Weaving computation and deduction

Claude Kirchner

In collaborations and interactions with
Paul Brauner, Guillaume Burel, Horatiu Cirstea,
Nachum Dershowitz, Gilles Dowek, Germain Faure,
Clément Houtmann, Luigi Liquori and Benjamin Wack

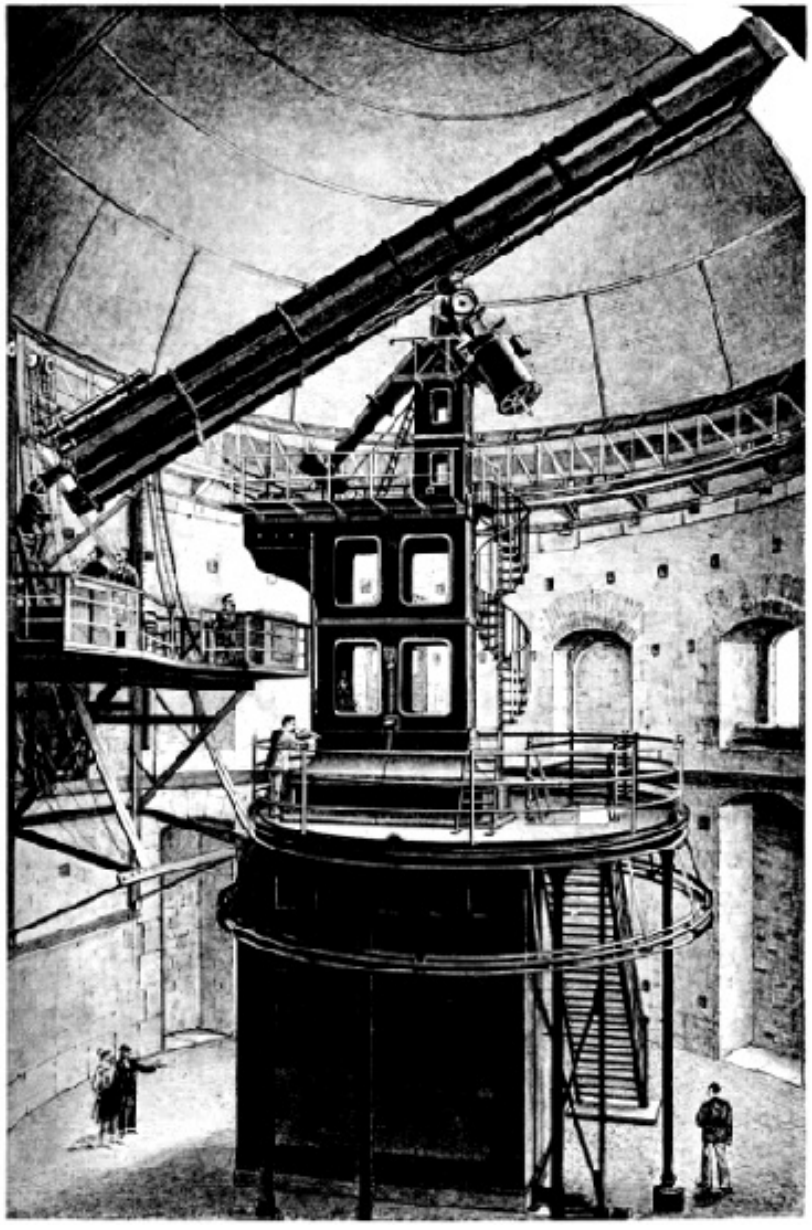
INRIA

Bordeaux - Sud-Ouest
France

Cachan, November 19, 2008

Workshop in honour of Hubert Comon

Sciences and instruments



–La Grande Lunette de l'Observatoire de Meudon. Héliogravure de Dujardin publiée en 1896–

The computational telescope

Computers profoundly change our way to make science

Roadrunner, the world's fastest supercomputer, performs one thousand trillion calculations per second ($1\,000\,000\,000\,000\,000$ (i.e. 10^{15}))

Computation and Security

Is that a real deal?

High-End Computing and NATIONAL SECURITY

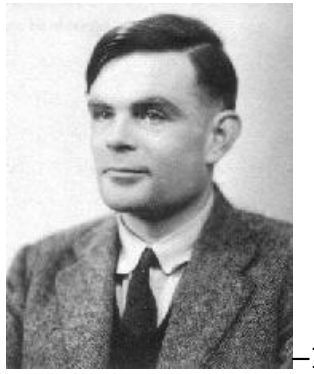
`http://www.scidacreview.org/0802/html/interview.html`
George Cotter (NSA and National Academy)

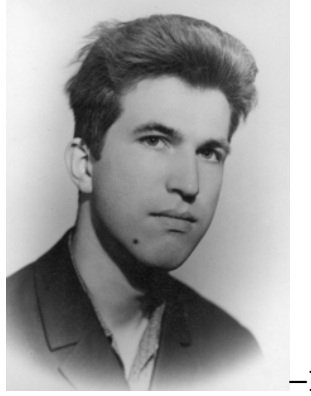
–2008–

Yes, the DNI [Director of National Intelligence] recognizes the criticality of supercomputing to my agency and several others and I'm expected to help find that greater role, and to foster greater collaboration across the intelligence community in HPC and other technologies.

But computation has a strong limit...

Everything cannot be reduced to computation

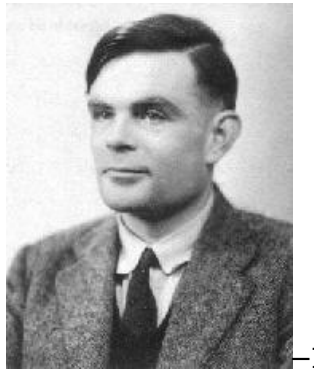
Halting problem [Alan Turing 1936]  –1951–

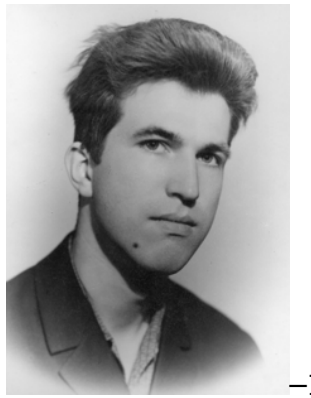
Hilbert's tenth problem [Yuri Matiyasevich 1970]  –1969–

Deduction is definitely needed,
but computation is a first class concept

But computation has a strong limit...

Everything cannot be reduced to computation

Halting problem [Alan Turing 1936]  –1951–



Hilbert's tenth problem [Yuri Matiyasevich 1970] –1969–

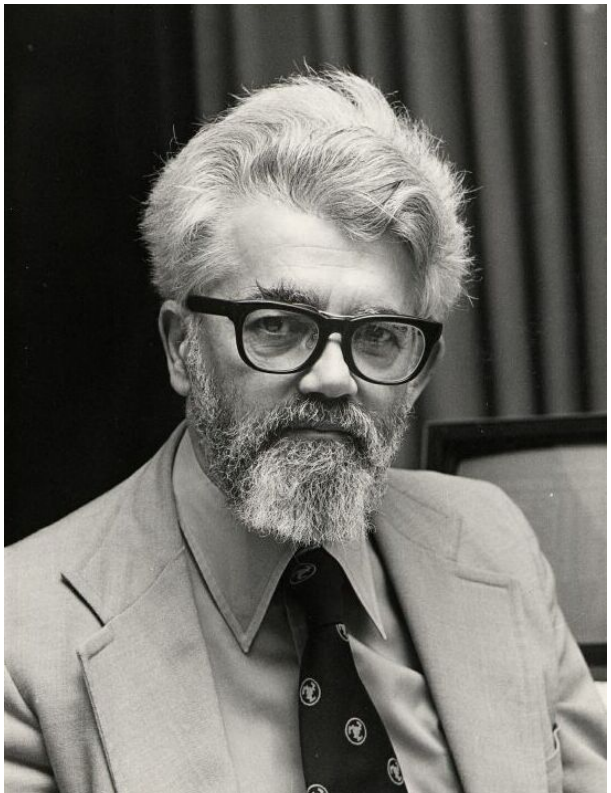
Deduction is definitely needed,
but computation is a first class concept

Computation *and* Deduction

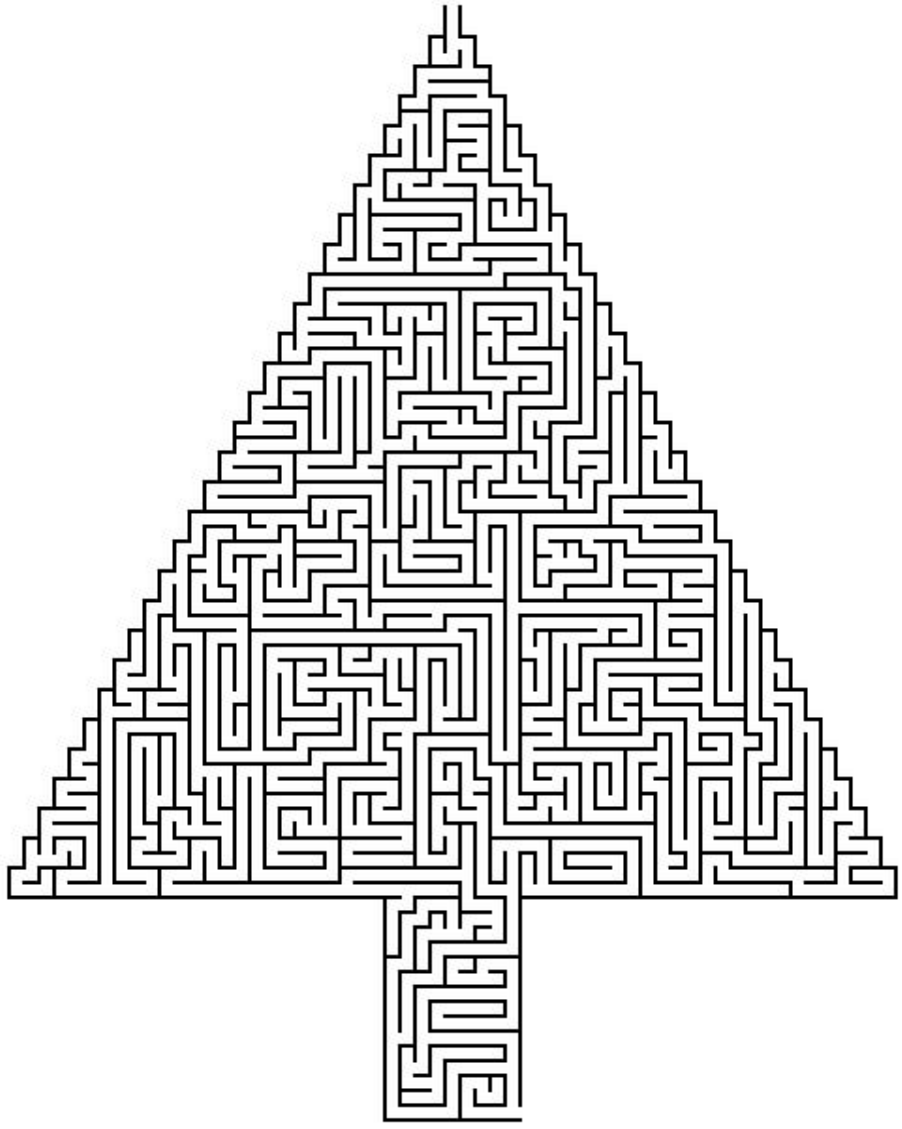
It is reasonable to hope that the relationship between computation and mathematical logic will be as fruitful in the next century as that between analysis and physics in the last.

