Computational Logic for Computer Science

# Mauricio Ayala-Rincón & Flávio L.C. de Moura Grupo de Teoria da Computação, Universidade de Brasília (UnB) Brasília D.F., Brazil

Research funded by

Brazilian Research Agencies: CNPq, CAPES and FAPDF

11° Seminário Informal (+ Formal!) GTC/UnB Brasília, Nov 28<sup>th</sup>, 29<sup>th</sup> 2013





#### Talk's Plan

Motivation: formalization - proofs & deduction

Computational proofs - logic & deduction Deduction Natural Deduction à la Gentzen

Formal proofs — Proofs in the Prototype Verification System - PVS

Formalizations versus programs

A very very simple case study: insertion sort

Conclusions and Future Work



## Motivation: classic teaching approach

Propositional logic

Semantic entailment vs deduction - completeness

Predicate logic

Semantic entailment vs deduction - completeness

Undecidability

Compactness and Löwenheim-Skolem theorems

Resolution

The focus on understanding formal logic notions gives no time for a careful analysis of deduction technologies and their usefulness in CS.



## Motivation: computational teaching approach

Propositional and Predicate logic
Semantic entailment vs deduction - completeness
Tableaux
Sat solvers
Resolution
Model cheking
Formal Verification

The focus on teaching a variety of deduction approaches gives no time for assimilating the related technologies - the more ..., the less ...

## Motivation: computational logic for CS

Induction and recursion

Classical and Intiutionistic Propositional and Predicate logic

Semantic entailment vs deduction - completeness

Natural deduction vs Gentzen Calculus

Program verification with induction and first-order deduction

Restricting computational logic to understand deduction and how this is applied in programming languages and program verification increases interest of CS and engineer students.



# Computational proofs - logic & deduction

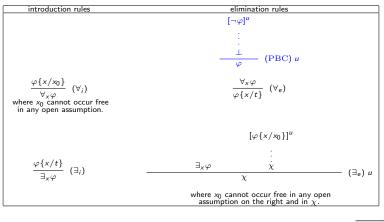
Table : NATURAL DEDUCTION FOR INTUITIONISTIC PROPOSITONAL LOGIC

introduction rules	elimination rules			
$rac{arphi^{}\psi^{}}{arphi^{}\wedge\psi^{}}$ ( $\wedge_i$ )	$rac{arphi\wedge\psi}{arphi}$ (^e)			
	$[\varphi]^{\mu}$ $[\psi]^{\nu}$			
$\frac{\varphi}{\varphi \lor \psi} (\lor_i)$		(∨ <sub>e</sub> ) u, v		
$\varphi \lor \psi$ $[\varphi]^{\mu}$	X			
$ \frac{\frac{\psi}{\psi}}{\varphi \to \psi} (\to_i) u $ $ [\varphi]^u $	$rac{arphi  arphi  ightarrow \psi}{\psi} \ \ ( ightarrow e)$			
$[\varphi]^u$ .				
$\stackrel{:}{\stackrel{\bot}{\neg \varphi}} (\neg_i) u$	$rac{arphi  \neg arphi}{\perp}  (\neg_e)$			

Universidade de Brasíli A D F A B F A B F A B F æ

# Computational proofs - logic & deduction

Table : NATURAL DEDUCTION FOR CLASSICAL PREDICATE LOGIC



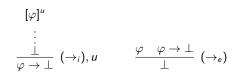
Universidade de Brasília э

(日)、

# Mathematical proofs - logic $\mathcal{E}$ deduction

#### Table : Encoding $\neg$ - Rules of natural deduction for CLASSICAL LOGIC

introduction rules	elimination rules
$[\varphi]^{u}$	
	$\varphi \neg \varphi$
$\frac{\perp}{\neg \varphi} (\neg_i), u$	$\frac{r}{ }$ $(\neg_e)$
<u>г</u>	_

