MPRI PRFA

Proof assistants

Yannick Forster

Théo Winterhalter

Goals

Ensure you are familiar enough with one proof assistant (Rocq) so that you can

- 🧑‍🎓 use Rocq in other courses
- 👨‍💻 use Rocq in an internship
- 👀 learn other proof assistants or become an expert Rocq user via self study
- 🧙‍♂️ ultimately use or study proof assistants as part of a PhD

We also cover meta-theory, in particular dependent type theory

(more in PRFSYS)

Important information on the course webpage

https://mpri-prfa.github.io/

Organisation

8 × 2 lectures

Fridays 10:45—11:45
⚠️ on 14 Nov: we start at 9:45 ⚠️

Mondays 09:45—11:45

Self-practice during the week(end)

Teachers

Yannick Forster

Théo Winterhalter

Project

One big Rocq exercise
with files to complete
and a report to write

3h exam

you may only bring
one handwritten A4 page
(both sides)

Organisation

Mondays

30 min | 1h | 30 min

Review | Lecture + live coding | Practice

Including advanced *optional* exercises

Fridays

1h

Lecture + live coding

Practice
is essential because
proving is programming

Every part of this course tries to help you practice:

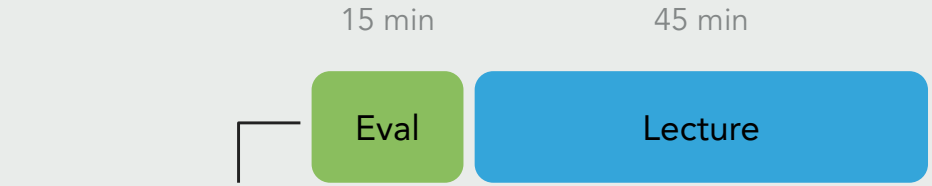
- Practical lectures with live coding
- Practice after the lecture
- Practice at home
- Practice during the project
- Self-evaluation

Organisation

Mondays

Including advanced *optional* exercises

Fridays



✍Review of the previous lesson

⚠️ on 14 Nov: we start at 9:45 ⚠️

Practice
is essential because
proving is programming

Every part of this course tries to help you practice:

- Practical lectures with live coding
- Practice after the lecture
- Practice at home
- Practice during the project
- Self-evaluation

Keeping in touch

Join the course's Discord server

send us an email to get the link

Discuss exercises, the project, other proof assistants…

We'll post about internships too!

Frequently asked questions will be added to… the FAQ

Keeping in touch

Join the course's Discord server
https://discord.gg/tPWWysfCT3

Discuss exercises, the project, other proof assistants…

We'll post about internships too!

Frequently asked questions will be added to… the FAQ

Useful resources

Rocq official website
rocq-prover.org

Links to everything you may need related to Rocq

Rocq 9.0 documentation
rocq-prover.org/doc/V9.0.0/refman/

Tactic index, command index and more…

Rocq discourse
discourse.rocq-prover.org

Forum for announcements and questions available in several languages

Rocq Zulip
rocq-prover.zulipchat.com

Chat where most of Rocq discussions happen nowadays

Outline of the course

Subject to change
The course webpage is authoritative

19, 22 Sept.	Intro.
26, 29 Sept.	Inductive types.
3, 6 Oct.	Proof terms and meta-theory.
10, 13 Oct.	Mathematical modelling. Automation.
17, 20 Oct.	Advanced elimination / induction.
24 Oct., 3 Nov.	Tactics and meta-programming.
7, 14 Nov.	Equality. ⚠️ on 14 Nov: we start at 9:45 ⚠️
17, 21 Nov.	Dependent functional programming.

Important dates

early Oct. Project handout

mid Nov. Project deadline

Nov. or Dec. Exam

Presentation round

Who are you?

Have you seen functional programming before?

Did you use a proof assistant before?

Past and planned internships?

Did you register?
framaforms.org/registration-for-mpri-course-prfa-2025-1753430092

Presentation round

Who are you?

Have you seen functional programming before?

Did you use a proof assistant before?

Past and planned internships?