The development of this relationship demands a concern for both applications and for mathematical elegance.

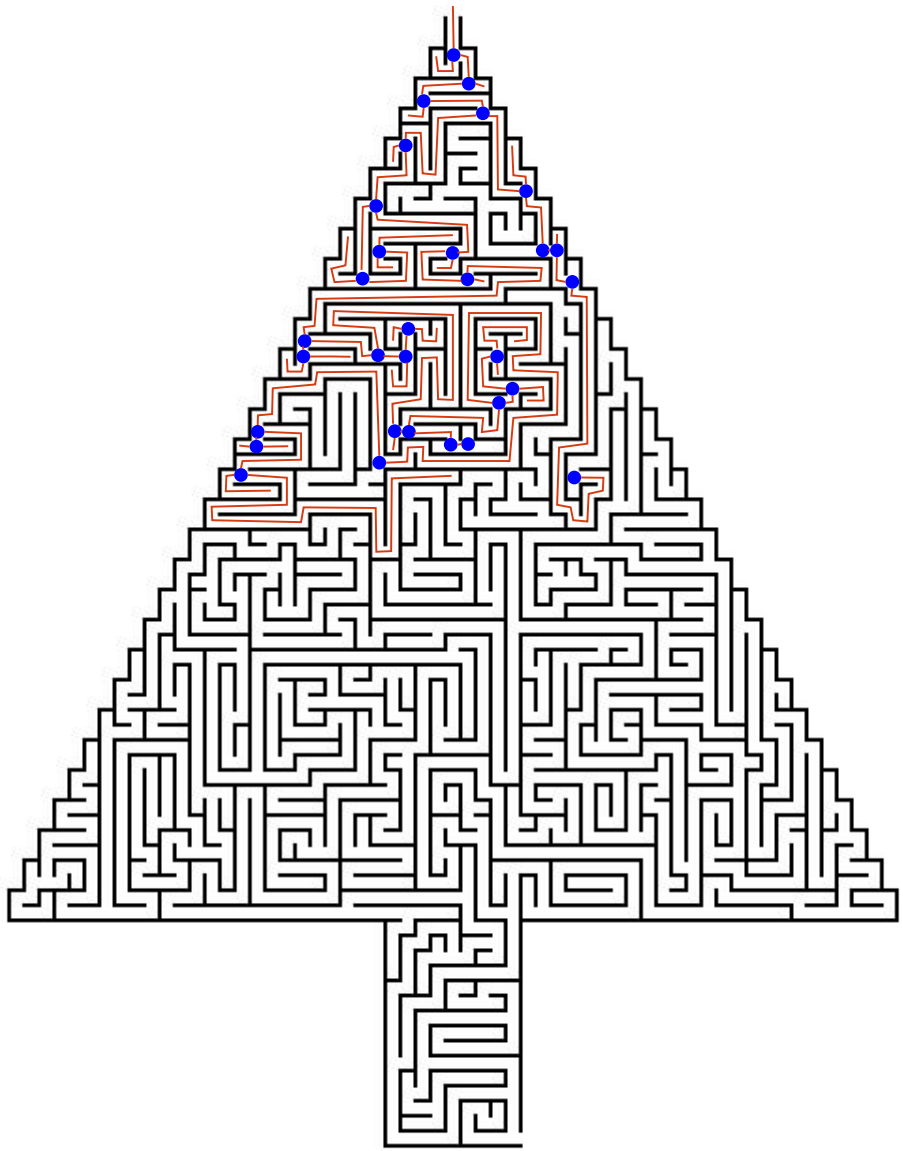
John McCarthy



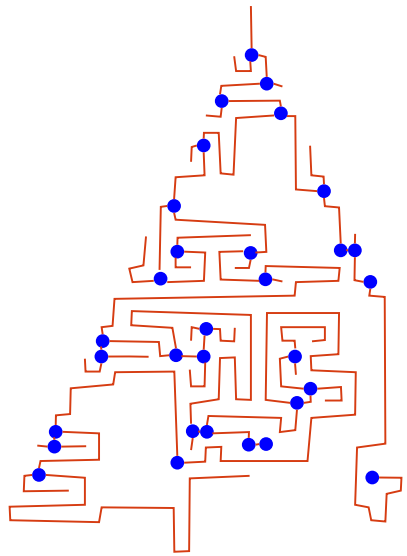
Amazing. . .



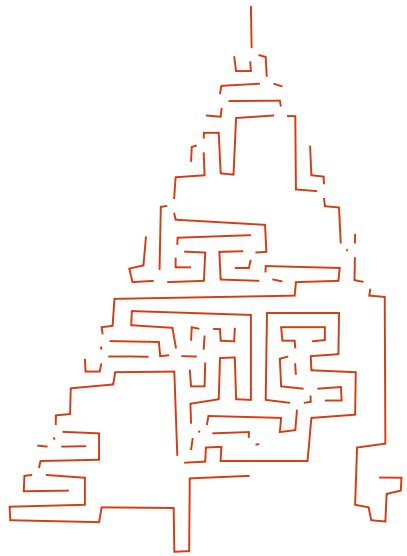
Amazing: Deduction versus Computation



Amazing: Deduction versus Computation



Amazing: just the computation level



Peano's axioms

(pp. 1-20)

ARITHMETICES PRINCIPIA.

§ 1. De numeris et de additione.

Explicationes.

Signo N significatur *numerus* (*integer positivus*).
» 1 » *unitas.*
» $a+1$ » *sequens a*, sive *a plus* 1.
» $=$ » *est aequalis.* Hoc ut novum signum considerandum est, etsi logicae signi figuram habeat.

Axiomata.

1. $1 \in N$.
2. $a \in N . \supset . a = a$.
3. $a, b \in N . \supset : a = b . = . b = a$.
4. $a, b, c \in N . \supset . \therefore a = b . b = c : \supset . a = c$.
5. $a = b . b \in N : \supset . a \in N$.
6. $a \in N . \supset . a + 1 \in N$.
7. $a, b \in N . \supset : a = b . = . a + 1 = b + 1$.
8. $a \in N . \supset . a + 1 - = 1$.
9. $k \in K . \therefore 1 \in k . \therefore x \in N . x \in k : \supset_x . x + 1 \in k :: \supset . N \supset k$.

Definitiones.

10. $2 = 1 + 1 ; 3 = 2 + 1 ; 4 = 3 + 1$; etc.

Theoremata.

11. $2 \in N$.

Demonstratio :

$P\ 1 . \supset :$	$1 \in N$	(1)
$1\ [a]\ (P\ 6) . \supset :$	$1 \in N . \supset . 1 + 1 \in N$	(2)
$(1)\ (2) . \supset :$	$1 + 1 \in N$	(3)
$P\ 10 . \supset :$	$2 = 1 + 1$	(4)
$(4) . (3) . (2, 1 + 1)\ [a, b]\ (P\ 5) : \supset :$	$2 \in N$	(Theorema).

Peano's axioms in modern notations

$$\begin{aligned} \forall y.(0 + y &= y) \\ \forall x.\forall y.(S(x) + y &= x + S(y)) \\ \forall x.(x &= x) \\ \forall x.\forall y.\forall z.((x = y \wedge y = z) &\Rightarrow x = z) \end{aligned}$$

Let us look at the well known example of $1 + 1 = 2$ (not speaking of $2 + 2 = 4$!)

written here $S(0) + S(0) = S(S(0))$

A proof of 1+1 = 2

$$
\dfrac{
\dfrac{
\dfrac{
\dfrac{
\dfrac{(Ax)\ \dfrac{}{\Gamma \vdash \forall x.\forall y.\forall z.(x = y \wedge y = z \Rightarrow x = z)}}{\Gamma \vdash \forall y.\forall z.(S(0) + S(0) = y \wedge y = z \Rightarrow S(0) + S(0) = z)}(\forall E)}{\Gamma \vdash \forall z.(S(0) + S(0) = 0 + S(S(0)) \wedge 0 + S(S(0)) = z \Rightarrow S(0) + S(0) = z)}(\forall E)}{\Gamma \vdash S(0) + S(0) = 0 + S(S(0)) \wedge 0 + S(S(0)) = S(S(0)) \Rightarrow S(0) + S(0) = S(S(0))}(\forall E)
\qquad
\dfrac{
\dfrac{
\dfrac{(Ax)\ \dfrac{}{\Gamma \vdash \forall x.\forall y.(S(x) + y = x + S(y))}}{\Gamma \vdash \forall y.(S(0) + y = 0 + S(y))}(\forall E)}{\Gamma \vdash S(0) + S(0) = 0 + S(S(0))}(\forall E)
\qquad
\dfrac{(Ax)\ \dfrac{}{\Gamma \vdash \forall y.(0 + y = y)}}{\Gamma \vdash 0 + S(S(0)) = S(S(0))}(\forall E)
}{\Gamma \vdash S(0) + S(0) = 0 + S(S(0)) \wedge 0 + S(S(0)) = S(S(0))}(\wedge I)
}{\Gamma \vdash S(0) + S(0) = S(S(0))}(\Rightarrow E)
$$

A proof of 1+1 = 2

$$
\dfrac{
\dfrac{
\dfrac{
\dfrac{
\dfrac{(Ax)\ \dfrac{}{\Gamma \vdash \forall x.\forall y.\forall z.(x = y \wedge y = z \Rightarrow x = z)}}{\Gamma \vdash \forall y.\forall z.(S(0) + S(0) = y \wedge y = z \Rightarrow S(0) + S(0) = z)}(\forall E)}{\Gamma \vdash \forall z.(S(0) + S(0) = 0 + S(S(0)) \wedge 0 + S(S(0)) = z \Rightarrow S(0) + S(0) = z)}(\forall E)}{\Gamma \vdash S(0) + S(0) = 0 + S(S(0)) \wedge 0 + S(S(0)) = S(S(0)) \Rightarrow S(0) + S(0) = S(S(0))}(\forall E)
\qquad
\dfrac{
\dfrac{
\dfrac{(Ax)\ \dfrac{}{\Gamma \vdash \forall x.\forall y.(S(x) + y = x + S(y))}}{\Gamma \vdash \forall y.(S(0) + y = 0 + S(y))}(\forall E)}{\Gamma \vdash S(0) + S(0) = 0 + S(S(0))}(\forall E)
\qquad
\dfrac{(Ax)\ \dfrac{}{\Gamma \vdash \forall y.(0 + y = y)}}{\Gamma \vdash 0 + S(S(0)) = S(S(0))}(\forall E)
}{\Gamma \vdash S(0) + S(0) = 0 + S(S(0)) \wedge 0 + S(S(0)) = S(S(0))}(\wedge I)
}{\Gamma \vdash S(0) + S(0) = S(S(0))}(\Rightarrow E)
$$

Integrating computation: a new proof

Instead of using the axioms:

$$\begin{array}{rcl} \forall y.(0 + y & = & y) \\ \forall x.\forall y.(S(x) + y & = & x + S(y)) \end{array}$$

Define the computation by:

$$\begin{array}{rcl} 0 + y & \rightarrowtriangle & y \\ S(x) + y & \rightarrowtriangle & x + S(y) \end{array}$$

The proof shrinks a lot and becomes:

$$(\forall E)\langle x,\ x{=}x,\ S(S(0))\rangle \dfrac{(Ax)\ \dfrac{}{\Gamma \vdash_{\cong} \forall x.(x = x)}}{\Gamma \vdash_{\cong} S(0) + S(0) = S(S(0))}$$