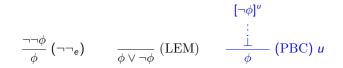




MOTIVATION: FORMALIZATION - PROOFS & DEDUCTION COMPUTATIONAL PROOFS - LOGIC & DEDUCTION FORMAL PROOFS - PR 00000000

## Mathematical proofs - logic $\mathcal{E}$ deduction

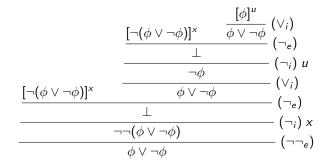
#### Interchangeable rules:





## Mathematical proofs - logic & deduction

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

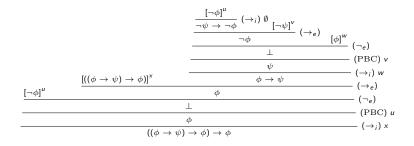


Notation:  $\neg \neg \phi \vdash \phi \lor \neg \phi$ 



# Mathematical proofs - logic & deduction

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



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



# Mathematical proofs - logic $\mathcal{E}$ deduction

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

$$\frac{\frac{[\neg\phi\{x/x_0\}]^u}{\exists x \neg \phi} (\exists_i) \quad [\neg\exists x \neg \phi]^v}{\frac{\frac{\bot}{\phi\{x/x_0\}} (\operatorname{PBC}) u}{\langle \forall_i \rangle} (\neg_e)} \frac{\frac{\Box}{\forall x \phi} (\forall_i) \quad \neg\forall x \phi}{\frac{\Box}{\exists x \neg \phi} (\operatorname{PBC}) v} (\neg_e)$$

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

$$\frac{[\neg\phi\{x/x_0\}]^u \quad \frac{[\forall x \phi]^v}{\phi\{x/x_0\}}}{\frac{\bot}{\neg\forall x \phi} \quad (\neg_i) v} \quad \neg e} \frac{\exists x \neg \phi \quad \frac{\bot}{\neg\forall x \phi} \quad (\neg_i) v}{(\exists_e) \ u}$$



# Mathematical proofs - logic $\mathcal{E}$ deduction

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

$$\frac{\left[\phi\{x/x_{0}\}\right]^{u}}{\frac{\exists x \phi}{\left(\exists_{i}\right)} \quad \neg \exists x \phi}{\left(\frac{\bot}{\neg \phi\{x/x_{0}\}} \quad (\neg_{i}) u\right)} \left(\neg_{e}\right)}$$

$$\frac{\frac{\bot}{\neg \phi\{x/x_{0}\}} \quad (\neg_{i}) u}{\forall x \neg \phi} \quad (\forall_{i})$$

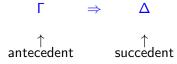
A derivation for  $\forall x \neg \phi \vdash \neg \exists x \phi$ 

$$\frac{\left[\exists x \phi\right]^{u}}{\frac{-\forall x \neg \phi}{\neg \phi\{x/x_{0}\}}} \stackrel{(\forall_{e})}{\left[\phi\{x/x_{0}\}\right]^{v}}{\frac{\bot}{\neg \exists x \phi}} \stackrel{(\neg_{e})}{(\neg_{i}) u} \stackrel{(\forall_{e})}{(\exists_{e}) v}$$



#### Gentzen Calculus

sequents:





## Gentzen Calculus

Table : Rules of deduction  $\dot{a}$  la Gentzen for predicate logic

left rules	right rules
Axioms:	
$\Gamma, \varphi \Rightarrow \varphi, \Delta$ (Ax)	$\bot, \Gamma \Rightarrow \Delta \ (L_{\bot})$
Structural rules:	
$\frac{\Gamma \Rightarrow \Delta}{\varphi, \Gamma \Rightarrow \Delta} \ (LW eakening)$	$rac{\Gamma\Rightarrow\Delta}{\Gamma\Rightarrow\Delta,arphi}$ (RWeakening)
$\frac{\varphi,\varphi,\Gamma\Rightarrow\Delta}{\varphi,\Gamma\Rightarrow\Delta} (LContraction)$	$\frac{\Gamma \Rightarrow \Delta, \varphi, \varphi}{\Gamma \Rightarrow \Delta, \varphi} \ (\textit{RContraction})$



