Formalization of Rewriting in PVS

Mauricio Ayala-Rincón

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

Brasília D.F., Brazil

Research funded by

Brazilian Research Agencies: CNPq, CAPES and FAPDF

International School on Rewriting ISR 2014
UTFSM Valparaíso, Chile - Aug 25^{th}-29^{th} 2014


André Luiz Galdino


Ana C. Rocha Oliveira


Andréia Borges Avelar

Talk's Plan

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

Formal proofs — Proofs in the Prototype Verification System - PVS
- Deduction à la Gentzen
- Exercise: predicate rewriting

Formalizations
- Exercise: more on rewriting
- Case study: rewriting
- Exercise: following proofs in the PVS theory `trs`
- Exercise: following proofs in the PVS theory `trs`

Conclusions and Future Work

Computational proofs - logic & deduction

Table : Natural deduction for classical propositional logic

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

Computational proofs - logic & deduction

Table : Natural deduction for classical predicate logic

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

Mathematical proofs - logic & deduction

Table : Encoding $\neg$ - Rules of natural deduction for classical logic

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

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

Mathematical proofs - logic & deduction

Interchangeable rules:

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

Mathematical proofs - logic & deduction

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

$$
\dfrac{\dfrac{[\neg(\phi\vee\neg\phi)]^x \qquad \dfrac{\dfrac{\dfrac{[\neg(\phi\vee\neg\phi)]^x \qquad \dfrac{[\phi]^u}{\phi\vee\neg\phi}\,(\vee_i)}{\bot}\,(\neg_e)}{\neg\phi}\,(\neg_i)\ u}{\phi\vee\neg\phi}\,(\vee_i)}{\bot}\,(\neg_e)}{\dfrac{\neg\neg(\phi\vee\neg\phi)}{\phi\vee\neg\phi}\,(\neg\neg_e)}\,(\neg_i)\ x
$$

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

Mathematical proofs - logic & deduction

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

$$
\begin{prooftree}
\AxiomC{$[\neg\phi]^u$}
\AxiomC{$[((\phi \to \psi) \to \phi)]^x$}
\AxiomC{$[\neg\phi]^u$}
\RightLabel{$(\to_i)\ \emptyset$}
\UnaryInfC{$\neg\psi \to \neg\phi$}
\AxiomC{$[\neg\psi]^v$}
\RightLabel{$(\to_e)$}
\BinaryInfC{$\neg\phi$}
\AxiomC{$[\phi]^w$}
\RightLabel{$(\neg_e)$}
\BinaryInfC{$\perp$}
\RightLabel{(PBC) v}
\UnaryInfC{ψ}
\RightLabel{$(\to_i)\ w$}
\UnaryInfC{$\phi \to \psi$}
\RightLabel{$(\to_e)$}
\BinaryInfC{ϕ}
\RightLabel{$(\neg_e)$}
\BinaryInfC{$\perp$}
\RightLabel{(PBC) u}
\UnaryInfC{ϕ}
\RightLabel{$(\to_i)\ x$}
\UnaryInfC{$((\phi \to \psi) \to \phi) \to \phi$}
\end{prooftree}
$$

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 $\neg\exists x\,\phi \vdash \forall x\,\neg\phi$

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

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

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

A first naive exercise: propositional rewriting

See the file propARS.pvs in:

www.mat.unb.br/∼ayala/propARS.pvs

or

www.cic.unb.br/∼ayala/propARS.pvs

Propositional analysis of rewriting properties

Theorem (Knuth-Bendix-Huet CP criterion)

CP joinability implies LC

Lemma (Newman)

SN implies LC if and only if CR

Thus,

Lemma (Knuth-Bendix CP criterion)

CP joinability and SN imply CR.

Where CP, LC, SN and CR abbreviate Critical Pair, Locally Confluent, Strongly Normalizing and Church-Rosser, as usual.
See exercise propARS.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 includes
  - 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, etc. etc.

The Prototype Verification System - PVS — Sequent calculus

- Sequents of the form: $\Gamma \vdash \Delta$.
  - Interpretation: from Γ one obtains Δ.
  - $A_1, A_2, ..., A_n \vdash B_1, B_2, ..., B_m$ interpreted as $A_1 \wedge A_2 \wedge ... \wedge A_n \vdash B_1 \vee B_2 \vee ... \vee B_m$.
- 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`.

