XIV Summer Workshop in Mathematics MAT/UnB

19o Seminário Informal (+Formal!) do Grupo de Teoria da Computação da UnB

Formalizing Theorems with PVS

Section 1: Logic and Formal Deduction

Thaynara Arielly de Lima (IME)

Mauricio Ayala-Rincón (CIC-MAT)

Funded by FAPDF DE grants 00193.0000.2144/2018-81, 00193-00000229/2021-21, and CNPq Research Grants 307672/2017-4 and 313290/2021-0

Jan 17 - 21 , 2022

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

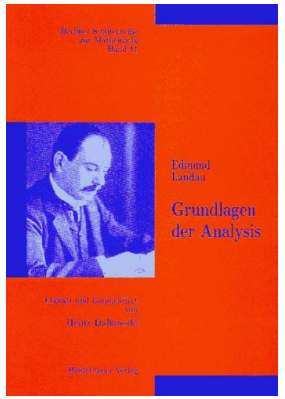
Formalizing Mathematics

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:

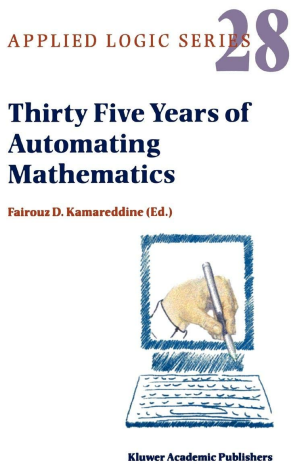
Mechanical verification of the famous Edmund Landau's (1877-1938) book *Grundlagen der Analysis*, Leipzig 1930.

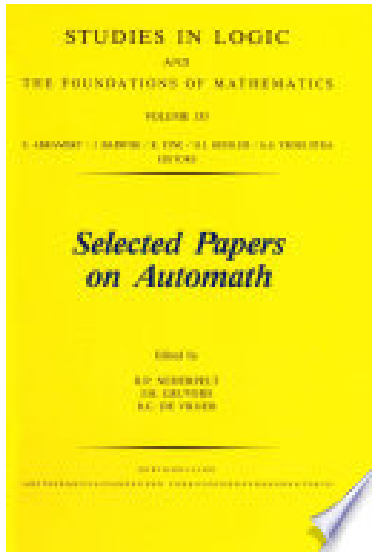


Formalizing Mathematics

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

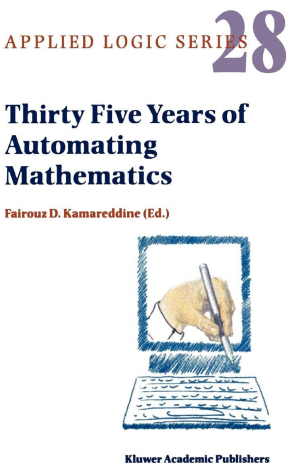
Automath is considered predecessor of modern proof assistants as: Coq, Nurpl, Isabelle, PVS ...





Formalizing Mathematics

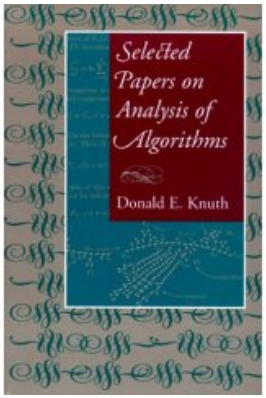
In Automath N.G. de Bruijn developed the first formalization of λ-calculus with intuitionistic types and explicit substitutions.



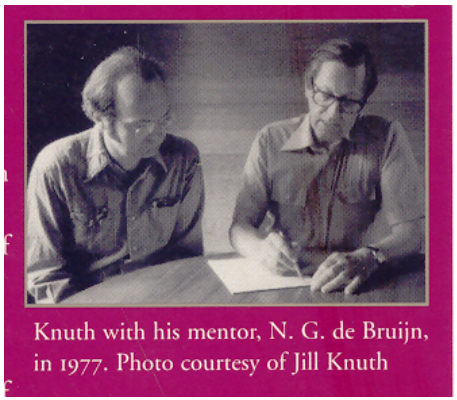
N.G. de Bruijn was a well established mathematician before deciding in 1967 at the age of 49 to work on a new direction related to Automating Mathematics. In the 1960s he became fascinated by the new computer technology and decided to start the new Automath project where he could check, with the help 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