Integrating computation: a new proof

Instead of using the axioms:

$$\begin{array}{rcl} \forall y.(0 + y & = & y) \\ \forall x.\forall y.(S(x) + y & = & x + S(y)) \end{array}$$

Define the computation by:

$$\begin{array}{rcl} 0 + y & \rightarrowtriangle & y \\ S(x) + y & \rightarrowtriangle & x + S(y) \end{array}$$

The proof shrinks a lot and becomes:

$$(\forall E)\langle x,\ x{=}x,\ S(S(0))\rangle \ \dfrac{(Ax)\ \dfrac{}{\Gamma \vdash_{\cong} \forall x.(x = x)}}{\Gamma \vdash_{\cong} S(0) + S(0) = S(S(0))}$$

Towards a new generation of proof assistants

Computation
and
Deduction
are both
first-class citizen

Gilles Dowek
Grand prix de philosophie de l'Académie Française 2007

Our goals

1. Adapt the computation and deduction systems to the specific domain we are interested in
   Domain Specific Model
2. Do this systematically and safely
   Domain Specific Logic
3. Make this usable
   Automated

In the context of urgent needs for

- Scalable safety and security (e.g. Embedded systems, privacy)
- Formal systems (e.g. proved cryptography)
- Education

Our goals

1. Adapt the computation and deduction systems to the specific domain we are interested in
   Domain Specific Model
2. Do this systematically and safely
   Domain Specific Logic
3. Make this usable
   Automated

In the context of urgent needs for

- Scalable safety and security (e.g. Embedded systems, privacy)
- Formal systems (e.g. proved cryptography)
- Education

Our goals

1. Adapt the computation and deduction systems to the specific domain we are interested in
   Domain Specific Model
2. Do this systematically and safely
   Domain Specific Logic
3. Make this usable
   Automated

In the context of urgent needs for

- Scalable safety and security (e.g. Embedded systems, privacy)
- Formal systems (e.g. proved cryptography)
- Education

Our goals

1. Adapt the computation and deduction systems to the specific domain we are interested in
   Domain Specific Model
2. Do this systematically and safely
   Domain Specific Logic
3. Make this usable
   Automated

In the context of urgent needs for

- Scalable safety and security (e.g. Embedded systems, privacy)
- Formal systems (e.g. proved cryptography)
- Education

Our goals

1. Adapt the computation and deduction systems to the specific domain we are interested in
   Domain Specific Model
2. Do this systematically and safely
   Domain Specific Logic
3. Make this usable
   Automated

In the context of urgent needs for

- Scalable safety and security (e.g. Embedded systems, privacy)
- Formal systems (e.g. proved cryptography)
- Education

Our goals

1. Adapt the computation and deduction systems to the specific domain we are interested in
   Domain Specific Model
2. Do this systematically and safely
   Domain Specific Logic
3. Make this usable
   Automated

In the context of urgent needs for

- Scalable safety and security (e.g. Embedded systems, privacy)
- Formal systems (e.g. proved cryptography)
- Education

Our goals

1. Adapt the computation and deduction systems to the specific domain we are interested in
   Domain Specific Model
2. Do this systematically and safely
   Domain Specific Logic
3. Make this usable
   Automated

In the context of urgent needs for

- Scalable safety and security (e.g. Embedded systems, privacy)
- Formal systems (e.g. proved cryptography)
- Education

Introduction

Deduction Modulo

Super Deduction

Conclusions

What is Deduction modulo?

[Dowek Hardin Kirchner 1998, 2003]

- A presentation of a deduction system,
- that works *modulo* a congruence on term *and* propositions,
- and that makes a clear distinction between *computation* and *deduction*.

Reasoning modulo

The standard modus ponens $\dfrac{A \Rightarrow B \qquad A}{B}$

Modus ponens modulo $\dfrac{A' \Rightarrow B \qquad A}{B}$ if $A' \cong A$

Modus ponens modulo, full power $\dfrac{C \qquad A}{B}$ if $C \cong A \Rightarrow B$

Reasoning modulo

The standard modus ponens

$$\frac{A \Rightarrow B \qquad A}{B}$$

Modus ponens modulo

$$\frac{A' \Rightarrow B \qquad A}{B} \quad \text{if } A' \cong A$$

Modus ponens modulo, full power

$$\frac{C \qquad A}{B} \quad \text{if } C \cong A \Rightarrow B$$

Reasoning modulo

The standard modus ponens

$$\frac{A \Rightarrow B \qquad A}{B}$$

Modus ponens modulo

$$\frac{A' \Rightarrow B \qquad A}{B} \quad \text{if } A' \cong A$$

Modus ponens modulo, full power

$$\frac{C \qquad A}{B} \quad \text{if } C \cong A \Rightarrow B$$

The same approach works for the sequent calculus

$$\frac{}{P \vdash_{\cong} Q} \text{ axiom } \text{ if } P \cong Q$$

$$\frac{\Gamma, Q_1, Q_2 \vdash_{\cong} \Delta}{\Gamma, P \vdash_{\cong} \Delta} \text{ contr-l } \text{ if } P \cong Q_1 \cong Q_2$$

$$\frac{\Gamma, P \vdash_{\cong} \Delta \quad \Gamma \vdash_{\cong} Q, \Delta}{\Gamma \vdash_{\cong} \Delta} \text{ cut } \text{ if } P \cong Q$$

The same approach works for the sequent calculus

$$\frac{}{P \vdash_{\cong} Q} \text{ axiom } \text{ if } P \cong Q$$

$$\frac{\Gamma, Q_1, Q_2 \vdash_{\cong} \Delta}{\Gamma, P \vdash_{\cong} \Delta} \text{ contr-l } \text{ if } P \cong Q_1 \cong Q_2$$

$$\frac{\Gamma, P \vdash_{\cong} \Delta \quad \Gamma \vdash_{\cong} Q, \Delta}{\Gamma \vdash_{\cong} \Delta} \text{ cut } \text{ if } P \cong Q$$

The same approach works for the sequent calculus

$$\frac{}{P \vdash_{\cong} Q} \text{ axiom } \text{ if } P \cong Q$$

$$\frac{\Gamma, Q_1, Q_2 \vdash_{\cong} \Delta}{\Gamma, P \vdash_{\cong} \Delta} \text{ contr-l } \text{ if } P \cong Q_1 \cong Q_2$$

$$\frac{\Gamma, P \vdash_{\cong} \Delta \quad \Gamma \vdash_{\cong} Q, \Delta}{\Gamma \vdash_{\cong} \Delta} \text{ cut } \text{ if } P \cong Q$$

$$\frac{}{P \vdash_{\cong} Q}\ \text{axiom}\ \ \text{if } P \cong Q$$

$$\frac{\Gamma, P \vdash_{\cong} \Delta \quad \Gamma \vdash_{\cong} Q, \Delta}{\Gamma \vdash_{\cong} \Delta}\ \text{cut}\ \ \text{if } P \cong Q$$

$$\frac{\Gamma, Q_1, Q_2 \vdash_{\cong} \Delta}{\Gamma, P \vdash_{\cong} \Delta}\ \text{contr-l}\ \ \text{if } P \cong Q_1 \cong Q_2$$

$$\frac{\Gamma \vdash_{\cong} Q_1, Q_2, \Delta}{\Gamma \vdash_{\cong} P, \Delta}\ \text{contr-r}\ \ \text{if } P \cong Q_1 \cong Q_2$$

$$\frac{\Gamma \vdash_{\cong} \Delta}{\Gamma, P \vdash_{\cong} \Delta}\ \text{weak-l}$$

$$\frac{\Gamma \vdash_{\cong} \Delta}{\Gamma \vdash_{\cong} P, \Delta}\ \text{weak-r}$$

$$\frac{\Gamma, P, Q \vdash_{\cong} \Delta}{\Gamma, R \vdash_{\cong} \Delta}\ \wedge\text{-l}\ \ \text{if } R \cong (P \wedge Q)$$

$$\frac{\Gamma \vdash_{\cong} P, \Delta \quad \Gamma \vdash_{\cong} Q, \Delta}{\Gamma \vdash_{\cong} R, \Delta}\ \wedge\text{-r}\ \ \text{if } R \cong (P \wedge Q)$$

$$\frac{\Gamma, P \vdash_{\cong} \Delta \quad \Gamma, Q \vdash_{\cong} \Delta}{\Gamma, R \vdash_{\cong} \Delta}\ \vee\text{-l}\ \ \text{if } R \cong (P \vee Q)$$

$$\frac{\Gamma \vdash_{\cong} P, Q, \Delta}{\Gamma \vdash_{\cong} R, \Delta}\ \vee\text{-r}\ \ \text{if } R \cong (P \vee Q)$$

$$\frac{\Gamma \vdash_{\cong} P, \Delta \quad \Gamma, Q \vdash_{\cong} \Delta}{\Gamma, R \vdash_{\cong} \Delta}\ \Rightarrow\text{-l}\ \ \text{if } R \cong (P \Rightarrow Q)$$

$$\frac{\Gamma, P \vdash_{\cong} Q, \Delta}{\Gamma \vdash_{\cong} R, \Delta}\ \Rightarrow\text{-r}\ \ \text{if } R \cong (P \Rightarrow Q)$$

$$\frac{\Gamma \vdash_{\cong} P, \Delta}{\Gamma, R \vdash_{\cong} \Delta}\ \neg\text{-l}\ \ \text{if } R \cong \neg P$$

$$\frac{\Gamma, P \vdash_{\cong} \Delta}{\Gamma \vdash_{\cong} R, \Delta}\ \neg\text{-r}\ \ \text{if } R \cong \neg P$$

$$\frac{}{\Gamma, P \vdash_{\cong} \Delta}\ \bot\text{-l}\ \ \text{if } P \cong \bot$$

$$\frac{\Gamma, Q\{t/x\} \vdash_{\cong} \Delta}{\Gamma, P \vdash_{\cong} \Delta}\,(Q, t)\ \forall\text{-l}\ \ \text{if } P \cong \forall x\ Q$$

$$\frac{\Gamma \vdash_{\cong} Q\{y/x\}, \Delta}{\Gamma \vdash_{\cong} P, \Delta}\,(Q, y)\ \forall\text{-r}\ \ \text{if } P \cong \forall x\ Q$$

$$\frac{\Gamma, Q\{y/x\} \vdash_{\cong} \Delta}{\Gamma, P \vdash_{\cong} \Delta}\,(Q, y)\ \exists\text{-l}\ \ \text{if } P \cong \exists x\ Q$$

$$\frac{\Gamma \vdash_{\cong} Q\{t/x\}, \Delta}{\Gamma \vdash_{\cong} P, \Delta}\,(Q, t)\ \exists\text{-r}\ \ \text{if } P \cong \exists x\ Q$$

Use rewriting to define $\cong$

Examples

- $\mathcal{R}$ rewrites Atomic Proposition to Proposition

$$x * y = 0 \rightarrowtriangle_{\mathcal{R}} x = 0 \vee y = 0$$

- $\mathcal{R}$ rewrites Term to Term

$$x * 0 \rightarrowtriangle_{\mathcal{R}} 0$$

- $\mathcal{E}$ equating Term to Term

$$(x * (y * z)) =_{\mathcal{E}} ((x * y) * z)$$
$$(x * y) =_{\mathcal{E}} (y * x)$$

