Formalization of Rewriting in PVS

Mauricio Ayala-Rincón

Grupo de Teoria da Computação, Universidade de Brasília (UnB)

Research funded by CNPq, CAPES and FAPDF

Nat@Logic 2015
UFRN Natal, Brazil - Aug 31st & Sep 01st 2015


André L Galdino

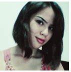

Ana CR Oliveira


Andréia B Avelar


Flávio LC Moura


Thiago MF Ramos


Yuri S Rêgo


Ariane A Almeida

Universidade de Brasília

Talk's Plan

Motivation: formalization - proofs & deduction
- Natural Deduction
- Exercise 1: propositional logic
- The Prototype Verification System PVS

Formal proofs — Proofs in the Prototype Verification System - PVS
- Deduction *à la* Gentzen
- Exercise 2: deduction in the predicate logic

Summary Gentzen versus PVS
- Exercise 3: correctness of algorithms

Conclusions and Future Work

Research in progress

Computational proofs - logic & deduction

Table : Natural deduction for classical propositional logic

introduction rules	elimination rules
$\dfrac{\varphi \quad \psi}{\varphi \wedge \psi}\ (\wedge_i)$	$\dfrac{\varphi \wedge \psi}{\varphi}\ (\wedge_e)$
$\dfrac{\varphi}{\varphi \vee \psi}\ (\vee_i)$	$\dfrac{\varphi \vee \psi \quad \begin{array}{c}[\varphi]^u \\ \vdots \\ \chi\end{array} \quad \begin{array}{c}[\psi]^v \\ \vdots \\ \chi\end{array}}{\chi}\ (\vee_e)\ u, v$
$\dfrac{\begin{array}{c}[\varphi]^u \\ \vdots \\ \psi\end{array}}{\varphi \rightarrow \psi}\ (\rightarrow_i)\ u$	$\dfrac{\varphi \quad \varphi \rightarrow \psi}{\psi}\ (\rightarrow_e)$
$\dfrac{\begin{array}{c}[\varphi]^u \\ \vdots \\ \perp\end{array}}{\neg\varphi}\ (\neg_i)\ u$	$\dfrac{\varphi \quad \neg\varphi}{\perp}\ (\neg_e)$

Computational proofs - logic & deduction

Table : Natural deduction for classical predicate logic

introduction rules	elimination rules
	$$\dfrac{\begin{array}{c}[\neg\varphi]^u \\ \vdots \\ \bot\end{array}}{\varphi}\ (\text{PBC})\ u$$
$$\dfrac{\varphi\{x/x_0\}}{\forall_x \varphi}\ (\forall_i)$$ where x_0 cannot occur free in any open assumption.	$$\dfrac{\forall_x \varphi}{\varphi\{x/t\}}\ (\forall_e)$$
$$\dfrac{\varphi\{x/t\}}{\exists_x \varphi}\ (\exists_i)$$	$$\dfrac{\exists_x \varphi \quad \begin{array}{c}[\varphi\{x/x_0\}]^u \\ \vdots \\ \chi\end{array}}{\chi}\ (\exists_e)\ u$$ where x_0 cannot occur free in any open assumption on the right and in χ.

Mathematical proofs - logic & deduction

Table : Encoding ¬ - Rules of natural deduction for classical logic

introduction rules	elimination rules
$\dfrac{\begin{matrix}[\varphi]^u \\ \vdots \\ \bot\end{matrix}}{\neg\varphi}\ (\neg_i), u$	$\dfrac{\varphi \quad \neg\varphi}{\bot}\ (\neg_e)$

$\dfrac{\begin{matrix}[\varphi]^u \\ \vdots \\ \bot\end{matrix}}{\varphi \to \bot}\ (\to_i), u$	$\dfrac{\varphi \quad \varphi \to \bot}{\bot}\ (\to_e)$

Mathematical proofs - logic & deduction

Interchangeable rules:

$$\frac{\neg\neg\phi}{\phi}\,(\neg\neg_e) \qquad \frac{}{\phi \vee \neg\phi}\,(\mathrm{LEM}) \qquad \frac{\begin{matrix}[\neg\phi]^u \\ \vdots \\ \bot\end{matrix}}{\phi}\,(\mathrm{PBC})\ u$$