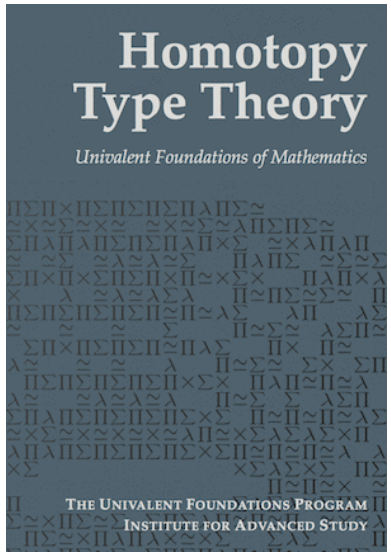
... I'm dedicating this book to N.G. "Dick" de Bruijn because his influence can be felt on every page. Ever since the 1960s he has been my chief mentor, the main person who would answer my questions when I was stuck on a problem that I had not been taught how to solve. I originally wrote Chapter 26 for 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`



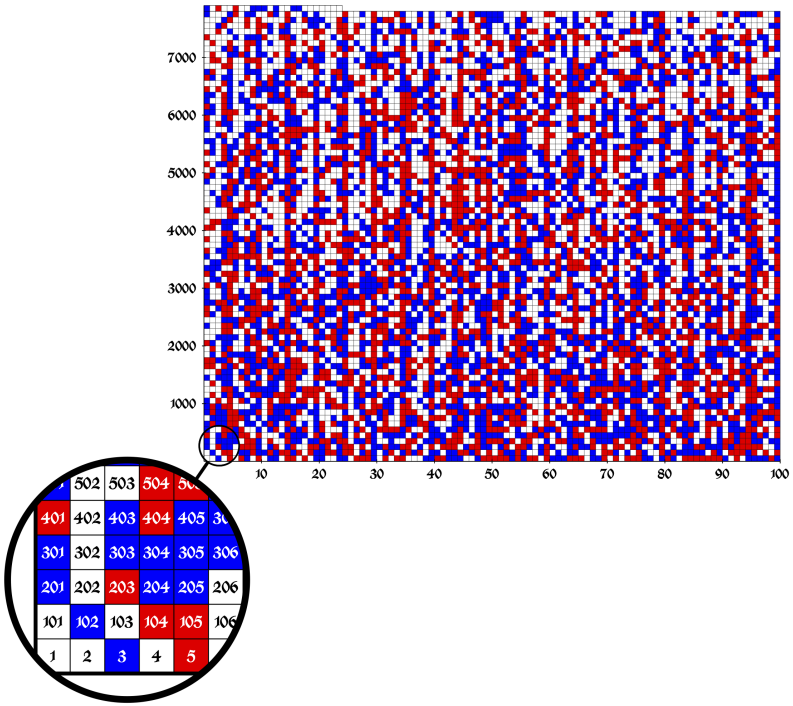
Formalizing Mathematics today

Avigad given examples are *"signs that such mechanical tools will allow a fundamental expansion of our capacities for discovering, verifying, and communicating mathematical knowledge."*

Jeremy Avigad *"The Mechanization of Mathematics"* - Notices of the ACM 2018

"The history of mathematics is a history of doing whatever it takes to extend our cognitive reach, and designing concepts and methods that augment our capacities to understand. The computer is nothing more than a tool in that respect, but it is one that fundamentally expands the range of structures we can discover and the kinds of truths we can reliably come to know."