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

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

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:

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

Gentzen Calculus

Cut rule:

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

Gentzen Calculus

Example of application of (Cut):

$$\frac{\Rightarrow \neg\neg(\psi \vee \neg\psi) \qquad \neg\neg(\psi \vee \neg\psi) \Rightarrow \psi \vee \neg\psi}{\Rightarrow \psi \vee \neg\psi}\ (Cut)$$

Gentzen Calculus

A derivation for the sequent $\Rightarrow \neg\neg(\psi \vee \neg\psi)$:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\psi \Rightarrow \psi, \bot \ (Ax)}{\Rightarrow \psi, \neg\psi}\,(R_\rightarrow)}{\Rightarrow \psi \vee \neg\psi, \neg\psi}\,(R_\vee)}{\Rightarrow \psi \vee \neg\psi, \psi \vee \neg\psi}\,(R_\vee)}{\Rightarrow \psi \vee \neg\psi}\,(RC) \qquad \psi \vee \neg\psi \Rightarrow \neg\neg(\psi \vee \neg\psi)}{\Rightarrow \neg\neg(\psi \vee \neg\psi)}\,(Cut)$$

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**:

$$
\begin{prooftree}
\AxiomC{$(Ax)\ \varphi, \Gamma \Rightarrow \Delta, \varphi, \varphi, \bot$}
\LeftLabel{$(\mathrm{R}_{\rightarrow})$}
\UnaryInfC{$\Gamma \Rightarrow \Delta, \varphi, \varphi, \neg\varphi$}
\AxiomC{$\bot, \Gamma \Rightarrow \Delta, \varphi, \varphi\ (L_{\bot})$}
\LeftLabel{$(\mathrm{L}_{\rightarrow})$}
\BinaryInfC{$\neg\neg\varphi, \Gamma \Rightarrow \Delta, \varphi, \varphi$}
\LeftLabel{$(\mathrm{R}_{\rightarrow})$}
\UnaryInfC{$\Gamma \Rightarrow \Delta, \varphi, \neg\neg\varphi \rightarrow \varphi$}
\AxiomC{$\neg\varphi, \Gamma \Rightarrow \Delta$}
\RightLabel{(RW)}
\UnaryInfC{$\neg\varphi, \Gamma \Rightarrow \Delta, \varphi, \bot$}
\RightLabel{$(\mathrm{R}_{\rightarrow})$}
\UnaryInfC{$\Gamma \Rightarrow \Delta, \varphi, \neg\neg\varphi$}
\AxiomC{$\varphi, \Gamma \Rightarrow \Delta, \varphi\ (Ax)$}
\BinaryInfC{$\neg\neg\varphi \rightarrow \varphi, \Gamma \Rightarrow \Delta, \varphi$}
\BinaryInfC{$\Gamma \Rightarrow \Delta, \varphi$}
\end{prooftree}
$$

Sufficiency:

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

$\square$

Gentzen versus Natural deduction

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

$$\vdash_G \Gamma \Rightarrow \varphi \text{ if, and only if } \Gamma \vdash_N \varphi$$

Propositional GC vs PVS rules - Regarding Ex.1

	(hide)	(copy)	(flatten)	(split)	(Skolem)	(Inst)	(lemma) (case)
(LW)	$\times$						
(LC)		$\times$					
($L_\wedge$)			$\times$				
($L_\vee$)				$\times$			
($L_\rightarrow$)				$\times$			
(RW)	$\times$						
(RC)		$\times$					
($R_\wedge$)				$\times$			
($R_\vee$)			$\times$				
($R_\rightarrow$)			$\times$				
(Cut)							$\times$

A second exercise: predicate rewriting

See the file predTRS.pvs in:

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

or

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

Analysis of rewriting properties - Exercise 2

Dealing with variables:

Theorem (Hindley-Rossen Theorem)

Commutation of $R1$ and $R2$ and both TRSs are CR imply CR of $R1 \cup R2$.

Thus,

Corollary (H-R application to prove CR)

For all TRS R, the existence of a commutative bipartition into CR TRSs (say $R1$ and $R2$, such that $CR(R1)$ and $CR(R2)$,) implies $CR(R)$.

