

Bonnes démonstrations en déduction modulo

Soutenance de thèse

Guillaume Burel

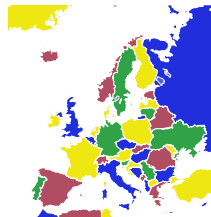
Université Henri Poincaré – Loria

23 mars 2009



Motivation

How to be sure of a complex mathematical proof?
(for instance: 4-color theorem)



How to certify complex software?

⇒ formalizing and automating proofs

Pure logic: well-studied (Frege, Hilbert, Gentzen, etc.)

But proofs are generally done within a theory

- ▶ first-order arithmetic
- ▶ pointer arithmetic
- ▶ etc.

How to present these theories to get better mechanized proof system?

Pure logic: well-studied (Frege, Hilbert, Gentzen, etc.)

But proofs are generally done within a theory

- ▶ first-order arithmetic
- ▶ pointer arithmetic
- ▶ etc.

How to present these theories to get better mechanized proof system?

Standard way of dealing with theories: axiomatization

- ▶ For instance, Peano's axioms for first-order arithmetic
- ▶ Not adapted for proof search!

$1+1=2$ In Γ :

$$\forall x, x + \mathbf{0} = x$$

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

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

$$\begin{array}{c} \frac{\frac{\frac{\Gamma, \underline{1} + \underline{1} = s(\underline{1} + \mathbf{0}) \vdash \underline{1} + \underline{1} = s(\underline{1} + \mathbf{0}), \underline{1} + \underline{1} = \underline{2}}{\vdash \underline{1} + \underline{1} = s(\underline{1} + \mathbf{0}), \underline{1} + \underline{1} = \underline{2}}}{\Rightarrow \vdash \underline{1} + \underline{1} = s(\underline{1} + \mathbf{0}), \underline{1} + \underline{1} = \underline{2}}}{\frac{\frac{\frac{\Gamma, \underline{1} + \underline{1} = \underline{2} \vdash \underline{1} + \underline{1} = \underline{2}}{\vdash \underline{1} + \underline{1} = \underline{2}}} \quad \frac{\Gamma, \underline{1} + \underline{1} = s(\underline{1} + \mathbf{0}) \Rightarrow \underline{1} + \underline{1} = \underline{2} \vdash \underline{1} + \underline{1} = \underline{2}}{\vdash \underline{1} + \underline{1} = \underline{2}}}{\vdash \underline{1} + \underline{1} = \underline{2}}} \\ \frac{\frac{\frac{\frac{\Gamma, \underline{1} + \mathbf{0} = \underline{1} \vdash \underline{1} + \mathbf{0} = \underline{1}, \underline{1} + \underline{1} = \underline{2}}{\vdash \underline{1} + \mathbf{0} = \underline{1}, \underline{1} + \underline{1} = \underline{2}}}{\Rightarrow \vdash \underline{1} + \mathbf{0} = \underline{1}, \underline{1} + \underline{1} = \underline{2}}} \quad \vdots}{\frac{\frac{\Gamma, \underline{1} + \mathbf{0} = \underline{1} \Rightarrow \underline{1} + \underline{1} = s(\underline{1} + \mathbf{0}) \Rightarrow \underline{1} + \underline{1} = \underline{2} \vdash \underline{1} + \underline{1} = \underline{2}}{\vdash \underline{1} + \underline{1} = \underline{2}}}{\vdash \underline{1} + \underline{1} = \underline{2}}}} \end{array}$$

$1+1=2$ In Γ :

$$\forall x, x + \mathbf{0} = x$$

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

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

$$\begin{array}{c} \frac{\frac{\frac{\Gamma, \underline{1} + \underline{1} = s(\underline{1} + \mathbf{0}) \vdash \underline{1} + \underline{1} = s(\underline{1} + \mathbf{0}), \underline{1} + \underline{1} = \underline{2}}{\vdash \underline{1} + \underline{1} = s(\underline{1} + \mathbf{0}), \underline{1} + \underline{1} = \underline{2}}}{\Rightarrow \vdash \underline{1} + \underline{1} = s(\underline{1} + \mathbf{0}), \underline{1} + \underline{1} = \underline{2}}}{\vdash \underline{1} + \underline{1} = s(\underline{1} + \mathbf{0}) \Rightarrow \underline{1} + \underline{1} = \underline{2} \vdash \underline{1} + \underline{1} = \underline{2}} \\ \frac{\frac{\frac{\Gamma, \underline{1} + \mathbf{0} = \underline{1} \vdash \underline{1} + \mathbf{0} = \underline{1}, \underline{1} + \underline{1} = \underline{2}}{\vdash \underline{1} + \mathbf{0} = \underline{1}, \underline{1} + \underline{1} = \underline{2}}}{\Rightarrow \vdash \underline{1} + \mathbf{0} = \underline{1} \Rightarrow \underline{1} + \underline{1} = s(\underline{1} + \mathbf{0}) \Rightarrow \underline{1} + \underline{1} = \underline{2} \vdash \underline{1} + \underline{1} = \underline{2}}}{\vdash \underline{1} + \mathbf{0} = \underline{1} \Rightarrow \underline{1} + \underline{1} = s(\underline{1} + \mathbf{0}) \Rightarrow \underline{1} + \underline{1} = \underline{2} \vdash \underline{1} + \underline{1} = \underline{2}} \end{array}$$

