Max-Planck-Institut für Informatik

Un environnement de démonstration universel

Talk at CPR

Guillaume Burel

Wednesday March 24th, 2010

Motivations

Given a theory $\mathcal{T}$, search for proof in $\mathcal{T}$

$\mathcal{T}$:

- arithmetic (fragment of)
- set theory
- pointer arithmetic
- lists
- higher order logic (Church's simple type theory)
- ...

Axiomatization

First approach: Use an axiomatization of the theory

For instance Peano's axioms for first-order arithmetic

Not adapted for proof search, in particular when the theory has a computational content!

1+1=2

In Γ:

$$\forall x,\ x + O = x$$

$$\forall x\ y,\ x + s(y) = s(x+y)$$

$$\forall x\ y,\ x = y \Rightarrow X(x) \Rightarrow X(y)$$

$$\forall\vdash\ \dfrac{\overset{\frown}{\vdash}\ \dfrac{}{\Gamma, \underline{1}+\underline{1} = s(\underline{1}+O) \vdash \underline{1}+\underline{1} = s(\underline{1}+O), \underline{1}+\underline{1} = \underline{2}}}{\Gamma \vdash \underline{1}+\underline{1} = s(\underline{1}+O), \underline{1}+\underline{1} = \underline{2}}$$

$$\Rightarrow\vdash\ \dfrac{\Gamma \vdash \underline{1}+\underline{1} = s(\underline{1}+O), \underline{1}+\underline{1} = \underline{2} \qquad \overset{\frown}{\vdash}\ \dfrac{}{\Gamma, \underline{1}+\underline{1} = \underline{2} \vdash \underline{1}+\underline{1} = \underline{2}}}{\Gamma, \underline{1}+\underline{1} = s(\underline{1}+O) \Rightarrow \underline{1}+\underline{1} = \underline{2} \vdash \underline{1}+\underline{1} = \underline{2}}$$

$$\forall\vdash\ \dfrac{\overset{\frown}{\vdash}\ \dfrac{}{\Gamma, \underline{1}+O = \underline{1} \vdash \underline{1}+O = \underline{1}, \underline{1}+\underline{1} = \underline{2}}}{\Gamma \vdash \underline{1}+O = \underline{1}, \underline{1}+\underline{1} = \underline{2}}$$

$$\forall\vdash\ \dfrac{\Rightarrow\vdash\ \dfrac{\Gamma \vdash \underline{1}+O = \underline{1}, \underline{1}+\underline{1} = \underline{2} \qquad \vdots}{\Gamma, \underline{1}+O = \underline{1} \Rightarrow \underline{1}+\underline{1} = s(\underline{1}+O) \Rightarrow \underline{1}+\underline{1} = \underline{2} \vdash \underline{1}+\underline{1} = \underline{2}}}{\Gamma \vdash \underline{1}+\underline{1} = \underline{2}}$$

Other approaches

- Satisfiability Modulo Theory: efficient proof search methods, not generic (theory = black box)

  DPLL(T) [Ganzinger, Hagen, Nieuwenhuis, Oliveras and Tinelli, 2004]

Other approaches

- Satisfiability Modulo Theory: efficient proof search methods, not generic (theory = black box)

  DPLL(T) [Ganzinger, Hagen, Nieuwenhuis, Oliveras and Tinelli, 2004]

- Dependent and Inductive Types: universal, hard to automatize

  Coq, Isabelle, etc.

