Automated Deduction Modulo

November 8, 2013

David Delahaye
David.Delahaye@cnam.fr

Cnam / Inria, Paris, France

PSATTT'13, École polytechnique, Palaiseau, France

Proof Search in Axiomatic Theories

Current Trends

- Axiomatic theories (Peano arithmetic, set theory, etc.);
- Decidable fragments (Presburger arithmetic, arrays, etc.);
- Applications of formal methods in industrial settings.

Place of the Axioms?

- Leave axioms wandering among the hypotheses?
- Induce a combinatorial explosion in the proof search space;
- Do not bear meaning usable by automated theorem provers.

Proof Search in Axiomatic Theories

A Solution

- A cutting-edge combination between:
  - First order automated theorem proving method (resolution);
  - Theory-specific decision procedures (SMT approach).

Drawbacks

- Specific decision procedure for each given theory;
- Decidability constraint over the theories;
- Lack of automatability and genericity.

Proof Search in Axiomatic Theories

Use of Deduction Modulo

- Transform axioms into rewrite rules;
- Turn proof search among the axioms into computations;
- Avoid unnecessary blowups in the proof search;
- Shrink the size of proofs (record only meaningful steps).

This Talk

- Introduce deduction modulo (and superdeduction);
- Present the experiments in automated deduction;
- Describe the applications in industrial settings.

Deduction Modulo & Superdeduction

Inclusion

$$\forall a \forall b\ ((a \subseteq b) \Leftrightarrow (\forall x\ (x \in a \Rightarrow x \in b)))$$

Proof in Sequent Calculus

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\ldots, x \in A \vdash A \subseteq A, x \in A}\ \text{Ax}}{\ldots \vdash A \subseteq A, x \in A \Rightarrow x \in A}\ \Rightarrow\text{R}}{\ldots \vdash A \subseteq A, \forall x\ (x \in A \Rightarrow x \in A)}\ \forall\text{R} \qquad \dfrac{}{\ldots, A \subseteq A \vdash A \subseteq A}\ \text{Ax}}{\dfrac{\ldots, (\forall x\ (x \in A \Rightarrow x \in A)) \Rightarrow A \subseteq A \vdash A \subseteq A}{\dfrac{A \subseteq A \Leftrightarrow (\forall x\ (x \in A \Rightarrow x \in A)) \vdash A \subseteq A}{\forall a \forall b\ ((a \subseteq b) \Leftrightarrow (\forall x\ (x \in a \Rightarrow x \in b))) \vdash A \subseteq A}\ \forall\text{L} \times 2}\ \wedge\text{L}}\ \Rightarrow\text{L}$$

Deduction Modulo & Superdeduction

Inclusion

$$\forall a \forall b\, ((a \subseteq b) \longrightarrow (\forall x\, (x \in a \Rightarrow x \in b)))$$

Rewrite Rule

$$(a \subseteq b) \longrightarrow (\forall x\, (x \in a \Rightarrow x \in b))$$

Proof in Deduction Modulo

$$\dfrac{\dfrac{\dfrac{}{x \in A \vdash x \in A}\,\mathrm{Ax}}{\vdash x \in A \Rightarrow x \in A}\,\Rightarrow\mathrm{R}}{\vdash A \subseteq A}\,\forall\mathrm{R},\ A \subseteq A \longrightarrow \forall x\, (x \in A \Rightarrow x \in A)$$

Deduction Modulo & Superdeduction

Inclusion

$$\forall a \forall b\ ((a \subseteq b) \longrightarrow (\forall x\ (x \in a \Rightarrow x \in b)))$$

Computation of the Superdeduction Rule

$$\frac{\Gamma \vdash \forall x\ (x \in a \Rightarrow x \in b), \Delta}{\Gamma \vdash a \subseteq b, \Delta}$$

Deduction Modulo & Superdeduction

Inclusion

$$\forall a \forall b\, ((a \subseteq b) \longrightarrow (\forall x\, (x \in a \Rightarrow x \in b)))$$

Computation of the Superdeduction Rule

$$\dfrac{\dfrac{\dfrac{\Gamma, x \in a \vdash x \in b, \Delta}{\Gamma \vdash x \in a \Rightarrow x \in b, \Delta} \Rightarrow\mathrm{R}}{\Gamma \vdash \forall x\, (x \in a \Rightarrow x \in b), \Delta} \forall\mathrm{R},\ x \notin \Gamma, \Delta}{\Gamma \vdash a \subseteq b, \Delta}$$

Deduction Modulo & Superdeduction

Inclusion

$$\forall a \forall b\ ((a \subseteq b) \longrightarrow (\forall x\ (x \in a \Rightarrow x \in b)))$$

Computation of the Superdeduction Rule

$$\frac{\Gamma, x \in a \vdash x \in b, \Delta}{\Gamma \vdash a \subseteq b, \Delta}\ \mathrm{IncR},\ x \notin \Gamma, \Delta$$

Proof in Superdeduction

$$\cfrac{\cfrac{}{x \in A \vdash x \in A}\ \mathrm{Ax}}{\vdash A \subseteq A}\ \mathrm{IncR}$$

From Axioms to Rewrite Rules

Difficulties

- Confluence and termination of the rewrite system;
- Preservation of the consistency;
- Preservation of the cut-free completeness;
- Automation of the transformation.

An Example

- Axiom $A \Leftrightarrow (A \Rightarrow B)$;
- Transformed into $A \longrightarrow A \Rightarrow B$;
- We want to prove: B.

