Logique et Fondements de l'Informatique

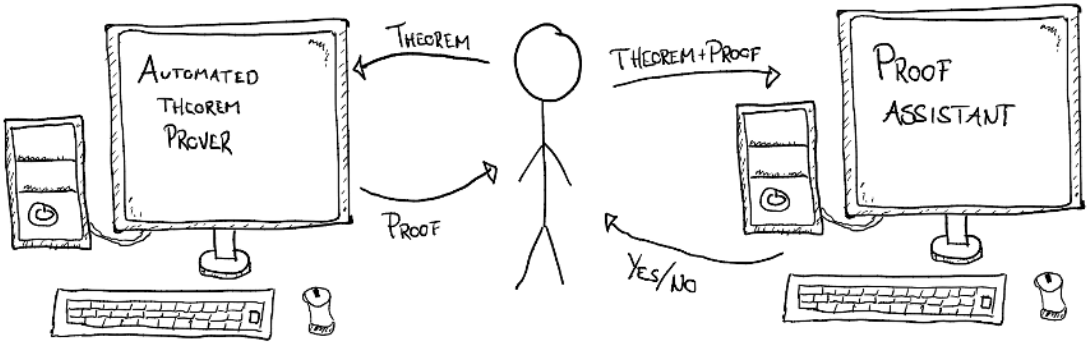
L'ère des Assistants de Preuve

Pablo Donato

Leçon 1

Qu'est-ce qu'un Assistant de Preuve ?

- L'humain fournit les idées (spécification, argumentation).
- La machine vérifie rigoureusement chaque étape de déduction.
- Aucun saut logique n'est toléré.



Une brève histoire de la mécanisation

- **1968 - Automath :** N.G. de Bruijn aux Pays-Bas. Première tentative de vérifier des mathématiques par ordinateur.
- **1972 - LCF :** Robin Milner invente le concept de "Tactiques" pour faciliter la construction à la fois automatique et interactive des preuves.
- **1984 - Calcul des Constructions :** Thierry Coquand et Gérard Huet (Inria).
- **1989 - Coq (aujourd'hui Rocq) :** Naissance du système que nous allons utiliser.
- **Aujourd'hui :** Rocq, Lean, Agda, Isabelle/HOL (et d'autres moins connus/plus spécialisés).

Le pont théorique : Curry-Howard

$$\text{Propositions} = \text{Types}$$
$$\text{Preuves} = \text{Programmes}$$

- **Théorie de la démonstration $\longleftrightarrow$ Théorie des types**
- Prouver un théorème, c'est écrire un programme (un λ-terme) qui "compile" avec le type de ce théorème.

Un raz-de-marée en Mathématiques

- **Vérification de l'état de l'art :**
  - **2005 - Théorème des Quatre Couleurs :** Preuve formelle en Rocq par Georges Gonthier [5].
  - **2014 - Conjecture de Kepler :** Projet Flyspeck (Isabelle/HOL Light) par Thomas Hales [6].
  - **2021 - Liquid Tensor Experiment :** Peter Scholze (Médaille Fields) fait vérifier son propre théorème par la communauté Lean [7].
- **Machine Learning & IA :**
  - **2024 - AlphaProof :** Médaille d'argent (avec AlphaProof) aux Olympiades Internationales de Mathématiques [8].
  - **2026 - Mathlib & Conjectures d'Erdős :** Résolution automatisée de problèmes ouverts via Lean par des startups et des mathématiciens [9].

Industrie et Méthodes Formelles

Comment s'assurer qu'un système critique (avion, métro, nucléaire) n'a **aucun** bug ? Les tests ne suffisent pas.

- **CompCert [1] :** Compilateur C certifié en Rocq (0 bug de compilation mathématiquement garanti).
- **seL4 [2] :** Noyau de système d'exploitation sécurisé et prouvé formellement dans Isabelle.
- **Ligne 14 du Métro Parisien [3] :** Développement des pilotes automatiques (projet METEOR) avec 0 bug détecté à l'aide de la Méthode B.
- **Signal & Protocole MLS [4] :** Le protocole de messagerie de groupe utilisé par Signal a été certifié par preuves formelles automatisées.