Other approaches

- Satisfiability Modulo Theory: efficient proof search methods, not generic (theory = black box)
  DPLL(T) [Ganzinger, Hagen, Nieuwenhuis, Oliveras and Tinelli, 2004]
- Dependent and Inductive Types: universal, hard to automatize
  Coq, Isabelle, etc.
- Deduction Modulo and Superdeduction
  [Dowek et al., 2003, Wack, 2005]

Poincaré's principle

In a proof, distinguish deduction from computation to better combine them

Deduction modulo: inference rules (deduction) are applied modulo a congruence (computation)

Universal model for computation: rewriting $\rightsquigarrow$ congruence based on a rewrite system over terms and formulæ

Example

$$x + \mathsf{O} \rightarrow x$$
$$x + s(y) \rightarrow s(x + y)$$

$$\mathsf{O} = \mathsf{O} \rightarrow \top$$
$$s(x) = s(y) \rightarrow x = y$$

$$\underline{1} + \underline{1} = \underline{2} \longrightarrow s(\underline{1} + \mathsf{O}) = \underline{2} \longrightarrow s(\underline{1}) = \underline{2} \longrightarrow^{+} \mathsf{O} = \mathsf{O} \longrightarrow \top$$

$$\vdash\top \frac{}{\vdash \underline{1} + \underline{1} = \underline{2}}$$

Compiling theories

$$Max(x,a) \rightarrow x \in a \wedge \forall y,\ y \in a \Rightarrow y \leq x$$

$$\vdash_{\longleftrightarrow^*} \dfrac{\vdash_{\wedge} \dfrac{\begin{array}{c}\vdots\\ \Gamma \vdash t \in b\end{array} \qquad \vdash_{\forall} \dfrac{\vdash_{\Rightarrow} \dfrac{\begin{array}{c}\vdots\\ \Gamma, y \in b \vdash y \leq t\end{array}}{\Gamma \vdash y \in b \Rightarrow y \leq t}}{\Gamma \vdash \forall y,\ y \in b \Rightarrow y \leq t}}{\Gamma \vdash t \in b \wedge \forall y,\ y \in b \Rightarrow y \leq t}}{\Gamma \vdash Max(t,b)}$$

$$\vdots$$

Compiling theories

$$Max(x,a) \rightarrow x \in a \wedge \forall y,\ y \in a \Rightarrow y \leq x$$

$$
\vdash_{\wedge}\ \dfrac{\dfrac{\vdots}{\Gamma \vdash t \in b} \qquad \vdash_{\forall}\ \dfrac{\vdash_{\Rightarrow}\ \dfrac{\dfrac{\vdots}{\Gamma, y \in b \vdash y \leq t}}{\Gamma \vdash y \in b \Rightarrow y \leq t}}{\Gamma \vdash \forall y,\ y \in b \Rightarrow y \leq t}}{\vdash_{\longleftrightarrow^*}\ \dfrac{\Gamma \vdash t \in b \wedge \forall y,\ y \in b \Rightarrow y \leq t}{\dfrac{\Gamma \vdash Max(t,b)}{\vdots}}}
$$

$$
\vdash_{Max^{\mathrm{def}}}\ \dfrac{\Gamma \vdash x \in a \qquad \Gamma, y \in a \vdash y \leq x}{\Gamma \vdash Max(x,a)}
$$

Superdeduction

New rules (superrules) from a proposition rewrite system

- Natural deduction $\rightsquigarrow$ supernatural deduction
  [Wack, 2005]
  Introduction and elimination superrules

- Sequent calculus $\rightsquigarrow$ extensible sequent calculus
  [Brauner et al., 2007]
  Left and right supperrules

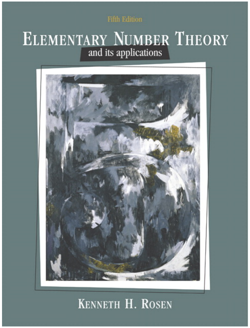
Term rewrite rules are still applied modulo

Outline

- Introduction
- Building Provers Adapted to Theories
  - From Theories to Rewrite Systems
  - Implementing a Prover
- Proof Length Speed-ups
- A Universal Framework
- Conclusion

From theories to provers

Given a theory $\mathcal{T}$, find a systematic way to obtain a prover adapted to that $\mathcal{T}$