- $\mathcal{RE}$ a class rewrite system $\rightarrowtriangle_{\mathcal{RE}}$

Assume the rewrite systems to be confluent and (weakly) terminating

Use rewriting to define $\cong$

Examples

- $\mathcal{R}$ rewrites Atomic Proposition to Proposition

$$x * y = 0 \rightarrowtriangle_{\mathcal{R}} x = 0 \vee y = 0$$

- $\mathcal{R}$ rewrites Term to Term

$$x * 0 \rightarrowtriangle_{\mathcal{R}} 0$$

- $\mathcal{E}$ equating Term to Term

$$(x * (y * z)) =_{\mathcal{E}} ((x * y) * z)$$
$$(x * y) =_{\mathcal{E}} (y * x)$$

- $\mathcal{RE}$ a class rewrite system $\rightarrowtriangle_{\mathcal{RE}}$

Assume the rewrite systems to be confluent and (weakly) terminating

Use rewriting to define $\cong$

Examples

- $\mathcal{R}$ rewrites Atomic Proposition to Proposition
$$x * y = 0 \rightarrow_{\mathcal{R}} x = 0 \vee y = 0$$
- $\mathcal{R}$ rewrites Term to Term
$$x * 0 \rightarrow_{\mathcal{R}} 0$$
- $\mathcal{E}$ equating Term to Term
$$(x * (y * z)) =_{\mathcal{E}} ((x * y) * z)$$
$$(x * y) =_{\mathcal{E}} (y * x)$$
- $\mathcal{RE}$ a class rewrite system $\rightarrow_{\mathcal{RE}}$

Assume the rewrite systems to be confluent and (weakly) terminating

Use rewriting to define $\cong$

Examples

- $\mathcal{R}$ rewrites Atomic Proposition to Proposition
$$x * y = 0 \rightarrowtriangle_{\mathcal{R}} x = 0 \vee y = 0$$
- $\mathcal{R}$ rewrites Term to Term
$$x * 0 \rightarrowtriangle_{\mathcal{R}} 0$$
- $\mathcal{E}$ equating Term to Term
$$(x * (y * z)) =_{\mathcal{E}} ((x * y) * z)$$
$$(x * y) =_{\mathcal{E}} (y * x)$$
- $\mathcal{RE}$ a class rewrite system $\rightarrowtriangle_{\mathcal{RE}}$

Assume the rewrite systems to be confluent and (weakly) terminating

Example: Integral rings

$\forall x, y \quad x * y = 0 \Longleftrightarrow x = 0 \vee y = 0$

$\mathcal{R} : x * y = 0 \rightarrowtriangle x = 0 \vee y = 0$

How to prove: (A) $\quad \exists z(a * a = z \Rightarrow a = z)$?

$$\dfrac{\dfrac{\dfrac{a = 0 \vdash_{\cong} a = 0 \quad a = 0 \vdash_{\cong} a = 0}{a = 0 \vee a = 0 \vdash_{\cong} a = 0}\ \vee\text{-l}}{\cong}}{\dfrac{a * a = 0 \vdash_{\cong} a = 0}{\vdash_{\cong} a * a = 0 \Rightarrow a = 0}\ \Rightarrow\text{-r}}$$

$$\dfrac{\vdash_{\cong} a * a = 0 \Rightarrow a = 0}{\vdash_{\cong} \exists z(a * a = z \Rightarrow a = z)}\ \exists\text{-r}$$

Security models

From Véronique's talk yesterday:

Composition rules

$$\frac{T \vdash u \quad T \vdash v}{T \vdash \langle u\,, v\rangle} \qquad \frac{T \vdash u \quad T \vdash v}{T \vdash \mathsf{enc}(u, v)} \qquad \frac{T \vdash u \quad T \vdash v}{T \vdash \mathsf{enca}(u, v)}$$

Decomposition rules

$$\frac{}{T \vdash u}\, u \in T \qquad \frac{T \vdash \langle u\,, v\rangle}{T \vdash u} \qquad \frac{T \vdash \langle u\,, v\rangle}{T \vdash v}$$

$$\frac{T \vdash \mathsf{enc}(u, v) \quad T \vdash v}{T \vdash u} \qquad \frac{T \vdash \mathsf{enca}(u, \mathsf{pub}(v)) \quad T \vdash \mathsf{priv}(v)}{T \vdash u}$$

Is deduction modulo a trivial extention of deduction?

- For each congruence $\cong$, there is a set of axioms $\mathcal{T}$ such that

$$\mathcal{T}, \Gamma \vdash \Delta \quad \text{if and only if} \quad \Gamma \vdash_{\cong} \Delta.$$

Hence the theorems are the same in the two formalisms

- The proofs are very different in the two formalisms
–remember the $1+1 = 2$ example–
and
Cut elimination does NOT hold in deduction modulo

Is deduction modulo a trivial extention of deduction?

- For each congruence $\cong$, there is a set of axioms $\mathcal{T}$ such that

$$\mathcal{T}, \Gamma \vdash \Delta \quad \text{if and only if} \quad \Gamma \vdash_{\cong} \Delta.$$

Hence the theorems are the same in the two formalisms

- The proofs are very different in the two formalisms
–remember the $1+1 = 2$ example–
and
Cut elimination does NOT hold in deduction modulo

Deep differences

- The congruence changes the proof logical structure (cut may remain)
- The congruence profoundly modifies the complexity of the proofs

Introduction

Deduction Modulo
- Definition
- Expressiveness of Deduction Modulo
- Changes in Proofs Logical Structure
- Change in proof complexities

Super Deduction

Conclusions

Cut-free proofs

Theorem (Gerhard Gentzen 1934)

In the sequent calculus, all theorems have a proof that does not involve the `cut` *deduction rule.*

Cut elimination is central in proof theory e.g. to demonstrate unprovability results, completeness of proof-search methods.

A sequent calculus without cut-elimination is like a car without engine

—J.-Y. Girard

Crabbé's example:
Find a proof where cut could not be removed

Rewrite system: $A \longrightarrow B \wedge \neg A$
derived from Crabbé's proof of the non-normalization of Zermelo's theory

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{B, A, B, B \vdash A}\,\mathrm{Ax}}{B, A, B, \neg A, B \vdash}\,\neg\text{-l}}{B, A, B \wedge \neg A, B \vdash}\,\wedge\text{-l}}{B, A, B \vdash}\,\uparrow\text{-l}}{B, A \wedge B \vdash}\,\wedge\text{-l} \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{B, A \vdash A}\,\mathrm{Ax}}{B \vdash A, B}\,\neg\text{-r} \qquad \dfrac{}{B \vdash A, \neg A}\,\mathrm{Ax}}{B \vdash A, B \wedge \neg A}\,\wedge\text{-r}}{B \vdash A, A \wedge \neg A}\,\uparrow\text{-r}}{B \vdash A}\,\wedge\text{-r} \qquad \dfrac{}{A \vdash A}\,\mathrm{Ax}$$

$$\dfrac{B, A \wedge B \vdash \qquad B \vdash A \wedge B}{\vdash}\,\mathrm{Cut}$$

Crabbé's example:
Find a proof where cut could not be removed

Rewrite system: $A \longrightarrow B \wedge \neg A$
derived from Crabbé's proof of the non-normalization of Zermelo's theory

$$\text{Cut}\ \frac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{B, A, B, B \vdash A}\text{Ax}}{B, A, B, \neg A, B \vdash}\neg\text{-l}}{B, A, B \wedge \neg A, B \vdash}\wedge\text{-l}}{B, A, B \vdash}\uparrow\text{-l}}{B, A \wedge B \vdash}\wedge\text{-l} \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{B \vdash A, B}\text{Ax} \qquad \dfrac{\dfrac{}{B, A \vdash A}\text{Ax}}{B \vdash A, \neg A}\neg\text{-r}}{B \vdash A, B \wedge \neg A}\wedge\text{-r}}{B \vdash A}\uparrow\text{-r} \qquad \dfrac{}{B \vdash B}\text{Ax}}{B \vdash A \wedge B}\wedge\text{-r}}{B \vdash}$$

Crabbé's example:
Find a proof where cut could not be removed

Rewrite system: $A \longrightarrow B \wedge \neg A$
derived from Crabbé's proof of the non-normalization of Zermelo's theory

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{B, A, B, B \vdash A}\,\text{Ax}}{B, A, B, \neg A, B \vdash}\,\neg\text{-l}}{B, A, B \wedge \neg A, B \vdash}\,\wedge\text{-l}}{B, A, B \vdash}\,\uparrow\text{-l}}{B, A \wedge B \vdash}\,\wedge\text{-l} \qquad \dfrac{\dfrac{\dfrac{}{B \vdash A, B}\,\text{Ax} \qquad \dfrac{\dfrac{}{B, A \vdash A}\,\text{Ax}}{B \vdash A, \neg A}\,\neg\text{-r}}{B \vdash A, B \wedge \neg A}\,\wedge\text{-r}}{B \vdash A}\,\uparrow\text{-r} \qquad \dfrac{}{B \vdash B}\,\text{Ax}}{B \vdash A \wedge B}\,\wedge\text{-r}}{B \vdash}\,\text{Cut}$$

Crabbé's example:
Find a proof where cut could not be removed

Rewrite system: $A \longrightarrow B \wedge \neg A$
derived from Crabbé's proof of the non-normalization of Zermelo's theory

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{B, A, B, B \vdash A}\,\text{Ax}}{B, A, B, \neg A, B \vdash}\,\neg\text{-l}}{B, A, B \wedge \neg A, B \vdash}\,\wedge\text{-l}}{B, A, B \vdash}\,\uparrow\text{-l}}{B, A \wedge B \vdash}\,\wedge\text{-l} \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{B \vdash A, B}\,\text{Ax} \qquad \dfrac{\dfrac{}{B, A \vdash A}\,\text{Ax}}{B \vdash A, \neg A}\,\neg\text{-r}}{B \vdash A, B \wedge \neg A}\,\wedge\text{-r}}{B \vdash A}\,\uparrow\text{-r} \qquad \dfrac{}{B \vdash B}\,\text{Ax}}{B \vdash A \wedge B}\,\wedge\text{-r}}{B \vdash}\,\text{Cut}$$

Crabbé's example:
Find a proof where cut could not be removed

Rewrite system: $A \longrightarrow B \wedge \neg A$
derived from Crabbé's proof of the non-normalization of Zermelo's theory

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{B, A, B, B \vdash A}\,\text{Ax}}{B, A, B, \neg A, B \vdash}\,\neg\text{-l}}{B, A, B \wedge \neg A, B \vdash}\,\wedge\text{-l}}{B, A, B \vdash}\,\uparrow\text{-l}}{B, A \wedge B \vdash}\,\wedge\text{-l} \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{B \vdash A, B}\,\text{Ax} \qquad \dfrac{\dfrac{}{B, A \vdash A}\,\text{Ax}}{B \vdash A, \neg A}\,\neg\text{-r}}{B \vdash A, B \wedge \neg A}\,\wedge\text{-r}}{B \vdash A}\,\uparrow\text{-r} \qquad \dfrac{}{B \vdash B}\,\text{Ax}}{B \vdash A \wedge B}\,\wedge\text{-r}}{B \vdash}\,\text{Cut}$$

Cut elimination

If there is no modulo, Cut is admissible (Gentzen's Hauptsatz)
There exists a cut elimination procedure:

$$\mathrm{Cut}\ \dfrac{\wedge\text{-l}\ \dfrac{\Gamma, A, B \vdash \Delta}{\Gamma, A \wedge B \vdash \Delta} \qquad \wedge\text{-l}\ \dfrac{\Gamma \vdash A, \Delta \qquad \Gamma \vdash B, \Delta}{\Gamma \vdash A \wedge B, \Delta}}{\Gamma \vdash \Delta}$$

becomes

$$\mathrm{Cut}\ \dfrac{\Gamma, A, B \vdash \Delta \qquad \Gamma, A \vdash B, \Delta}{\mathrm{Cut}\ \dfrac{\Gamma, A \vdash \Delta \qquad\qquad \Gamma \vdash A, \Delta}{\Gamma \vdash \Delta}}$$

Proofs with cuts are replaced by smaller counterexamples

Crabbé's counterexample (Cont.)