### Gentzen Calculus

Table : RULES OF DEDUCTION à la GENTZEN FOR PREDICATE LOGIC

left rules	right rules
Logical rules:	
$\frac{\varphi_{i\in\{1,2\}},\Gamma\Rightarrow\Delta}{\varphi_1\land\varphi_2,\Gamma\Rightarrow\Delta} (L_{\wedge})$	$\frac{\Gamma \Rightarrow \Delta, \varphi \ \Gamma \Rightarrow \Delta, \psi}{\Gamma \Rightarrow \Delta, \varphi \land \psi} \ (R_{\wedge})$
$\frac{\varphi, \Gamma \Rightarrow \Delta}{\varphi \lor \psi, \Gamma \Rightarrow \Delta}  (\mathcal{L}_{\lor})$	$\frac{\Gamma \Rightarrow \Delta, \varphi_{i \in \{1,2\}}}{\Gamma \Rightarrow \Delta, \varphi_1 \lor \varphi_2} \ (R_{\lor})$
$\frac{\Gamma \Rightarrow \Delta, \varphi \ \psi, \Gamma \Rightarrow \Delta}{\varphi \to \psi, \Gamma \Rightarrow \Delta} \ (L_{\to})$	$\frac{\varphi, \Gamma \Rightarrow \Delta, \psi}{\Gamma \Rightarrow \Delta, \varphi \rightarrow \psi} \ (R_{\rightarrow})$
$\frac{\varphi[\mathbf{x}/t], \Gamma \Rightarrow \Delta}{\forall_{\mathbf{x}}\varphi, \Gamma \Rightarrow \Delta} \ (L_{\forall})$	$\frac{\Gamma \Rightarrow \Delta, \varphi[x/y]}{\Gamma \Rightarrow \Delta, \forall_x \varphi} \ (R_\forall),  y \not\in \texttt{fv}(\Gamma, \Delta)$
$\frac{\varphi[x/y], \Gamma \Rightarrow \Delta}{\exists_x \varphi, \Gamma \Rightarrow \Delta} \ (L_{\exists}),  y \not\in fv(\Gamma, \Delta)$	$\frac{\Gamma \Rightarrow \Delta, \varphi[x/t]}{\Gamma \Rightarrow \Delta, \exists_x \varphi} \ (R_{\exists})$
	-

(日) (同) (日) (日)

#### Gentzen Calculus

Derivation of the Peirce's law:

$$(RW) \frac{\varphi \Rightarrow \varphi (Ax)}{\varphi \Rightarrow \varphi, \psi} \\ (R_{\rightarrow}) \frac{\varphi \Rightarrow \varphi, \psi}{\varphi \Rightarrow \varphi, \psi} \qquad \varphi \Rightarrow \varphi (Ax) \\ \frac{\varphi \Rightarrow \varphi, \varphi \rightarrow \psi}{(\varphi \rightarrow \psi) \rightarrow \varphi \Rightarrow \varphi} (R_{\rightarrow}) \\ \frac{\varphi \Rightarrow \psi}{\varphi \Rightarrow \psi} (L_{\rightarrow})$$



MOTIVATION: FORMALIZATION - PROOFS & DEDUCTION COMPUTATIONAL PROOFS - LOGIC & DEDUCTION FORMAL PROOFS - PR

### Gentzen Calculus

Cut rule:

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



#### Gentzen Calculus

Example of application of (Cut):  $\vdash \Rightarrow \neg \neg (\psi \lor \neg \psi)$ .