Applications en philosophie analytique

La formalisation permet de débusquer les prémisses cachées, les contradictions, et de désambiguïser les concepts :

- **Métaphysique :** Formalisation par Zalta des "objets abstraits" via la logique computationnelle dans Isabelle/HOL [10].
- **Théologie :** Vérification formelle par ordinateur des arguments ontologiques (Anselme, variante modale de Gödel) [11].
- **Éthique classique :** Tentatives de mécanisation de l'*Éthique* de Spinoza (écrite initialement *more geometrico*) en Coq [12].

Plan du Cours (6 semaines)

- **Semaine 1 :** Du λ-calcul à Rocq
- **Semaine 2 :** Logique propositionnelle, intuitionniste, classique
- **Semaine 3 :** Épistémologie des types de données
- **Semaine 4 :** Logique des prédicats et égalité
- **Semaine 5 :** Programmation fonctionnelle et types inductifs
- **Semaine 6 :** Vérification de programmes et fiabilité de l'IA

Au menu aujourd'hui : Anatomie de Rocq

Rocq est composé de **3 niveaux de langage** :

1. **Le Vernaculaire** : Les commandes pour "parler" au système (`Section`, `Definition`, `Lemma`).
2. **Gallina** : Le langage fonctionnel pur (λ-calcul) où vivent les termes et les types. L'artefact final de vérité.
3. **Ltac** : Le langage de "Tactiques" (`intros`, `apply`). Un méta-langage interactif pour aider l'humain à construire le terme Gallina étape par étape.

Bibliographie I

[1] Xavier Leroy.
Formal verification of a realistic compiler.
Communications of the ACM, 2009.
https://dl.acm.org/doi/10.1145/1538788.1538814

[2] Gerwin Klein et al.
seL4 : Formal verification of an OS kernel.
SOSP, 2009.
https://dl.acm.org/doi/10.1145/1629575.1629596

[3] P. Behm, P. Benoit, A. Faivre, and J.-M. Meynadier.
Météor : A successful application of B in a large project.
FM'99—Formal Methods, 1999.
https://link.springer.com/chapter/10.1007/3-540-48119-2_22

Bibliographie II

[4] Théophile Wallez.
Cadriciel de vérification pour la messagerie de groupe sécurisée.
Thèse de doctorat, Inria Paris, 2025.
https://theses.fr/s275565

[5] Georges Gonthier.
Formal Proof—The Four Color Theorem.
Notices of the AMS, 2008.
https://www.ams.org/notices/200811/tx081101382p.pdf

[6] Thomas Hales et al.
A Formal Proof of the Kepler Conjecture.
Forum of Mathematics, Pi, 2017.
https://doi.org/10.1017/fmp.2017.1

Bibliographie III

[7] Peter Scholze et al.
Liquid Tensor Experiment.
2021.
https://leanprover-community.github.io/liquid/
https://pat2023.icube.unistra.fr/slides/slides-pat2023-commelin-talk2.pdf

[8] Google DeepMind.
AlphaProof and AlphaGeometry 2 solve IMO problems.
2024.
https://deepmind.google/discover/blog/ai-solves-imo-problems-at-silver-medal-level/

Bibliographie IV

[9] natsothanaphan.
AI contributions to Erdős problems.
2026.
https://github.com/teorth/erdosproblems/wiki/AI-contributions-to-Erd%C5%91s-problems/

[10] Daniel Kirchner.
Representation and Partial Automation of the Principia Logico-Metaphysica in Isabelle/HOL.
Archive of Formal Proofs, 2017.
https://www.isa-afp.org/entries/PLM.html

Bibliographie V

[11] Christoph Benzmüller and Bruno Woltzenlogel Paleo.
Formalization, Mechanization and Automation of Gödel's Proof of God's Existence.
arXiv preprint, 2013.
https://arxiv.org/abs/1308.4526

[12] Luc Pommeret.
Preuves en déduction naturelle du livre I de l'Éthique de Spinoza.
Document de travail (lucpommeret.com), 2025.
https://lucpommeret.com/assets/De_Deo.pdf