$A \longrightarrow B \wedge \neg A$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{B, A, B, B \vdash A}\,\text{Ax}}{B, A, B, \neg A, B \vdash}\,\neg\text{-l}}{B, A, B \wedge \neg A, B \vdash}\,\wedge\text{-l}}{B, A, B \vdash}\,\uparrow\text{-l} \qquad \dfrac{}{B, A \vdash B}\,\text{Ax}}{B, A \vdash}\,\text{Cut} \qquad \dfrac{\dfrac{\dfrac{}{B \vdash B, A}\,\text{Ax} \qquad \dfrac{\dfrac{}{B, A \vdash A}\,\text{Ax}}{B \vdash \neg A, A}\,\neg\text{-r}}{B \vdash B \wedge \neg A, A}\,\wedge\text{-r}}{B \vdash A}\,\uparrow\text{-r}}{B \vdash}\,\text{Cut}$$

= minimal counterexample

Crabbé's counterexample (Cont.)

$A \longrightarrow B \wedge \neg A$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{B, A, B \vdash A}\,\text{Ax}}{B, A, B, \neg A \vdash}\,\neg\text{-l}}{B, A, B \wedge \neg A \vdash}\,\wedge\text{-l}}{B, A \vdash}\,\uparrow\text{-l} \qquad \dfrac{\dfrac{\dfrac{}{B \vdash B, A}\,\text{Ax} \qquad \dfrac{\dfrac{}{B, A \vdash A}\,\text{Ax}}{B \vdash \neg A, A}\,\neg\text{-r}}{B \vdash B \wedge \neg A, A}\,\wedge\text{-r}}{B \vdash A}\,\uparrow\text{-r}}{B \vdash}\,\text{Cut}$$

= minimal counterexample

Crabbé's counterexample (Cont.)

$A \longrightarrow B \wedge \neg A$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{B, A, B \vdash A}\,\text{Ax}}{B, A, B, \neg A \vdash}\,\neg\text{-l}}{B, A, B \wedge \neg A \vdash}\,\wedge\text{-l}}{B, A \vdash}\,\uparrow\text{-l} \qquad \dfrac{\dfrac{\dfrac{}{B \vdash B, A}\,\text{Ax} \qquad \dfrac{\dfrac{}{B, A \vdash A}\,\text{Ax}}{B \vdash \neg A, A}\,\neg\text{-r}}{B \vdash B \wedge \neg A, A}\,\wedge\text{-r}}{B \vdash A}\,\uparrow\text{-r}}{B \vdash}\,\text{Cut}$$

= minimal counterexample

Regaining Cut admissibility

Need to build a proof of $B \vdash$

Add a new rule to your system: $B \longrightarrow \bot$

$$\dfrac{\dfrac{}{\bot \vdash}\ \bot\text{-I}}{B \vdash}\ \uparrow\text{-I}$$

$\begin{cases} A \longrightarrow B \wedge \neg A \\ B \longrightarrow \bot \end{cases}$ admits Cut

Regaining Cut admissibility

Need to build a proof of $B \vdash$

Add a new rule to your system: $B \rightarrow \bot$

$$\dfrac{\dfrac{}{\bot \vdash}\ \bot\text{-I}}{B \vdash}\ \uparrow\text{-I}$$

$\begin{cases} A \rightarrow B \wedge \neg A \\ B \rightarrow \bot \end{cases}$ admits Cut

How does it work in general?

The gap is important as the admissibility of the cut rule is undecidable when one works modulo.

To recover cut admissibility, we should complete the congruence.

1. manualy (difficult!)
2. automatically by computing the appropriate critical pairs

How does it work in general?

The gap is important as the admissibility of the cut rule is undecidable when one works modulo.

To recover cut admissibility, we should complete the congruence.

1. manualy (difficult!)
2. automatically by computing the appropriate critical pairs
   2.1 Cut admissibility is equivalent to the confluence of the rewrite system, provided only **terms** are rewritten [Dowek, RTA'03]
   2.2 A completion process can be designed to achieve cut admissibility provided quantifier free rules are considered in the congruence [Dowek, RTA'03]
   2.3 The general case is solved in [Burel&Kirchner, LFCS'06, '08] using the Abstract Canonical System framework [Dershowitz&Kirchner, LICS'03, '06], [Dershowitz&Bonacina,2005]

How does it work in general?

The gap is important as the admissibility of the cut rule is undecidable when one works modulo.

To recover cut admissibility, we should complete the congruence.

1. manualy (difficult!)
2. automatically by computing the appropriate critical pairs
   - 2.1 Cut admissibility is equivalent to the confluence of the rewrite system, provided only **terms** are rewritten [Dowek, RTA'03]
   - 2.2 A completion process can be designed to achieve cut admissibility provided **quantifier free** rules are considered in the congruence [Dowek, RTA'03]
   - 2.3 The **general case** is solved in [BurelKirchner, LFCS'06, 08] using the *Abstract Canonical System* framework [DershowitzKirchner, LICS'03, 06], [DershowitzBonacina,2005]

How does it work in general?

The gap is important as the admissibility of the cut rule is undecidable when one works modulo.

To recover cut admissibility, we should complete the congruence.

1. manualy (difficult!)
2. automatically by computing the appropriate critical pairs
   - 2.1 Cut admissibility is equivalent to the confluence of the rewrite system, provided only **terms** are rewritten [Dowek, RTA'03]
   - 2.2 A completion process can be designed to achieve cut admissibility provided **quantifier free** rules are considered in the congruence [Dowek, RTA'03]
   - 2.3 The **general case** is solved in [BurelKirchner, LFCS'06, 08] using the *Abstract Canonical System* framework [DershowitzKirchner, LICS'03, 06], [DershowitzBonacina,2005]

How does it work in general?

The gap is important as the admissibility of the cut rule is undecidable when one works modulo.

To recover cut admissibility, we should complete the congruence.

1. manualy (difficult!)
2. automatically by computing the appropriate critical pairs
   - 2.1 Cut admissibility is equivalent to the confluence of the rewrite system, provided only **terms** are rewritten [Dowek, RTA'03]
   - 2.2 A completion process can be designed to achieve cut admissibility provided **quantifier free** rules are considered in the congruence [Dowek, RTA'03]
   - 2.3 The **general case** is solved in [BurelKirchner, LFCS'06, 08] using the *Abstract Canonical System* framework [DershowitzKirchner, LICS'03, 06], [DershowitzBonacina,2005]

How does it work in general?

The gap is important as the admissibility of the cut rule is undecidable when one works modulo.

To recover cut admissibility, we should complete the congruence.

1. manualy (difficult!)
2. automatically by computing the appropriate critical pairs
   - 2.1 Cut admissibility is equivalent to the confluence of the rewrite system, provided only **terms** are rewritten [Dowek, RTA'03]
   - 2.2 A completion process can be designed to achieve cut admissibility provided **quantifier free** rules are considered in the congruence [Dowek, RTA'03]
   - 2.3 The **general case** is solved in [BurelKirchner, LFCS'06, 08] using the *Abstract Canonical System* framework [DershowitzKirchner, LICS'03, 06], [DershowitzBonacina,2005]

Extended compatibility

The Rew algorithm on $\Gamma \vdash \Delta$:
build partial sequent proofs

1. "remove" negation; for instance
   $A, \neg B \vdash \neg C, \neg\neg D$ becomes $A, C \vdash B, D$
2. "isolate last"; for instance
   - $\Gamma \vdash Q_1, \ldots, Q_m$ becomes $\Gamma, \neg Q_1, \ldots, \neg Q_{m-1} \vdash Q_m$
   - $P_1, \ldots, P_n \vdash \exists x.\ Q$ becomes $P_1, \ldots, P_n, \neg\underline{\exists x.\ Q} \vdash Q[t/x]$ where t can be any term (x for instance)
   - $P_1, \ldots, P_n \vdash A$ generates $A \longrightarrow \exists x_1, \ldots, x_p.\ (P_1 \wedge \cdots \wedge P_n)$
3. "isolate first"; dual of "isolate first"

Proposition

If there is a proof of a sequent $\Gamma' \vdash \Delta'$ *modulo* $Rew(\Gamma \vdash \Delta) \cup R$, *there is a proof of* $\mathcal{P}(\Gamma, \Delta), \Gamma' \vdash \Delta'$ *modulo* R.

Internalize a theory

Theorem

For all finite set of formulæ Γ and rewrite systems R, there exists a rewrite system R' such that for all finite set of formulæ Δ:

$$\Gamma \vdash_R \Delta \quad \textit{iff} \quad \vdash_{R'} \Delta$$

Proof.

Simply take $R' = R \cup \bigcup_{P \in \Gamma} \mathit{Rew}(\vdash P)$. □

Weave computation and deduction

1. internalize the theory using strong compatibility,
2. complete the resulting congruence (proposition rewrite system) to get cut admissibility.

Applications to Arithmetic

Introduction

Deduction Modulo
- Definition
- Expressiveness of Deduction Modulo
- Changes in Proofs Logical Structure
- Change in proof complexities

Super Deduction

Conclusions

Proofs in HOL

Theorem ([Dowek et al., 2001])

If Γ, Δ *are sets of propositions in HOL-*λ *then*

$$\Gamma \vdash \Delta \quad \textit{iff} \quad \varepsilon(\Gamma_F) \vdash_{\lambda\sigma\mathcal{L}} \varepsilon(\Delta_F)$$

and the proof lengths are the same.

Proof Complexities: Towards the essence of proofs

Theorem (Buss (conjectured by Gödel))

Let $i \geq 0$. Then there is an infinite family $\mathcal{F}$ of $\prod_1^0$-formulas such that

1. *for all $\phi \in \mathcal{F}$, $Z_i \vdash \phi$*
2. *there is a fixed $k \in \mathbb{N}$ such that for all $\phi \in \mathcal{F}$, $Z_{i+1} \vdash_{\overline{k \text{ steps}}} \phi$*
3. *there is no fixed $k \in \mathbb{N}$ such that for all $\phi \in \mathcal{F}$, $Z_i \vdash_{\overline{k \text{ steps}}} \phi$*

Theorem (Burel, CSL 2007)

For all $i \geq 0$, there exists a (finite) rewrite system $\rightarrow_i$ such that for all formulæ P, if $Z_{i+1} \vdash_k P$ then $Z_i \vdash_{\overline{O(k)\ \rightarrow_i}} P$.

Proof Complexities: Towards the essence of proofs

Theorem (Buss (conjectured by Gödel))

Let $i \geq 0$. Then there is an infinite family $\mathcal{F}$ of $\prod_1^0$-formulas such that

1. *for all $\phi \in \mathcal{F}$, $Z_i \vdash \phi$*
2. *there is a fixed $k \in \mathbb{N}$ such that for all $\phi \in \mathcal{F}$, $Z_{i+1} \vdash_{\overline{k \text{ steps}}} \phi$*
3. *there is no fixed $k \in \mathbb{N}$ such that for all $\phi \in \mathcal{F}$, $Z_i \vdash_{\overline{k \text{ steps}}} \phi$*

Theorem (Burel, CSL 2007)

For all $i \geq 0$, there exists a (finite) rewrite system $\rightarrow_i$ such that for all formulæ P, if $Z_{i+1} \vdash_{\overline{k}} P$ then $Z_i \vdash_{\overline{O(k)}\ \rightarrow_i} P$.