$\Rightarrow$

```
picard:~/cvs/slud gburel$ ./slud
  Slud, theorem proving modulo
> include(number.theo).
- : number.theo included
> fof(fermat, conjecture,
       ! [N] : N > 2 =>
         ~ ? [A,B,C] :
           A ^ N + B ^ N = C ^ N).
proving...
% SZS status Theorem for fermat
- : fermat proved
```

Idea

1. Transform the presentation of the theory into a rewrite system
2. Use the rewrite system in a prover based on deduction modulo

For the prover to be complete, the rewrite system has to imply cut-elimination

Automation

Problem: rewrite rules of the form *atomic formula* $\longrightarrow$ *formula*

corresponds to *atomic formula* $\Leftrightarrow$ *formula*

Idea: decompose the axiom by applying inference rules of a sequent calculus

Automation

Problem: rewrite rules of the form *atomic formula* $\rightarrow$ *formula*

corresponds to *atomic formula* $\Leftrightarrow$ *formula*

Idea: decompose the axiom by applying inference rules of a sequent calculus

From set of axioms Θ to a rewrite system $\mathcal{R}(\Theta)$

$\Theta \vdash P$ iff $\vdash_{\mathcal{R}(\Theta)} P$: use only invertible rules (system G4 of Kleene)

Examples

$$\vdash_{\Rightarrow} \frac{A \Rightarrow B \vdash A}{\vdash (A \Rightarrow B) \Rightarrow A} \rightsquigarrow A \longrightarrow^{+} A \Rightarrow B$$

Examples

$$\vdash_{\Rightarrow} \frac{A \Rightarrow B \vdash A}{\vdash (A \Rightarrow B) \Rightarrow A} \rightsquigarrow A \longrightarrow^{+} A \Rightarrow B$$

$$\vdash_{\forall} \dfrac{\vdash_{\vee} \dfrac{\vdash_{\exists} \dfrac{\vdash A_1(x_1, t), \exists y.\ A_1(x_1, y), \exists y.\ A_2(x_2, y)}{\vdash \exists y.\ A_1(x_1, y), \exists y.\ A_2(x_2, y)}}{\vdash \exists y.\ A_1(x_1, y) \vee \exists y.\ A_2(x_2, y)}}{\vdash \forall x_1\ x_2.\ \exists y.\ A_1(x_1, y) \vee \exists y.\ A_2(x_2, y)}$$

$$\rightsquigarrow A_1(x_1, t) \longrightarrow^{+} \exists x_2.\ (\neg \exists y.\ A_1(x_1, y) \wedge \neg \exists y.\ A_2(x_2, y))$$

The cut rule

$$\frac{\Gamma, P \vdash \Delta \qquad \Gamma \vdash P, \Delta}{\Gamma \vdash \Delta} \ \vdash\!\!\smile$$

Cut admissibility: $\Gamma \vdash \Delta$ provable iff provable without Cut

Without modulo, cut admissible (Gentzen's *Hauptsatz*)

Importance of the cut admissibility

- Implies the consistency of the theory defined by the congruence
- Is equivalent to the completeness of the proof-search procedures based on deduction modulo:
  - Extended Narrowing And Resolution and its variant Polarized Resolution Modulo [Dowek 2009]: equational resolution + extended narrowing rules:

$$\text{Ext. Narr.}\ \frac{C, A}{C, P}\ A \longrightarrow P$$

  - TaMed, a tableau method [Bonichon and Hermant, 2006]

Inadmissibility in deduction modulo

$A \to A \Rightarrow B$

Let us search a “minimal” counter-example:

Inadmissibility in deduction modulo

$$A \rightarrow A \Rightarrow B$$

Let us search a "minimal" counter-example:

$$\underset{\smile}{\vdash}\;\dfrac{{\uparrow\vdash}\;\dfrac{A \Rightarrow B, A \vdash}{A \vdash} \qquad {\vdash\uparrow}\;\dfrac{\vdash A, A \Rightarrow B}{\vdash A}}{\vdash}$$