See predTRS.pvs

A third exercise: HO rewriting

See the files predCommutation.pvs and predCommutation.prf in:

www.mat.unb.br/∼ayala/predCommutation.pvs / prf

or

www.cic.unb.br/∼ayala/predCommutation.pvs / prf

Analysis of rewriting properties - Exercise 3

Dealing with HO variables, quantifying binary relations, and induction:

Theorem (CR vr C)

Confluence and CR are equivalent properties

See predCommutation.pvs

Case Study: rewriting - ARSs • Binary relations

```
relations_closure[T : TYPE] : THEORY
BEGIN
    IMPORTING    orders@closure_ops[T],    sets_lemmas[T]
                          ⋮
    S, R: VAR pred[[T, T]]
    n: VAR nat
    p: VAR posnat
                          ⋮
    RC(R): reflexive = union(R, =)
    SC(R): symmetric = union(R, converse(R))
    TC(R): transitive = IUnion(LAMBDA p: iterate(R, p))
    RTC(R): reflexive_transitive = IUnion(LAMBDA n: iterate(R, n))
    EC(R): equivalence = RTC(SC(R))
                          ⋮
END relations_closure
```

Case Study: rewriting - ARSs • Binary relations

change_to_TC : LEMMA transitive_closure(R) = TC(R)

R_subset_TC :LEMMA subset?(R, TC(R))

TC_converse: LEMMA TC(converse(R)) = converse(TC(R))

TC_idempotent : LEMMA TC(TC(R)) = TC(R)

TC_characterization : LEMMA transitive?(S) $\Leftrightarrow$ (S = TC(S))

Case Study: rewriting - ARSs • Hierarchy

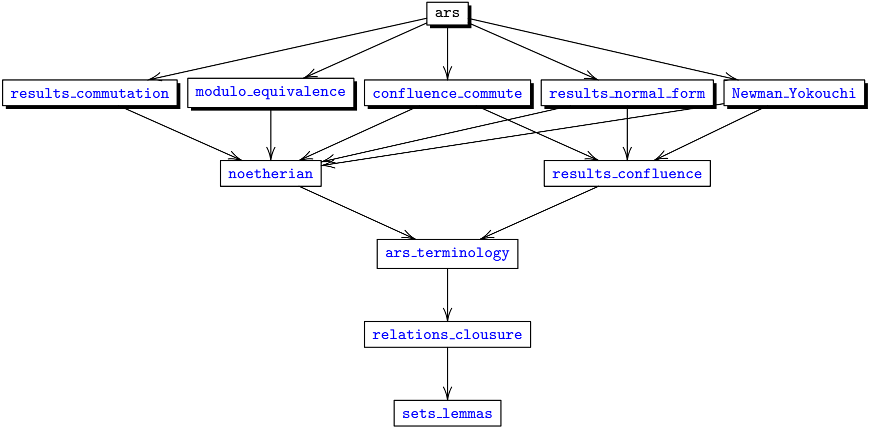


Figure : Hierarchy ars

Case Study: rewriting - ARSs • Newman Lemma

noetherian?(R): bool = well_founded?(converse(R))

joinable?(R)(x,y): bool = EXISTS z: RTC(R)(x,z) & RTC(R)(y, z)

locally_confluent?(R): bool =
FORALL x, y, z: R(x,y) & R(x,z) ⇒ joinable?(R)(y,z)

confluent?(R): bool =
FORALL x, y, z: RTC(R)(x,y) & RTC(R)(x,z) ⇒ joinable?(R)(y,z)

Newman_lemma: THEOREM
noetherian?(R) ⇒ (confluent?(R) ⇔ locally_confluent?(R))