From Axioms to Rewrite Rules

An Example (Continued)

- In sequent calculus, we have a cut-free proof:

$$\dfrac{\dfrac{\dfrac{\dfrac{\sim \Pi}{A \Rightarrow (A \Rightarrow B), A \vdash B, B}}{A \Rightarrow (A \Rightarrow B) \vdash B, A \Rightarrow B}\Rightarrow\mathrm{R} \qquad \dfrac{\Pi}{A \Rightarrow (A \Rightarrow B), A \vdash B}}{A \Rightarrow (A \Rightarrow B), (A \Rightarrow B) \Rightarrow A \vdash B}\Rightarrow\mathrm{L}}{A \Leftrightarrow (A \Rightarrow B) \vdash B}\Leftrightarrow\mathrm{L}$$

Where Π is:

$$\dfrac{\dfrac{}{A \vdash B, A}\mathrm{ax} \qquad \dfrac{\dfrac{}{A \vdash B, A}\mathrm{ax} \qquad \dfrac{}{A, B \vdash B}\mathrm{ax}}{A, A \Rightarrow B \vdash B}\Rightarrow\mathrm{L}}{A \Rightarrow (A \Rightarrow B), A \vdash B}\Rightarrow\mathrm{L}$$

From Axioms to Rewrite Rules

An Example (Continued)

- In deduction modulo, we have to cut A to get a proof:

$$\dfrac{\dfrac{\Pi}{A \vdash B} \qquad \dfrac{\dfrac{\Pi}{A \vdash B}}{\vdash A}\Rightarrow\mathrm{R},\ A \longrightarrow A \Rightarrow B}{\vdash B}\ \mathrm{cut}$$

Where Π is:

$$\dfrac{\dfrac{}{A \vdash A}\ \mathrm{ax} \qquad \dfrac{\dfrac{}{A \vdash A}\ \mathrm{ax} \qquad \dfrac{}{A, B \vdash B}\ \mathrm{ax}}{A, A \vdash B}\Rightarrow\mathrm{L},\ A \longrightarrow A \Rightarrow B}{A \vdash B}\ \mathrm{cut}$$

Some References for Deduction Modulo

Seminal Papers

- Deduction Modulo:

  G. Dowek, T. Hardin, C. Kirchner. *Theorem Proving Modulo*. JAR (2003).

- Superdeduction:

  P. Brauner, C. Houtmann, C. Kirchner. *Principles of Superdeduction.* LICS (2007).

Theories Modulo

- Arithmetic:

  G. Dowek, B. Werner. *Arithmetic as a Theory Modulo.* RTA (2005).

- Set Theory:

  G. Dowek, A. Miquel. *Cut Elimination for Zermelo Set Theory.* Draft (2007).

Some References for Deduction Modulo

Proof Search Methods

- Resolution: ENAR (Extended Narrowing and Resolution)
  G. Dowek, T. Hardin, C. Kirchner. *Theorem Proving Modulo*. JAR (2003).
- Tableaux: TaMeD (Tableau Method for Deduction Modulo)
  R. Bonichon. *TaMeD: A Tableau Method for Deduction Modulo.* IJCAR (2004).

Experiments

- Resolution: iProver Modulo (based on iProver)
  G. Burel. *Experimenting with Deduction Modulo.* CADE (2011).
- Tableaux: (extensions based on Zenon)
  - Superdeduction: Super Zenon
  - Deduction Modulo: Zenon Modulo

Some References for Deduction Modulo

Proof Search Methods

- Resolution: ENAR (Extended Narrowing and Resolution)
- Tableaux: TaMeD (Tableau Method for Deduction Modulo)

Experiments

- Resolution: iProver Modulo (based on iProver)
- Tableaux: (extensions based on Zenon)
  - Superdeduction: Super Zenon
    M. Jacquel, K. Berkani, D. Delahaye, C. Dubois. *Tableaux Modulo Theories Using Superdeduction: An Application to the Verification of B Proof Rules with the Zenon Automated Theorem Prover.* IJCAR (2012).
  - Deduction Modulo: Zenon Modulo
    D. Delahaye, D. Doligez, F. Gilbert, P. Halmagrand, O. Hermant. *Zenon Modulo: When Achilles Outruns the Tortoise using Deduction Modulo.* LPAR (2013).

The Zenon Automated Theorem Prover

Features of Zenon

- First order logic with equality;
- Tableau-based proof search method;
- Extensible by adding new deductive rules;
- Certifying, 3 outputs: Coq, Isabelle, Dedukti;
- Used by other systems: Focalize, TLA.

Zenon

- Reference:

  R. Bonichon, D. Delahaye, D. Doligez. *Zenon: An Extensible Automated Theorem Prover Producing Checkable Proofs.* LPAR (2007).
- Freely available (BSD license);
- Developed by D. Doligez;
- Download: `http://focal.inria.fr/zenon/`

The Zenon Automated Theorem Prover

The Tableau Method

- We start from the negation of the goal (no clausal form);
- We apply the rules in a top-down fashion;
- We build a tree whose each branch must be closed;
- When the tree is closed, we have a proof of the goal.