Inadmissibility in deduction modulo

$$A \rightarrow A \Rightarrow B$$

Let us search a “minimal” counter-example:

$$\dfrac{\uparrow\vdash\ \dfrac{\Rightarrow\vdash\ \dfrac{A, B \vdash \qquad \dfrac{}{A \vdash A}\ \text{ax}}{A \Rightarrow B, A \vdash}}{A \vdash} \qquad \vdash\uparrow\ \dfrac{\vdash\Rightarrow\ \dfrac{\dfrac{}{A \vdash A, B}\ \text{ax}}{\vdash A, A \Rightarrow B}}{\vdash A}}{\vdash}\ \text{cut}$$

Inadmissibility in deduction modulo

$$A \rightarrow A \Rightarrow B$$

Let us search a "minimal" counter-example:

$$\overset{\frown}{\vdash}\;\dfrac{\uparrow\vdash\;\dfrac{\Rightarrow\vdash\;\dfrac{\overset{\frown}{\vdash}\;\dfrac{}{A, B \vdash B} \qquad \overset{\frown}{\vdash}\;\dfrac{}{A \vdash A, B}}{A \Rightarrow B, A \vdash B}}{A \vdash B} \qquad \vdash\uparrow\;\dfrac{\vdash\Rightarrow\;\dfrac{\overset{\frown}{\vdash}\;\dfrac{}{A \vdash A, B}}{\vdash A, A \Rightarrow B, B}}{\vdash A, B}}{\vdash B}$$

Inadmissibility in deduction modulo

$A \rightarrow A \Rightarrow B$

Let us search a “minimal” counter-example:

$$
\cfrac{
  \cfrac{
    \cfrac{
      \cfrac{}{A, B \vdash B}\,\widehat{\vdash}
      \qquad
      \cfrac{}{A \vdash A, B}\,\widehat{\vdash}
    }{A \Rightarrow B, A \vdash B}\,{\Rightarrow}{\vdash}
  }{A \vdash B}\,{\uparrow}{\vdash}
  \qquad
  \cfrac{
    \cfrac{
      \cfrac{}{A \vdash A, B}\,\widehat{\vdash}
    }{\vdash A, A \Rightarrow B, B}\,{\vdash}{\Rightarrow}
  }{\vdash A, B}\,{\vdash}{\uparrow}
}{\vdash B}\,\underset{\smile}{\vdash}
$$

Proof term: $(\lambda x.\ x\ x)\ (\lambda x.\ x\ x)$

A completion procedure

How to recover cut admissibility?

- If only terms are rewritten: cut admissibility = confluence [Dowek, 2003]

  Recover confluence using standard completion [Knuth and Bendix, 1970]

- If propositions are rewritten: need for a generalization of standard completion

  Complete $A \to A \Rightarrow B$ with $B \to \top$: cut admissibility recovered

A completion procedure to recover cut admissibility

Using the framework of the abstract canonical systems [Dershowitz and Kirchner, 2006, Bonacina and Dershowitz, 2007]

Critical proofs:

$$\dfrac{\dfrac{\overset{\pi}{\Gamma, A, P \vdash \Delta}}{\Gamma, A \vdash \Delta}\uparrow\vdash\ A \longrightarrow P \qquad \dfrac{\overset{\pi'}{\Gamma \vdash Q, A, \Delta}}{\Gamma \vdash A, \Delta}\uparrow\vdash\ A \longrightarrow Q}{\Gamma \vdash \Delta}\ \text{cut}$$

Deduce: Add rewrite rules corresponding to $\Gamma \vdash \Delta$

If terminates, DM the resulting system admits cuts [LFCS'07]

In intuitionistic logic

Some theories cannot be transformed into a rewrite system where the cut admissibility holds $(A \vee B)$

Extension of the completion procedure, mixing **Deduce** and the rules to get a rewrite system from axioms [FroCoS'09]

Implementing a prover for deduction modulo

From scratch?