Case Study: rewriting - ARSs • Newman Lemma

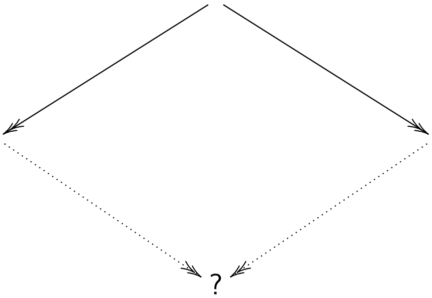


Figure : Proof's Sketch of Newman Lemma

Case Study: rewriting - ARSs • Newman Lemma

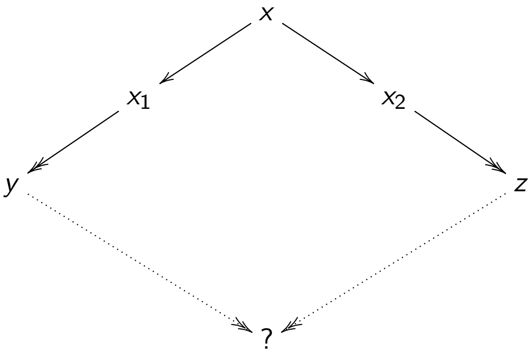


Figure : Proof's Sketch of Newman Lemma

Case Study: rewriting - ARSs • Newman Lemma

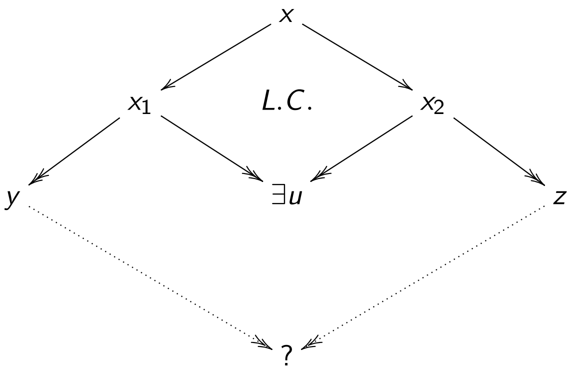


Figure : Proof's Sketch of Newman Lemma

Case Study: rewriting - ARSs • Newman Lemma

$$
\begin{tikzcd}
 & & x \arrow[dl] \arrow[dr] & & \\
 & x_1 \arrow[dl] \arrow[dr] & & x_2 \arrow[dl] \arrow[dr] & \\
y \arrow[dr] & I.N. & u \arrow[dl] & & z \arrow[ddll, dotted] \\
 & \exists v \arrow[dr, dotted] & & & \\
 & & ? & &
\end{tikzcd}
$$

Figure : Proof's Sketch of Newman Lemma

Case Study: rewriting - ARSs • Newman Lemma

$$
\begin{tikzcd}
 & & x \arrow[dl] \arrow[dr] & & \\
 & x_1 \arrow[dl] \arrow[dr] & & x_2 \arrow[dl] \arrow[dr] & \\
y \arrow[dr] & & u \arrow[dl] & & z \arrow[ddll] \\
 & v \arrow[dr] & I.N. & & \\
 & & \exists w & &
\end{tikzcd}
$$

Figure : Proof's Sketch of Newman Lemma

Case Study: rewriting - ARSs • Newman Lemma

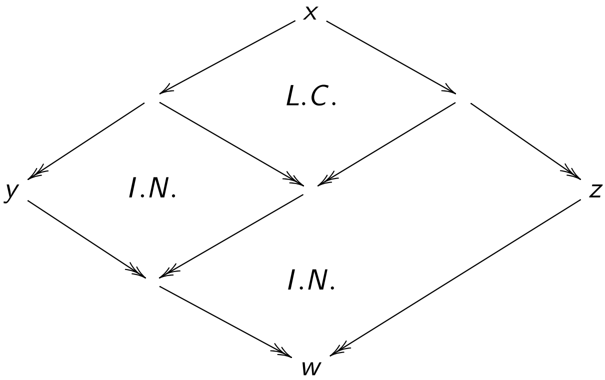


Figure : Proof's Sketch of Newman Lemma

Case Study: rewriting - ARSs • Newman Lemma

A few used lemmas:

```
R_subset_RC : LEMMA subset?(R, RC(R))
iterate_RTC: LEMMA FORALL n : subset?(iterate(R, n), RTC(R))
R_is_Noet_iff_TC_is: LEMMA noetherian?(R) ⇔ noetherian?(TC(R))
R_subset_TC :LEMMA subset?(R, TC(R))
```

```
noetherian_induction: LEMMA
   (FORALL (R: noetherian, P):
    (FORALL x:
     (FORALL y: TC(R)(x, y) ⇒ P(y))
      ⇒ P(x))
    ⇒
     (FORALL x: P(x)))
```

A final exercise: follow Newman's lemma proof in the PVS theory `ars`

1. Change context in PVS through the command `change-context`.
2. Accordingly to your instalation of the NASA PVS libraries you should change context to `../nasalib/TRS`.
3. Open the file `../nasalib/TRS/newman_yokouchi.pvs` .
4. Use the command `x-step-proof`.
5. By the key combination `tab` and `1` the proof can be followed step by step.

