XXIX Semana do IME/UFG e VI Seminário de Pesquisa e Pós-Graduação do IME/UFG

Formalização de Teoremas em Assistentes de Prova

Section 1: Lógica e Dedução Formal

Thaynara Arielly de Lima (IME) UFG
Mauricio Ayala-Rincón (CIC-MAT) UnB

Funded by FAPDF DE grant 00193.0000.2144/2018-81, CNPq Research Grant 307672/2017-4

Oct 6 -8 , 2021

Talk's Plan

1. Section 1
   - Formalizing Mathematics
   - Gentzen's Calculus
   - The Prototype Verification System (PVS)
   - Gentzen Deductive Rules vs PVS Proof Commands
   - Preliminary Exercises

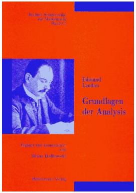
Formalizing Mathematics

Since the early development of computers, implementing mathematical deduction was a very important challenge:

Nicolaas Govert de Bruijn (1918-2012).
Dutch mathematician leader of the
Automath project.

Automath started in 1967:

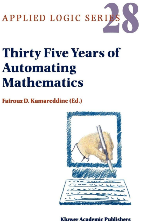
Mechanical verification of the famous Edmund Landau's (1877-1938) book *Grundlagen der Analysis*, Leipzig 1930.


Berliner Studienreihe zur Mathematik Band 11
Edmund Landau
Grundlagen der Analysis
Ergänzt und kommentiert von
Heinz Dalkowski
Heldermann Verlag

Formalizing Mathematics

https://www.win.tue.nl/automath/

Automath is considered predecessor of modern proof assistants as: Coq, Nurpl, Isabelle, PVS ...


APPLIED LOGIC SERIES 28

Thirty Five Years of Automating Mathematics

Fairouz D. Kamareddine (Ed.)

Kluwer Academic Publishers

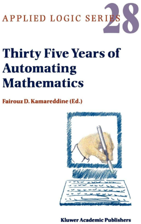
STUDIES IN LOGIC

THE FOUNDATIONS OF MATHEMATICS

Selected Papers on Automath

Formalizing Mathematics

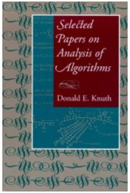
In Automath N.G. de Bruijn developed the first formalization of λ-calculus with intuitionistic types and explicit substitutions.


APPLIED LOGIC SERIES 28
Thirty Five Years of Automating Mathematics
Fairouz D. Kamareddine (Ed.)
Kluwer Academic Publishers

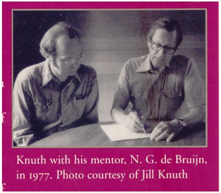
N.G. de Bruijn was a well established mathematician before deciding in 1967 at the age of 49 to work on a new direction related to Automating Mathematics. In the 1960s he became fascinated by the new computer technology and decided to start the new Automath project where he could check, with the help of the computer, the correctness of books of mathematics. Through his work on Automath, de Bruijn started a revolution in using the computer for verification, and since, we have seen more and more proof-checking and theorem-proving systems.

Formalizing Mathematics

N.G. De Bruijn's influence in computing is not restricted to Automath.



Donald Knuth dedicates his book to his mentor, N. G. de Bruijn.


Knuth with his mentor, N. G. de Bruijn, in 1977. Photo courtesy of Jill Knuth

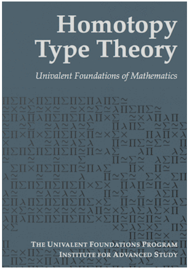
... I'm dedicating this book to N.G. "Dick" de Bruijn because his influence can be felt on every page. Ever since the 1960s he has been my chief mentor, the main person who would answer my questions when I was stuck on a problem that I had not been taught how to solve. I originally wrote Chapter 26 for his $(3 \cdot 4 \cdot 5)$*th birthday; now he is* 3^4 *years young as I gratefully present him with this book.*