Did you register?
framaforms.org/registration-for-mpri-course-prfa-2025-1753430092

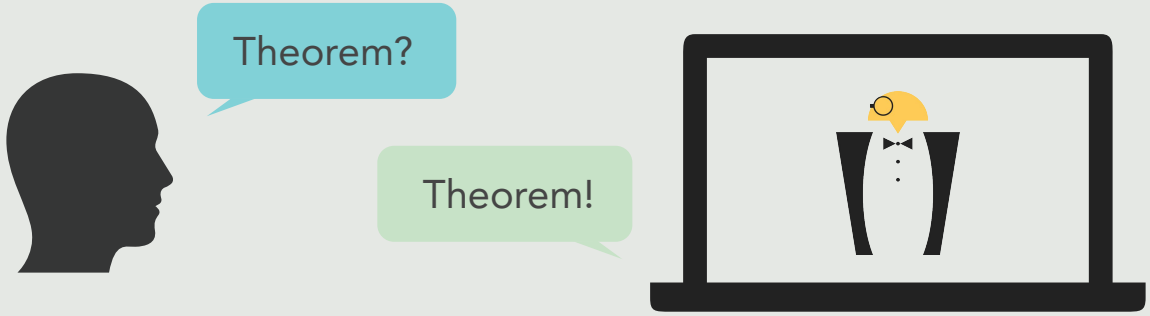
Today

Short introduction to proof assistants

Getting acquainted with Rocq

Proving things in propositional logic

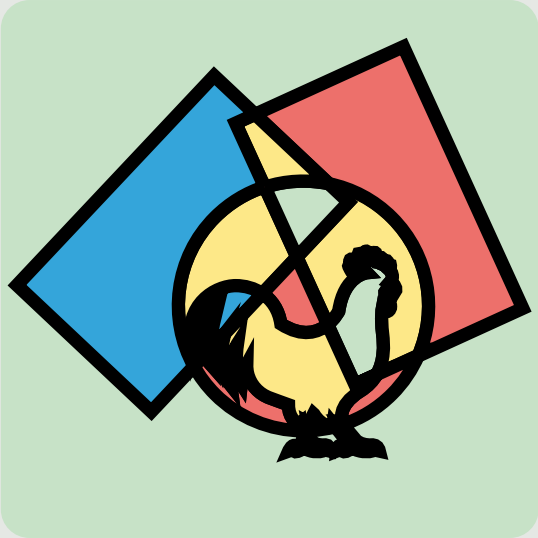
What is a proof assistant?



A piece of software for stating and proving mathematical theorems

It helps you build proofs interactively by giving you feedback,
inferring missing information, and, crucially, checking proofs!

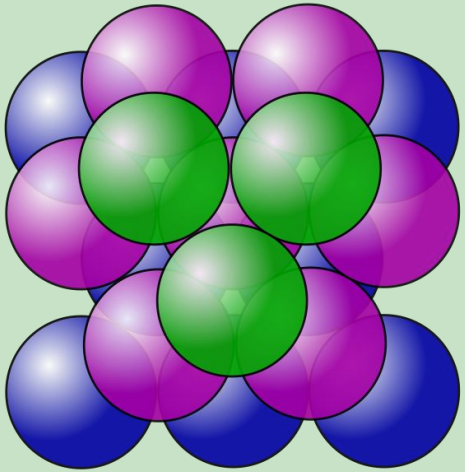
Why a proof assistant?



Mathematics

Program verification

Kepler conjecture



Source: [wikipedia]

1611 Conjecture by Johannes Kepler

1998 Proof by Tom Hales
300 pages, 400k lines of code

1999 12 reviewers, 99% certain

1999 Continued reviewing,
2002 still 99% certain

2005 Published in Annals of Mathematics,
"without complete certification from the referees"

2003 Formal proof project in Isabelle and HOL light
-2015 "Flyspeck"

2017 Published formal proof in Forum of Mathematics

Four colour theorem

1852 Conjecture by Francis Guthrie

1879 Initial proof by Alfred Kempe

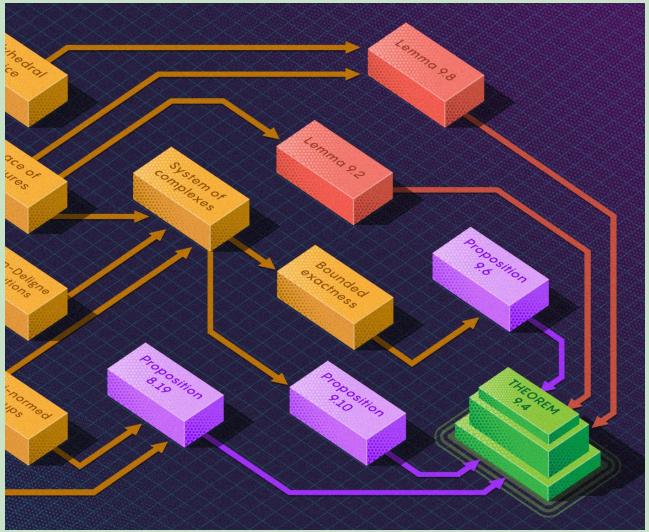
1890 Percy Heawood finds a mistake
(5 colour theorem)

1976 Proof by Kenneth Appel and Wolfgang Haken
Proof idea: find an “unavoidable”, “reducible” set of configurations
Reducibility: Checked by computer, took 2 days
Unavoidability: 400 pages of microfiche, checked manually by Appel, Haken, and Haken’s teenage daughter Dorothea Blostein