A final exercise: follow other proofs in the PVS theory `trs`

1. Critical Pair theorem.
   Load the file `../nasalib/TRS/newman_yokouchi.pvs` .
2. Confluence of orthogonal TRSs.
   Load the file `../nasalib/TRS/orthogonality.pvs`.
3. Etc.

Final exercice: conclude the proof of the last exercise in the third list of exercises by applying Noetherian Induction as in the formalization of Newman Lemma.

Summary - Gentzen Deductive Rules vs Proof Commads

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
$\dfrac{\varphi_1, \varphi_2, \Gamma \Rightarrow \Delta}{\varphi_1 \wedge \varphi_2, \Gamma \Rightarrow \Delta}\ (L_\wedge)$	$\dfrac{\varphi_1 \wedge \varphi_2, \Gamma \vdash \Delta}{\varphi_{i\in\{1,2\}}, \Gamma \vdash \Delta}\ (\mathit{Flatten})$
$\dfrac{\varphi, \Gamma \Rightarrow \Delta \quad \psi, \Gamma \Rightarrow \Delta}{\varphi \vee \psi, \Gamma \Rightarrow \Delta}\ (L_\vee)$	$\dfrac{\varphi \vee \psi, \Gamma \vdash \Delta}{\varphi, \Gamma \vdash \Delta \quad \psi, \Gamma \vdash \Delta}\ (\mathit{Split})$
$\dfrac{\Gamma \Rightarrow \Delta, \varphi \quad \psi, \Gamma \Rightarrow \Delta}{\varphi \to \psi, \Gamma \Rightarrow \Delta}\ (L_\to)$	$\dfrac{\varphi \to \psi, \Gamma \vdash \Delta}{\Gamma \vdash \Delta, \varphi \quad \psi, \Gamma \vdash \Delta}\ (\mathit{Split})$
$\dfrac{\varphi[x/t], \Gamma \Rightarrow \Delta}{\forall_x \varphi, \Gamma \Rightarrow \Delta}\ (L_\forall)$	$\dfrac{\forall_x \varphi, \Gamma \vdash \Delta}{\varphi[x/t], \Gamma \vdash \Delta}\ (\mathit{Instantiate})$
$\dfrac{\varphi[x/y], \Gamma \Rightarrow \Delta}{\exists_x \varphi, \Gamma \Rightarrow \Delta}\ (L_\exists), \quad y \notin \mathtt{fv}(\Gamma, \Delta)$	$\dfrac{\exists_x \varphi, \Gamma \vdash \Delta}{\varphi[x/y], \Gamma \vdash \Delta}\ (\mathit{Skolem}), \quad y \notin \mathtt{fv}(\Gamma, \Delta)$

Summary - Gentzen Deductive Rules vs Proof Commads

Table : Logical Right Rules vs Proof Commands

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

Summary - Completing the GC vs PVS rules

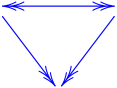
	(hide)	(copy)	(flatten)	(split)	(Skolem)	(Inst)	(lemma) (case)
(LW)	×						
(LC)		×					
($L_\wedge$)			×				
($L_\vee$)				×			
($L_\rightarrow$)				×			
($L_\forall$)						×	
($L_\exists$)					×		
(RW)	×						
(RC)		×					
($R_\wedge$)				×			
($R_\vee$)			×				
($R_\rightarrow$)			×				
($R_\forall$)					×		
($R_\exists$)						×	
(Cut)							×

Exercises — ARSs

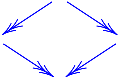
$$\begin{tikzcd}[column sep=small] & \bullet \arrow[dl] \arrow[dr] & \\ \bullet \arrow[dr] & & \bullet \arrow[dl, "\leq 1"] \\ & \bullet & \end{tikzcd} \quad \textit{IMPLIES} \quad \begin{tikzcd}[column sep=small] & \bullet \arrow[dl] \arrow[dr] & \\ \bullet \arrow[dr] & & \bullet \arrow[dl] \\ & \bullet & \end{tikzcd}$$

```
Strong_Confl_implies_Confl: COROLLARY
        strong_confluent?(R) ⇒ confluent?(R)
```

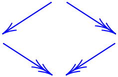
Exercises — ARSs



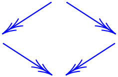
IFF



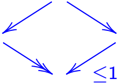
CR_iff_Confluent: THEOREM church_rosser?(R) $\Leftrightarrow$ confluent?(R)