Donald E. Knuth

Formalizing Mathematics

Vladimir Voevodsky (1966-2017) (2002) popularised the Univalent Foundations that use classical predicate logic as the underlying deductive sytem, categorical approaches, and intuitionistic types, indeed the so called

`https://homotopytypetheory.org`



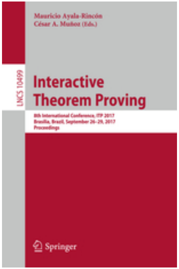
Homotopy Type Theory

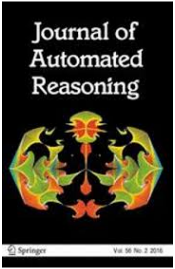
Univalent Foundations of Mathematics

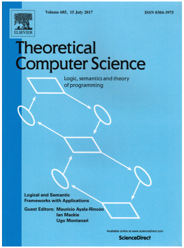
The Univalent Foundations Program
Institute for Advanced Study

Formalizing Mathematics

Some related conferences/journals:


Mauricio Ayala-Rincón
César A. Muñoz (Eds.)
LNCS 10499
Interactive Theorem Proving
8th International Conference, ITP 2017
Brasília, Brazil, September 26–29, 2017
Proceedings
Springer




Volume 685, 15 July 2017
ISSN 0304-3975
ELSEVIER
Theoretical Computer Science
Logic, semantics and theory of programming
Logical and Semantic Frameworks with Applications
Guest Editors: Mauricio Ayala-Rincón
Ian Mackie
Ugo Montanari
Available online at www.sciencedirect.com
ScienceDirect

MSCS
Mathematical Structures in Computer Science
A journal in the applications of categorical, algebraic and geometric methods in Computer Science
CAMBRIDGE

Formalized Mathematics by GTC members

- Rewriting Theory

  `https://github.com/nasa/pvslib/tree/master/TRS`

- Termination

  `https://github.com/nasa/pvslib/tree/master/PVS0`

  `https://github.com/nasa/pvslib/tree/master/CCG`

- Nominal equational reasoning

  `nominal.cic.unb.br`

- Group and Ring's Theories

  `https://github.com/nasa/pvslib/tree/master/algebra`

Formalized Mathematics by GTC members:

Term Rewriting trs.cic.unb.br

- Termination — Ariane Almeida (PhD Informatics 2021), Thiago Ramos (PhD Inf student)

  Science of Computer Programming
  "Formalizing the Dependency Pair Criterion for Innermost Termination"

- Confluence — André Galdino (PhD Math 2008), Ana Cristina Oliveira (PhD Inf 2016)

  JFR (2008) *"A Formalization of Newman's and Yokouchi's Lemmas in a Higher-Order Language"*

  Journal of Automated Reasoning
  (2017) *"Confluence of Orthogonal Term Rewriting Systems in the Prototype Verification System"*

- Knuth-Bendix Critical Pairs Theorem — André Galdino (PhD Math 2008)

  Journal of Automated Reasoning
  (2010) *"A Formalization of the Knuth-Bendix(-Huet) Critical Pair Theorem"*

- Existence of First-order Unification Theorem — Andréia Avelar (PhD Math 2014)

  Logic Journal of the IGPL
  (2014) *"First-order unification in the PVS proof assistant"*

Formalized Mathematics by GTC members:

Termination Analysis

- Formalization of the Computational Theory of a functional language - Thiago Ramos (PhD Inf Student), Andreia Avelar (PhD Math 2014), Ariane Alves (PhD Math 2021), Mariano Moscato & César Muñoz (NIA / NASA LaRC FM)

  (2018) *"Formalization of the Undecidability of the Halting Problem for a Functional Language"*

- TRS Termination by Dependency Pairs Criteria Theorem — Ariane Alves Almeida (PhD Inf UnB Student)

  (2020) *"Formalizing the Dependency Pair Criterion for Innermost Termination"*

- (Submitted 2021) *"Formal Verification of Termination Criteria for First-Order Recursive Functions"*. Presented in (2021) .