Closure and Cut Rules

$$\frac{\bot}{\odot}\ \odot_\bot \qquad \frac{\neg\top}{\odot}\ \odot_{\neg\top} \qquad \frac{}{P \mid \neg P}\ \text{cut}$$

$$\frac{\neg R_r(t, t)}{\odot}\ \odot_r \qquad \frac{P \qquad \neg P}{\odot}\ \odot \qquad \frac{R_s(a, b) \qquad \neg R_s(b, a)}{\odot}\ \odot_s$$

Analytic Rules

$$\frac{\neg\neg P}{P}\ \alpha_{\neg\neg} \qquad \frac{P \Leftrightarrow Q}{\neg P, \neg Q \mid P, Q}\ \beta_{\Leftrightarrow} \qquad \frac{\neg(P \Leftrightarrow Q)}{\neg P, Q \mid P, \neg Q}\ \beta_{\neg\Leftrightarrow}$$

$$\frac{P \wedge Q}{P, Q}\ \alpha_{\wedge} \qquad \frac{\neg(P \vee Q)}{\neg P, \neg Q}\ \alpha_{\neg\vee} \qquad \frac{\neg(P \Rightarrow Q)}{P, \neg Q}\ \alpha_{\neg\Rightarrow}$$

$$\frac{P \vee Q}{P \mid Q}\ \beta_{\vee} \qquad \frac{\neg(P \wedge Q)}{\neg P \mid \neg Q}\ \beta_{\neg\wedge} \qquad \frac{P \Rightarrow Q}{\neg P \mid Q}\ \beta_{\Rightarrow}$$

$$\frac{\exists x\ P(x)}{P(\epsilon(x).P(x))}\ \delta_{\exists} \qquad \frac{\neg\forall x\ P(x)}{\neg P(\epsilon(x).\neg P(x))}\ \delta_{\neg\forall}$$

The Zenon Automated Theorem Prover

γ-Rules

$$\frac{\forall x\ P(x)}{P(X)}\ \gamma_{\forall M} \qquad \frac{\neg\exists x\ P(x)}{\neg P(X)}\ \gamma_{\neg\exists M}$$

$$\frac{\forall x\ P(x)}{P(t)}\ \gamma_{\forall \text{inst}} \qquad \frac{\neg\exists x\ P(x)}{\neg P(t)}\ \gamma_{\neg\exists \text{inst}}$$

Relational Rules

- Equality, reflexive, symmetric, transitive rules;
- Are not involved in the computation of superdeduction rules.

Example of Proof Search

$$\frac{\forall x\ (P(x) \lor Q(x))\ ,\ \ \neg P(a)\ ,\ \ \neg Q(a)}{}$$

The Zenon Automated Theorem Prover

Example of Proof Search

$$\dfrac{\forall x\ (P(x) \lor Q(x))\ ,\quad \neg P(a)\ ,\quad \neg Q(a)}{P(X) \lor Q(X)}\ \gamma_{\forall M}$$

le cnam
Inria
BWare

Example of Proof Search

$$\dfrac{\dfrac{\forall x\ (P(x) \lor Q(x))\ ,\ \ \neg P(a)\ ,\ \ \neg Q(a)}{P(X) \lor Q(X)}\ \gamma_{\forall M}}{P(X) \qquad\qquad Q(X)}\ \beta_{\lor}$$

Example of Proof Search

$$\dfrac{\dfrac{\forall x\,(P(x) \lor Q(x))\ ,\quad \neg P(a)\ ,\quad \neg Q(a)}{P(X) \lor Q(X)}\,\gamma_{\forall M}}{P(X) \qquad\qquad Q(X)}\,\beta_{\lor}$$

Example of Proof Search

$$\dfrac{\forall x\ (P(x) \lor Q(x))\ ,\quad \neg P(a)\ ,\quad \neg Q(a)}{\dfrac{P(X) \lor Q(X)}{\dfrac{P(X)}{P(a) \lor Q(a)}\gamma_{\forall\mathrm{inst}} \qquad Q(X)}\beta_{\lor}}\gamma_{\forall M}$$

Example of Proof Search

$$\cfrac{\forall x\,(P(x) \lor Q(x))\ ,\quad \neg P(a)\ ,\quad \neg Q(a)}{\cfrac{P(X) \lor Q(X)}{\cfrac{\cfrac{P(X)}{\cfrac{P(a) \lor Q(a)}{\color{red}{P(a) \qquad Q(a)}}\ \color{red}{\beta_\lor}}\ \gamma_{\forall\mathrm{inst}} \qquad\qquad Q(X)}{}}\ \beta_\lor}\ \gamma_{\forall M}$$

Example of Proof Search

$$
\dfrac{\dfrac{\forall x\,(P(x) \lor Q(x))\ ,\ \ \neg P(a)\ ,\ \ \neg Q(a)}{P(X) \lor Q(X)}\ \gamma_{\forall M}}{\dfrac{\dfrac{P(X)}{P(a) \lor Q(a)}\ \gamma_{\forall\mathrm{inst}}}{\dfrac{P(a)}{\odot}\ \odot \qquad \dfrac{Q(a)}{}}\ \beta_{\lor} \qquad Q(X)}\ \beta_{\lor}
$$

Example of Proof Search

$$\dfrac{\forall x\ (P(x) \lor Q(x))\ ,\quad \neg P(a)\ ,\quad {\color{red}\neg Q(a)}}{P(X) \lor Q(X)}\ \gamma_{\forall M}$$

$$\dfrac{P(X) \lor Q(X)}{P(X) \qquad\qquad Q(X)}\ \beta_{\lor}$$

$$\dfrac{P(X)}{P(a) \lor Q(a)}\ \gamma_{\forall\mathrm{inst}}$$

$$\dfrac{P(a) \lor Q(a)}{P(a) \qquad {\color{red}Q(a)}}\ \beta_{\lor}$$

$$\dfrac{P(a)}{\odot}\ \odot \qquad \dfrac{{\color{red}Q(a)}}{{\color{red}\odot}}\ {\color{red}\odot}$$