Formalizing Mathematics today

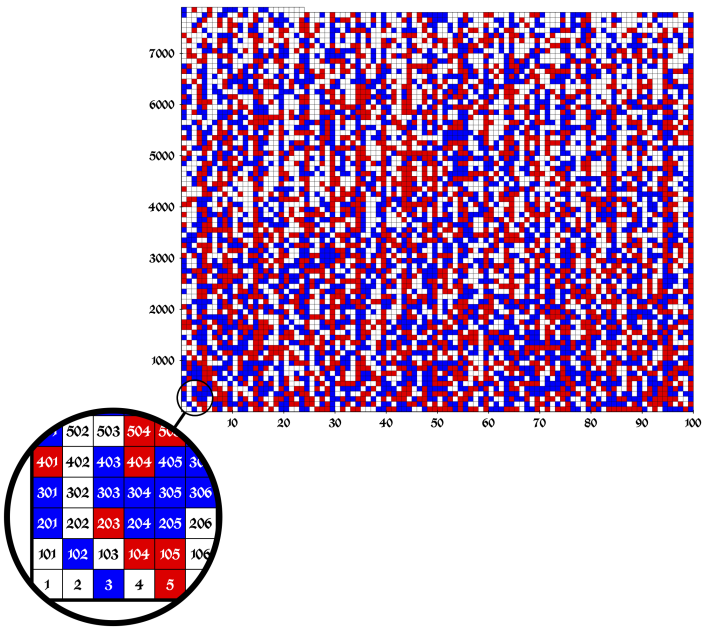


Can all positive naturals be colored red and blue, so that all *Pythagorean Triples* use different colors?

A Pythagorean triple (i, j, k) satisfies $i^2 + j^2 = k^2$, as $(3, 4, 5)$, and $(6, 8, 10)$.

Marijn Heule, Oliver Kullmann and Victor W. Marek (SAT 2016) solved the so called **Boolean Pythagorean triples problem** proving that such a coloring is only possible up to the number 7824.

Formalizing Mathematics today



Essentially, they proved mechanically that $\Phi(n)$ holds for $n \leq 7824$, but not for $\Phi(7825)$, where:

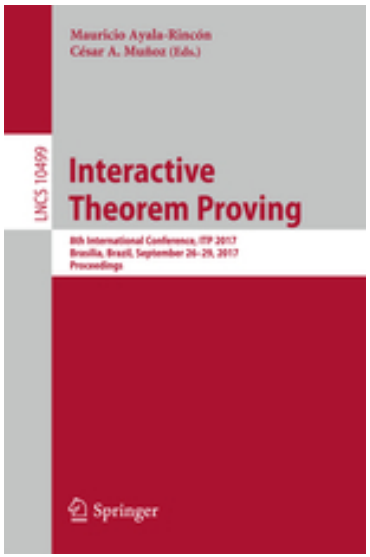
$$\Phi(n) := \bigwedge_{\substack{1 \leq i < j < k \leq n \\ i^2 + j^2 = k^2}} (x_i \vee x_j \vee x_k) \wedge (\bar{x_i} \vee \bar{x_j} \vee \bar{x_k})$$

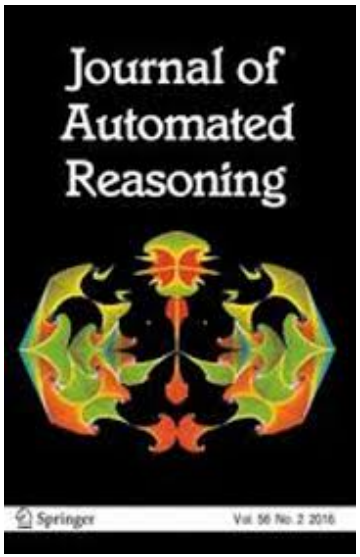
Boolean solvers may be applied to prove satisfiability of $\Phi(n)$ for $n \leq 7824$, but proving that unsatisfiability of $\Phi(7825)$ required too much effort[†]: checking that all of the 2^{7825} possible bi-partitions of the set $\{1, \ldots, 7825\}$ includes a Pythagorean triple in one of the two sets.

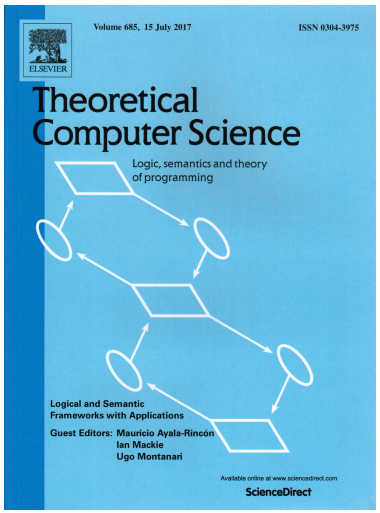
[†]Creating the proof took about 4 CPU years on a cluster with 800 cores in about 2 days. It is the largest proof ever: almost 200 terabytes in size, from which it was extracted a compressed certificate of 68 gigabytes.

