PVS for Computer Scientists

Tutorial

Part 1: propositional and predicate logic

Mauricio Ayala-Rincón
Universidade de Brasília
Joint tutorial with César Muñoz and Mariano Moscato

Collocated with ITP/Tableaux/FroCoS 2017
Brasília, Brazil - Sep 25^{th} 2017

Talk's Plan

Motivation: formalization - proofs & deduction
- Deduction *à la* Gentzen
- Exercise 1: propositional logic

Formal proofs — Proofs in the Prototype Verification System - PVS
- Exercise 2: deduction in the predicate logic

Summary Gentzen versus PVS
- Additional Exercise: correctness of algorithms

Gentzen Calculus

sequents:

$$\Gamma \quad \Rightarrow \quad \Delta$$

$\uparrow$ antecedent $\qquad$ $\uparrow$ succedent

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:

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

Gentzen Calculus

Cut rule:

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

Gentzen Calculus - dealing with negation: c-equivalence

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

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

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**:

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

Sufficiency:

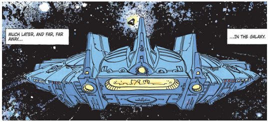
$$
\dfrac{\Gamma \Rightarrow \Delta, \varphi \qquad \bot, \Gamma \Rightarrow \Delta}{\neg\varphi, \Gamma \Rightarrow \Delta}\,(\mathrm{L}_{\rightarrow})
$$

□

There is no Plan B - Exercise 1

There can be no Plan B because' there is no "planet B,"
UN Secretary-General Ban Ki-moon

Avatar's Interstellar VS
J. Cameron



Valérian's Spaceship
P. Christin & J.-C. Mézières

Going to "Planet B"

Pandora
Alflolol
Edena

Moebius' Word of Edena

There is no Plan B - Exercise 1

See the file `planetB.pvs` in:

`www.mat.unb.br/∼ayala/planetB.pvs`

or

`www.cic.unb.br/∼ayala/planetB.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*:
   - HO functional language;
   - a type system based on Church's simple theory of types augmented with *subtypes* and *dependent types*.
2. an *interactive theorem prover*:
   - based on Gentzen's **sequent calculus**: $\boldsymbol{\Gamma} \vdash \boldsymbol{\Delta}$, where $\boldsymbol{\Gamma}$ and $\boldsymbol{\Delta}$ are finite sequences of formulae, with the usual Gentzen semantics.

Sequent calculus in PVS

- Representation of $A_1, A_2, ..., A_m \vdash B_1, B_2, ..., B_n$:

$$\begin{array}{l} [-1]\ A_1 \\ \vdots \\ [-m]\ A_m \\ |\text{---} \\ [1]\ B_1 \\ \vdots \\ [n]\ B_n \end{array}$$

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

$$\frac{\Gamma\ |\text{---}\ \Delta}{\Gamma_1\ |\text{---}\ \Delta_1 \ldots \Gamma_n\ |\text{---}\ \Delta_n}\ \textbf{(Rule Name)}$$

Some inference rules in PVS

- Structural:

$$\begin{array}{l} \vdots \\ [-i]\ A \wedge \neg B \\ \vdots \\ \texttt{|---} \\ \vdots \\ [j]\ \neg C \rightarrow D \\ \vdots \end{array} \qquad (\texttt{copy} - \texttt{i}) \qquad \begin{array}{l} [-1]\ A \wedge \neg B \\ \vdots \\ [-(i+1)]\ A \wedge \neg B \\ \vdots \\ \texttt{|---} \\ \vdots \\ [j]\ \neg C \rightarrow D \\ \vdots \end{array}$$

Some inference rules in PVS

- Structural:

$$
\begin{array}{lcl}
[-1]\ A \wedge \neg B & & [-1]\ A \wedge \neg B \\
\vdots & & \vdots \\
[-(i+1)]\ A \wedge \neg B & & |--- \\
\vdots & (\texttt{hide} - (\texttt{i}+1)) & \vdots \\
|--- & & [j]\ \neg C \to D \\
\vdots & & \vdots \\
[j]\ \neg C \to D & & \\
\vdots & &
\end{array}
$$

Some inference rules in PVS

- Propositional:

|---
[1] $A \wedge B \rightarrow (C \vee D \rightarrow C \vee (A \wedge C))$

`(flatten)`

[−1] A
[−2] B
[−3] $C \vee D$
|---
[1] C
[2] $A \wedge C$

Some inference rules in PVS

- Propositional:

[−1] $(A \to B) \to A$
|---
[1] A

(split −1)

[−1] A
|---
[1] A

|---
[1] $A \to B$
[2] A

Some inference rules in PVS

- Propositional - semantics of PVS instructions:

$$\dfrac{\dfrac{a, \Gamma \,|\text{---}\, \Delta, b}{\Gamma \,|\text{---}\, \Delta, a \to b}\ (\texttt{flatten}) \qquad \dfrac{\Gamma \,|\text{---}\, \Delta, a, c}{\Gamma \,|\text{---}\, \Delta, \neg a \to c}\ (\texttt{flatten})}{\Gamma \,|\text{---}\, \Delta, \textbf{if } a \textbf{ then } b \textbf{ else } c \textbf{ endif}}\ (\texttt{split})$$

$$\dfrac{\dfrac{a, b, \Gamma \,|\text{---}\, \Delta}{a \wedge b, \Gamma \,|\text{---}\, \Delta}\ (\texttt{flatten}) \qquad \dfrac{c, \Gamma \,|\text{---}\, \Delta, a}{\neg a \wedge c, \Gamma \,|\text{---}\, \Delta}\ (\texttt{flatten})}{\textbf{if } a \textbf{ then } b \textbf{ else } c \textbf{ endif}, \Gamma \,|\text{---}\, \Delta}\ (\texttt{split})$$