The Zenon Automated Theorem Prover

Example of Proof Search

$$\dfrac{\forall x\,(P(x)\lor Q(x))\ ,\quad \neg P(a)\ ,\quad \neg Q(a)}{\dfrac{\color{red}{P(X)\lor Q(X)}}{\dfrac{\color{red}{P(X)}}{\dfrac{P(a)\lor Q(a)}{\dfrac{P(a)}{\odot}\ \odot \qquad \dfrac{Q(a)}{\odot}\ \odot}\ \beta_\lor}\ \gamma_{\forall\mathrm{inst}} \qquad \color{red}{Q(X)}}\ \beta_\lor}\ \gamma_{\forall M}$$

The Zenon Automated Theorem Prover

Example of Proof Search

$$\dfrac{\dfrac{\forall x\ (P(x) \lor Q(x))\ ,\quad \neg P(a)\ ,\quad \neg Q(a)}{P(a) \lor Q(a)}\ \gamma_{\forall\mathrm{inst}}}{\dfrac{P(a)}{\odot}\ \odot \qquad \dfrac{Q(a)}{\odot}\ \odot}\ \beta_{\lor}$$

Integrating Superdeduction to Zenon

Computation of Superdeduction Rules

- $\mathcal{S} \equiv$ closure rules, analytic rules, $\gamma_{\forall M}$ and $\gamma_{\neg\exists M}$ rules;
- Axiom: $R : P \longrightarrow \varphi$;
- A positive superdeduction rule R (and a negative one $\neg R$):
  - Initialize the procedure with the formula φ;
  - Apply the rules of $\mathcal{S}$ until there is no applicable rule anymore;
  - Collect the premises and the conclusion, and replace φ by P.
- If metavariables, add an instantiation rule R_{inst} (or $\neg R_{\mathrm{inst}}$).

Integrating Superdeduction to Zenon

Example (inclusion)

$$\dfrac{\dfrac{\forall x\ (x \in a \Rightarrow x \in b)}{X \in a \Rightarrow X \in b}\ \gamma_{\forall M}}{X \notin a \mid X \in b}\ \beta_{\Rightarrow}$$

$$\dfrac{\dfrac{\neg\forall x\ (x \in a \Rightarrow x \in b)}{\neg(\epsilon_x \in a \Rightarrow \epsilon_x \in b)}\ \delta_{\neg\forall}}{\epsilon_x \in a, \epsilon_x \notin b}\ \alpha_{\neg\Rightarrow}$$

with $\epsilon_x = \epsilon(x).\neg(x \in a \Rightarrow x \in b)$

$$\dfrac{a \subseteq b}{X \notin a \mid X \in b}\ \mathrm{Inc}$$

$$\dfrac{a \not\subseteq b}{\epsilon_x \in a, \epsilon_x \notin b}\ \neg\mathrm{Inc}$$

with $\epsilon_x = \epsilon(x).\neg(x \in a \Rightarrow x \in b)$

$$\dfrac{a \subseteq b}{t \notin a \mid t \in b}\ \mathrm{Inc}_{\mathrm{inst}}$$

Integrating Superdeduction to Zenon

Example of Proof Search

- With regular rules of Zenon:

$$
\dfrac{\forall a \forall b\,((a \subseteq b) \Leftrightarrow (\forall x\,(x \in a \Rightarrow x \in b))), A \not\subseteq A}{\dfrac{(X \subseteq Y) \Leftrightarrow (\forall x\,(x \in X \Rightarrow x \in Y))}{\dfrac{X \subseteq Y, \forall x\,(x \in X \Rightarrow x \in Y)}{\dfrac{(A \subseteq A) \Leftrightarrow (\forall x\,(x \in A \Rightarrow x \in A))}{\dfrac{A \subseteq A, \forall x\,(x \in A \Rightarrow x \in A)}{\odot}\ \odot \qquad \Pi}\ \beta_{\Leftrightarrow}}\ \gamma_{\forall\text{inst}} \times 2 \qquad \Pi'}\ \beta_{\Leftrightarrow}}\ \gamma_{\forall M} \times 2
$$

Where Π is:

$$
\dfrac{A \not\subseteq A, \neg\forall x\,(x \in A \Rightarrow x \in A)}{\dfrac{\neg(\epsilon_x \in A \Rightarrow \epsilon_x \in A)}{\dfrac{\epsilon_x \in A, \epsilon_x \notin A}{\odot}\ \odot}\ \alpha_{\neg\Rightarrow}}\ \delta_{\neg\forall}
$$

with $\epsilon_x = \epsilon(x).\neg(x \in A \Rightarrow x \in A)$

Integrating Superdeduction to Zenon

Example of Proof Search

- With regular rules of Zenon:

$$\dfrac{\dfrac{\forall a \forall b\,((a \subseteq b) \Leftrightarrow (\forall x\,(x \in a \Rightarrow x \in b))), A \not\subseteq A}{(A \subseteq A) \Leftrightarrow (\forall x\,(x \in A \Rightarrow x \in A))}\ \gamma_{\forall\text{inst}} \times 2}{\dfrac{A \subseteq A, \forall x\,(x \in A \Rightarrow x \in A)}{\odot}\ \odot \qquad \Pi}\ \beta_{\Leftrightarrow}$$

Where Π is:

$$\dfrac{A \not\subseteq A, \neg\forall x\,(x \in A \Rightarrow x \in A)}{\dfrac{\neg(\epsilon_x \in A \Rightarrow \epsilon_x \in A)}{\dfrac{\epsilon_x \in A, \epsilon_x \notin A}{\odot}\ \odot}\ \alpha_{\neg\Rightarrow}}\ \delta_{\neg\forall}$$

with $\epsilon_x = \epsilon(x).\neg(x \in A \Rightarrow x \in A)$

Integrating Superdeduction to Zenon

Example of Proof Search

- With superdeduction rules:

$$\dfrac{\dfrac{A \not\subseteq A}{\epsilon_x \in A, \epsilon_x \notin A}\ \neg\text{Inc}}{\odot}\ \odot$$

with $\epsilon_x = \epsilon(x).\neg(x \in A \Rightarrow x \in A)$

Superdeduction for the B Method

Collaboration between Cnam and Siemens

- M. Jacquel, K. Berkani, D. Delahaye, C. Dubois;
- Meteor line at Paris (line 14), opened 15 years ago;
- VAL, automatic metro systems, optical guidance for buses/trolleybuses.