Formalized Mathematics by GTC members:

Nominal Equational Reasoning nominal.cic.unb.br

equality check: $s = t$? matching: $\exists\sigma : s\sigma = t$? unification: $\exists\sigma : s\sigma = t\sigma$?

- Formalization of Functional Nominal Unification — Ana Cristina Oliveira (PhD Inf 2016)

  (2015) *“Completeness in PVS of a Nominal Unification Algorithm”*

- Formalization of Rule-Inference Nominal Unification and Matching Modulo C — Washington de Carvalho Segundo (PhD Inf 2019)

  (2017) *“Nominal C-Unification”*

- Formalization of Functional Nominal Equality Check Modulo AC — W. de Carvalho

  (2019) *“A formalisation of nominal α-equivalence with A, C, and AC function symbols”*

- Formalization of Functional Nominal Unification and Matching Modulo C — W. de Carvalho and Gabriel Silva (PhD Inf student)

  (2019) (2021) *“Functional Formalisation of Nominal C-Unification and Matching with Protected Variables”*

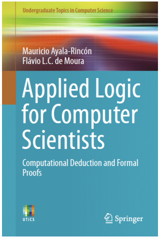
Formalized Mathematics by GTC members: Groups and Rings

- Thaynara A. de Lima (UFG), André Galdino (UFCat), Andréia Avelar (UnB), M. Ayala-Rincón (UnB)

  (2021) *"Formalization of Ring Theory in PVS - Isomorphism Theorems, Principal, Prime and Maximal Ideals,Chinese Remainder Theorem"*

Formalizing Mathematics at GTC (PPGs Math & Inf — UnB)

You are welcome!



Undergraduate Topics in Computer Science

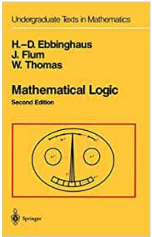
Mauricio Ayala-Rincón
Flávio L.C. de Moura

Applied Logic for Computer Scientists

Computational Deduction and Formal Proofs

UTICS

Springer



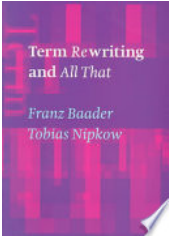
Undergraduate Texts in Mathematics

H.-D. Ebbinghaus
J. Flum
W. Thomas

Mathematical Logic

Second Edition

Springer



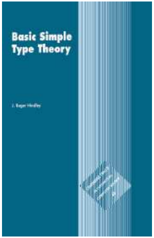
Term Rewriting and *All That*

Franz Baader
Tobias Nipkow

Basic Proof Theory

Second Edition

A. S. Troelstra and
H. Schwichtenberg



Basic Simple Type Theory

J. Roger Hindley

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: $\vdash ((\varphi \rightarrow \psi) \rightarrow \psi) \rightarrow \psi$

$$\varphi \Rightarrow \varphi \ \ (Ax)$$

Gentzen Calculus

Derivation of the Peirce's law: $\vdash ((\varphi \to \psi) \to \psi) \to \psi$

$$(RW)\ \dfrac{\varphi \Rightarrow \varphi \ \ (Ax)}{\varphi \Rightarrow \varphi, \psi}$$

Gentzen Calculus

Derivation of the Peirce's law: $\vdash ((\varphi \to \psi) \to \psi) \to \psi$

$$\begin{array}{rc}
(RW) & \dfrac{\varphi \Rightarrow \varphi \quad (Ax)}{\varphi \Rightarrow \varphi, \psi} \\
(R_{\to}) & \dfrac{}{\Rightarrow \varphi, \varphi \to \psi}
\end{array}$$

Gentzen Calculus

Derivation of the Peirce's law: $\vdash ((\varphi \rightarrow \psi) \rightarrow \psi) \rightarrow \psi$