Other approaches

- ▶ Satisfiability Modulo Theory: efficient proof search methods, not generic
DPLL(T) [Ganzinger, Hagen, Nieuwenhuis, Oliveras and Tinelli, 2004]

Other approaches

- ▶ Satisfiability Modulo Theory: efficient proof search methods, not generic
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
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, Hardin and Kirchner, 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

$$x + \mathbf{0} \rightarrow x$$

$$x + s(y) \rightarrow s(x + y)$$

$$\mathbf{0} = \mathbf{0} \rightarrow \top$$

$$s(x) = s(y) \rightarrow x = y$$

$$\underline{1} + \underline{1} = \underline{2} \longrightarrow s(\underline{1} + \mathbf{0}) = \underline{2} \longrightarrow s(\underline{1}) = \underline{2} \xrightarrow{+} \mathbf{0} = \mathbf{0} \longrightarrow \top$$

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

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, Houtmann and Kirchner, 2007]
Left and right superrules

Term rewrite rules are still applied modulo

Goal

How do deduction modulo and superdeduction help produce better proofs from the mechanised-theorem-proving viewpoint?

- ① more direct
- ② shorter
- ③ universal

Goal

How do deduction modulo and superdeduction help produce better proofs from the mechanised-theorem-proving viewpoint?

- ① more direct: **Cut admissibility**
- ② shorter
- ③ universal

Goal

How do deduction modulo and superdeduction help produce better proofs from the mechanised-theorem-proving viewpoint?

- ① more direct: **Cut admissibility**
- ② shorter: **Proof length**
- ③ universal

Goal

How do deduction modulo and superdeduction help produce better proofs from the mechanised-theorem-proving viewpoint?

- ① more direct: **Cut admissibility**
- ② shorter: **Proof length**
- ③ universal: **Logical framework**

Outline

- ① Cut admissibility
 - Example
 - Undecidability
 - A completion procedure
 - Implementation
- ② Proof length
- ③ Logical framework

The cut rule

$$\text{c} \frac{\Gamma, P \vdash \Delta \quad \Gamma \vdash P, \Delta}{\Gamma \vdash \Delta}$$

Proof search procedures complete iff cut admissible

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

Inadmissibility in deduction modulo

$$A \rightarrow A \Rightarrow B$$

Let us search a “minimal” counter-example:

Inadmissibility in deduction modulo

$$A \rightarrow A \Rightarrow B$$

Let us search a “minimal” counter-example:

$$\uparrow \vdash \frac{A \Rightarrow B, A \vdash}{\vdash A \vdash} \quad \vdash \uparrow \frac{\vdash A, A \Rightarrow B}{\vdash A}$$

$$\vdash$$

Inadmissibility in deduction modulo

$$A \rightarrow A \Rightarrow B$$

Let us search a “minimal” counter-example:

$$\Rightarrow \vdash \frac{A, B \vdash \quad \widehat{\vdash} \frac{}{A \vdash A}}{\uparrow \vdash \frac{A \Rightarrow B, A \vdash}{A \vdash}} \quad \vdash \Rightarrow \frac{\widehat{\vdash} \frac{}{A \vdash A, B}}{\vdash \uparrow \frac{\vdash A, A \Rightarrow B}{\vdash A}}}{\vdash}$$

Inadmissibility in deduction modulo

$$A \rightarrow A \Rightarrow B$$

Let us search a “minimal” counter-example:

$$\begin{array}{c}
 \widehat{\vdash} \frac{}{A, B \vdash B} \quad \widehat{\vdash} \frac{}{A \vdash A, B} \quad \widehat{\vdash} \frac{}{A \vdash A, B} \\
 \Rightarrow \vdash \frac{}{A \Rightarrow B, A \vdash B} \quad \vdash \Rightarrow \frac{}{\vdash A, A \Rightarrow B, B} \\
 \uparrow \vdash \frac{}{A \vdash B} \quad \vdash \uparrow \frac{}{\vdash A, B} \\
 \vdash \frac{}{A \vdash B} \quad \vdash \frac{}{\vdash A, B} \\
 \vdash B
 \end{array}$$

Undecidability of cut admissibility

Theorem 1 ([LFCS07]).