Metro Line 14

New York Subway

Use of the B Method

The B Method

- Defined in the B-Book (1996) by J.-R. Abrial;
- Based on a (typed) set theory;
- Generation of executable code from formal specifications;
- Notion of machines, refined until implementations;
- Generation of proof obligations (consistency, refinement);
- Supporting tool: Atelier B (ClearSy).

Proof Activity with Atelier B

- Automated proofs (pp);
- Interactive proofs: apply tactics, add rules (axioms).
- If the added rule is wrong then:
  - The proof of the proof obligation may be unsound;
  - The generated code may contain some bugs.

Use of the B Method

The B Method

- Defined in the B-Book (1996) by J.-R. Abrial;
- Based on a (typed) set theory;
- Generation of executable code from formal specifications;
- Notion of machines, refined until implementations;
- Generation of proof obligations (consistency, refinement);
- Supporting tool: Atelier B (ClearSy).

Figures

- Meteor: 27,800 proof obligations, 1,400 added rules;
- Currently about 5,300 rules in the database of Siemens.

Verification of B Proof Rules with Zenon

Approach with Zenon

- Preliminary normalization to get rid of set constructs;
- Formulas with only the "$\in$" (uninterpreted) symbol;
- Call of Zenon and Coq used as a backend;
- See the SEFM'11 paper for more details:
  M. Jacquel, K. Berkani, D. Delahaye, C. Dubois. *Verifying B Proof Rules Using Deep Embedding and Automated Theorem Proving.* SEFM (2011).

Problems

- Preliminary normalization:
  - Incomplete approach;
  - Weak performances in terms of time.
- Solution: reason modulo the B set theory!

Superdeduction Rules for the B Set Theory

Axioms (4 over 6)

$$
\begin{array}{l}
(x, y) \in a \times b \Leftrightarrow x \in a \wedge y \in b \\
a \in \mathbb{P}(b) \Leftrightarrow \forall x\ (x \in a \Leftrightarrow x \in b) \\
x \in \{\ y \mid P(y)\ \} \Leftrightarrow P(x) \\
a = b \Leftrightarrow \forall x\ (x \in a \Rightarrow x \in b)
\end{array}
$$

Superdeduction Rules (Comprehension and Equality)

$$\frac{x \in \{\ y \mid P(y)\ \}}{P(x)}\ \{|\} \qquad \frac{a = b}{X \notin a, X \notin b \mid X \in a, X \in b}\ =$$

$$\frac{x \notin \{\ y \mid P(y)\ \}}{\neg P(x)}\ \neg\{|\} \qquad \frac{a \neq b}{\epsilon_x \notin a, \epsilon_x \in b \mid \epsilon_x \in a, \epsilon_x \notin b}\ \neq$$

with $\epsilon_x = \epsilon(x).\neg(x \in a \Leftrightarrow x \in b)$

Superdeduction Rules for the B Set Theory

Axioms (4 over 6)

$$
\begin{array}{l}
(x, y) \in a \times b \longrightarrow x \in a \wedge y \in b \\
a \in \mathbb{P}(b) \longrightarrow \forall x\, (x \in a \Rightarrow x \in b) \\
x \in \{\, y \mid P(y) \,\} \longrightarrow P(x) \\
a = b \longrightarrow \forall x\, (x \in a \Leftrightarrow x \in b)
\end{array}
$$

Superdeduction Rules (Comprehension and Equality)

$$
\frac{x \in \{\, y \mid P(y) \,\}}{P(x)}\ \{|\}
\qquad
\frac{a = b}{X \notin a, X \notin b \mid X \in a, X \in b}\ =
$$

$$
\frac{x \notin \{\, y \mid P(y) \,\}}{\neg P(x)}\ \neg\{|\}
\qquad
\frac{a \neq b}{\epsilon_x \notin a, \epsilon_x \in b \mid \epsilon_x \in a, \epsilon_x \notin b}\ \neq
$$

with $\epsilon_x = \epsilon(x).\neg(x \in a \Leftrightarrow x \in b)$

Superdeduction Rules for the B Set Theory

Definitions

$$E \triangleq F$$
$$R : x \in E \longrightarrow x \in F$$
$$a \cup b \triangleq \{ x \mid x \in a \vee x \in b \}$$
$$a \cap b \triangleq \{ x \mid x \in a \wedge x \in b \}$$
$$\cup : x \in a \cup b \longrightarrow x \in \{ x \mid x \in a \vee x \in b \}$$
$$\cap : x \in a \cap b \longrightarrow x \in \{ x \mid x \in a \wedge x \in b \}$$

Superdeduction Rules (Union and Intersection)

$$\frac{x \in a \cup b}{x \in a \mid x \in b}\ \cup \qquad \frac{x \in a \cap b}{x \in a, x \in b}\ \cap$$