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

Gentzen Calculus

Derivation of the Peirce's law: $\vdash ((\varphi \to \psi) \to \psi) \to \psi$

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

Gentzen Calculus

Derivation of the Peirce's law: $\vdash ((\varphi \to \psi) \to \psi) \to \psi$

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

Gentzen Calculus

Derivation of: $\vdash \exists_x \neg\varphi \Rightarrow \neg\forall_x\varphi$

$$\cfrac{\cfrac{\cfrac{(L_\forall)\ \cfrac{\varphi[x/t] \Rightarrow \varphi[x/t]}{\forall_x\varphi \Rightarrow \varphi[x/t]}}{\neg\varphi[x/t], \forall_x\varphi \Rightarrow}\ (\text{C-EQUIV})}{\neg\varphi[x/t] \Rightarrow \neg\forall_x\varphi}\ (\text{C-EQUIV})}{\exists_x\neg\varphi \Rightarrow \neg\forall_x\varphi}\ (L_\exists)$$

Gentzen Calculus

Cut rule:

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

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 1 (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.

1. **Necessity**:

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

Sufficiency:

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

Gentzen Calculus - dealing with negation

(ii) **Necessity**:

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

$$\overline{\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad}\ (\textsc{Cut})$$

$$\Gamma\Rightarrow\Delta,\varphi$$

Sufficiency:

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

□

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 $\mathbf{\Gamma} \vdash \mathbf{\Delta}$, where $\mathbf{\Gamma}$ and $\mathbf{\Delta}$ are finite sequences of formulae, with the usual Gentzen semantics.

The Prototype Verification System (PVS) — Libraries

- The `prelude` library
  - It is a collection of basic *theories* containing specifications about:
    - functions;
    - sets;
    - predicates;
    - logic; among others.
  - The theories in the prelude library are visible in all PVS contexts;
  - It provides the infrastructure for the PVS typechecker and prover, as well as much of the basic mathematics needed to support specification and verification of systems.

The Prototype Verification System (PVS) — Libraries

- NASA LaRC PVS library (`nasalib`)
  - It includes the *theories*
    - ★ 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 `nasalib` is maintaned by the NASA LaRC formal methods group;
  - The `nasalib` is result of research developed by the NASA LaRC formal methods group and the cientific comunity in general.

Sequent Calculus in PVS

A sequent of the form $\Gamma \vdash \Delta$ (or $A_1, A_2, ..., A_n \vdash B_1, B_2, ..., B_m$, since Γ and Δ are finite sequences of formulae) is:

- interpreted as:
  $A_1 \wedge A_2 \wedge ... \wedge A_n \vdash B_1 \vee B_2 \vee ... \vee B_m$,
  that is, from the conjunction of the antecedent formulae one obtains the disjunction of the succedent formulae.

- represented in PVS as:

$$\begin{array}{c} \texttt{[-1]}\ \ A_1 \\ \vdots \\ \texttt{[-n]}\ \ A_n \\ \texttt{|----------} \\ \texttt{[1]}\ \ B_1 \\ \vdots \\ \texttt{[m]}\ \ B_m \end{array}$$

Sequent Calculus in PVS

- Inference rules
  - Premises and conclusions are simultaneously constructed:

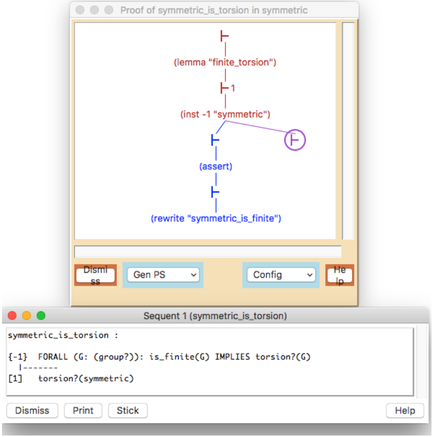
$$\frac{\Gamma \vdash \Delta}{\Gamma' \vdash \Delta'}$$

  - A PVS proof command corresponds to the application of an inference rule. In general:

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

- Goal: $\vdash \Delta$.