The problem:

Given a rewrite system \mathcal{R} , does the sequent calculus modulo \mathcal{R} admits cut?

is undecidable

Sketch of proof: P valid iff the sequent calculus modulo $A \rightarrow A \Rightarrow P$ admits cut

Completion

Recover confluence using standard completion

[Knuth and Bendix, 1970]

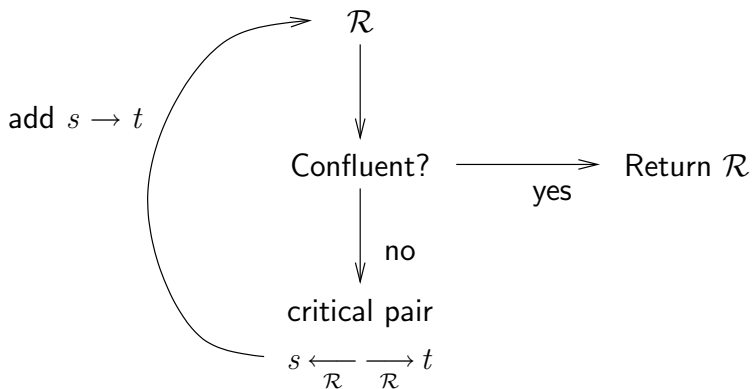
Complete $A \rightarrow A \Rightarrow B$ with $B \rightarrow \top$: cut admissibility recovered

If only terms are rewritten: cut admissibility = confluence

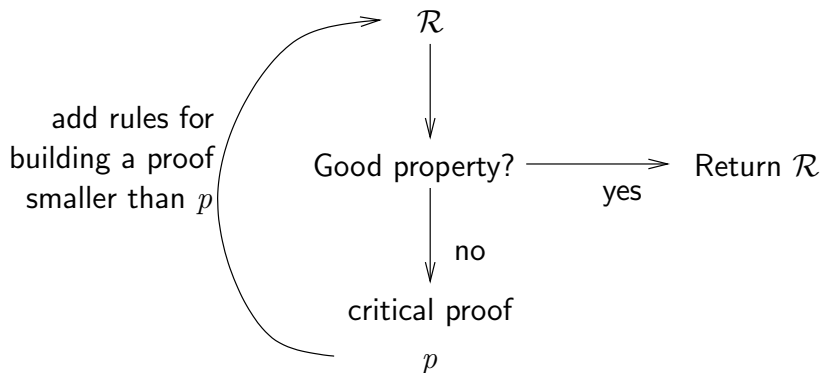
[Dowek, 2003]

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

Basic mechanism of completion (w/o simplification)



Basic mechanism of completion (w/o simplification)



Abstract canonical systems

[Dershowitz and Kirchner, 2006, Bonacina and Dershowitz, 2007]

Order on proofs

- ~> critical proofs (minimal counter-examples)
- ~> completion procedure

Instances: ground completion, standard completion, Moore families, Horn theories, ...

Deduction modulo as an ACS

Polarized unfolding sequent calculus:

$$\uparrow \vdash \frac{\Gamma, A, P \vdash \Delta}{\Gamma, A \vdash \Delta} A \rightarrow^- P \qquad \vdash \uparrow \frac{\Gamma \vdash P, A, \Delta}{\Gamma \vdash A, \Delta} A \rightarrow^+ P$$

Equivalent to the sequent calculus modulo, especially w.r.t. cuts

Order on proofs: RPO with precedence $\vdash > \uparrow > r$ and $\vdash(A \Rightarrow B) > \vdash(A)$

Well adapted to the cut elimination procedure

If the completion terminates, the limit admits cut

Critical proofs

$$\uparrow \vdash \frac{\frac{\uparrow \vdash \frac{\Gamma, A, P \vdash \Delta}{\Gamma, A \vdash \Delta} \pi \quad A \longrightarrow P}{\uparrow \vdash \frac{\Gamma \vdash Q, A, \Delta}{\Gamma \vdash A, \Delta} \pi' \quad A \longrightarrow Q}}{\Gamma \vdash \Delta} \text{cut}}{\Gamma \vdash \Delta} \text{cut}$$

where

- ▶ π and π' without cut
- ▶ π and π' without useless application of rules
- ▶ π and π' apply $\hat{\vdash}$ an atomic formulæ only
- ▶ Γ contains only atomic or universally quantified formulæ $\neq A$
- ▶ Dual for Δ
- ▶ All formulæ in Γ, Δ are used somewhere

Completing formulæ

Find a proof smaller than a critical proof of $\Gamma \vdash \Delta$

\rightsquigarrow Find a rewrite system \mathcal{R} s.t. $\Gamma \vdash_{\mathcal{R}} \Delta$ w/o cut

An algorithm *Rew* from sequents to rewrite systems

Theorem 2.