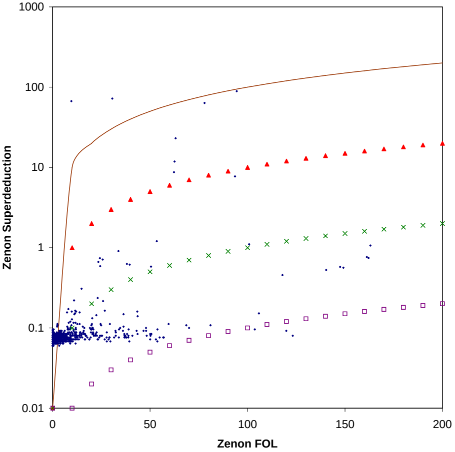
$$\frac{x \notin a \cup b}{x \notin a, x \notin b}\ \neg\cup \qquad \frac{x \notin a \cap b}{x \notin a \mid x \notin b}\ \neg\cap$$

Benchmarks

Superdeduction vs Pre-Normalization (Time)

1,397 rules

Intel Core i5 3.3GHz

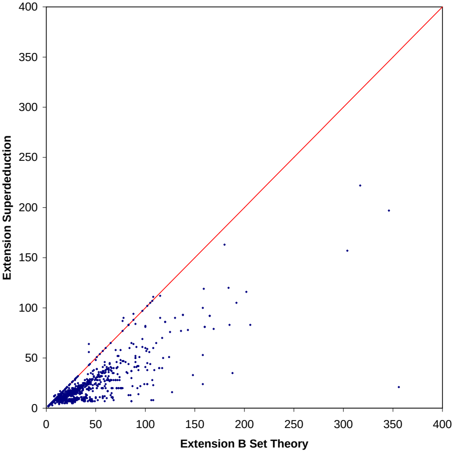


le cnam
Inria
BWare

Superdeduction vs Prawitz's Approach (Nodes)

1,397 rules

Intel Core i5 3.3GHz



Benchmarks

Figures

- Number of rules that can be handled: 1,397 rules;
- Initial approach (with Zenon): 1,145 proved rules (82%);
- With Zenon extended to superdeduction:
  - 1,340 proved rules (96%);
  - On average, proved 67 times faster (best ratio: 1,540).
- With Zenon à la Prawitz:
  - 1,340 proved rules (96%);
  - On average, 1.6 times more nodes (best ratio: 6.25).
- See the IJCAR'12 paper for more details:

  M. Jacquel, K. Berkani, D. Delahaye, C. Dubois. *Tableaux Modulo Theories Using Superdeduction: An Application to the Verification of B Proof Rules with the Zenon Automated Theorem Prover.* IJCAR (2012).

Benchmarks

Figures

- Number of rules that can be handled: 1,397 rules;
- Initial approach (with Zenon): 1,145 proved rules (82%);
- With Zenon extended to superdeduction:
  - 1,340 proved rules (96%);
  - On average, proved 67 times faster (best ratio: 1,540).
- With Zenon à la Prawitz:
  - 1,340 proved rules (96%);
  - On average, 1.6 times more nodes (best ratio: 6.25).
- See the IJCAR'12 paper for more details.

Remarks

- Approach with Zenon: problems due to pre-normalization.
- Narrowing not implemented (incompleteness).

Generalization of the Approach

For any First Order Theory

- Automated orientation of the theories;
- Not oriented axioms left as axioms;
- Computation using other superdeduction rules;
- New tool: Superdeduction + Zenon = Super Zenon !

Heuristic

- Axiom $\forall \bar{x}\ (P \Leftrightarrow \varphi)$: $R : P \rightarrow \varphi\ (R, \neg R)$;
- Axiom $\forall \bar{x}\ (P \Rightarrow P')$: $R : P \rightarrow P'\ (R)$, $R' : \neg P' \rightarrow \neg P\ (R')$;
- Axiom $\forall \bar{x}\ (P \Rightarrow \varphi)$: $R : P \rightarrow \varphi\ (R)$;
- Axiom $\forall \bar{x}\ (\varphi \Rightarrow P)$: $R : \neg P \rightarrow \neg \varphi\ (R)$;
- Axiom $\forall \bar{x}\ P$: $R : \neg P \rightarrow \bot\ (R)$.

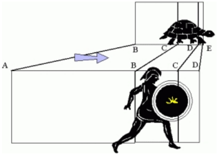
Generalization of the Approach

Figures

TPTP Category (v5.3.0)	Zenon	Super Zenon
FOF 6,644 problems	1,646	1,765 (7.2%)
SET 462 problems	147	202 (37.4%)

Super Zenon

- Freely available (GPL license);
- Collaboration Cnam and Siemens;
- Download:
  `http://cedric.cnam.fr/~delahaye/super-zenon/`



Integrating Deduction Modulo to Zenon

Goals

- Improve the proof search in axiomatic theories;
- Reduce the proof size;
- New tool: Zenon + Deduction Modulo = Zenon Modulo!

Compared to Super Zenon

- Compare deduction modulo and superdeduction in practice;
- Rewrite rules over propositions and terms;
- Normalization strategies (efficiency);
- Light integration (metavariable management);
- No trace of computation in the proofs.

Class Rewrite System

Definition

A class rewrite system is a pair consisting of:

- $\mathcal{R}$: a set of proposition rewrite rules;
- $\mathcal{E}$: a set of term rewrite rules (and equational axioms).

Rewrite Rules

- Proposition rewrite rule: $l \longrightarrow r$, where l is an atomic proposition and $FV(r) \subseteq FV(l)$;
- Term rewrite rule: $l \longrightarrow r$, where $FV(r) \subseteq FV(l)$.

Congruence

- $=_{\mathcal{R}\mathcal{E}} \equiv$ congruence generated by the set $\mathcal{R} \cup \mathcal{E}$.