- Proof tree: each node is labelled by a sequent


Proof of symmetric_is_torsion in symmetric
⊢
(lemma "finite_torsion")
⊢1
(inst -1 "symmetric")
⊢
⊢
(assert)
⊢
(rewrite "symmetric_is_finite")
Dismiss
Gen PS
Config
Help
Sequent 1 (symmetric_is_torsion)
Dismiss
Print
Stick
Help

```
symmetric_is_torsion :

{-1}  FORALL (G: (group?)): is_finite(G) IMPLIES torsion?(G)
  |-------
[1]   torsion?(symmetric)
```

Some inference rules in PVS

- Structural:

Deduction rule	PVS command
$\dfrac{\varphi, \varphi, \Gamma \Rightarrow \Delta}{\varphi, \Gamma \Rightarrow \Delta}\ (LContraction)$	$\dfrac{\varphi, \Gamma \vdash \Delta}{\varphi, \varphi, \Gamma \vdash \Delta}\ (copy)$

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

Some inference rules in PVS

- Structural:

Deduction rule	PVS command
$\dfrac{\Gamma \Rightarrow \Delta}{\varphi, \Gamma \Rightarrow \Delta}\ (LWeakening)$	$\dfrac{\varphi, \Gamma \vdash \Delta}{\Gamma \vdash \Delta}\ (hide)$

$[-1]\ A \wedge \neg B$

$\vdots$

$[-(i+1)]\ A \wedge \neg B$

$\vdots$

$|\ -\ -\ -\ -\ -\ -$

$\vdots$

$[j]\ \neg C \to D$

$\vdots$

$(\mathtt{hide} - (\mathtt{i}+\mathtt{1}))$ $\rightsquigarrow$

$[-1]\ A \wedge \neg B$

$\vdots$

$|\ -\ -\ -\ -\ -\ -$

$\vdots$

$[j]\ \neg C \to D$

$\vdots$

Some inference rules in PVS

- Propositional:

$$|-------$$

$$[1]\ A \wedge B \to (C \vee D \to C \vee (A \wedge C))$$

$$\downarrow \textcolor{red}{(\texttt{flatten})}$$

$$[-1]\ A$$

$$[-2]\ B$$

$$[-3]\ C \vee D$$

$$|-------$$

$$[1]\ C$$

$$[2]\ A \wedge C$$

Deduction rule	PVS command $(flatten)$
$\dfrac{\varphi, \Gamma \Rightarrow \Delta, \psi}{\Gamma \Rightarrow \Delta, \varphi \to \psi}\ (R_\to)$	$\dfrac{\Gamma \vdash \Delta, \varphi \to \psi}{\varphi, \Gamma \vdash \Delta, \psi}$
$\dfrac{\varphi_1, \varphi_2, \Gamma \Rightarrow \Delta}{\varphi_1 \wedge \varphi_2, \Gamma \Rightarrow \Delta}\ (L_\wedge)$	$\dfrac{\varphi_1 \wedge \varphi_2, \Gamma \vdash \Delta}{\varphi_{i\in\{1,2\}}, \Gamma \vdash \Delta}$
$\dfrac{\Gamma \Rightarrow \Delta, \varphi_{i\in\{1,2\}}}{\Gamma \Rightarrow \Delta, \varphi_1 \vee \varphi_2}\ (R_\vee)$	$\dfrac{\Gamma \vdash \Delta, \varphi_1 \vee \varphi_2}{\Gamma \vdash \Delta, \varphi_1, \varphi_2}$

Some inference rules in PVS

- Propositional:

Deduction rule	PVS command
$\dfrac{\Gamma \Rightarrow \Delta, \varphi \quad \psi, \Gamma \Rightarrow \Delta}{\varphi \to \psi, \Gamma \Rightarrow \Delta}\ (L_{\to})$	$\dfrac{\varphi \to \psi, \Gamma \vdash \Delta}{\Gamma \vdash \Delta, \varphi \quad \psi, \Gamma \vdash \Delta}\ (split)$

$$[-1]\ (A \to B) \to A$$

$|- - - - - -$ $\qquad$ $\color{red}{(\texttt{split}\ -1)}$

$$[1]\ A$$

↙ ↘

$[-1]\ A$

$|- - - - - -$

$[1]\ A$

$|- - - - - -$

$[1]\ A \to B$

$[2]\ A$

Some inference rules in PVS

- Propositional:

$$
\begin{array}{l}
|------ \quad \texttt{(case "m} \geq \texttt{n")}\\
[1]\ \ \mathrm{gcd}(m,n) = \mathrm{gcd}(n,m)
\end{array}
\rightsquigarrow
\begin{array}{l}
[-1]\ m \geq n\\
|------\\
[1]\ \ \mathrm{gcd}(m,n) = \mathrm{gcd}(n,m)\\
\\
|------\\
[1]\ m \geq n\\
[2]\ \ \mathrm{gcd}(m,n) = \mathrm{gcd}(n,m)
\end{array}
$$

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 (propax):

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

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

Some inference rules in PVS

- Predicate:

Deduction rule	PVS command
$\dfrac{\varphi[x/y], \Gamma \Rightarrow \Delta}{\exists_x \varphi, \Gamma \Rightarrow \Delta}\ (L_\exists), \quad y \notin \texttt{fv}(\Gamma, \Delta)$	$\dfrac{\exists_x \varphi, \Gamma \vdash \Delta}{\varphi[x/y], \Gamma \vdash \Delta}\ (skolem), \quad y \notin \texttt{fv}(\Gamma, \Delta)$
$\dfrac{\varphi[x/t], \Gamma \Rightarrow \Delta}{\forall_x \varphi, \Gamma \Rightarrow \Delta}\ (L_\forall)$	$\dfrac{\forall_x \varphi, \Gamma \vdash \Delta}{\varphi[x/t], \Gamma \vdash \Delta}\ (inst)$

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

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

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
$\dfrac{\Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta, \varphi}$ $(RW\textit{eakening})$	$\dfrac{\Gamma \vdash \Delta, \varphi}{\Gamma \vdash \Delta}$ $(hide)$
$\dfrac{\Gamma \Rightarrow \Delta, \varphi, \varphi}{\Gamma \Rightarrow \Delta, \varphi}$ $(RC\textit{ontraction})$	$\dfrac{\Gamma \vdash \Delta, \varphi}{\Gamma \vdash \Delta, \varphi, \varphi}$ $(copy)$

Summary - Gentzen Deductive Rules vs Proof Commads

Table: Logical Left Rules vs Proof Commands

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

Summary - Gentzen Deductive Rules vs Proof Commads

Table: Logical Right Rules vs Proof Commands

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

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 - infinity of Primes

See the file `preliminaries.pvs` in Exercises directory

Infinitude dos Primos: Argumento de Fürstenberg

Topologia

Uma topologia sobre um conjunto X é uma coleção τ de subconjuntos de X com as propriedades:

i) $\emptyset$ e X pertencem a τ;

ii) A união de elementos de qualquer subcoleção de τ pertence a τ;

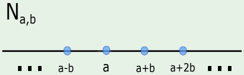
iii) A interseção dos elementos de qualquer subcoleção finita de τ está em τ.

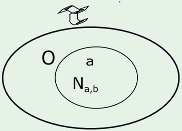
- Um conjunto X munido de uma topologia τ é dito **espaço topológico**.
- Um subconjunto U de um espaço topológico X, que pertece à coleção τ, é chamado um **aberto de** X.