Mathematical proofs - logic & deduction

Examples of deductions. Assuming $(\neg\neg_e)$, (LEM) holds:

$$
\begin{prooftree}
\AxiomC{$[\neg(\phi \vee \neg\phi)]^x$}
\AxiomC{$[\neg(\phi \vee \neg\phi)]^x$}
\AxiomC{$[\phi]^u$}
\RightLabel{$(\vee_i)$}
\UnaryInfC{$\phi \vee \neg\phi$}
\RightLabel{$(\neg_e)$}
\BinaryInfC{$\bot$}
\RightLabel{$(\neg_i)\ u$}
\UnaryInfC{$\neg\phi$}
\RightLabel{$(\vee_i)$}
\UnaryInfC{$\phi \vee \neg\phi$}
\RightLabel{$(\neg_e)$}
\BinaryInfC{$\bot$}
\RightLabel{$(\neg_i)\ x$}
\UnaryInfC{$\neg\neg(\phi \vee \neg\phi)$}
\RightLabel{$(\neg\neg_e)$}
\UnaryInfC{$\phi \vee \neg\phi$}
\end{prooftree}
$$

Notation: $\neg\neg\phi \vdash \phi \vee \neg\phi$

Mathematical proofs - logic & deduction

A derivation of Peirce's law, $((\phi \to \psi) \to \phi) \to \phi$:

$$
\dfrac{\dfrac{[\neg\phi]^u \qquad \dfrac{[((\phi \to \psi) \to \phi)]^x \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{[\neg\phi]^u}{\neg\psi \to \neg\phi}\,(\to_i)\ \emptyset \qquad [\neg\psi]^v}{\neg\phi}\,(\to_e) \qquad [\phi]^w}{\perp}\,(\neg_e)}{\psi}\,(\mathrm{PBC})\ v}{\phi \to \psi}\,(\to_i)\ w}{\phi}\,(\to_e)}{\dfrac{\perp}{\phi}\,(\mathrm{PBC})\ u}\,(\neg_e)}{((\phi \to \psi) \to \phi) \to \phi}\,(\to_i)\ x
$$

Notation: $\vdash ((\phi \to \psi) \to \phi) \to \phi$

Mathematical proofs - logic & deduction

More examples. A derivation for $\neg\forall x\,\phi \vdash \exists x\,\neg\phi$

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{[\neg\phi\{x/x_0\}]^u}{\exists x\,\neg\phi}\,(\exists_i) \qquad [\neg\exists x\,\neg\phi]^v}{\bot}\,(\neg_e)}{\phi\{x/x_0\}}\,(\mathrm{PBC})\ u}{\forall x\,\phi}\,(\forall_i) \qquad \neg\forall x\,\phi}{\bot}\,(\neg_e)}{\exists x\,\neg\phi}\,(\mathrm{PBC})\ v$$

A derivation for $\exists x\,\neg\phi \vdash \neg\forall x\,\phi$

$$\dfrac{\exists x\,\neg\phi \qquad \dfrac{\dfrac{[\neg\phi\{x/x_0\}]^u \qquad \dfrac{[\forall x\,\phi]^v}{\phi\{x/x_0\}}\,(\forall_e)}{\bot}\,\neg e}{\neg\forall x\,\phi}\,(\neg_i)\ v}{\neg\forall x\,\phi}\,(\exists_e)\ u$$

Mathematical proofs - logic & deduction

More examples. A derivation for $\neg\exists x\,\phi \vdash \forall x\,\neg\phi$

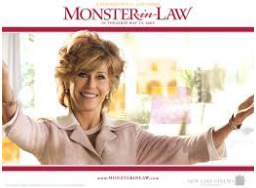
$$\dfrac{\dfrac{\dfrac{\dfrac{[\phi\{x/x_0\}]^u}{\exists x\,\phi}(\exists_i) \quad \neg\exists x\,\phi}{\bot}(\neg_e)}{\neg\phi\{x/x_0\}}(\neg_i)\ u}{\forall x\,\neg\phi}(\forall_i)$$

A derivation for $\forall x\,\neg\phi \vdash \neg\exists x\,\phi$

$$\dfrac{\dfrac{[\exists x\,\phi]^u \quad \dfrac{\dfrac{\forall x\,\neg\phi}{\neg\phi\{x/x_0\}}(\forall_e) \quad [\phi\{x/x_0\}]^v}{\bot}(\neg_e)}{\bot}(\exists_e)\ v}{\neg\exists x\,\phi}(\neg_i)\ u$$

Research Conditions - Exercise 1

Your loud uncle: you're a *smart* guy. Aren't you? What are you doing? There are phantastic well-payed employement opportunities . . . Don't waste your time in research!