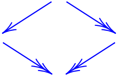
IMPLIES



Semi_implies_CR: THEOREM semi_confluent?(R) $\Rightarrow$ church_rosser?(R)

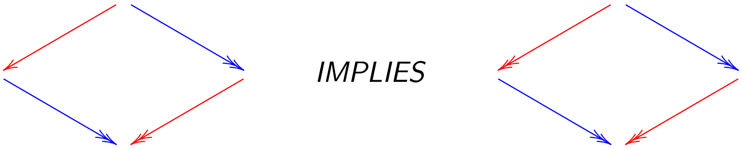


IMPLIES



Str_Confl_implies_Semi_Confl: THEOREM
strong_confluent?(R) $\Rightarrow$ semi_confluent?(R)

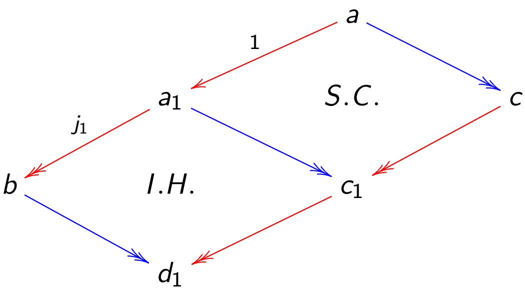
Exercises — ARSs



Ex. 1.3.6 [Staples 1975], terese: semi-commutation implies commutation.

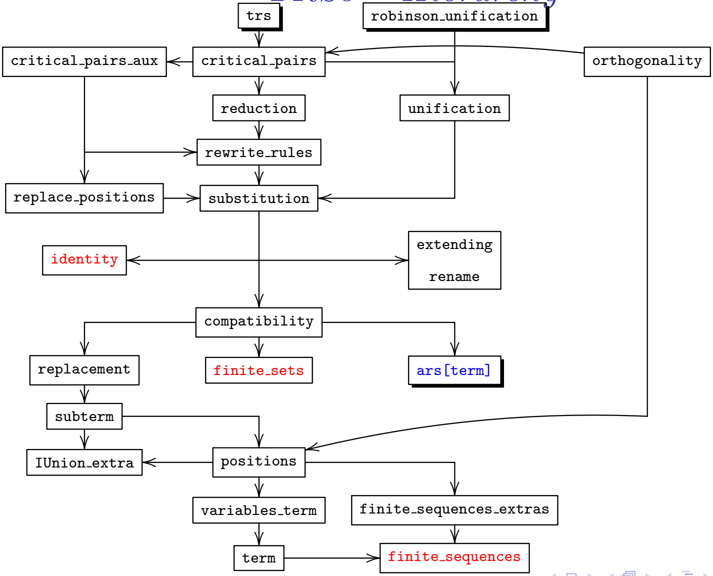
```
semi_commute?(R1,R2): bool =
  FORALL x, y, z: R1(x,y) & RTC(R2)(x,z) ⇒
         EXISTS r: RTC(R2)(y,r) & RTC(R1)(z,r)

commute?(R1,R2): bool =
  FORALL x, y, z: RTC(R1)(x,y) & RTC(R2)(x,z) ⇒
         EXISTS r: RTC(R2)(y,r) & RTC(R1)(z,r)
```

Exercises — ARSs



semi_comm_implies_comm: LEMMA
semi_commute?(R1,R2) ⇒ commute?(R1,R2)

TRSs • Hierarchy



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.

Future Work

- A myriad of elaborated theorems could 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*

Developments of the GTC at UnB - References

A. B. Avelar, F.L.C. de Moura, A. L. Galdino, and M. Ayala-Rincón.
Verification of the Completeness of Unification Algorithms à la Robinson.
In *Proc. 17th Int. Workshop on Logic, Language, Information and Computation (WoLLIC)*, volume 6188 of *Lecture Notes in Computer Science*, pages 110–124. Springer, 2010.

A.B. Avelar, A.L. Galdino, F.L.C. de Moura, and M. Ayala-Rincón.
A Formalization of the Theorem of Existence of First-Order Most General Unifiers.
In *Proceedings 6th Workshop on Logical and Semantic Frameworks with Applications, LSFA'11*, volume 81 of *EPTCS*, pages 63–78, 2011.

A.B. Avelar, A.L. Galdino, F.L.C. de Moura, and M. Ayala-Rincón.
First-order unification in the PVS proof assistant.
Logic Journal of the IGPL, 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.