Towards Good Proofs

Proofs are fundamental certificates

Certificates should be informative, communicable and checkable

So proof should be good...

What is a good proof?
Towards a canonical notion of proofs where computational arguments are handled appropriately.

Deduction Modulo Summary

1. Cut elimination does not hold anymore in general.
2. For a large class of theory, it does hold:
   - $HOL_{\lambda\sigma}$ provides a first-order presentation modulo of *HOL* [Dowek et al., 2001],
   - Arithmetic can be presented as a theory modulo [Dowek and Werner, 2005],
   - PTS too [Cousineau and Dowek, 2007] [Burel, 2008],
   - For congruences defined only on terms,
   - For congruences defined by terminating and confluence rewrite systems, either quantifier free or positive.

   and then:

   2.1 The concept of logical framework now explicitely integrates computations

   2.2 Generalized resolution provides a correct and complete proof search: ENAR [Dowek et al., 2003]

   2.3 A tableau method is also given: TaMed [Bonichon, 2004]

Introduction

Deduction Modulo

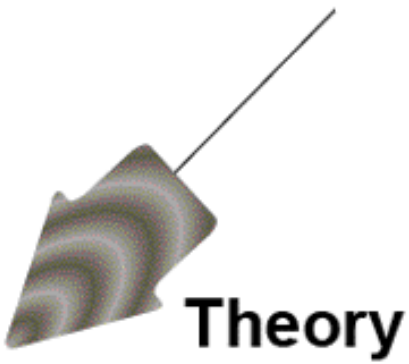
Super Deduction

Conclusions

Super Deduction?

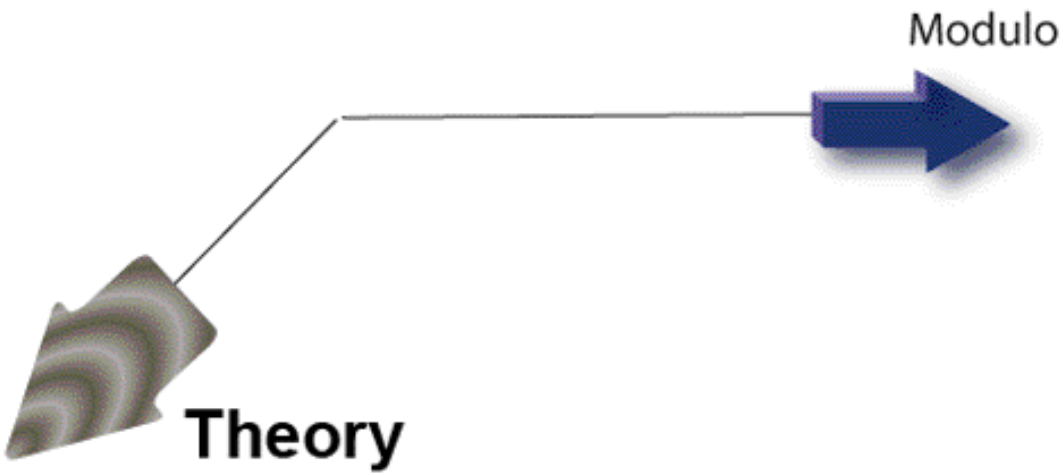
What do we want?

1D state



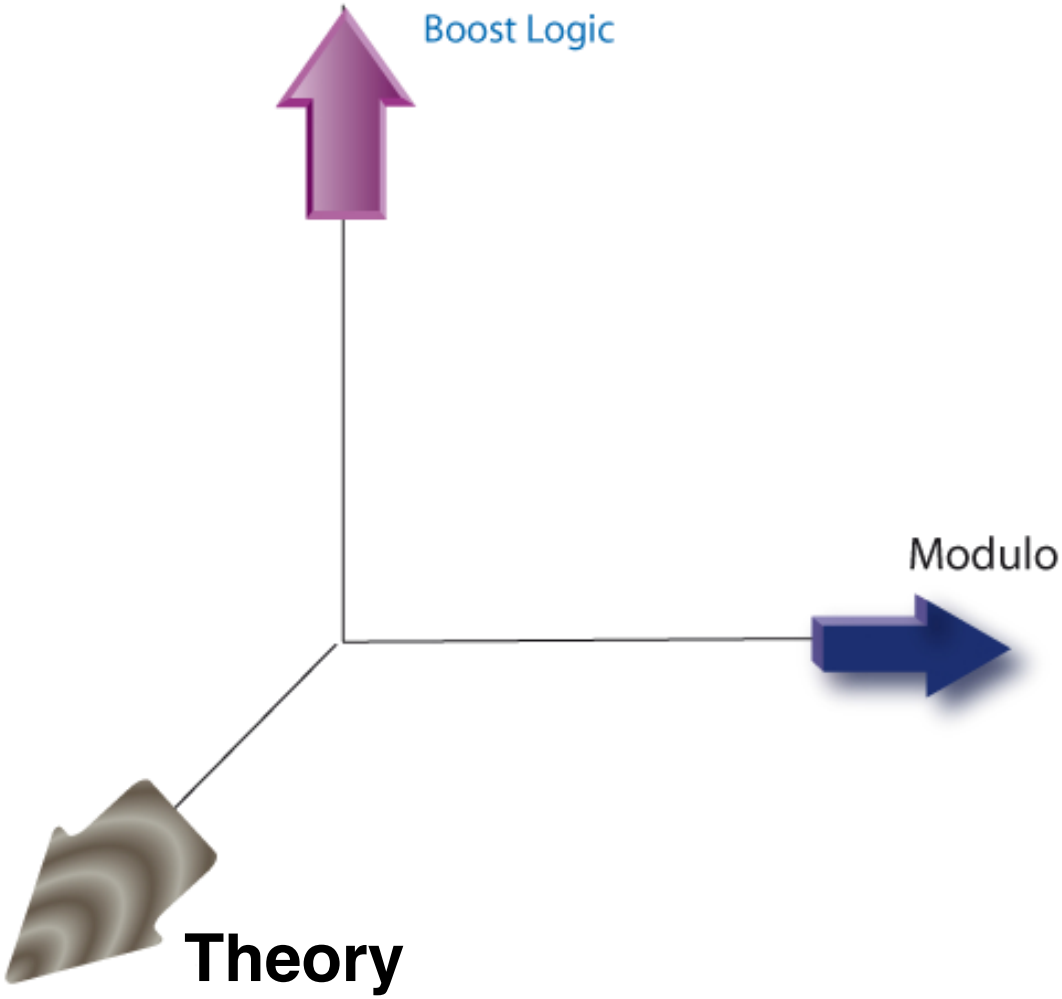
Sequent calculus
Natural deduction
CoC

2D state



Sequent calculus
Natural deduction
CoC
Modulo

3D



Super
Sequent calculus
Natural deduction
CoC
Modulo

A "small" proof

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\ldots, x \in A \vdash A \subseteq A, x \in A}\,\text{Ax}}{\ldots \vdash A \subseteq A, x \in A \Rightarrow x \in A}\,\Rightarrow\text{-R}}{\ldots \vdash A \subseteq A, \forall x.(x \in A \Rightarrow x \in A)}\,\forall\text{-R} \qquad \dfrac{}{\ldots, A \subseteq A \vdash A \subseteq A}\,\text{Ax}}{\ldots, \forall x.(x \in A \Rightarrow x \in A) \Rightarrow A \subseteq A \vdash A \subseteq A}\,\Rightarrow\text{-L}}{(A \subseteq A) \Leftrightarrow \forall x.(x \in A \Rightarrow x \in A) \vdash A \subseteq A}\,\wedge\text{-L}}{\forall Y.(A \subseteq Y) \Leftrightarrow \forall x.(x \in A \Rightarrow x \in Y) \vdash A \subseteq A}\,\forall\text{-L}}{\forall X.\forall Y.(X \subseteq Y) \Leftrightarrow \forall x.(x \in X \Rightarrow x \in Y) \vdash A \subseteq A}\,\forall\text{-L}$$

Loading and instanciating

A "small" proof

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\ldots, x \in A \vdash A \subseteq A, x \in A}\,\text{Ax}}{\ldots \vdash A \subseteq A, x \in A \Rightarrow x \in A}\,\Rightarrow\text{-R}}{\ldots \vdash A \subseteq A, \forall x.(x \in A \Rightarrow x \in A)}\,\forall\text{-R} \qquad \dfrac{}{\ldots, A \subseteq A \vdash A \subseteq A}\,\text{Ax}}{\ldots, \forall x.(x \in A \Rightarrow x \in A) \Rightarrow A \subseteq A \vdash A \subseteq A}\,\Rightarrow\text{-L}}{(A \subseteq A) \Leftrightarrow \forall x.(x \in A \Rightarrow x \in A) \vdash A \subseteq A}\,\wedge\text{-L}}{\forall Y.(A \subseteq Y) \Leftrightarrow \forall x.(x \in A \Rightarrow x \in Y) \vdash A \subseteq A}\,\forall\text{-L}}{\forall X.\forall Y.(X \subseteq Y) \Leftrightarrow \forall x.(x \in X \Rightarrow x \in Y) \vdash A \subseteq A}\,\forall\text{-L}$$

Loading and instanciating

Custom new deduction rules

With the customized deduction rule:

$$\subseteq\text{-R}\ \frac{\Gamma, x \in A \vdash x \in B, \Delta}{\Gamma \vdash A \subseteq B, \Delta}\ x \notin \mathcal{FV}(\Gamma, \Delta)$$

We can build a much shorter (and readable) proof:

$$\subseteq\text{-R}\ \frac{Ax\ \dfrac{}{x \in A \vdash x \in A}}{\vdash A \subseteq A}$$

Custom new deduction rules

With the customized deduction rule:

$$\subseteq\text{-R}\ \frac{\Gamma, x \in A \vdash x \in B, \Delta}{\Gamma \vdash A \subseteq B, \Delta}\ x \notin \mathcal{FV}(\Gamma, \Delta)$$

We can build a much shorter (and readable) proof:

$$\subseteq\text{-R}\ \frac{Ax\ \dfrac{}{x \in A \vdash x \in A}}{\vdash A \subseteq A}$$

Consequences

What are the consequences of adding such a rule ?

- Is it sound?
- Is it complete wrt. the theory?
- Could it be done automatically?
- Do we still have a cut elimination procedure and does it strongly normalise?

Shaping the proof system

$$\mathcal{T}h = \mathcal{T}h_1 \cup \mathcal{T}h_2 \cup \mathcal{T}h_3$$

instead of seeking for a proof of

$$\mathcal{T}h_1 \cup \mathcal{T}h_2 \cup \mathcal{T}h_3 \vdash \varphi$$

We are building a proof of

$$\mathcal{T}h_3 \vdash^{+\mathcal{T}h_2}_{\cong_{\mathcal{T}h_1}} \varphi$$

i.e. we use the theory $\mathcal{T}h_3$ to prove φ using the extended deduction system modulo the congruence $\cong_{\mathcal{T}h_1}$.

We assume that the propositions in $\mathcal{T}h_2$ are all proposition rewrite rules, *i.e.* are of the form $\forall \overline{x}.(P \Leftrightarrow \varphi)$, where P is atomic.