$\Theta \vdash P$ iff $\vdash_{\text{Rew}(\{\vdash H : H \in \Theta\})} P$

Transforms axiomatic presentations of a theory into rewrite systems

Search for critical proofs

$$\uparrow \vdash \frac{\uparrow \vdash \frac{\frac{\Gamma, A, P \vdash \Delta}{\Gamma, A \vdash \Delta} \pi \quad A \longrightarrow P}{\Gamma \vdash \Delta} \quad \uparrow \vdash \frac{\frac{\Gamma \vdash Q, A, \Delta}{\Gamma \vdash A, \Delta} \pi' \quad A \longrightarrow Q}{\Gamma \vdash \Delta}}{\Gamma \vdash \Delta}$$

Search for critical proofs

$$\Gamma, A, P \vdash^{\pi} \Delta$$

$$\Gamma \vdash^{\pi'} Q, A, \Delta$$

Search for critical proofs

$$A, P \vdash^{\pi} \qquad \vdash^{\pi'} Q, A$$

Search for a cut-free proof, complete branch respecting conditions of critical proofs to find Γ, Δ

Search for critical proofs

$$A, P \stackrel{\pi}{\vdash} \quad \quad \quad \vdash Q, A \stackrel{\pi'}{\quad}$$

Search for a cut-free proof, complete branch respecting conditions of critical proofs to find Γ, Δ

Implementation of the tableaux method TaMed
[Bonichon and Hermant, 2006]

Contributions [LFCS07]

- ▶ Undecidability of cut admissibility in deduction modulo
- ▶ Completion procedure to recover it
- ▶ Algorithm to transform axiomatic presentations into rewrite systems used modulo
- ▶ Implementation in TOM/OCaml

Outline

- ① Cut admissibility
- ② Proof length
 - Motivation
 - First results
 - Application to higher-order arithmetic
- ③ Logical framework

Speed-ups in higher-order arithmetic

Second-order arithmetic proves more than first-order arithmetic, but also more quickly:

Theorem 3 ([Gödel, 1936, Buss, 1994]).

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

- ▶ *for all j , $A_1 \vdash P_j$*
- ▶ *there exists k such that for all j , $A_2 \vdash_k P_j$*
- ▶ *there exists no k such that for all j , $A_1 \vdash_k P_j$*

True for all orders i over $i - 1$

Higher-order logic in deduction modulo

[Dowek, Hardin and Kirchner, 2001] $HOL_{\lambda\sigma}$ encodes the higher-order logic based on simple type theory

Same proof length in $HOL-\lambda$ and in the sequent calculus modulo $HOL_{\lambda\sigma}$

Possibility to encode higher-order arithmetic without increasing proof length?

No restrictions on the rewrite system?

\mathcal{R} : rewrite system such that $P \xrightarrow[\mathcal{R}]{}^* \top$ for all first-order tautology P

All proofs can be abridged to $\vdash^{\top} \frac{}{\vdash P}$

Are those really proofs?

Proof checking is not decidable

A formal framework

If interested with links with complexity theory, proof checking must be performed in polynomial time
[Cook and Reckhow, 1979]

In deduction modulo: the congruence should be decidable in polynomial time

Simple example

$$Add \stackrel{\text{def}}{=} \begin{cases} Add(\mathbf{0}, y, y) \rightarrow \top \\ Add(s(x), y, s(z)) \rightarrow Add(x, y, z) \end{cases}$$

Proposition 4.