$$\frac{\frac{\psi \Rightarrow \psi, \bot (Ax)}{\Rightarrow \psi, \neg \psi} (R_{\rightarrow})}{\frac{\Rightarrow \psi, \neg \psi}{\Rightarrow \psi \lor \neg \psi, \neg \psi} (R_{\lor})} = \frac{(Ax) \psi \lor \neg \psi \Rightarrow \psi \lor \neg \psi \quad \bot \Rightarrow \bot (L_{\bot})}{\psi \lor \neg \psi, \neg (\psi \lor \neg \psi) \Rightarrow \bot} (R_{\rightarrow}) = \frac{(Ax) \psi \lor \neg \psi \Rightarrow \psi \lor \neg \psi}{\psi \lor \neg \psi, \neg (\psi \lor \neg \psi) \Rightarrow \bot} (R_{\rightarrow}) = \frac{(Ax) \psi \lor \neg \psi \Rightarrow \neg (\psi \lor \neg \psi) \Rightarrow \bot}{\psi \lor \neg \psi \Rightarrow \neg \neg (\psi \lor \neg \psi)} (Cut)$$



# The Prototype Verification System - PVS

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

- 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.

Universidade de

э

A D F A B F A B F A B F

- an interactive theorem prover:
  - based on Gentzen sequent calculus.

## 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.



## Sequent calculus

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

• Example: 
$$\frac{\Gamma \vdash \Delta}{\Gamma' \vdash \Delta'}$$

• Goal:  $\vdash \Delta$ .



## Sequent calculus in PVS

- Proof tree: each node is labelled by a sequent.
- A PVS proof command (*R*) corresponds to the reverse application of an inference rule.

• In general:

$$\frac{\Gamma \vdash \Delta}{\Gamma_1 \vdash \Delta_1 ... \Gamma_n \vdash \Delta_n} \ (R)$$

[n] B<sub>n</sub>



## Some inference rules in PVS

• <u>Structural</u>:

$$\frac{\Gamma, \Gamma' \vdash \Delta, \Delta'}{\Gamma \vdash \Delta} \text{ (hide) } (W) \qquad \frac{\Gamma, \Gamma' \vdash \Delta, \Delta'}{\Gamma, \Gamma', \Gamma' \vdash \Delta, \Delta', \Delta'} \text{ (copy) } (C)$$

• <u>Axioms</u>:

$$\Gamma, A \vdash A, \Delta$$
 (Ax) (Ax)

$$[\Gamma, FALSE \vdash \Delta \text{ (False} \vdash) (L_{\perp})] \quad [\Gamma \vdash TRUE, \Delta; (\vdash True) (L_{\perp})]$$



## Some inference rules in PVS

#### • Logical (propositional) rules:

$$\frac{\Gamma \vdash \Delta, \psi \lor \varphi}{\Gamma, \vdash \Delta, \psi, \varphi} \text{ (flatten) } (R_{\lor})$$
$$\frac{\psi \land \varphi, \Gamma, \vdash \Delta}{\psi, \varphi, \Gamma \vdash \Delta} \text{ (flatten) } (L_{\land})$$
$$\frac{\Gamma \vdash \Delta, \psi \rightarrow \varphi}{\psi, \Gamma \vdash \Delta, \varphi} \text{ (flatten) } (R_{\rightarrow})$$



## Some inference rules in PVS

• Logical (propositional) rules:

$$\begin{array}{c} \frac{\psi \lor \varphi, \Gamma \vdash \Delta}{\psi, \Gamma \vdash \Delta} \hspace{0.1cm} \textbf{(split)} \hspace{0.1cm} (L_{\vee}) \\ \\ \frac{\Gamma \vdash \Delta, \psi \land \varphi}{\Gamma \vdash \Delta, \psi} \hspace{0.1cm} \textbf{(split)} \hspace{0.1cm} (L_{\wedge}) \\ \\ \frac{\psi \rightarrow \varphi, \Gamma \vdash \Delta}{\varphi, \Gamma \vdash \Delta, \Gamma \vdash \Delta, \psi} \hspace{0.1cm} \textbf{(split)} \hspace{0.1cm} (L_{\rightarrow}) \end{array} \end{array}$$