Superdeduction

- Internalize a theory into an **enriched** deduction system
- New deduction rules are systematically derived from the axioms
- Good properties of the deduction system are ensured

Notation:

- Axioms of the form: $\forall \overline{x}.(P \Leftrightarrow \varphi)$ with P atomic.
- We note them $P \longrightarrow \varphi$ and call them proposition rewrite rules.

Superdeduction

- Internalize a theory into an **enriched** deduction system
- New deduction rules are systematically derived from the axioms
- Good properties of the deduction system are ensured

Notation:

- Axioms of the form: $\forall \overline{x}.(P \Leftrightarrow \varphi)$ with P atomic.
- We note them $P \longrightarrow \varphi$ and call them proposition rewrite rules.

Building the rules

$\subseteq_{def}$: $A \subseteq B \rightarrow \forall x.(x \in A \Rightarrow x \in B)$

$$\Rightarrow\text{-R}\ \frac{x \in A \vdash x \in B}{\vdash x \in A \Rightarrow x \in B}$$

$$\forall\text{-R}\ \frac{\vdash x \in A \Rightarrow x \in B}{\vdash \forall x.(x \in A \Rightarrow x \in B)}$$

$$\downarrow$$

$$\subseteq_{def}\text{-R}\ \frac{\Gamma, x \in A \vdash_+ x \in B, \Delta}{\Gamma \vdash_+ A \subseteq B, \Delta}\ x \notin \mathcal{FV}(\Delta, \Gamma)$$

Building the rules

$\subseteq_{def}$: $A \subseteq B \rightarrow \forall x.(x \in A \Rightarrow x \in B)$

$$\forall\text{-R}\ \frac{\Rightarrow\text{-R}\ \dfrac{x \in A \vdash x \in B}{\vdash x \in A \Rightarrow x \in B}}{\vdash \forall x.(x \in A \Rightarrow x \in B)}$$

$$\downarrow$$

$$\subseteq_{def}\text{-R}\ \frac{\Gamma, x \in A \vdash_+ x \in B, \Delta}{\Gamma \vdash_+ A \subseteq B, \Delta}\ x \notin \mathcal{FV}(\Delta, \Gamma)$$

Building the rules

$\subseteq_{def}$: $A \subseteq B \rightarrow \forall x.(x \in A \Rightarrow x \in B)$

$$\forall\text{-R}\ \dfrac{\Rightarrow\text{-R}\ \dfrac{x \in A \vdash x \in B}{\vdash x \in A \Rightarrow x \in B}}{\vdash \forall x.(x \in A \Rightarrow x \in B)}$$

$$\downarrow$$

$$\subseteq_{def}\ \text{-R}\ \dfrac{\Gamma, x \in A \vdash_+ x \in B, \Delta}{\Gamma \vdash_+ A \subseteq B, \Delta}\ x \notin \mathcal{FV}(\Delta, \Gamma)$$

Building the rules

$\subseteq_{def}$: $A \subseteq B \rightarrow \forall x.(x \in A \Rightarrow x \in B)$

$$\forall\text{-L}\ \frac{\Rightarrow\text{-L}\ \dfrac{\vdash t \in A \qquad t \in B \vdash}{t \in A \Rightarrow t \in B \vdash}}{\forall x.(x \in A \Rightarrow x \in B) \vdash}$$

$$\downarrow$$

$$\subseteq_{def}\text{-L}\ \frac{\Gamma \vdash_+ t \in A, \Delta \qquad \Gamma, t \in B \vdash_+ \Delta}{\Gamma, A \subseteq B \vdash_+ \Delta}$$

Building the rules

$\subseteq_{def}$: $A \subseteq B \rightarrow \forall x.(x \in A \Rightarrow x \in B)$

$$\forall\text{-L}\ \dfrac{\Rightarrow\text{-L}\ \dfrac{\vdash t \in A \qquad t \in B \vdash}{t \in A \Rightarrow t \in B \vdash}}{\forall x.(x \in A \Rightarrow x \in B) \vdash}$$

$$\downarrow$$

$$\subseteq_{def}\text{-L}\ \dfrac{\Gamma \vdash_+ t \in A, \Delta \qquad \Gamma, t \in B \vdash_+ \Delta}{\Gamma, A \subseteq B \vdash_+ \Delta}$$

Permutability problem, eigen variables

$$\dfrac{\dfrac{\dfrac{}{P(x_0) \vdash P(x_0)}\,\text{Ax}}{\forall x.P(x) \vdash P(x_0)}\,\forall\text{-L}}{\forall x.P(x) \vdash \forall x.P(x)}\,\forall\text{-R} \qquad\qquad \dfrac{P(t) \vdash \forall x.P(x)}{\forall x.P(x) \vdash \forall x.P(x)}\,\forall\text{-L}$$

- Problems : $\forall$-R then $\forall$-L or $\exists$-R, etc.
- Solution : *focussing*

$$R : P \to \forall x.(\forall y.A(x,y) \Rightarrow B(x))$$

$$\downarrow$$

$$\begin{array}{rcl} R & : & P \to \forall x.(Q(x) \Rightarrow B(x)) \\ R_1 & : & Q(x) \to \forall y.A(x,y) \end{array}$$

Permutability problem, eigen variables

$$\dfrac{\dfrac{\dfrac{}{P(x_0) \vdash P(x_0)}\,\text{Ax}}{\forall x.P(x) \vdash P(x_0)}\,\forall\text{-L}}{\forall x.P(x) \vdash \forall x.P(x)}\,\forall\text{-R} \qquad\qquad \dfrac{P(t) \vdash \forall x.P(x)}{\forall x.P(x) \vdash \forall x.P(x)}\,\forall\text{-L}$$

- Problems : $\forall$-R then $\forall$-L or $\exists$-R, etc.
- Solution : *focussing*

$$R : P \rightarrow \forall x.(\forall y.A(x,y) \Rightarrow B(x))$$

$$\downarrow$$

$$\begin{array}{rcccl} R & : & P & \rightarrow & \forall x.(Q(x) \Rightarrow B(x)) \\ R_1 & : & Q(x) & \rightarrow & \forall y.A(x,y) \end{array}$$

Application : arithmetic

- Natural numbers definition $\longrightarrow$ induction principle
- "cleaning" the rule:

$$\in_{\mathbb{N}}: n \in \mathbb{N} \longrightarrow \forall P.(0 \in P \Rightarrow \forall m.(m \in P \Rightarrow S(m) \in P) \Rightarrow n \in P)$$

$$\downarrow$$

$$\begin{array}{rcrcl} \in_{\mathbb{N}} & : & n \in \mathbb{N} & \longrightarrow & \forall P.(0 \in P \Rightarrow H(P) \Rightarrow n \in P) \\ \mathit{hered} & : & H(P) & \longrightarrow & \forall m.(m \in P \Rightarrow S(m) \in P) \end{array}$$

Application : arithmetic

New deduction rules for *hered*:

$$\textit{hered}\text{-L}\ \frac{\Gamma \vdash_+ m \in P, \Delta \qquad \Gamma, S(m) \in P \vdash_+ \Delta}{\Gamma, H(P) \vdash_+ \Delta}$$

$$\textit{hered}\text{-R}\ \frac{\Gamma, m \in P \vdash_+ S(m) \in P, \Delta}{\Gamma \vdash_+ H(P), \Delta}\ m \notin \mathcal{FV}(\Gamma, \Delta)$$

"if from P(n) we can deduce P(n+1) then P is hereditary"

Application : arithmetic

New deduction rules for *hered*:

$$\textit{hered}\text{-L}\ \frac{\Gamma \vdash_+ m \in P, \Delta \qquad \Gamma, S(m) \in P \vdash_+ \Delta}{\Gamma, H(P) \vdash_+ \Delta}$$

$$\textit{hered}\text{-R}\ \frac{\Gamma, m \in P \vdash_+ S(m) \in P, \Delta}{\Gamma \vdash_+ H(P), \Delta}\ m \notin \mathcal{FV}(\Gamma, \Delta)$$

"if from P(n) we can deduce P(n+1) then P is hereditary"

Application : arithmetic

New deduction rules fo $\in_{\mathbb{N}}$:

$$\in_{\mathbb{N}}\text{-L}\ \frac{\Gamma \vdash_+ 0 \in P, \Delta \qquad \Gamma \vdash_+ H(P), \Delta \qquad \Gamma, n \in P \vdash_+ \Delta}{\Gamma, n \in \mathbb{N} \vdash_+ \Delta}$$

$$\in_{\mathbb{N}}\text{-R}\ \frac{0 \in P, H(P) \vdash_+ n \in P, \Delta}{\Gamma \vdash_+ n \in \mathbb{N}, \Delta}\ P \notin \mathcal{FV}(\Gamma, \Delta)$$

Application : arithmetic

New deduction rules fo $\in_{\mathbb{N}}$:

$$\in_{\mathbb{N}}\text{-L}\ \frac{\Gamma \vdash_+ 0 \in P, \Delta \qquad \Gamma \vdash_+ H(P), \Delta \qquad \Gamma, n \in P \vdash_+ \Delta}{\Gamma, n \in \mathbb{N} \vdash_+ \Delta}$$

$$\in_{\mathbb{N}}\text{-R}\ \frac{0 \in P, H(P) \vdash_+ n \in P, \Delta}{\Gamma \vdash_+ n \in \mathbb{N}, \Delta}\ P \notin \mathcal{FV}(\Gamma, \Delta)$$

Metaproperties

Theorem (Soundness and completeness of superdeduction (Brauner Houtmann Kirchner, LICS'07))

Every proof $\vdash_+ \phi$ in super sequent calculus can be translated into a proof of $\mathcal{T}h \vdash \phi$ in sequent calculus (soundness) and conversely (completeness).

Theorem (cutfreeness preservation (Houtmann'08))

Cut-elimination holds for deduction modulo $\mathcal{T}h_1 \cup \mathcal{T}h_2$ if and only if it holds for superdeduction modulo associated with $(\mathcal{T}h_1, \mathcal{T}h_2)$

Introduction

Deduction Modulo

Super Deduction

Conclusions

Conclusions

Central role of proof design, quality, communication and verification for

- system security, safety, understandability
- formal design quality and elegance, typically in information sciences or in mathematics
- education

We now understand better how logical systems can be customized in a safe and clear way by making

1. computation a first class component
2. the deduction system adapted to the user needs and culture

Towards 3D-proof systems, technically relying on extensions of deduction modulo and on the rewriting calculus

Conclusions

Central role of proof design, quality, communication and verification for

- system security, safety, understandability
- formal design quality and elegance, typically in information sciences or in mathematics
- education

We now understand better how logical systems can be customized in a safe and clear way by making

1. computation a first class component
2. the deduction system adapted to the user needs and culture

Towards 3D-proof systems, technically relying on extensions of deduction modulo and on the rewriting calculus

Conclusions

Central role of proof design, quality, communication and verification for

- system security, safety, understandability
- formal design quality and elegance, typically in information sciences or in mathematics
- education

We now understand better how logical systems can be customized in a safe and clear way by making

1. computation a first class component
2. the deduction system adapted to the user needs and culture

Towards 3D-proof systems, technically relying on extensions of deduction modulo and on the rewriting calculus

Conclusions

- development of the prototype Lemuridae
  http://rho.loria.fr/lemuridae.html
- original treatment of inductive definitions

And this (re)open new research questions like

1. what is a theory?
2. what are good proofs and good-proof theory?
3. what is the right place of constraints?
4. how much deep inference do we want?
5. interaction with *current standard* proof-assistants (Coq, Isabelle...)
6. ...