Argumento de Fürstenberg

Exemplo

Considere os conjuntos $X = \mathbb{Z}$ e $N_{a,b} = \{a + n \cdot b; n \in \mathbb{Z}\}$, $a, b \in \mathbb{Z}$, com $b > 0$. Um conjunto $O \subseteq \mathbb{Z}$ é dito aberto se $O = \emptyset$ ou se para todo $a \in O$, existe $b > 0$ inteiro tal que $N_{a,b} \subseteq O$.





$\mathcal{T}$, induzida pelos abertos O, é uma topologia sobre $\mathbb{Z}$:

i) $\emptyset$ e $\mathbb{Z}$ pertencem a $\mathcal{T}$;

ii) Pela definição dos elementos de $\mathcal{T}$, a união arbitrária de subconjuntos de $\mathcal{T}$ está em $\mathcal{T}$;

iii) Se O_1 e O_2 estão em $\mathcal{T}$ então $O_1 \cap O_2$ está em $\mathcal{T}$.

- De fato, seja $a \in O_1 \cap O_2$. Existem b_1 e b_2 tais que $N_{a,b_1} \subseteq O_1$ e $N_{a,b_2} \subseteq O_2$. Logo, $N_{a,b_1 \cdot b_2} \subseteq O_1 \cap O_2$.

Argumento de Fürstenberg

- **Fato 1:** Qualquer conjunto aberto não vazio é infinito.
  - Pela própria definição dos abertos da topologia.
- **Fato 2:** $N_{a,b}$ é aberto, quaisquer que sejam $a, b \in \mathbb{Z}$ e $b > 0$.

Argumento de Fürstenberg

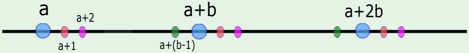
- **Fato 1:** Qualquer conjunto aberto não vazio é infinito.
  - Pela própria definição dos abertos da topologia.
- **Fato 2:** $N_{a,b}$ é aberto, quaisquer que sejam $a, b \in \mathbb{Z}$ e $b > 0$.

Conjuntos fechados

Um subconjunto A de um espaço topológico X é dito fechado se e somente se é o complementar de um aberto em X.

- **Fato 3:** $N_{a,b}$ é fechado, quaisquer que sejam a e b.

$$N_{a,b} = \mathbb{Z} \setminus \bigcup_{i=1}^{b-1} N_{a+i,b}$$



e $\bigcup_{i=1}^{b-1} N_{a+i,b}$ é aberto.

Argumento de Fürstenberg

Propriedades sobre conjuntos fechados

Se X é um espaço topológico então:

P1. $\emptyset$ e X são fechados;

P2. A união finita de fechados é fechado;

- Considere A_i, $1 \leq i \leq n$ fechados. Assim,

$$X \setminus \bigcup_{i=1}^{n} A_i = \bigcap_{i=1}^{n} (X \setminus A_i) \text{ é aberto}$$

P3. A interseção arbitrária de fechados é fechado.

- Considere A_α, família de fechados. Assim,

$$X \setminus \bigcap A_\alpha = \bigcup (X \setminus A_\alpha) \text{ é aberto}$$

Argumento de Fürstenberg

- **Fato 4:** Qualquer número inteiro k, tal que $k \neq 1$ e $k \neq -1$ tem um divisor primo. Portanto, $k \in N_{0,p}$ para algum primo p. Logo,

  $$\mathbb{Z} \setminus \{-1, 1\} = \bigcup_{p \in \mathbb{P}} N_{0,p}, \text{ em que } \mathbb{P} \text{ denota o conjunto dos números primos}$$

Se $\mathbb{P}$ é finito, então:

- $\bigcup_{p \in \mathbb{P}} N_{0,p}$ é fechado (**Fato 3 + P1**);
- Portanto, $\{-1, 1\}$ é aberto.
- De onde $\{-1, 1\}$ é infinito. (**Fato 1**)

Portanto, o conjunto $\mathbb{P}$ dos números primos é infinito.