg$	Negation
`|`	$\vee$	Disjunction
`&`	$\wedge$	Conjunction
`=>`	$\Rightarrow$	Material implication
`<=>`	$\Leftrightarrow$	Equivalence
`$O(p/q)`	$O(p/q)$	Dyadic deontic obligation (It ought to be p given that q)
`$box(p)`	$\Box(p)$	In all worlds p

Input statements: `ddl(<name>, <role>, <formula>).`

Dyadic Deontic Logic cont.

Input statements: ddl(<name>, <role>, <formula>).

where <role> provides meta-logical information
(*Extracted from Christoph's experiments*):

- axiom *assumed, globally valid*
- localAxiom *assumed, valid in current world*
- conjecture *global consequence?*
- localConjecture *consequence in current world?*

Example

This problem can directly be given to Leo-III:

```
ddl(a1, axiom, $O(processDataLawfully)).
ddl(a2, axiom, (~processDataLawfully) => $O(eraseData)).
ddl(a3, localAxiom, ~processDataLawfully).

ddl(c1, conjecture, $O(eraseData)).
```

... giving ...

```
% SZS status Theorem for gdpr.p : 2248 ms resp. 1008 ms w/o parsing
```

Dyadic Deontic Logic cont.

Input statements: ddl(<name>, <role>, <formula>).

where <role> provides meta-logical information
(*Extracted from Christoph's experiments*):

- axiom *assumed, globally valid*
- localAxiom *assumed, valid in current world*
- conjecture *global consequence?*
- localConjecture *consequence in current world?*

Example

This problem can directly be given to Leo-III:

```
ddl(a1, axiom, $O(processDataLawfully)).
ddl(a2, axiom, (~processDataLawfully) => $O(eraseData)).
ddl(a3, localAxiom, ~processDataLawfully).

ddl(c1, conjecture, $O(eraseData)).
```

... giving ...

```
% SZS status Theorem for gdpr.p : 2248 ms resp. 1008 ms w/o parsing
```

Summary

Leo-III is an new HO ATP with

- good performance
- verifiable proof certificates
- and high compatibility with TPTP/TSTP standards

Claim *(please dispute if wrong!)*
No other ATP system is directly applicable to

- more TPTP dialects and/or
- more non-classical logics

How to get it (BSD-3 license): **github.com/leoprover**

- Leo-III 1.2: `leoprover/leoprover`
- LeoPARD: `leoprover/LeoPARD`
- Modal Logic Converter: `leoprover/embedModal`
- Dyadic Deontic Logic Converter: `leoprover/ddl2thf`

Summary

Leo-III is an new HO ATP with

- good performance
- verifiable proof certificates
- and high compatibility with TPTP/TSTP standards

Claim *(please dispute if wrong!)*
No other ATP system is directly applicable to

- more TPTP dialects and/or
- more non-classical logics

How to get it (BSD-3 license): **github.com/leoprover**

- Leo-III 1.2: `leoprover/leoprover`
- LeoPARD: `leoprover/LeoPARD`
- Modal Logic Converter: `leoprover/embedModal`
- Dyadic Deontic Logic Converter: `leoprover/ddl2thf`

Live Demo?