Conclusions

- development of the prototype Lemuridae
  `http://rho.loria.fr/lemuridae.html`
- original treatment of inductive definitions

And this (re)open new research questions like

1. what is a theory?
2. what are good proofs and good-proof theory?
3. what is the right place of constraints?
4. how much deep inference do we want?
5. interaction with *current standard* proof-assistants (Coq, Isabelle...)
6. ...

Conclusions

- development of the prototype Lemuridae
  `http://rho.loria.fr/lemuridae.html`
- original treatment of inductive definitions

And this (re)open new research questions like

1. what is a theory?
2. what are good proofs and good-proof theory?
3. what is the right place of constraints?
4. how much deep inference do we want?
5. interaction with *current standard* proof-assistants (Coq, Isabelle...)
6. ...

Conclusions

- development of the prototype Lemuridae
  `http://rho.loria.fr/lemuridae.html`
- original treatment of inductive definitions

And this (re)open new research questions like

1. what is a theory?
2. what are good proofs and good-proof theory?
3. what is the right place of constraints?
4. how much deep inference do we want?
5. interaction with *current standard* proof-assistants (Coq, Isabelle...)
6. ...

Conclusions

- development of the prototype Lemuridae
  `http://rho.loria.fr/lemuridae.html`
- original treatment of inductive definitions

And this (re)open new research questions like

1. what is a theory?
2. what are good proofs and good-proof theory?
3. what is the right place of constraints?
4. how much deep inference do we want?
5. interaction with *current standard* proof-assistants (Coq, Isabelle...)
6. ...

Conclusions

- development of the prototype Lemuridae
  `http://rho.loria.fr/lemuridae.html`
- original treatment of inductive definitions

And this (re)open new research questions like

1. what is a theory?
2. what are good proofs and good-proof theory?
3. what is the right place of constraints?
4. how much deep inference do we want?
5. interaction with *current standard* proof-assistants (Coq, Isabelle...)
6. ...

Conclusions

- development of the prototype Lemuridae
  `http://rho.loria.fr/lemuridae.html`
- original treatment of inductive definitions

And this (re)open new research questions like

1. what is a theory?
2. what are good proofs and good-proof theory?
3. what is the right place of constraints?
4. how much deep inference do we want?
5. interaction with *current standard* proof-assistants (Coq, Isabelle...)
6. ...

Conclusions

- development of the prototype Lemuridae
  `http://rho.loria.fr/lemuridae.html`
- original treatment of inductive definitions

And this (re)open new research questions like

1. what is a theory?
2. what are good proofs and good-proof theory?
3. what is the right place of constraints?
4. how much deep inference do we want?
5. interaction with *current standard* proof-assistants (Coq, Isabelle...)
6. ...


Bonichon, R. (2004).
TaMeD: A tableau method for deduction modulo.
In Basin, D. A. and Rusinowitch, M., editors, *IJCAR*, volume 3097 of *LNCS*, pages 445–459. Springer.


Burel, G. (2008).
Superdeduction as a logical framework.
In Pfenning, F., editor, *Proceedings of LICS'08*, Pittsburgh, PA, USA.


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.


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):21–45.


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


Dowek, G. and Werner, B. (2005).
Arithmetic as a theory modulo.
In Giesl, J., editor, *Term Rewriting and Applications, 16th International Conference, RTA 2005, Nara, Japan, April 19-21, 2005, Proceedings*, volume 3467 of *Lecture notes in Computer Science*, pages 423–437.

Prototype

Lemuridae : a proof assistant for superdeduction modulo

- Rewrite rules on terms and propositions
- Proof building in the extendible sequent calculus
- Interactive matching rules presentation
- Basic automatic tactics
- Tiny proofchecker

Lemu's "kernel"

$$\wedge\text{-R}\ \frac{\Gamma \vdash \Delta_1, A, \Delta_2 \qquad \Gamma \vdash \Delta_1, B, \Delta_2}{\Gamma \vdash \Delta_1, \underline{A \wedge B}, \Delta_2}$$

```
rule(
  andRightInfo[],
  ( p1@rule(_,_,sequent(g,(d1*,A,d2*)),_),
    p2@rule(_,_,sequent(g,(d1*,B,d2*)),_)
  ),
  sequent(g,(d1*,a,d2*)),
  a@and(A,B)
)
```

Lemu's "kernel"

$$\wedge\text{-R}\ \frac{\Gamma \vdash \Delta_1, A, \Delta_2 \qquad \Gamma \vdash \Delta_1, B, \Delta_2}{\Gamma \vdash \Delta_1, \underline{A \wedge B}, \Delta_2}$$

```
rule(
  andRightInfo[],
  ( p1@rule(_,_,sequent(g,(d1*,A,d2*)),_),
    p2@rule(_,_,sequent(g,(d1*,B,d2*)),_)
  ),
  sequent(g,(d1*,a,d2*)),
  a@and(A,B)
)
```

Lemu's "kernel"

$$\wedge\text{-R}\ \frac{\Gamma \vdash \Delta_1, A, \Delta_2 \qquad \Gamma \vdash \Delta_1, B, \Delta_2}{\Gamma \vdash \Delta_1, \underline{A \wedge B}, \Delta_2}$$

```
rule(
  andRightInfo[],
  ( p1@rule(_,_,sequent(g,(d1*,A,d2*)),_),
    p2@rule(_,_,sequent(g,(d1*,B,d2*)),_)
  ),
  sequent(g,(d1*,a,d2*)),
  a@and(A,B)
)
```

Lemu's "kernel"

$$\wedge\text{-R}\ \frac{\Gamma \vdash \Delta_1, A, \Delta_2 \qquad \Gamma \vdash \Delta_1, B, \Delta_2}{\Gamma \vdash \Delta_1, \underline{A \wedge B}, \Delta_2}$$

```
rule(
  andRightInfo[],
  ( p1@rule(_,_,sequent(g,(d1*,A,d2*)),_),
    p2@rule(_,_,sequent(g,(d1*,B,d2*)),_)
  ),
  sequent(g,(d1*,a,d2*)),
  a@and(A,B)
)
```

Lemu's "kernel"

$$\wedge\text{-R}\ \frac{\Gamma \vdash \Delta_1, A, \Delta_2 \qquad \Gamma \vdash \Delta_1, B, \Delta_2}{\Gamma \vdash \Delta_1, \underline{A \wedge B}, \Delta_2}$$

```
rule(
  andRightInfo[],
  ( p1@rule(_,_,sequent(g,(d1*,A,d2*)),_),
    p2@rule(_,_,sequent(g,(d1*,B,d2*)),_)
  ),
  sequent(g,(d1*,a,d2*)),
  a@and(A,B)
)
```

Lemu's "kernel"

$$\wedge\text{-R}\ \frac{\Gamma \vdash \Delta_1, A, \Delta_2 \qquad \Gamma \vdash \Delta_1, B, \Delta_2}{\Gamma \vdash \Delta_1, \underline{A \wedge B}, \Delta_2}$$

```
rule(
  andRightInfo[],
  ( p1@rule(_,_,sequent(g,(d1*,A,d2*)),_),
    p2@rule(_,_,sequent(g,(d1*,B,d2*)),_)
  ),
  sequent(g,(d1*,a,d2*)),
  a@and(A,B)
)
```

Lemu's "kernel"

$$\wedge\text{-R}\ \frac{\Gamma \vdash \Delta_1, A, \Delta_2 \qquad \Gamma \vdash \Delta_1, B, \Delta_2}{\Gamma \vdash \Delta_1, \underline{A \wedge B}, \Delta_2}$$

```
rule(
  andRightInfo[],
  ( p1@rule(_,_,sequent(g,(d1*,A,d2*)),_),
    p2@rule(_,_,sequent(g,(d1*,B,d2*)),_)
  ),
  sequent(g,(d1*,a,d2*)),
  a@and(A,B)
)
```

Lemu's "kernel"

$$\wedge\text{-R}\ \frac{\Gamma \vdash \Delta_1, A, \Delta_2 \qquad \Gamma \vdash \Delta_1, B, \Delta_2}{\Gamma \vdash \Delta_1, \underline{A \wedge B}, \Delta_2}$$

```
rule(
  andRightInfo[],
  ( p1@rule(_,_,sequent(g,(d1*,A,d2*)),_),
    p2@rule(_,_,sequent(g,(d1*,B,d2*)),_)
  ),
  sequent(g,(d1*,a,d2*)),
  a@and(A,B)
)
```

Lemu's "kernel"

$$\wedge\text{-R}\ \frac{\Gamma \vdash \Delta_1, A, \Delta_2 \qquad \Gamma \vdash \Delta_1, B, \Delta_2}{\Gamma \vdash \Delta_1, \underline{A \wedge B}, \Delta_2}$$

```
rule(
  andRightInfo[],
  ( p1@rule(_,_,sequent(g,(d1*,A,d2*)),_),
    p2@rule(_,_,sequent(g,(d1*,B,d2*)),_)
  ),
  sequent(g,(d1*,a,d2*)),
  a@and(A,B)
)
```

Lemu's "kernel"

$$\wedge\text{-R}\ \frac{\Gamma \vdash \Delta_1, A, \Delta_2 \qquad \Gamma \vdash \Delta_1, B, \Delta_2}{\Gamma \vdash \Delta_1, \underline{A \wedge B}, \Delta_2}$$

```
rule(
  andRightInfo[],
  ( p1@rule(_,_,sequent(g,(d1*,A,d2*)),_),
    p2@rule(_,_,sequent(g,(d1*,B,d2*)),_)
  ),
  sequent(g,(d1*,a,d2*)),
  a@and(A,B)
)
```

Lemu's "kernel"

$$\wedge\text{-R}\ \frac{\Gamma \vdash \Delta_1, A, \Delta_2 \qquad \Gamma \vdash \Delta_1, B, \Delta_2}{\Gamma \vdash \Delta_1, \underline{A \wedge B}, \Delta_2}$$

```
rule(
  andRightInfo[],
  ( p1@rule(_,_,sequent(g,(d1*,A,d2*)),_),
    p2@rule(_,_,sequent(g,(d1*,B,d2*)),_)
  ),
  sequent(g,(d1*,a,d2*)),
  a@and(A,B)
)
```

Lemu's "kernel"

$$\wedge\text{-R}\ \frac{\Gamma \vdash \Delta_1, A, \Delta_2 \qquad \Gamma \vdash \Delta_1, B, \Delta_2}{\Gamma \vdash \Delta_1, \underline{A \wedge B}, \Delta_2}$$

```
rule(
  andRightInfo[],
  ( p1@rule(_,_,sequent(g,(d1*,A,d2*)),_),
    p2@rule(_,_,sequent(g,(d1*,B,d2*)),_)
  ),
  sequent(g,(d1*,a,d2*)),
  a@and(A,B)
)
```

Lemu's "kernel"

$$\wedge\text{-R}\ \frac{\dfrac{p_1}{\Gamma \vdash \Delta_1, A, \Delta_2} \qquad \dfrac{p_2}{\Gamma \vdash \Delta_1, B, \Delta_2}}{\Gamma \vdash \Delta_1, \underline{A \wedge B}, \Delta_2}$$

```
rule(
  andRightInfo[],
  ( p1@rule(_,_,sequent(g,(d1*,A,d2*)),_),
    p2@rule(_,_,sequent(g,(d1*,B,d2*)),_)
  ),
  sequent(g,(d1*,a,d2*)),
  a@and(A,B)
)
-> { return (proofcheck(‘p1) && proofcheck(‘p2)); }
```