Your beloved mother-in-law: (since you are the sole person doing nothing relevant) Hallo my dear, could you pick me up from the airport/mall/... Yes, yes just now?

Research Conditions - Exercise 1

See the file `research_conditions.pvs` in:

`www.mat.unb.br/~ayala/research_conditions.pvs`

or

`www.cic.unb.br/~ayala/research_conditions.pvs`

The Prototype Verification System - PVS

PVS is a verification system, developed by the SRI International Computer Science Laboratory, which consists of

1. a *specification language*:
   - based on *higher-order logic*;
   - a type system based on Church's simple theory of types augmented with *subtypes* and *dependent types*.
2. an *interactive theorem prover*:
   - based on **sequent calculus**; that is, goals in PVS are sequents of the form $\Gamma \vdash \Delta$, where Γ and Δ are finite sequences of formulae, with the usual Gentzen semantics.

The Prototype Verification System - PVS — Libraries

- NASA LaRC PVS library
  http://shemesh.larc.nasa.gov/fm/ftp/larc/PVS-library/pvslib.html
  - Structures, analysis, algebra, Graphs, Digraphs,
  - real arithmetic, floating point arithmetic, groups, interval arithmetic,
  - linear algebra, measure integration, metric spaces,
  - orders, probability, series, sets, topology,
  - term rewriting systems, unification, termination etc etc
    http://trs.cic.unb.br


André L Galdino


Ana CR Oliveira


Andréia B Avelar


Thiago MF Ramos

- Other recommended tutorials
  - NASA/NIA PVS class 2012: http://shemesh.larc.nasa.gov/PVSClass2012
  - Formalisation in PVS of Rewriting Properties - ISR 2014: http://isr2014.inf.utfsm.cl

The Prototype Verification System - PVS — Sequent calculus

- Sequents of the form: $\Gamma \vdash \Delta$.
  - Interpretation: from Γ one obtains Δ.
  - $A_1, A_2, \ldots, A_n \vdash B_1, B_2, \ldots, B_m$ interpreted as $A_1 \wedge A_2 \wedge \ldots \wedge A_n \vdash B_1 \vee B_2 \vee \ldots \vee B_m$.
- Inference rules
  - Premises and conclusions are simultaneously constructed:

$$\frac{\Gamma \vdash \Delta}{\Gamma' \vdash \Delta'}$$

- Goal: $\vdash \Delta$.

Sequent calculus in PVS

- Representation of $A_1, A_2, ..., A_n \vdash B_1, B_2, ..., B_m$:

```
  [-1]  A_1
     :
  [-n]  A_n
|----------
  [1]  B_1
     :
  [n]  B_n
```

- Proof tree: each node is labelled by a sequent.
- A PVS proof command corresponds to the application of an inference rule.
    - In general:

$$\frac{\Gamma \vdash \Delta}{\Gamma_1 \vdash \Delta_1 ... \Gamma_n \vdash \Delta_n} \textbf{ (Rule Name)}$$

Some inference rules in PVS

- Structural:

$$\boxed{\frac{\Gamma_2 \vdash \Delta_2}{\Gamma_1 \vdash \Delta_1} \textbf{ (W)}, \text{ if } \Gamma_1 \subseteq \Gamma_2 \text{ and } \Delta_1 \subseteq \Delta_2}$$

- Propositional:

$$\boxed{\frac{\Gamma, A \vdash A, \Delta}{} \textbf{ (Ax)}} \qquad \boxed{\frac{\Gamma, \mathit{FALSE} \vdash \Delta}{} \textbf{ (FALSE} \vdash \textbf{)}}$$

$$\boxed{\frac{\Gamma \vdash \mathit{TRUE}, \Delta}{} \textbf{ (} \vdash \textbf{TRUE)}}$$

Some inference rules in PVS

- Cut:
  - Corresponds to the case and lemma proof commands.

$$\boxed{\frac{\Gamma \vdash \Delta}{\Gamma, A \vdash \Delta \qquad \Gamma \vdash A, \Delta}\ \textbf{(Cut)}}$$

- Conditional: IF-THEN-ELSE.