- probably inefficient

Integrate deduction modulo into a existing prover, benefits from:

- term indexing
- literal selection
- clause simplification

Polarized Resolution Modulo

Ext. Narr. $\dfrac{C, Q}{C, D}\, Q \longrightarrow^{-} D$

Polarized Resolution Modulo

Ext. Narr. $\dfrac{C, Q}{C, D}\, Q \longrightarrow^- D$

Resolution $\dfrac{C, Q \qquad \neg Q, D}{C, D}$

Polarized Resolution Modulo

Ext. Narr. $\dfrac{C, Q}{C, D}\; Q \longrightarrow^{-} D$ Resolution $\dfrac{C, Q \qquad \neg Q, D}{C, D}$

$Q \rightarrow^{-} D$ viewed as clause $\neg Q, D$

- Only $\neg Q$ can be used in a resolution
- Two clauses coming from polarized rules cannot be resolved one with the other

[Dowek 2009]: "one-way clauses"

One-Way Clauses as Known Techniques

Polarized Resolution Modulo =
Set of Support
+ Literal Selection in its complement

Easy to integrate in existing provers thanks to the given-clause algorithm

Benefits from term indexing

Refinement of polarized resolution modulo

Literal Selection in the one-way clauses
What about the other clauses?

Using **order-based literal selection**
and **simplification rules** such as

- strict subsumption elimination
- demodulation

preserves completeness

Implementation

Used the resolution prover within iprover [Korovin 2008]

Tested in on the encoding of the TPTP HOL problems

Category	#Problems	TPS 3.27022008		iprover_mod	
TNE	50	45	42.93s	15	21.56s
THE	150	125	16.06s	30	14.92s
THF	200	170	23.18s	45	17.14s

Outline

Shorter proofs

Deduction modulo may lead to arbitrarily shorter proofs [CSL'2007]

Compare

$$
\forall\vdash\;\dfrac{\Rightarrow\vdash\;\dfrac{\forall\vdash\;\dfrac{\overset{\frown}{\vdash}\;\dfrac{}{\Gamma, \underline{1}+\mathsf{O}=\underline{1} \vdash \underline{1}+\mathsf{O}=\underline{1}, \underline{1}+\underline{1}=\underline{2}}}{\Gamma \vdash \underline{1}+\mathsf{O}=\underline{1}, \underline{1}+\underline{1}=\underline{2}} \qquad \dfrac{\Rightarrow\vdash\;\dfrac{\forall\vdash\;\dfrac{\overset{\frown}{\vdash}\;\dfrac{}{\Gamma, \underline{1}+\underline{1}=s(\underline{1}+\mathsf{O}) \vdash \underline{1}+\underline{1}=s(\underline{1}+\mathsf{O}), \underline{1}+\underline{1}=\underline{2}}}{\Gamma \vdash \underline{1}+\underline{1}=s(\underline{1}+\mathsf{O}), \underline{1}+\underline{1}=\underline{2}} \qquad \overset{\frown}{\vdash}\;\dfrac{}{\Gamma, \underline{1}+\underline{1}=\underline{2} \vdash \underline{1}+\underline{1}=\underline{2}}}{\Gamma, \underline{1}+\underline{1}=s(\underline{1}+\mathsf{O}) \Rightarrow \underline{1}+\underline{1}=\underline{2} \vdash \underline{1}+\underline{1}=\underline{2}}}{\vdots}}{\Gamma, \underline{1}+\mathsf{O}=\underline{1} \Rightarrow \underline{1}+\underline{1}=s(\underline{1}+\mathsf{O}) \Rightarrow \underline{1}+\underline{1}=\underline{2} \vdash \underline{1}+\underline{1}=\underline{2}}}{\Gamma \vdash \underline{1}+\underline{1}=\underline{2}}
$$

to

$$
\vdash\top\;\dfrac{}{\vdash \underline{1}+\underline{1}=\underline{2}}
$$

Application to higher-order arithmetic

Theorem ([Gödel, 1936, Buss, 1994])