1981 Mistakes found by Master’s student but fixed

2004 Proof in Rocq by Georges Gonthier with Benjamin Werner

The liquid tensor experiment



Samuel Velasco/Quanta Magazine; Johan Commelin

https://www.quantamagazine.org/lean-computer-program-confirms-peter-scholze-proof-20210728/

2019 July: Scholze works out proof of central theorem of Scholze-Clausen liquid mathematics, mainly in his head

2019 Proof is written up, but Scholze is unsure about parts

2020 Scholze writes post on Kevin Buzzard's blog about "liquid tensor experiment", lead by Johan Commelin: A mechanisation of the proof in Lean

Working mode: Commelin works on the main proof, technical lemmas are outsourced to community via online chat

2021 May: Main argument mechanised

2022 July 14: complete proof mechanised

BusyBeaver(5)

https://www.quantamagazine.org/amateur-mathematicians-find-fifth-busy-beaver-turing-machine-20240702

https://www.lemonde.fr/sciences/article/2024/07/17/mathematiques-le-defi-du-castor-affaire-resolu_6251337_1650684.html

1962 Tibor Radó introduces “The busy beaver game”: BB(n) is the maximal number of steps a Turing machine with n states can take

1966 Allen Brady discovers 4 state machine taking 107 steps

1974 Brady proves BB(4) = 107

But there are 17 trillion possible 5-state Turing machines…

1989 Heiner Marxen and Jürgen Buntrock find 5-state machine taking 47,176,870 steps

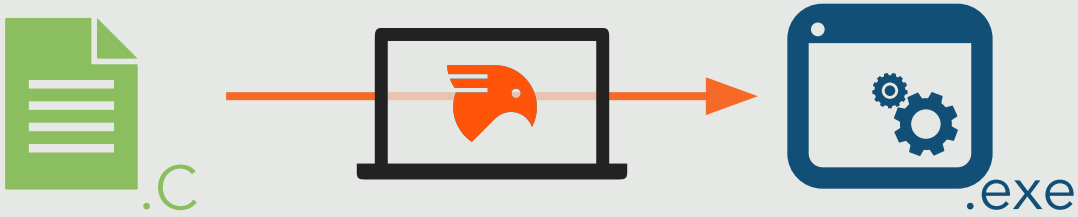
2020 Scott Aaronson conjectures BB(5) = 47,176,870

2021 Busy beaver challenge started by Tristan Stérin

2024 Rocq proof of BB(5) = 47,176,870
by anonymous contributors

Compiling 1 million random C programs will result in miscompilation unless
Optimisations are disabled
Or the compiler is formally verified
John Regehr

CompCert: Fully verified C compiler with optimisations

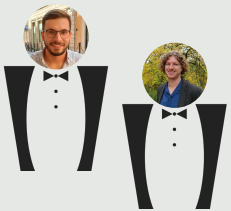


More verified software: CakeML, sel4, FiatCrypto, Google boring SSL, …

Proof assistants can help when…

… proofs are too big to be reviewed
… proofs rely on complicated computer programs
… proofs are too complicated to be trusted by their authors
… proofs are contributed by anonymous hobbyist mathematicians
… computer programs are too complicated to be trusted
… students want to develop a deeper understanding what is a proof

We can help…



… you get started learning in the lectures
… you practice at home
… you assess your current level constantly
… deepen your understanding of Rocq – advanced exercises for all levels

Proving is programming: This course will take you more time than others!

Proof assistants can help when…

… proofs are too big to be reviewed
… proofs rely on complicated computer programs
… proofs are too complicated to be trusted by their authors
… proofs are contributed by anonymous hobbyist mathematicians
… computer programs are too complicated to be trusted
… students want to develop a deeper understanding what is a proof

We can help…

… you to get started learning in the lectures
… you to practice at home
… you to assess your current level constantly
… deepen your understanding of Rocq – advanced exercises for all levels

Proving is programming: This course will take you more time than others!

Why Rocq?

All proof assistants are beautiful!

But we have 8 x 3 hours to teach you

Option 1:

Reach limited proficiency in several proof assistant without understanding concepts deeply

Goals

👩‍🎓 use PAs in other courses

👨‍💻 use a PA in an internship

👀 learn other proof assistants or become an expert PA user via self study

🧙‍♂️ ultimately use or study proof assistants as part of a PhD

Why Rocq?

All proof assistants are beautiful!

But we have 8 x 3 hours to teach you

Option 1:

Reach limited proficiency in several proof assistant without understanding concepts deeply

Option 2:

Reach very good proficiency in one proof assistant, be able to learn others on your own

Explored in the HOTT course!

Now

Live-coding with Rocq

+

Practice