$$\boxed{\frac{\Gamma, \mathbf{IF}(A, B, C) \vdash \Delta}{\Gamma, A, B \vdash \Delta \qquad \Gamma, C \vdash A, \Delta}\ \textbf{(IF} \vdash \textbf{)}}$$

$$\boxed{\frac{\Gamma \vdash \mathbf{IF}(A, B, C)\Delta}{\Gamma, A \vdash B, \Delta \qquad \Gamma \vdash A, C, \Delta}\ \textbf{(} \vdash \textbf{IF)}}$$

Gentzen Calculus

sequents:

$$\begin{array}{ccc} \Gamma & \Rightarrow & \Delta \\ \uparrow & & \uparrow \\ \text{antecedent} & & \text{succedent} \end{array}$$

Gentzen Calculus

Table : Rules of deduction *à la* Gentzen for predicate logic

left rules	right rules
Axioms:	
$\Gamma, \varphi \Rightarrow \varphi, \Delta \;\; (Ax)$	$\bot, \Gamma \Rightarrow \Delta \;\; (L_\bot)$
Structural rules:	
$\dfrac{\Gamma \Rightarrow \Delta}{\varphi, \Gamma \Rightarrow \Delta} \; (LWeakening)$	$\dfrac{\Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta, \varphi} \; (RWeakening)$
$\dfrac{\varphi, \varphi, \Gamma \Rightarrow \Delta}{\varphi, \Gamma \Rightarrow \Delta} \; (LContraction)$	$\dfrac{\Gamma \Rightarrow \Delta, \varphi, \varphi}{\Gamma \Rightarrow \Delta, \varphi} \; (RContraction)$

Gentzen Calculus

Table : Rules of deduction *à la* Gentzen for predicate logic

left rules	right rules
Logical rules:	
$\dfrac{\varphi_{i\in\{1,2\}}, \Gamma \Rightarrow \Delta}{\varphi_1 \wedge \varphi_2, \Gamma \Rightarrow \Delta}\ (L_\wedge)$	$\dfrac{\Gamma \Rightarrow \Delta, \varphi \quad \Gamma \Rightarrow \Delta, \psi}{\Gamma \Rightarrow \Delta, \varphi \wedge \psi}\ (R_\wedge)$
$\dfrac{\varphi, \Gamma \Rightarrow \Delta \quad \psi, \Gamma \Rightarrow \Delta}{\varphi \vee \psi, \Gamma \Rightarrow \Delta}\ (L_\vee)$	$\dfrac{\Gamma \Rightarrow \Delta, \varphi_{i\in\{1,2\}}}{\Gamma \Rightarrow \Delta, \varphi_1 \vee \varphi_2}\ (R_\vee)$
$\dfrac{\Gamma \Rightarrow \Delta, \varphi \quad \psi, \Gamma \Rightarrow \Delta}{\varphi \rightarrow \psi, \Gamma \Rightarrow \Delta}\ (L_\rightarrow)$	$\dfrac{\varphi, \Gamma \Rightarrow \Delta, \psi}{\Gamma \Rightarrow \Delta, \varphi \rightarrow \psi}\ (R_\rightarrow)$
$\dfrac{\varphi[x/t], \Gamma \Rightarrow \Delta}{\forall_x \varphi, \Gamma \Rightarrow \Delta}\ (L_\forall)$	$\dfrac{\Gamma \Rightarrow \Delta, \varphi[x/y]}{\Gamma \Rightarrow \Delta, \forall_x \varphi}\ (R_\forall), \quad y \notin \mathtt{fv}(\Gamma, \Delta)$
$\dfrac{\varphi[x/y], \Gamma \Rightarrow \Delta}{\exists_x \varphi, \Gamma \Rightarrow \Delta}\ (L_\exists), \quad y \notin \mathtt{fv}(\Gamma, \Delta)$	$\dfrac{\Gamma \Rightarrow \Delta, \varphi[x/t]}{\Gamma \Rightarrow \Delta, \exists_x \varphi}\ (R_\exists)$

Gentzen Calculus

Derivation of the Peirce's law:

$$
\dfrac{\dfrac{(R_{\to})\;\dfrac{(RW)\;\dfrac{\varphi \Rightarrow \varphi \;\; (Ax)}{\varphi \Rightarrow \varphi, \psi}}{\Rightarrow \varphi, \varphi \to \psi} \qquad \varphi \Rightarrow \varphi \;\; (Ax)}{(\varphi \to \psi) \to \varphi \Rightarrow \varphi}\;(L_{\to})}{\Rightarrow ((\varphi \to \psi) \to \varphi) \to \varphi}\;(R_{\to})
$$