There exists a family $(P_j)_{j \in \mathbb{N}}$ such that

- *for all j, $FOA \vdash P_j$*
- *there exists k such that for all j, $SOA \vdash_{\overline{k}} P_j$*
- *there exists no k such that for all j, $FOA \vdash_{\overline{k}} P_j$*

True for all orders i over $i - 1$

Application to higher-order arithmetic

Theorem ([Gödel, 1936, Buss, 1994])

There exists a family $(P_j)_{j\in\mathbb{N}}$ *such that*

- *for all* j, $FOA \vdash P_j$
- *there exists* k *such that for all* j, $SOA \vdash_{\overline{k}} P_j$
- *there exists no* k *such that for all* j, $FOA \vdash_{\overline{k}} P_j$

True for all orders i over $i-1$

[CSL'07] Encoding higher order with deduction modulo, it is possible to stay in first order without increasing the length

Outline

Expressiveness

Theories expressed in deduction modulo:

- simple type theory (HOL) [Dowek et al., 2001]
- Peano's arithmetic [Dowek and Werner, 2005]
- Zermelo's set theory [Dowek and Miquel, 2006]

Can deduction modulo be used as a universal proof environment?

Encoding pure type systems

Pure Type Systems: generic type systems for the lambda calculus with dependant types

- bases of many proof assistants
- can often be used as logical framework

[Cousineau and Dowek, 2007]: encoding of every functional PTS in $\lambda\Pi$-modulo

[LICS'08]: encoding of every functional PTS into (first-order) superdeduction

Ideas

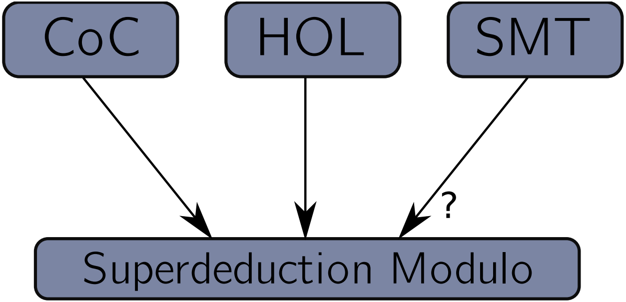
- Use a λ-calculus with explicit substitutions
- Simulation typing derivation rules by superrules:

$$\text{Product}\ \frac{\Gamma \vdash A : s_1 \qquad \Gamma, x : A \vdash B : s_2}{\Gamma \vdash \Pi x : A,\ B : s_3}\ (s_1, s_2, s_3) \in R$$

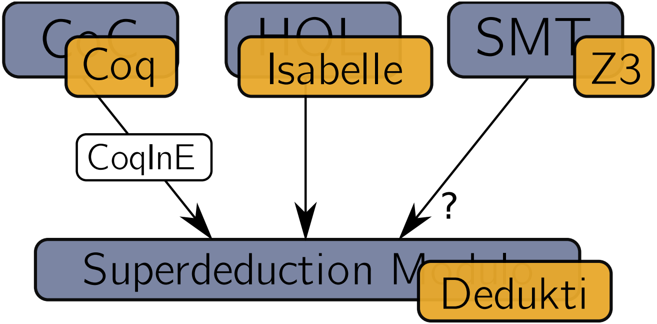
$$\epsilon\left(\dot{\pi}_{\langle s_1, s_2, s_3 \rangle}(a, b), \dot{s}_3\right) \longrightarrow \epsilon(a, \dot{s}_1) \wedge \forall z.\ \epsilon(z, a) \Rightarrow \epsilon(b[cons(z)], \dot{s}_2) \quad \text{(prod)}$$

$$\vdash_{\text{(prod)}}\ \frac{\Gamma \vdash \epsilon(a, \dot{s}_1) \qquad \Gamma, \epsilon(z, a) \vdash \epsilon(b[cons(z)], \dot{s}_2)}{\Gamma \vdash \epsilon\left(\dot{\pi}_{\langle s_1, s_2, s_3 \rangle}(a, b), \dot{s}_3\right)}\ z \notin FV(\Gamma, a, b)$$