- ▶ $\vdash_{\mathcal{I}_{Add}} Add(\underline{i}, \underline{i}, \underline{2i})$
- ▶ $\Theta \vdash_{\mathcal{O}(i)} Add(\underline{i}, \underline{i}, \underline{2i})$ for all finite compatible presentations Θ

$\overset{*}{\longleftrightarrow}_{Add}$ is decidable in polynomial time

Encoding higher order in deduction modulo

Higher-order arithmetic



Higher order: \mathcal{HO}_i

Remaining axioms: fA

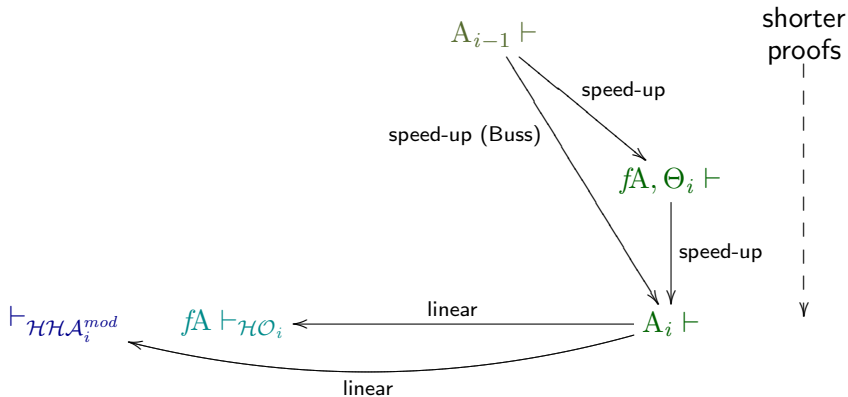
=

comprehension schema +
rules encoding formulæ by terms
[Kirchner, 2006]

Theorem 5.

$$A_i \vdash_k P \rightsquigarrow fA \vdash_{\mathcal{O}(k) \mathcal{HO}_i} P$$

“0th order” 1st order \dots $i - 1$ st order i th order



Contributions [CSL07]

- ▶ Simple speed-ups in deduction modulo
- ▶ Even when counting rewrite steps
(using deep inference [Bruscoli and Guglielmi, 2008])
- ▶ Length-preserving simulation of higher-order arithmetic in first order modulo
- ▶ Purely computational presentation of higher-order arithmetic

Outline

- ① Cut admissibility
- ② Proof length
- ③ Logical framework
 - Motivations
 - Application to the functional pure type systems

Encoding higher-order systems in first order modulo?

- ▶ well studied
- ▶ existing efficient proof search procedures
- ▶ near to implementation
- ▶ universal (tool cooperation)

Encoding higher-order systems in first order modulo?

- ▶ well studied
- ▶ existing efficient proof search procedures
- ▶ near to implementation
- ▶ **universal** (tool cooperation)

Logical framework

[Pfenning, 1996]:

“a meta-language for the specification of deductive systems”

most famous: ELF (based on $\lambda\Pi$)

HOL $\lambda\sigma$ = specification of HOL- λ

Deduction modulo as a logical framework?

Logical framework

[Pfenning, 1996]:

“a meta-language for the specification of deductive systems”

most famous: ELF (based on $\lambda\Pi$)

HOL $\lambda\sigma$ = specification of HOL- λ

Superdeduction as a logical framework?

Logical framework

[Pfenning, 1996]:

“a meta-language for the specification of deductive systems”

most famous: ELF (based on $\lambda\Pi$)

HOL $\lambda\sigma$ = specification of HOL- λ

Superdeduction as a logical framework?

A methodology to specify a deductive system

\rightsquigarrow Application to functional pure type systems

Pure type systems [Geuvers and Nederhof, 1991]

$$T ::= x \mid \lambda x : T, T \mid T T \mid \Pi x : T, T$$

A pure type system is given by

- ▶ sorts \mathcal{S}
- ▶ axioms $\mathbf{A} \subseteq \mathcal{S} \times \mathcal{S}$
- ▶ rules $\mathbf{R} \subseteq \mathcal{S} \times \mathcal{S} \times \mathcal{S}$

Functional if \mathbf{A} and \mathbf{R} are graphs defining functions

Typing system

$$\text{Empty} \frac{}{\square \text{ well-formed}}$$

$$\text{Declaration} \frac{\Gamma \text{ well-formed} \quad \Gamma \vdash A : s}{\Gamma, x : A \text{ well-formed}} \quad s \in \mathbf{S} \text{ and } x \text{ not in } \Gamma$$

$$\text{Sort} \frac{\Gamma \text{ well-formed}}{\Gamma \vdash s_1 : s_2} \quad (s_1, s_2) \in \mathbf{A}$$

$$\text{Variable} \frac{\Gamma \text{ well-formed}}{\Gamma \vdash x : A} \quad x : A \in \Gamma$$

Typing system (cont.)