### Some inference rules in PVS

• Logical (classical) rules:  $\frac{\forall_{x}\psi, \Gamma \vdash \Delta}{\psi[x/t], \Gamma \vdash \Delta} \text{ (inst) } (L_{\forall}) \qquad \frac{\Gamma \vdash \Delta, \forall_{x}\psi}{\Gamma \vdash \Delta, \psi[x/x_{0}]} \text{ (skolem) } (R_{\forall})$   $\frac{\Gamma \vdash \Delta, \exists_{x}\psi}{\Gamma \vdash \Delta, \psi[x/t]} \text{ (inst) } (R_{\exists}) \qquad \frac{\exists_{x}\psi, \Gamma \vdash \Delta}{\psi[x/x_{0}], \Gamma \vdash \Delta} \text{ (skolem) } (L_{\exists})$ 



## PVS vs Gentzen Rules

	(flatten)	(split)	(inst)	(skolem)
$(L_{\vee})$		×		
$(R_{\vee})$	Х			
$(L_{\wedge})$	×			
$(R_{\wedge})$		×		
$(L_{\rightarrow})$		×		
$(R_{\rightarrow})$	×			
$(L_{\forall})$			×	
$(R_{\forall})$				×
( <i>L</i> ∃)				×
( <i>R</i> ∃)			×	



## PVS propositional derivation example

Derivation of the Peirce's law:

$$(R_{\rightarrow}) \frac{\varphi \Rightarrow \varphi, \psi (Ax)}{\Rightarrow \varphi, \varphi \rightarrow \psi} \qquad \varphi \Rightarrow \varphi (Ax) \\ \frac{\varphi \Rightarrow \varphi, \varphi \rightarrow \psi}{(\varphi \rightarrow \psi) \rightarrow \varphi \Rightarrow \varphi} \qquad (R_{\rightarrow}) \\ \hline \Rightarrow ((\varphi \rightarrow \psi) \rightarrow \varphi) \rightarrow \varphi \qquad (L_{\rightarrow})$$

$$(\textit{flatten}) \frac{ \begin{array}{c} \vdash ((\varphi \rightarrow \psi) \rightarrow \varphi) \rightarrow \varphi \\ \hline (\varphi \rightarrow \psi) \rightarrow \varphi \vdash \varphi \\ \hline \hline (\varphi \rightarrow \psi) \rightarrow \varphi \vdash \varphi \\ \hline \varphi \vdash \varphi, \varphi \rightarrow \psi \\ \hline \varphi \vdash \varphi, \psi \quad (Ax) \\ \end{array}}{(\textit{flatten})} \frac{ \begin{array}{c} \vdash (\varphi, \varphi \rightarrow \psi) \\ \hline \varphi \vdash \varphi, \psi \quad (Ax) \\ \hline \varphi \vdash \varphi \quad (Ax) \\ \end{array}}$$

Universidade de Brasília

æ

・ロト ・四ト ・モト ・モト

### Additional rules in PVS

#### • <u>Case</u>:

• Corresponds to the rule (Cut).

$$\frac{\Gamma\vdash\Delta}{\Gamma,A\vdash\Delta\quad\Gamma\vdash A,\Delta} \ \textbf{(Case ``A'')}$$

• <u>Conditional</u>: IF-THEN-ELSE.

$$\left| \frac{\Gamma, A \land B \vdash \Delta \quad \Gamma, \neg A \land C \vdash \Delta}{\Gamma, \operatorname{IF}(A, B, C) \vdash \Delta} \right|$$
(split)

$$\frac{\Gamma, A \to B \vdash \Delta \quad \Gamma, \neg A \to C \vdash \Delta}{\Gamma \vdash \mathrm{IF}(A, B, C)\Delta} \text{ (split)}$$