Some inference rules in PVS

- Propositional:

```
|---
[1] gcd(m, n) = gcd(n, m)
```

(case "$m \geq n$")

```
[-1] m ≥ n
|---
[1] gcd(m, n) = gcd(n, m)
```

```
|---
[1] m ≥ n
[2] gcd(m, n) = gcd(n, m)
```

Some inference rules in PVS

- Propositional (propax):

$$\frac{\Gamma, A \,|\!\!-\!\!-\!\!- A, \Delta}{} \ \textbf{(Ax)}$$

$$\frac{\Gamma, FALSE \vdash \Delta}{} \ \textbf{(FALSE}\,|\!\!-\!\!-\!\!-\,\textbf{)}$$

$$\frac{\Gamma \,|\!\!-\!\!-\!\!- TRUE, \Delta}{} \ \textbf{(}\vdash \textbf{TRUE)}$$

Some inference rules in PVS

- Predicate:

$$\begin{array}{l}[-1]\ \forall_{x:T} : P(x)\\ [-2]\ \exists_{x:T} : \neg P(x)\\ |\texttt{---}\end{array} \qquad (\texttt{skolem}\ -2\ \text{“z”}) \qquad \begin{array}{l}[-1]\ \forall_{x:T} : P(x)\\ |\texttt{---}\\ [1]\ P(z)\end{array}$$

$$\begin{array}{l}[-1]\ \forall_{x:T} : P(x)\\ |\texttt{---}\\ [1]\ P(z)\end{array} \qquad (\texttt{inst}\ -1\ \text{“z”}) \qquad \left(\begin{array}{c}[-1]\ P(z)\\ |\texttt{---}\\ [1]\ P(z)\end{array}\right)\ \texttt{Q.E.D.}$$

Analysis of GCD properties - Exercise 2

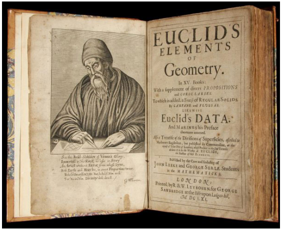
Dealing with variables:

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
$\dfrac{\Gamma \Rightarrow \Delta}{\varphi, \Gamma \Rightarrow \Delta}$ (LW)	$\dfrac{\varphi, \Gamma \vdash \Delta}{\Gamma \vdash \Delta}$ (*hide*)
$\dfrac{\varphi, \varphi, \Gamma \Rightarrow \Delta}{\varphi, \Gamma \Rightarrow \Delta}$ (LC)	$\dfrac{\varphi, \Gamma \vdash \Delta}{\varphi, \varphi, \Gamma \vdash \Delta}$ (*copy*)

Summary - Gentzen Deductive Rules vs Proof Commads

Table: Structural Right Rules vs Proof Commands

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

Summary - Gentzen Deductive Rules vs Proof Commads

Table: Logical Left Rules vs Proof Commands

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

Summary - Gentzen Deductive Rules vs Proof Commads

Table: Logical Right Rules vs Proof Commands

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

PVS proof rules versus Gentzen SC 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 - Additional Exercise

See the files `gcd.pvs` in:

`www.mat.unb.br/∼ayala/pred_gcd.pvs / .prf`

or

`www.cic.unb.br/∼ayala/pred_gcd.pvs / .prf`

Checking algorithmic properties - Addional Exercise

```
euclidean_gcd(m, (n | NOT m=0 OR NOT n=0)) :  RECURSIVE posnat =
      IF abs(m) = abs(n) THEN abs(m)
      ELSE IF (m = 0 OR n = 0) THEN abs(m+n)
           ELSE IF (abs(n) > abs(m)) THEN
                      euclidean_gcd(rem(abs(m))-abs(n), abs(m))
                ELSE euclidean_gcd(rem(abs(n))(abs(m)),abs(n))
                ENDIF
           ENDIF
      ENDIF
MEASURE abs(m)+abs(n)
```

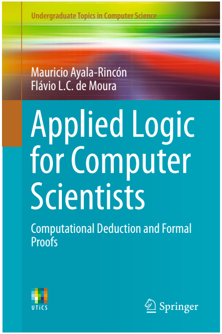
It works?

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

Checking algorithmic properties - Addional Exercise

```
euclidean_gcd_is_correct :  COROLLARY
 FORALL (m, (n | NOT m=0 OR NOT n=0)) :
   divides(euclidean_gcd(m,n),m) AND
   divides(euclidean_gcd(m,n),n) AND
   FORALL (k) :  divides(k,m) AND divides(k,n) =>
                        k <= euclidean_gcd(m,n)
```

References



2017

Logic for CS with applications to algorithm verification and details on the relations between Gentzen DN and SC rules and PVS proof commands