Formalizing Mathematics

Some related conferences/journals:







Formalized Mathematics by GTC members

- Term Rewriting Theory: TRS NASA PVS Library
- Program Termination: PVS0 and CCG NASA PVS Library
- Nominal Equational Reasoning: Nominal PVS theory
- Group and Ring's Theories: Algebra NASA PVS Library

Formalized Mathematics by GTC members:

Term Rewriting theory - TRS trs.cic.unb.br

- Termination — Ariane Alves Almeida (PhD Informatics 2021)

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

- Confluence — André Luiz Galdino (PhD Mathematics 2008), Ana Cristina Oliveira (PhD Informatics 2016)

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

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

- Knuth-Bendix Critical Pairs — André Luiz Galdino (PhD Mathematics 2008)

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

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

  
  (2014) *"First-order unification in the PVS proof assistant"*

Formalized Mathematics by GTC members:

Program Termination Analysis - PVS0 and CCG

- Formalization of the Computational Theory of a functional language - Thiago Ramos (PhD Informatics Student), Ariane Alves Almeida (PhD Informatics 2021), Andréia Borges Avelar (PhD Math 2014) Mariano Moscato & César Muñoz, Aaron Dutle, Anthony Narkawicz (NIA / NASA LaRC FM)

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

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

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

 (In press 2022) *"Formalization of the Computational Theory of a Turing Complete Functional Language Model"*

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 Equality Check, matching, and Unification modulo C, A, and AC —
  Ana Cristina Oliveira (PhD Informatics 2016), Washington de Carvalho Segundo (PhD Informatics 2019), Gabriel Silva (PhD Informatics student).

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

 (2017) *"Nominal C-Unification"*

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

 (2019) *"A Certified Functional Nominal C-Unification Algorithm"*

 (2021) *"Formalising nominal C-unification generalised with protected variables"*

Formalized Mathematics by GTC members:

Groups and Rings algebra

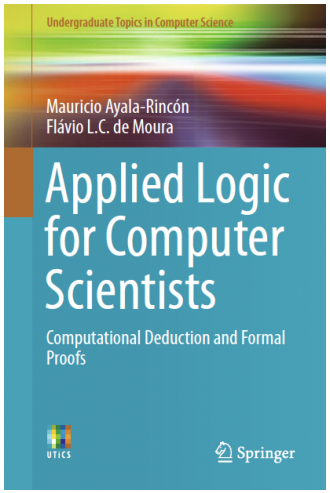
- Thaynara A. de Lima (UFG), André Galdino (UFCat), Andréia Avelar (UnB), Mauricio Ayala-Rincón (UnB)

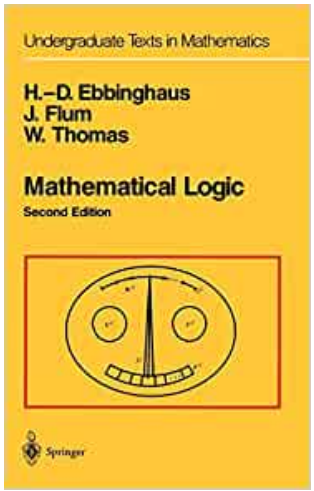

(2018) *"Formalizing Ring Theory in PVS "*

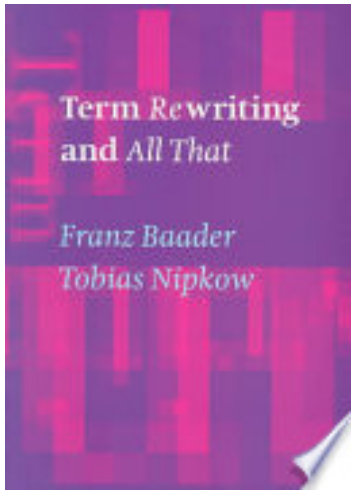

(2021) *"Formalization of Ring Theory in PVS - Isomorphism Theorems, Principal, Prime and Maximal Ideals,Chinese Remainder Theorem"*

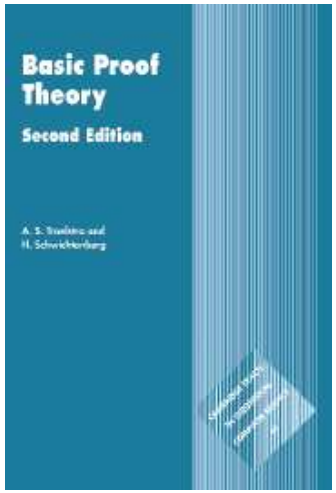
Formalizing Mathematics at GTC (PPGs in Mathematics & Informatics — UnB)