Gentzen Calculus

Cut rule:

$$\boxed{\frac{\Gamma \Rightarrow \Delta, \varphi \quad \varphi, \Gamma' \Rightarrow \Delta'}{\Gamma\Gamma' \Rightarrow \Delta\Delta'}\ (Cut)}$$

Gentzen Calculus - dealing with negation: c-equivalence

$$\varphi, \Gamma \Rightarrow \Delta \text{ one-step c-equivalent } \Gamma \Rightarrow \Delta, \neg\varphi$$

$$\Gamma \Rightarrow \Delta, \varphi \text{ one-step c-equivalent } \neg\varphi, \Gamma \Rightarrow \Delta$$

The c-equivalence is the equivalence closure of this relation.

Lemma (One-step c-equivalence)

(i) $\vdash_G \varphi, \Gamma \Rightarrow \Delta$, *iff* $\vdash_G \Gamma \Rightarrow \Delta, \neg\varphi$;

(ii) $\vdash_G \neg\varphi, \Gamma \Rightarrow \Delta$, *iff* $\vdash_G \Gamma \Rightarrow \Delta, \varphi$.

Gentzen Calculus - dealing with negation

Proof.

(i) **Necessity**:

$$\dfrac{\dfrac{\varphi, \Gamma \Rightarrow \Delta}{\varphi, \Gamma \Rightarrow \Delta, \bot}\ (\mathrm{RW})}{\Gamma \Rightarrow \Delta, \neg\varphi}\ (\mathrm{R}_{\rightarrow})$$

Sufficiency:

$$\dfrac{(\mathrm{LW})\ \dfrac{\Gamma \Rightarrow \Delta, \neg\varphi}{\varphi, \Gamma \Rightarrow \Delta, \neg\varphi} \qquad \dfrac{(Ax)\ \varphi, \Gamma \Rightarrow \Delta, \varphi \qquad \bot, \varphi, \Gamma \Rightarrow \Delta\ (L_{\bot})}{\neg\varphi, \varphi, \Gamma \Rightarrow \Delta}\ (\mathrm{L}_{\rightarrow})}{\varphi, \Gamma \Rightarrow \Delta}\ (\mathrm{Cut})$$

Gentzen Calculus - dealing with negation

(ii) **Necessity**:

$$
\dfrac{
(\mathrm{R}_\to)\ \dfrac{
(\mathrm{L}_\to)\ \dfrac{
(\mathrm{R}_\to)\ \dfrac{(Ax)\ \varphi, \Gamma \Rightarrow \Delta, \varphi, \varphi, \bot}{\Gamma \Rightarrow \Delta, \varphi, \varphi, \neg\varphi}
\qquad \bot, \Gamma \Rightarrow \Delta, \varphi, \varphi\ (L_\bot)
}{\neg\neg\varphi, \Gamma \Rightarrow \Delta, \varphi, \varphi}
}{\Gamma \Rightarrow \Delta, \varphi, \neg\neg\varphi \to \varphi}
\qquad
\dfrac{
\dfrac{\dfrac{\neg\varphi, \Gamma \Rightarrow \Delta}{\neg\varphi, \Gamma \Rightarrow \Delta, \varphi, \bot}\ (\mathrm{RW})}{\Gamma \Rightarrow \Delta, \varphi, \neg\neg\varphi}\ (\mathrm{R}_\to)
\quad \varphi, \Gamma \Rightarrow \Delta, \varphi\ (Ax)
}{\neg\neg\varphi \to \varphi, \Gamma \Rightarrow \Delta, \varphi}
}{\Gamma \Rightarrow \Delta, \varphi}
$$

Sufficiency:

$$
\dfrac{\Gamma \Rightarrow \Delta, \varphi \qquad \bot, \Gamma \Rightarrow \Delta}{\neg\varphi, \Gamma \Rightarrow \Delta}\ (\mathrm{L}_\to)
$$

$\square$

Gentzen versus Natural deduction

Theorem (Natural vs deduction à la Gentzen for the classical logic)

$\vdash_G \Gamma \Rightarrow \varphi$ *if, and only if* $\Gamma \vdash_N \varphi$

Analysis of GCD properties - Exercise 2

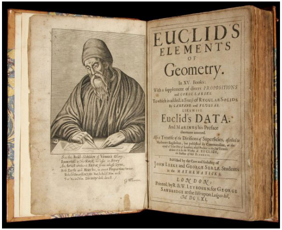
Dealing with variables:

Definition (GCD)

For all $m, n \in \mathbb{Z} \setminus (0,0)$ the greatest common divisor of m and n, denoted as $gcd(m,n)$ is the smallest number that divides both m and n.