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

$$\text{Application} \frac{\Gamma \vdash T : \Pi x : A, B \quad \Gamma \vdash U : A}{\Gamma \vdash (T U) : \{U/x\}B}$$

$$\text{Abstraction} \frac{\Gamma \vdash \Pi x : A, B : s \quad \Gamma, x : A \vdash T : B}{\Gamma \vdash \lambda x : A, T : \Pi x : A, B}$$

$$\text{Conversion} \frac{\Gamma \vdash T : A \quad \Gamma \vdash B : s}{\Gamma \vdash T : B} \quad s \in \mathbf{S} \text{ and } A \overset{*}{\longleftrightarrow}_{\beta} B$$

Encoding the λ -terms

Binary predicate $\epsilon(t, u)$ to encode $T : U$ (shallow encoding)

λ -calculus with explicit substitutions [Kesner, 2000]

+ constants \dot{s} for all sorts $s \in \mathbf{S}$

+ binary function $\dot{\pi}_{\langle s_1, s_2, s_3 \rangle}$ for all rules $(s_1, s_2, s_3) \in \mathbf{R}$

additional term rewrite rules:

$$\begin{aligned} \dot{s} [t] &\rightarrow \dot{s} \\ \dot{\pi}_{\langle s_1, s_2, s_3 \rangle} (a, b) [s] &\rightarrow \dot{\pi}_{\langle s_1, s_2, s_3 \rangle} (a [s], b [\mathit{lift}(s)]) \end{aligned}$$

Encoding the inference rules through superrules

Find a rewrite rule of which one superrule correspond to the inference rule

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

$$\epsilon(\dot{\pi}_{\langle s_1, s_2, s_3 \rangle}(a, b), \dot{s}_3) \rightarrow \epsilon(a, \dot{s}_1) \wedge \forall z. \epsilon(z, a) \Rightarrow \epsilon(b[\text{cons}(z)], \dot{s}_2) \quad (2)$$

$$\stackrel{(2)}{\vdash} \frac{\Gamma \vdash \epsilon(a, \dot{s}_1) \quad \Gamma, \epsilon(z, a) \vdash \epsilon(b[\text{cons}(z)], \dot{s}_2)}{\Gamma \vdash \epsilon(\dot{\pi}_{\langle s_1, s_2, s_3 \rangle}(a, b), \dot{s}_3)} \quad z \notin FV(\Gamma, a, b)$$

Correctness

$\mathcal{PTS}_{(S,A,R)}$: explicit substitutions +

$$\epsilon(\dot{s}_1, \dot{s}_2) \rightarrow \top \quad (s_1, s_2) \in \mathbf{A} \quad (1)$$

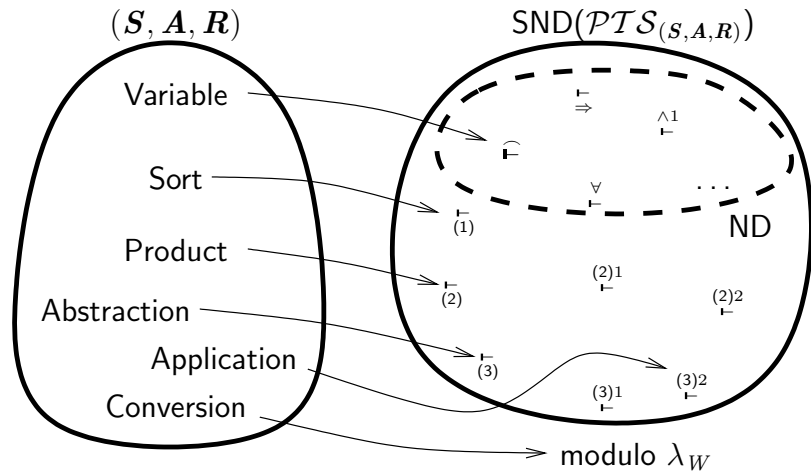
$$\epsilon(\dot{\pi}_{\langle s_1, s_2, s_3 \rangle}(a, b), \dot{s}_3) \rightarrow \epsilon(a, \dot{s}_1) \wedge \forall z. \epsilon(z, a) \Rightarrow \epsilon(b[\text{cons}(z)], \dot{s}_2) \quad (2)$$

$$\epsilon(t, \dot{\pi}_{\langle s_1, s_2, s_3 \rangle}(a, b)) \rightarrow \epsilon(\dot{\pi}_{\langle s_1, s_2, s_3 \rangle}(a, b), \dot{s}_3) \wedge \forall z. \epsilon(z, a) \Rightarrow \epsilon(t z, b[\text{cons}(z)]) \quad (3)$$