A universal proof checker



A universal proof checker



Dedukti: proof checker for $\lambda\Pi$-modulo, M. Boespflug

Outline

- Introduction
- Building Provers Adapted to Theories
- Proof Length Speed-ups
- A Universal Framework
- Conclusion

Conclusion

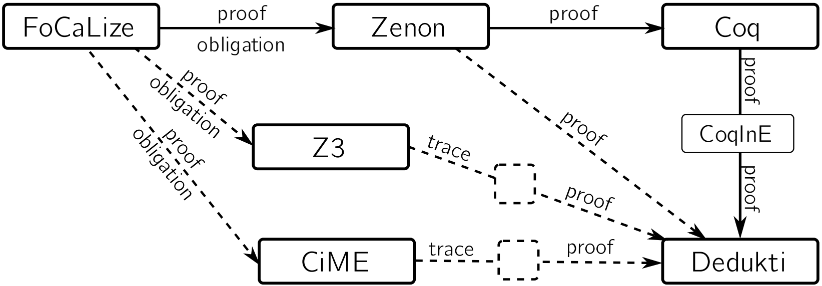
Superdeduction modulo good for:

- reasoning with computations
- reducing proof length
- expressing non-trivial theories and inference systems
- systematically producing provers adapted to a given theory

Further Work

- deduction modulo and equality
- decision procedures from refinement of PRM
- a modular proof environment

Extending FoCaLize with provers?



Bonacina, M. and Dershowitz, N. (2007).
Abstract canonical inference.
ACM Transactions on Computational Logic, 8(1).

Bonichon, R. and Hermant, O. (2006).
A semantic completeness proof for TaMed.
In Hermann, M. and Voronkov, A., editors, *LPAR*, volume 4246 of *LNCS*, pages 167–181. Springer.

Brauner, P., Houtmann, C., and Kirchner, C. (2007).
Principle of superdeduction.
In Ong, L., editor, *Proceedings of LICS*, pages 41–50.

Buss, S. R. (1994).
On Gödel's theorems on lengths of proofs I: Number of lines and speedup for arithmetics.
The Journal of Symbolic Logic, 59(3):737–756.

Cousineau, D. and Dowek, G. (2007).
Embedding pure type systems in the lambda-pi-calculus modulo.
In Ronchi Della Rocca, S., editor, *TLCA*, volume 4583 of *LNCS*, pages 102–117. Springer.

Dershowitz, N. and Kirchner, C. (2006).
Abstract canonical presentations.
Theoretical Computer Science, 357:53–69.

Dowek, G. (2003).
Confluence as a cut elimination property.
In Nieuwenhuis, R., editor, *RTA*, volume 2706 of *LNCS*, pages 2–13. Springer.

Dowek, G., Hardin, T., and Kirchner, C. (2001).

HOL-$\lambda\sigma$ an intentional first-order expression of higher-order logic.
Mathematical Structures in Computer Science, 11(1):1–25.

Dowek, G., Hardin, T., and Kirchner, C. (2003).
Theorem proving modulo.
Journal of Automated Reasoning, 31(1):33–72.

Dowek, G. and Miquel, A. (2006).
Cut elimination for Zermelo's set theory.
Available on authors' web page.

Dowek, G. and Werner, B. (2005).
Arithmetic as a theory modulo.
In Giesl, J., editor, *RTA*, volume 3467 of *LNCS*, pages 423–437. Springer.

Gödel, K. (1936).
Über die Länge von Beweisen.
Ergebnisse eines Mathematischen Kolloquiums, 7:23–24.
English translation in [?].

Knuth, D. E. and Bendix, P. B. (1970).
Simple word problems in universal algebras.
In Leech, J., editor, *Computational Problems in Abstract Algebra*, pages 263–297. Pergamon Press, Oxford.

Wack, B. (2005).
Typage et Déduction dans le Calcul de Réécriture.
PhD thesis, Université Henri Poincaré – Nancy 1.