Theorem (Improved Euclid Theorem ∼300 BC- Gabriel Lamé 1844)

$\forall (m,n) : \mathbb{Z} \setminus (0,0) : GCD(m,n) = GCD(rem(n)(m), n)$



Analysis of GCD properties - Exercise 2

See the file pred_gcd.pvs in:

www.mat.unb.br/∼ayala/pred_gcd.pvs

or

www.cic.unb.br/∼ayala/pred_gcd.pvs

Summary - Gentzen Deductive Rules vs Proof Commands

Table : Structural Left Rules vs Proof Commands

Structural left rules	PVS commands
$\dfrac{\Gamma \Rightarrow \Delta}{\varphi, \Gamma \Rightarrow \Delta}\ (LWeakening)$	$\dfrac{\varphi, \Gamma \vdash \Delta}{\Gamma \vdash \Delta}\ (hide)$
$\dfrac{\varphi, \varphi, \Gamma \Rightarrow \Delta}{\varphi, \Gamma \Rightarrow \Delta}\ (LContraction)$	$\dfrac{\varphi, \Gamma \vdash \Delta}{\varphi, \varphi, \Gamma \vdash \Delta}\ (copy)$

Summary - Gentzen Deductive Rules vs Proof Commads

Table : Structural Right Rules vs Proof Commands

Structural right rules	PVS commands
$\dfrac{\Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta, \varphi}$ $(RWeakening)$	$\dfrac{\Gamma \vdash \Delta, \varphi}{\Gamma \vdash \Delta}$ $(hide)$
$\dfrac{\Gamma \Rightarrow \Delta, \varphi, \varphi}{\Gamma \Rightarrow \Delta, \varphi}$ $(RContraction)$	$\dfrac{\Gamma \vdash \Delta, \varphi}{\Gamma \vdash \Delta, \varphi, \varphi}$ $(copy)$

Summary - Gentzen Deductive Rules vs Proof Commads

Table : Logical Left Rules vs Proof Commands

left rules	PVS commands
$\dfrac{\varphi_1, \varphi_2, \Gamma \Rightarrow \Delta}{\varphi_1 \wedge \varphi_2, \Gamma \Rightarrow \Delta} \ (L_\wedge)$	$\dfrac{\varphi_1 \wedge \varphi_2, \Gamma \vdash \Delta}{\varphi_{i\in\{1,2\}}, \Gamma \vdash \Delta} \ (\mathit{flatten})$
$\dfrac{\varphi, \Gamma \Rightarrow \Delta \quad \psi, \Gamma \Rightarrow \Delta}{\varphi \vee \psi, \Gamma \Rightarrow \Delta} \ (L_\vee)$	$\dfrac{\varphi \vee \psi, \Gamma \vdash \Delta}{\varphi, \Gamma \vdash \Delta \quad \psi, \Gamma \vdash \Delta} \ (\mathit{split})$
$\dfrac{\Gamma \Rightarrow \Delta, \varphi \quad \psi, \Gamma \Rightarrow \Delta}{\varphi \rightarrow \psi, \Gamma \Rightarrow \Delta} \ (L_\rightarrow)$	$\dfrac{\varphi \rightarrow \psi, \Gamma \vdash \Delta}{\Gamma \vdash \Delta, \varphi \quad \psi, \Gamma \vdash \Delta} \ (\mathit{split})$
$\dfrac{\varphi[x/t], \Gamma \Rightarrow \Delta}{\forall_x \varphi, \Gamma \Rightarrow \Delta} \ (L_\forall)$	$\dfrac{\forall_x \varphi, \Gamma \vdash \Delta}{\varphi[x/t], \Gamma \vdash \Delta} \ (\mathit{inst})$
$\dfrac{\varphi[x/y], \Gamma \Rightarrow \Delta}{\exists_x \varphi, \Gamma \Rightarrow \Delta} \ (L_\exists), \quad y \notin \mathtt{fv}(\Gamma, \Delta)$	$\dfrac{\exists_x \varphi, \Gamma \vdash \Delta}{\varphi[x/y], \Gamma \vdash \Delta} \ (\mathit{skolem}), \quad y \notin \mathtt{fv}(\Gamma, \Delta)$