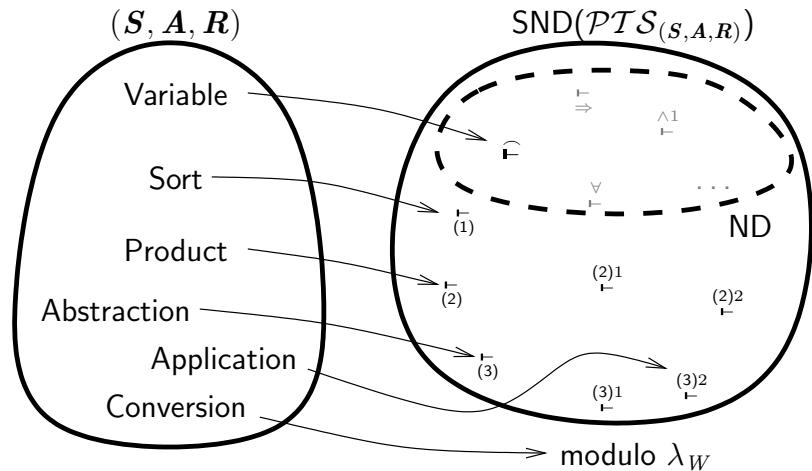
Theorem 6.

If $\Gamma \vdash^{(S,A,R)} T : A$ then $|\Gamma| \vdash^{+\mathcal{PTS}_{(S,A,R)}} \epsilon(|T|^\Gamma, |A|^\Gamma)$

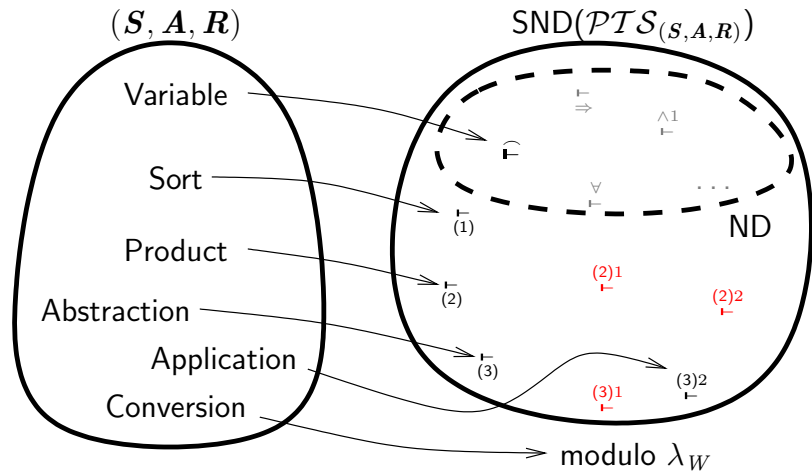
Extra rules



Extra rules



Extra rules



Conservativeness

Extra rules:

$$\stackrel{(2)_1}{\vdash} \frac{\Gamma \vdash \epsilon \left(\dot{\pi}_{\langle s_1, s_2, s_3 \rangle} (a, b), \dot{s}_3 \right)}{\Gamma \vdash \epsilon (a, \dot{s}_1)}$$

By correctness of the translation, if

$|\Pi x : A, B| = \dot{\pi}_{\langle s_1, s_2, s_3 \rangle} (a, b)$ then $A : s_1$

Theorem 7.

If Γ well formed and $|\Gamma| \vdash^{+PT\mathcal{S}(S,A,R)} \epsilon (a, b)$

there exists A and B such that

$$a \xrightarrow[*]{PT\mathcal{S}(S,A,R)} |A| \qquad b \xrightarrow[*]{PT\mathcal{S}(S,A,R)} |B| \qquad \Gamma \vdash^{\underline{(S,A,R)}} A : B$$

Contributions [LICS08]

- ▶ Methodology to encode deductive systems in superdeduction
- ▶ Correct and conservative encoding of functional pure type systems
- ▶ Proof search in PTS via the extensible sequent calculus
- ▶ New insight on normalization in PTS

Outline

- 0 Introduction
- 1 Cut admissibility
- 2 Proof length
- 3 Logical framework
- 4 Conclusion
 - Further work

Other simplicity criteria

- ▶ Normalization

- ▶ new instance of abstract canonical system?
- ▶ simplification rules?

$$A \rightarrow A \Rightarrow B ; B \rightarrow \top \rightsquigarrow A \rightarrow \top ; B \rightarrow \top$$

- ▶ Decidability of the congruence

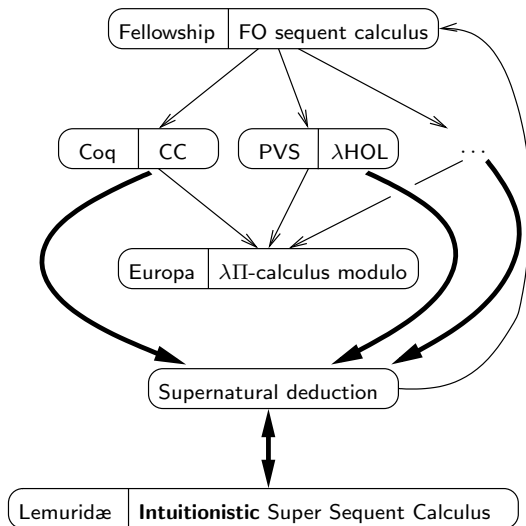
- ▶ decidable proof checking
- ▶ decidability in polynomial time