Case Study: insertion sort

```
insert(x : \mathbb{N}, l : list[\mathbb{N}]) : RECURSIVE list[\mathbb{N}] = if null?(l) then 
| cons(x, null) else 
| if <math>x \le car(l) then 
| cons(x, l) 
| else 
| cons(car(l), insert(x, cdr(l))) 
| end 
end 
MEASURE length(l)
```



```
Case Study: insertion sort
```

```
in_sort(1: list[N]): RECURSIVE list[N] =
if null?(1) then
| null
else
| insert(car(1), in_sort(cdr(1)))
end
MEASURE length(1)
```



```
Insertion sort — correctness
```

```
insert_preserves_order : LEMMA \forall (I : list[nat], x : nat) :
is_sorted?(I) \rightarrow is_sorted?(insert(x, I))
```



#### Insertion sort — correctness formalization

The proof is by induction on |I|. Induction hypothesis (IH):  $\forall (l', x') : |l'| < |l| \rightarrow (is\_sorted?(l') \rightarrow is\_sorted?(insert(x', l'))$ Sequent:

 $\forall (l', x') : |l'| < |l| \rightarrow (is\_sorted?(l') \rightarrow is\_sorted?(insert(x', l')) \Rightarrow$ is\\_sorted?(l)  $\rightarrow$  is\\_sorted?(insert(x, l))

... two interesting sequents should be proved:

 $\begin{array}{ll} null?(I), \texttt{is\_sorted}?(I), IH & \texttt{is\_sorted}?(I), IH \\ \Rightarrow & \texttt{and} & \Rightarrow \\ \texttt{is\_sorted}?(\texttt{insert}(x, I)) & null?(I), \texttt{is\_sorted}?(\texttt{insert}(x, I)) \end{array}$ 

Universidade de

#### Insertion sort — correctness formalization

The former sequent is easily proved. For the latter sequent, insert is expanded obtaining:

```
is\_sorted?(I), IH

\Rightarrow

null?(I),

is\_sorted?(if x \le car(I) then cons(x, I) else

cons(car(I), insert(x, cdr(I))))
```

Applying logical commands such as (lift-if) and (prop), that guided an application of the (Cut) rule by the guard  $x \le car(l)$  of the **if-then-else** command this gives two sequents:

```
x \leq car(l), is\_sorted?(l), IH \Rightarrow null?(l), is\_sorted?(cons(x, l))
```

and is\_sorted?(I),  $IH \Rightarrow null?(I)$ ,  $x \le car(I)$ , is\_sorted?(cons(car(I), insert(x, cdr(I))))



#### Insertion sort — correctness formalization

The former sequent is easily proved. For the latter one, the *IH* is used by applying the PVS instantiation command (inst) which corresponds to  $(L_{\exists})$ , obtaining the sequent:

```
\begin{split} &\text{is\_sorted?}(I), \\ &|cdr(I)| < |I| \rightarrow (\text{is\_sorted?}(cdr(I)) \rightarrow \\ &\text{is\_sorted?}(\text{insert}(x, cdr(I))) \\ \Rightarrow \\ &null?(I), x \leq car(I) \\ &\text{is\_sorted?}(cons(car(I), \text{insert}(x, cdr(I)))) \end{split}
```



#### Insertion sort — correctness formalization

By applications of the command (prop), guided applications of  $(L_{\rightarrow})$  by the premises of the implications in the antecedent, that is, |cdr(I)| < |I| and is\_sorted?(cdr(I)), are done, obtaining the interesting sequent below that follows easily.

```
is\_sorted?(I), |cdr(I)| < |I|, \\is\_sorted?(cdr(I)), \\is\_sorted?(insert(x, cdr(I))) \\ \Rightarrow \\null?(I), x \le car(I) \\is\_sorted?(cons(car(I), insert(x, cdr(I))))
```



# Conclusions

- Nowadays, computational logic is intensively applied in formal methods.
- In computer sciences, a useful 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.



## Work in Progress

- Textbook on truly computational logic with concrete applications.
  - M.Ayala-Rincón & F.L.C.de Moura *Applied Logic for Computer Scientists: computational deduction and formal proofs*, 2014, UTiCS series, Springer.

GTC at the Universidade de Brasília.