Summary - Gentzen Deductive Rules vs Proof Commads

Table : Logical Right Rules vs Proof Commands

right rules	PVS commands
$\dfrac{\Gamma \Rightarrow \Delta, \varphi \quad \Gamma \Rightarrow \Delta, \psi}{\Gamma \Rightarrow \Delta, \varphi \wedge \psi}\ (R_\wedge)$	$\dfrac{\Gamma \vdash \Delta, \varphi \wedge \psi}{\Gamma \vdash \Delta, \varphi \quad \Gamma \vdash \Delta, \psi}\ (\mathit{split})$
$\dfrac{\Gamma \Rightarrow \Delta, \varphi_{i \in \{1,2\}}}{\Gamma \Rightarrow \Delta, \varphi_1 \vee \varphi_2}\ (R_\vee)$	$\dfrac{\Gamma \vdash \Delta, \varphi_1 \vee \varphi_2}{\Gamma \vdash \Delta, \varphi_1, \varphi_2}\ (\mathit{flatten})$
$\dfrac{\varphi, \Gamma \Rightarrow \Delta, \psi}{\Gamma \Rightarrow \Delta, \varphi \rightarrow \psi}\ (R_\rightarrow)$	$\dfrac{\Gamma \vdash \Delta, \varphi \rightarrow \psi}{\varphi, \Gamma \vdash \Delta, \psi}\ (\mathit{flatten})$
$\dfrac{\Gamma \Rightarrow \Delta, \varphi[x/y]}{\Gamma \Rightarrow \Delta, \forall_x \varphi}\ (R_\forall), \quad y \notin \mathtt{fv}(\Gamma, \Delta)$	$\dfrac{\Gamma \vdash \Delta, \forall_x \varphi}{\Gamma \vdash \Delta, \varphi[x/y]}\ (\mathit{skolem}), \quad y \notin \mathtt{fv}(\Gamma, \Delta)$
$\dfrac{\Gamma \Rightarrow \Delta, \varphi[x/t]}{\Gamma \Rightarrow \Delta, \exists_x \varphi}\ (R_\exists)$	$\dfrac{\Gamma \vdash \Delta, \exists_x \varphi}{\Gamma \vdash \Delta, \varphi[x/t]}\ (\mathit{inst})$

Gentzen Calculus inference rules vs PVS proof rules

	(hide)	(copy)	(flatten)	(split)	(skolem)	(inst)	(lemma) (case)
(Ax)			×	×			
($L_\perp$)			×	×			
(LW)	×						
(LC)		×					
($L_\wedge$)			×				
($L_\vee$)				×			
($L_\rightarrow$)				×			
($L_\forall$)						×	
($L_\exists$)					×		
(RW)	×						
(RC)		×					
($R_\wedge$)				×			
($R_\vee$)			×				
($R_\rightarrow$)			×				
($R_\forall$)					×		
($R_\exists$)						×	
(Cut)							×

GCD algorithm correctness - Exercise 3

See the files `gcd.pvs` in:

`www.mat.unb.br/~ayala/gcd.pvs / prf`

or

`www.cic.unb.br/~ayala/gcd.pvs / prf`

Verification of algorithmic properties - Exercise 3

```
gcd(n, m) :  RECURSIVE nat =
      IF abs(n) = abs(m) THEN abs(n)
      ELSE IF (n = 0 OR m = 0) THEN abs(n+m)
           ELSE IF (abs(n) > abs(m)) THEN
                     gcd(abs(n)-abs(m), abs(m))
                ELSE gcd(abs(m)-abs(n),abs(n))
                ENDIF
           ENDIF
      ENDIF
MEASURE abs(n)+abs(m)
```

It works?

Does this specification compute correctly the ``GCD'' of the definition?

Verification of algorithmic properties - Exercise 3

```
gcd_is_correct :  COROLLARY
    (m /= 0 OR n /=0) =>
      divides(gcd(m,n),m) AND
      divides(gcd(m,n),n) AND
      FORALL (k) :  (divides(k,m) AND divides(k,n) =>
           k <= gcd(m,n))
```

Conclusions

- Nowadays, computational logic is intensively applied in formal methods.
- In computer sciences, a reasonable training on "computational" logic should focus on derivation/proof techniques.
- Understanding proof theory is essential to mastering proof assistants:
  - to provide mathematical proofs of robustness of computational systems and
  - well-accepted quality certificates.
- The deductive framework - proof assistant - is important but irrelevant.