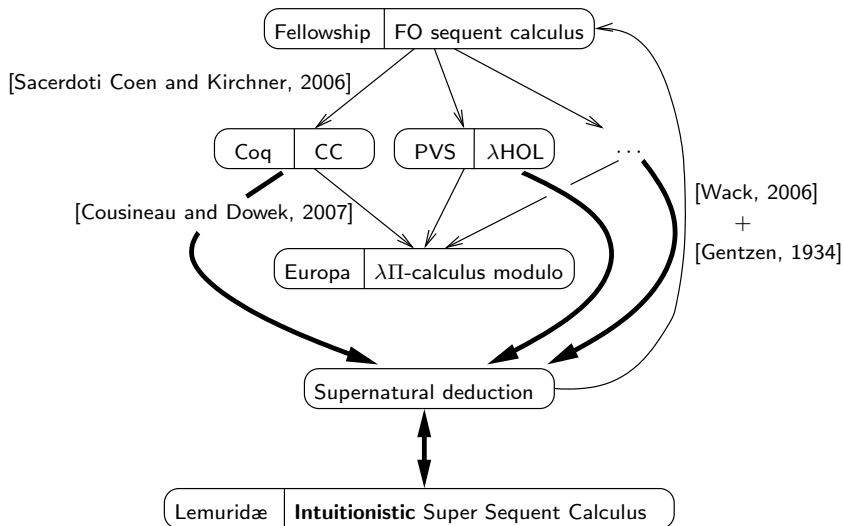
- ▶ Proof length





- ▶ A formal framework for proof complexity
- ▶ Link deduction modulo – Tseytin's extensions





Automating the logical framework

- ▶ From axiomatic presentations to rewrite systems:
 - ▶ automate
 - ▶ ensure the good properties
 - ▶ not always possible in intuitionistic logic
- ▶ Automated theorem proving
 - ▶ term rewrite rule strategies for the modulo
 - ▶ superrules application strategies
 - ▶ automated or user specified?
- ▶ A universal proof environment
 - ▶ share proof developments from different tools
 - ▶ modular deduction modulo
 - ▶ inductive types, subtyping, ...





-  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.
-  Bruscoli, P. and Guglielmi, A. (2008).
On the proof complexity of deep inference.
ACM Transactions on Computational Logic.
To appear.

-  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.
-  Cook, S. A. and Reckhow, R. A. (1979).
The relative efficiency of propositional proof systems.
The Journal of Symbolic Logic, 44(1):36–50.
-  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.



Gentzen, G. (1934).

Untersuchungen über das logische Schliessen.

Mathematische Zeitschrift, 39:176–210, 405–431.

Translated in Szabo, editor., *The Collected Papers of Gerhard Gentzen* as “Investigations into Logical Deduction” .



Gödel, K. (1936).

Über die Länge von Beweisen.

Ergebnisse eines Mathematischen Kolloquiums, 7:23–24.

English translation in [Gödel, 1986].



Gödel, K. (1986).

On the length of proofs.

In Feferman, S. et al., editors, *Kurt Gödel: Collected Works*, volume 1, pages 396–399. Oxford University Press, Oxford.



Kesner, D. (2000).

Confluence of extensional and non-extensional λ -calculi with explicit substitutions.

Theoretical Computer Science, 238(1–2):183–220.



Kirchner, F. (2006).

A finite first-order theory of classes.

In Altenkirch, T. and McBride, C., editors, *TYPES*, volume 4502 of *LNCS*, pages 188–202. Springer.



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.



Pfenning, F. (1996).

The practice of logical frameworks.

In *CAAP*, volume 1059 of *LNCS*, pages 119–134. Springer.

-  Sacerdoti Coen, C. and Kirchner, F. (2006).
Fellowship.
<http://www.lix.polytechnique.fr/Labo/Florent.Kirchner/fellowship/>.
-  Wack, B. (2005).
Typage et Dédution dans le Calcul de Réécriture.
PhD thesis, Université Henri Poincaré – Nancy 1.
-  Wack, B. (2006).
Supernatural deduction.
Manuscript, available at <http://www.loria.fr/~wack/papers/supernatural.ps.gz>.