Rules of Zenon Modulo

Closure and Cut Rules

$$\frac{P \qquad \neg Q}{\odot} \odot \text{ if } P =_{\mathcal{RE}} Q \qquad \frac{}{P \mid \neg Q} \text{cut} \text{ if } P =_{\mathcal{RE}} Q$$

$$\frac{P}{\odot} \odot_{\bot} \text{ if } P =_{\mathcal{RE}} \bot \qquad \frac{\neg P}{\odot} \odot_{\neg\top} \text{ if } P =_{\mathcal{RE}} \top$$

$$\frac{\neg P}{\odot} \odot_{r} \text{ if } P =_{\mathcal{RE}} R_r(t,t) \qquad \frac{P \qquad \neg Q}{\odot} \odot_{s} \text{ if } P =_{\mathcal{RE}} R_s(a,b) \text{ and } Q =_{\mathcal{RE}} R_s(b,a)$$

Where R_r is a reflexive relation, and R_s a symmetric relation.

Rules of Zenon Modulo

α/β-Rules

$$\frac{\neg S}{P}\ \alpha_{\neg\neg} \quad \text{if } S =_{\mathcal{RE}} \neg P$$

$$\frac{S}{P, Q}\ \alpha_{\wedge} \quad \text{if } S =_{\mathcal{RE}} P \wedge Q \qquad\qquad \frac{\neg S}{\neg P \mid \neg Q}\ \beta_{\neg\wedge} \quad \text{if } S =_{\mathcal{RE}} P \wedge Q$$

$$\frac{S}{P \mid Q}\ \beta_{\vee} \quad \text{if } S =_{\mathcal{RE}} P \vee Q \qquad\qquad \frac{\neg S}{\neg P, \neg Q}\ \alpha_{\neg\vee} \quad \text{if } S =_{\mathcal{RE}} P \vee Q$$

$$\frac{S}{\neg P \mid Q}\ \beta_{\Rightarrow} \quad \text{if } S =_{\mathcal{RE}} P \Rightarrow Q \qquad\qquad \frac{\neg S}{P, \neg Q}\ \alpha_{\neg\Rightarrow} \quad \text{if } S =_{\mathcal{RE}} P \Rightarrow Q$$

$$\frac{S}{\neg P, \neg Q \mid P, Q}\ \beta_{\Leftrightarrow} \quad \text{if } S =_{\mathcal{RE}} P \Leftrightarrow Q$$

$$\frac{\neg S}{\neg P, Q \mid P, \neg Q}\ \beta_{\neg\Leftrightarrow} \quad \text{if } S =_{\mathcal{RE}} P \Leftrightarrow Q$$

Rules of Zenon Modulo

δ/γ-Rules

$$\frac{S}{P(\epsilon(x).P(x))}\ \delta_{\exists}\quad \text{if } S =_{\mathcal{RE}} \exists x\ P(x)$$

$$\frac{\neg S}{\neg P(\epsilon(x).\neg P(x))}\ \delta_{\neg\forall}\quad \text{if } S =_{\mathcal{RE}} \forall x\ P(x)$$

$$\frac{S}{P(X)}\ \gamma_{\forall M}\quad \text{if } S =_{\mathcal{RE}} \forall x\ P(x) \qquad \frac{\neg S}{\neg P(X)}\ \gamma_{\neg\exists M}\quad \text{if } S =_{\mathcal{RE}} \exists x\ P(x)$$

$$\frac{S}{P(t)}\ \gamma_{\forall\text{inst}}\quad \text{if } S =_{\mathcal{RE}} \forall x\ P(x) \qquad \frac{\neg S}{\neg P(t)}\ \gamma_{\neg\exists\text{inst}}\quad \text{if } S =_{\mathcal{RE}} \exists x\ P(x)$$

Experimental Results over the TPTP Library

Figures

TPTP Category	Zenon	Zenon Mod. (Prop. Rew.)	Zenon Mod. (Term/Prop. Rew.)
FOF 6,659 prob.	1,586	1,626 (2.5%) +114 (7.2%) -74 (4.7%)	1,616 (1.9%) +170 (10.7%) -140 (8.8%)
SET 462 prob.	149	219 (47%) +78 (52.3%) -8 (5.4%)	222 (49%) +86 (57.7%) -13 (8.7%)

- TPTP Library v5.5.0;
- Intel Xeon X5650 2.67GHz;
- Timeout 300 s, memory limit 1 GB.

Experimental Results over the TPTP Library

Figures

TPTP Category	Zenon	Zenon Mod. (Prop. Rew.)	Zenon Mod. (Term/Prop. Rew.)
FOF 6,659 prob.	1,586	1,626 (2.5%) +114 (7.2%) -74 (4.7%)	1,616 (1.9%) +170 (10.7%) -140 (8.8%)
SET 462 prob.	149	219 (47%) +78 (52.3%) -8 (5.4%)	222 (49%) +86 (57.7%) -13 (8.7%)

- 29 difficult problems (TPTP ranking);
- 29 with a ranking ≥ 0.7;
- 9 with a ranking ≥ 0.8;
- 1 with a ranking ≥ 0.9.

Proof Compression

Experiment

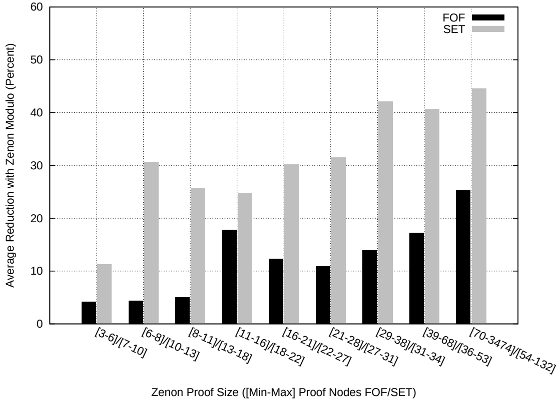
- 1,446 problems proved by both Zenon and Zenon Modulo;
- 624 FOF problems and 110 SET problems;
- Subset of proofs where rewriting occurs;
- Measure: number of proof nodes of the resulting proof.