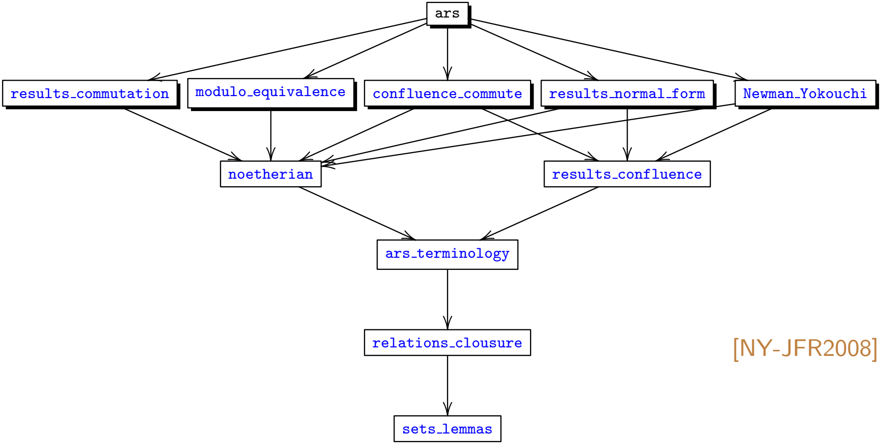
Future Work

- A myriad of elaborated mathematical theorems are to be formalized.
- Termination analysis including more sophisticated termination semantics such as the one based on the *size change termination* principle.
- New mechanisms to apply the theory to verify rewriting based specifications.

Developments of the GTC at UnB - References

Figure : The *Grupo de Teoria da Computação* at *Universidade de Brasília*

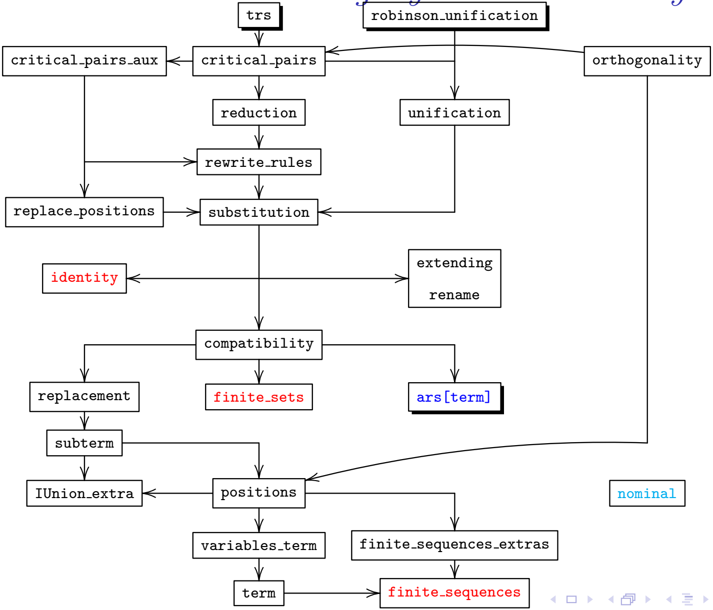
The Abstract Redution Systems Hierarchy - ars



[NY-JFR2008]

Figure : Hierarchy ars

The Term Rewriting Systems Hierarchy - trs



[CP-JAR2010]

[U-IGPl2014]

[CO-EP2013]

[NU-LSFA2015]

Developments of the GTC at UnB - References

Andréia Borges Avelar, André Luiz Galdino, Flávio Leonardo Cavalcanti de Moura, and Mauricio Ayala-Rincón.
First-order unification in the PVS proof assistant.
Logic Journal of the IGPL, 22(5):758–789, 2014.

A. L. Galdino and M. Ayala-Rincón.
A Formalization of Newman's and Yokouchi Lemmas in a Higher-Order Language.
Journal of Formalized Reasoning, 1(1):39–50, 2008.

A. L. Galdino and M. Ayala-Rincón.
A Formalization of the Knuth-Bendix(-Huet) Critical Pair Theorem.
J. of Automated Reasoning, 45(3):301–325, 2010.

A. C. Rocha Oliveira and M. Ayala-Rincón.
Formalizing the confluence of orthogonal rewriting systems.
CoRR, abs/1303.7335, 2013.

A. C. Rocha Oliveira, M. Fernández, and M. Ayala-Rincón.
Completeness in PVS of a Nominal Unification Algorithm.
In *Pre-proc. LSFA*, 2015.