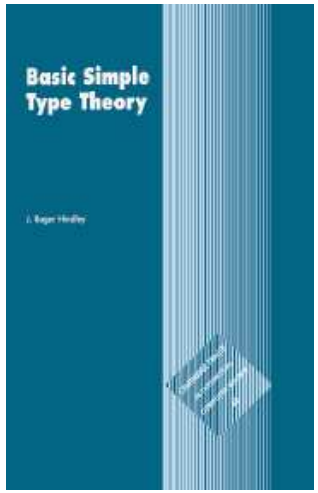
You are welcome!











Gentzen Calculus

Sequents:

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

Γ: antecedent

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

Gentzen Calculus

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

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

Gentzen Calculus

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

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

Gentzen Calculus

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

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

$$(R_{\to})\ \frac{\varphi \Rightarrow \varphi, \psi}{\Rightarrow \varphi, \varphi \to \psi}$$

Gentzen Calculus

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

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

$$(R_{\to})\ \frac{\varphi \Rightarrow \varphi, \psi}{\Rightarrow \varphi, \varphi \to \psi} \qquad \varphi \Rightarrow \varphi\ \ (Ax)$$

Gentzen Calculus

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

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

Gentzen Calculus

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

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

Gentzen Calculus

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

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

Gentzen Calculus

Cut rule:

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

Gentzen Calculus - dealing with negation: c-equivalence

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

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

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

Lemma 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.

Necessity:

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

Sufficiency:

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

$$\dfrac{\qquad\qquad\qquad\qquad}{\varphi, \Gamma \Rightarrow \Delta}\,(\textsc{Cut})$$

Gentzen Calculus - dealing with negation

Necessity:

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

$$\dfrac{\Gamma \Rightarrow \Delta, \varphi \qquad \bot, \Gamma \Rightarrow \Delta}{\Gamma \Rightarrow \Delta, \varphi}\ (\mathrm{CUT})$$

Sufficiency:

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

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:

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

Sequent Calculus in PVS

- Inference rules
  - Premises and conclusions are simultaneously constructed:

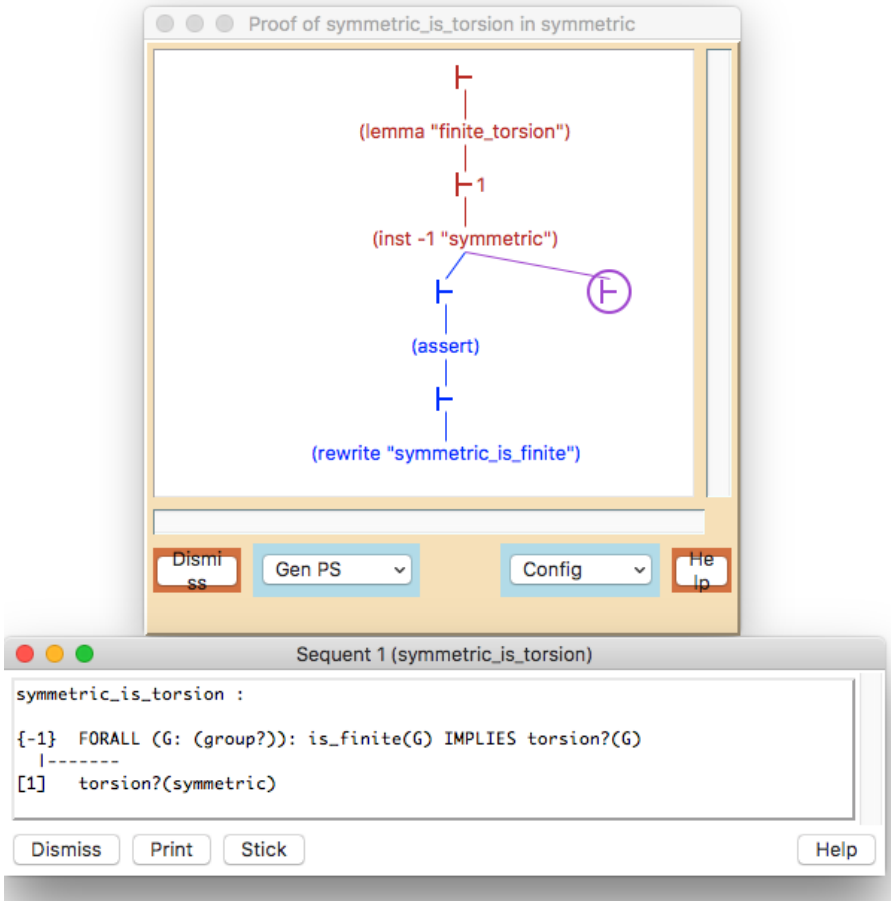
$$\frac{\Gamma \vdash \Delta}{\Gamma' \vdash \Delta'}$$