Figures

TPTP Category	Average Reduction	Maximum Reduction
FOF 624 problems	6.8%	91.4%
SET 110 problems	21.6%	84.6%

Proof Compression

Figures



A Backend for Zenon Modulo

Using the Existing Backends

- Create special inference nodes for rewriting rules;
- Record rewrite steps in the proof traces;
- Extend the existing backends of Zenon;
- Prove the rewriting lemmas in Coq and Isabelle.

Problems of this Approach

- Possible large number of rewrite steps to record;
- May Lead to memory explosion;
- Against the Poincaré principle;
- Loss of deduction modulo benefits.

Using the Dedukti Universal Proof Checker

Features of Dedukti

- Universal proof checker for the $\lambda\Pi$-calculus modulo;
- Propositions/types and proofs/λ-terms (Curry-Howard);
- Native support of rewriting;
- Only need to provide the set of rewrite rules.

Dedukti

- Freely available (CeCILL-B license);
- Developed by Deducteam;
- Download:
  `https://www.rocq.inria.fr/deducteam/Dedukti/`

Using the Dedukti Universal Proof Checker

From Zenon Modulo Proofs to Dedukti

- From classical to intuitionistic logic;
- Based on a double-negation translation;
- Optimized to minimize the number of double-negations;
- 54% of the TPTP proofs already intuitionistic;
- See the LPAR'13 paper for more details:

  D. Delahaye, D. Doligez, F. Gilbert, P. Halmagrand, O. Hermant. *Zenon Modulo: When Achilles Outruns the Tortoise using Deduction Modulo.* LPAR (2013).

Dedukti

- Freely available (CeCILL-B license);
- Developed by Deducteam;
- Download:

  `https://www.rocq.inria.fr/deducteam/Dedukti/`

Proof Verification with Dedukti

le cnam
Inria
BWare

Figures

FOF 624 prob.	Dedukti Success	Dedukti Failure	Backend Issue
Problems	559	5	60
Rate	89.6%	0.8%	9.6%

Failures

- Dedukti: rewrite system (termination, confluence, etc.);
- Backend: minimization of the double-negations.

The BWare Project

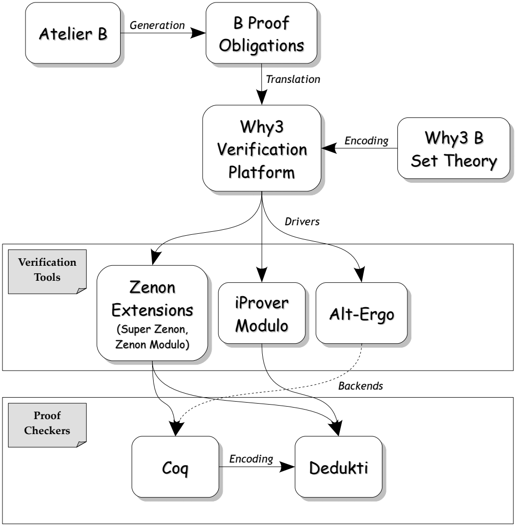
The Project

- INS prog. of the French National Research Agency (ANR);
- Academics: Cnam, LRI, Inria;
- Companies: Mitsubishi, ClearSy, OCamlPro.

Goals

- Mechanized framework for automated verification of B PO;
- Generic platform (several automated deduction tools);
- First order tools and SMT solvers;
- Production of proof objects (certificates).

The BWare Project



Deduction Modulo in the BWare Project

Tools

- Super Zenon, Zenon Modulo (extensions of Zenon);
- iProver Modulo (extension of iProver);
- Backend for these tools: Dedukti.

Adequacy of the Tools

- Build a B set theory modulo (manually);
- Comprehension scheme (higher order) hard-coded;
- Good results of Super Zenon for B proof rules;
- Good results of Zenon Modulo in the SET category of TPTP.

Conclusion

Deduction Modulo in Automated Tools

- Resolution: iProver Modulo (based on iProver);
- Tableaux: Super Zenon, Zenon Modulo (based on Zenon);
- Appropriate backend: Dedukti ($\lambda\Pi$-calculus modulo).

Experimental Results

- Performances increased for generic benchmarks (TPTP);
- Successful use in industrial settings (B method):
  - Collaboration Cnam/Siemens: verification of B proof rules;
  - BWare project: verification of B PO (work in progress).

Automated Deduction

Automated Generation of Theories Modulo

- Generation of theories modulo “on the fly”;
- Preservation of “good” properties (cut-free completeness);
- Difficulties for term rewrite rules (heuristics);
- Use of external tools to study the rewrite system;
- Integration of the equational axioms (rewriting modulo).

Set Theory Modulo

- Good experimental results for set theory;
- Results of Super Zenon (B), Zenon Modulo (TPTP);
- Ability to prove difficult problems in this domain;
- Promising for the BWare project;
- Problem of large formulas, large contexts (PO).

Proof Checking

Proof Checking for Automated Tools

- $\lambda\Pi$-calculus modulo appropriate to encode theories;
- Suitable framework to certify deduction modulo proofs;
- High quality proof certificates (size in particular);
- Dedukti as a backend for several automated tools:
  - Zenon Modulo (extension of Zenon);
  - iProver Modulo (extension of iProver).

Interoperability between Proof Systems

- Shallow embeddings of theories;
- Dedukti embeddings:
  - CoqInE (from Coq);
  - Holide (from HOL);
  - Focalide (from Focalize).