

# Assistants de preuve et Apprentissage du raisonnement

---

Isabelle Dubois

20/03/2024

Laboratoire Institut Elie Cartan de Lorraine - UFR MIM - Metz - Université de Lorraine

# Histoire de commencer

---

- Enseignement depuis 7 ans d'un cours sur la logique et le raisonnement en L1 Maths/Info
- Rapide découverte de Coq lors d'une initiation proposée par un collègue informaticien
- Recherche, lecture de témoignages d'utilisation d'un assistant de preuve, en particulier à l'étranger

*Un assistant de preuve pour apprendre à faire des preuves ?*

# Emballement ! Rencontres et expérimentations

- Lecture de l'article de la gazette de la SMF paru en octobre 2022 :  
[https://smf.emath.fr/system/files/filepdf/G174-BD.pdf#section\\*.148](https://smf.emath.fr/system/files/filepdf/G174-BD.pdf#section*.148)
- Congrès PAT Juin 2023 :  
<https://pat2023.icube.unistra.fr/>
- Projet de recherche APPAM :  
<https://appam.icube.unistra.fr/>
- Premiers tests personnels avec une stagiaire troisième, quelques étudiants du parcours PPPE ; observation et appui pédagogique lors de 2 séances de TPs à Paris 13 Villetaneuse.

*Toute une communauté d'enseignants et de chercheurs croit aux potentiels de l'utilisation d'un assistant de preuve dans l'enseignement !*

# Logiciel Assistant de preuve

---

Qu'est-ce qu'un logiciel assistant de preuve ?

## Qu'est-ce ? (Attention : je ne suis pas une experte)

Un logiciel assistant de preuve est un logiciel permettant de **vérifier l'exactitude de preuves mathématiques formelles**, portant sur des assertions mathématiques modélisées dans le logiciel.

C'est en général un langage de programmation spécifique, dont la validité a elle-même été vérifiée (jusqu'à un certain point). A noter qu'un tel logiciel est également employé pour la vérification de programmes informatiques.

Les plus connus, utilisés, et libres :

- **Coq** (1989 - développé en O'Caml, essentiellement par l'INRIA)
- **Isabelle** (1986 - développé en Standard ML et Scala)
- **Lean** (2013 - développé en C++), emparé par les mathématiciens chercheurs depuis quelques années.

Voici quelques résultats mathématiques certifiés formellement :

- Théorème des 4 couleurs ("coloriage de cartes"), certifié en 2004 par G. Gonthier et B. Werner en utilisant Coq.
- Conjecture de Képler/Théorème de Hales ("empilement optimal de sphères"), certifié de 2004 à 2014 par le projet Flyspeck, en utilisant HOL/Light, Coq, Isabelle.
- Théorème de Feit-Thomson ("Tout groupe d'ordre impair est résoluble"), équipe de G. Gonthier de 2006 à 2012 avec Coq.
- Problème d'Erdős-Graham ("sous-ensemble d'entiers et fractions égyptiennes") démontré par T.F. Bloom en 2021 et vérifié en 2023 en Lean. [Lien vers l'article.](#)

Remarque : Coq a été utilisé dans d'autres contextes : certification d'une plateforme de cartes à puce, développement d'un compilateur certifié pour des programmes C.

# Limitations

Tout n'est pas simple :

- Un assistant de preuve ne trouve pas une démonstration par lui-même, il faut l'assister et lui dire quoi faire. Il vérifie ensuite la validité de la démarche.

Toutefois, il y a certaines phases d'automatisation.

- Langage spécifique, parfois éloigné des pratiques mathématiques. Non lisibilité d'une preuve formelle.
- Certaines parties des mathématiques sont difficiles à implémenter. On doit les adapter.
- Problèmes des fondements et/ou philosophiques : Pertinence des modèles utilisés : logique, théorie des types, théorie des ensembles, isomorphisme de Curry-Howard... ?  
Sont-ce les mêmes mathématiques ? Comprend-on mieux les démonstrations ?

# Enseignement de la démonstration

---

Que peut apporter un logiciel assistant de  
preuve dans l'enseignement ?

Dans un contexte d'apprentissage de la logique et du raisonnement, et de l'appropriation des concepts mathématiques :

- **Interactivité et rétro-action** d'un assistant de preuve.  
Démonstration construite **pas à pas**, en laissant éventuellement des trous que l'on admet.
- Encourage **l'autonomie** et la **la prise d'initiative** de l'apprenant.
- Apprentissage et mise en valeur de la nécessité du **langage formel** des mathématiques, et de la **logique**.
- Apprentissage de la **structure d'une démonstration**, notamment en lien avec l'énoncé à démontrer.
- Acquisition **d'automatismes**, notamment pour démarrer une démonstration.
- Affichage et évolution du **contexte de la preuve** : état de preuve, objets mathématiques en jeu, hypothèses faites...

# Limitations

Bien sûr, des écueils ont déjà été pointés dans leur usage en enseignement :

- Focalisation sur des détails inintéressants
- Formalisation trop poussée
- Perte des idées principales, perte de sens
- Construire des automatismes sans les comprendre
- Passage de la machine au papier/crayon pouvant être difficile

Etudes didactiques en cours !

## Exemple en enseignement du raisonnement

Frédéric Le Roux, enseignant-chercheur à Sorbonne Université (Jussieu), enseigne une UE "Raisonnement mathématique avec le logiciel Deaduction" en L1 Mathématiques.

Voici un exemple d'activité qu'il propose, articulant apport du logiciel/analyse d'une démonstration papier :

Fichier

# Présentation de quelques assistants de preuve - Exemples d'exercices

---

Créé par une start-up française, basé sur Coq.

Interface Point and Click, plus vraiment mis à jour, non libre, quelques erreurs.

Pour l'enseignement secondaire et le début du supérieur.

Site : <https://edukera.com/>

Exemple : Passage du langage naturel à sa formalisation,

"Tout le monde n'a pas lu Les Misérables"

Libre.

Interface jsCoq intéressante pour proposer un cours/TP sur un même fichier.

Complicé à installer, et à utiliser (notamment en tant que créateur de cours).

Page de P. Rousselin, TPs de L1 Math/Info :

<https://www.math.univ-paris13.fr/~rousselin/ipf.html>

Exemple : Fichier du premier TP, Logique naturelle

Libre, développé par une communauté de mathématiciens. A installer pour du LEAN pur et dur. Existe une version en ligne.

Initiatives pour une utilisation en situation d'apprentissage :

1. En ligne : Natural Number Game, plutôt pour apprendre le langage Lean. Site : <https://adam.math.hhu.de/>
2. A installer, ou en ligne : exercices pour la L1 utilisant "Lean Verbose", créé par P. Massot  
Lien Page P. Massot  
Version en ligne sous GitPod : [Lien](#)
3. Logiciel DEDUCTION, Interface Point and Click créée par F. Le Roux : [Lien Page](#)

Toujours en cours de développement.

Facile d'utilisation, pour l'apprenant et pour l'enseignant.

Possibilité de créer ses propres feuilles d'exercices facilement.

Exemple : Des fichiers tutoriel que j'ai créés. Version Deaduction v0.39

A vous de jouer !