  - 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



Some inference rules in PVS

- Structural:

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

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

Some inference rules in PVS

- Structural:

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

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

Some inference rules in PVS

- Propositional:

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

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

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

$$[-1]\ A$$

$$[-2]\ B$$

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

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

$$[1]\ C$$

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

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

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

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

$$[1]\ A$$

$$\swarrow \qquad\qquad \searrow$$

$$[-1]\ A \qquad\qquad |\ - - - - - - -$$

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

$$[1]\ A \qquad\qquad [2]\ A$$

Some inference rules in PVS

- Propositional:

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

$$\begin{array}{l} |------ \\ [1]\ m \geq n \\ [2]\ \gcd(m,n) = \gcd(n,m) \end{array}$$

Some inference rules in PVS

- Propositional - semantics of PVS instructions:

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

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

Some inference rules in PVS

- Propositional (propax):

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

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

Some inference rules in PVS

- Predicate:

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

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

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

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

Summary - Gentzen Deductive Rules vs Proof Commads

Table: Logical Right Rules vs Proof Commands

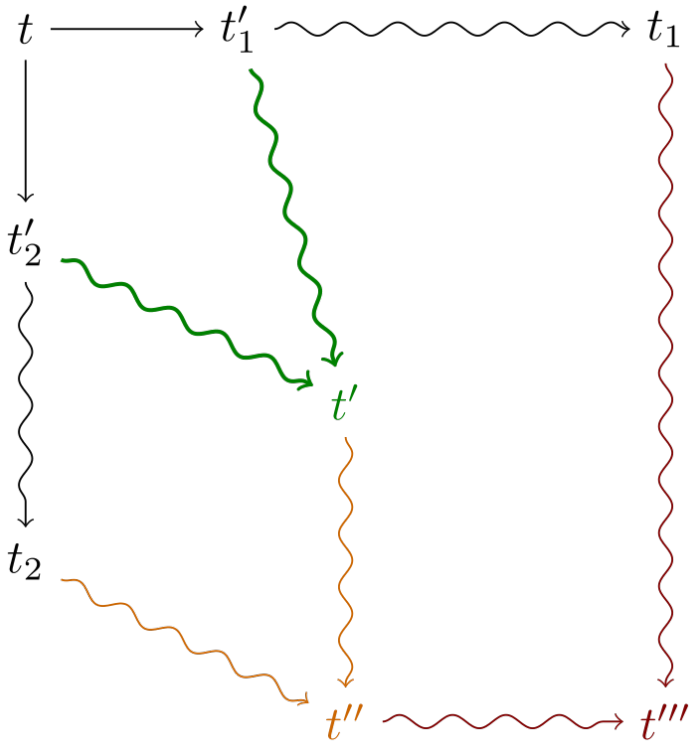
Right rules	PVS commands
$\dfrac{\Gamma \Rightarrow \Delta, \varphi \quad \Gamma \Rightarrow \Delta, \psi}{\Gamma \Rightarrow \Delta, \varphi \wedge \psi}\ (R_\wedge)$	$\dfrac{\Gamma \vdash \Delta, \varphi \wedge \psi}{\Gamma \vdash \Delta, \varphi \quad \Gamma \vdash \Delta, \psi}\ (split)$
$\dfrac{\Gamma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

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 - Newman's Lemma (Diamond Lemma)

A terminating rewriting system is confluent if it is locally confluent



See the file `predCommutation.pvs` in Exercises directory