Pourrait-on vérifier toutes les mathématiques sur ordinateur ?

Riccardo Brasca

Conférence *Des preuves et des programmes*

1er décembre 2023

Introduction

On utilise des ordinateurs en mathématique depuis des décennies

Introduction

On utilise des ordinateurs en mathématique depuis des décennies

Considérons par exemple la conjecture suivante

Conjecture (Goldbach)

Tout nombre pair $n > 2$ est la somme de deux nombres premiers.

Introduction

On utilise des ordinateurs en mathématique depuis des décennies

Considérons par exemple la conjecture suivante

Conjecture (Goldbach)

Tout nombre pair $n > 2$ est la somme de deux nombres premiers.

On peut utiliser l'ordinateur pour tester cette conjecture et faire des mathématiques « expérimentales »

Introduction

On utilise des ordinateurs en mathématique depuis des décennies

Considérons par exemple la conjecture suivante

Conjecture (Goldbach)

Tout nombre pair $n > 2$ est la somme de deux nombres premiers.

On peut utiliser l'ordinateur pour tester cette conjecture et faire des mathématiques « expérimentales »

Peut-on *démontrer* la conjecture avec un ordinateur ?

Introduction

On utilise des ordinateurs en mathématique depuis des décennies

Considérons par exemple la conjecture suivante

Conjecture (Goldbach)

Tout nombre pair $n > 2$ est la somme de deux nombres premiers.

On peut utiliser l'ordinateur pour tester cette conjecture et faire des mathématiques « expérimentales »

Peut-on *démontrer* la conjecture avec un ordinateur ?

On peut aujourd'hui faire des experiences *autour des preuves*

Introduction

On utilise des ordinateurs en mathématique depuis des décennies

Considérons par exemple la conjecture suivante

Conjecture (Goldbach)

Tout nombre pair $n > 2$ est la somme de deux nombres premiers.

On peut utiliser l'ordinateur pour tester cette conjecture et faire des mathématiques « expérimentales »

Peut-on *démontrer* la conjecture avec un ordinateur ?

On peut aujourd'hui faire des experiences *autour des preuves*

Les « laboratoires » pour ces experiences sont les *assistants de preuves*

Les assistants de preuves

Un assistant de preuve est un logiciel qui est capable de comprendre, et de vérifier, des démonstrations mathématiques

Les assistants de preuves

Un assistant de preuve est un logiciel qui est capable de comprendre, et de vérifier, des démonstrations mathématiques

Il ne s'agit pas d'utiliser l'ordinateur pour faire des calculs

Les assistants de preuves

Un assistant de preuve est un logiciel qui est capable de comprendre, et de vérifier, des démonstrations mathématiques

Il ne s'agit pas d'utiliser l'ordinateur pour faire des calculs, mais pour *raisonner*

Les assistants de preuves

Un assistant de preuve est un logiciel qui est capable de comprendre, et de vérifier, des démonstrations mathématiques

Il ne s'agit pas d'utiliser l'ordinateur pour faire des calculs, mais pour *raisonner*

Il existe plusieurs assistants de preuve

Les assistants de preuves

Un assistant de preuve est un logiciel qui est capable de comprendre, et de vérifier, des démonstrations mathématiques

Il ne s'agit pas d'utiliser l'ordinateur pour faire des calculs, mais pour *raisonner*

Il existe plusieurs assistants de preuve, avec des bibliothèques de théorèmes plus ou moins développées

Les assistants de preuves

Un assistant de preuve est un logiciel qui est capable de comprendre, et de vérifier, des démonstrations mathématiques

Il ne s'agit pas d'utiliser l'ordinateur pour faire des calculs, mais pour *raisonner*

Il existe plusieurs assistants de preuve, avec des bibliothèques de théorèmes plus ou moins développées

On va voir dans la pratique un assistant de preuve, *Lean*, en utilisant aussi sa bibliothèque mathématique *Mathlib*

L'assistant de preuve s'occupe de plusieurs aspects de la formalisation :

L'assistant de preuve s'occupe de plusieurs aspects de la formalisation :

- il traduit ce qu'on écrit en un langage totalement précis

L'assistant de preuve s'occupe de plusieurs aspects de la formalisation :

- il traduit ce qu'on écrit en un langage totalement précis
- il peut effectuer automatiquement certaines tâches

L'assistant de preuve s'occupe de plusieurs aspects de la formalisation :

- il traduit ce qu'on écrit en un langage totalement précis
- il peut effectuer automatiquement certaines tâches
- il vérifie l'exactitude des démonstrations, en partant des axiomes

Les deux premières parties sont très complexes

Les deux premières parties sont très complexes, le but étant d'écrire les maths comme « au tableau »

Les deux premières parties sont très complexes, le but étant d'écrire les maths comme « au tableau »

La vérification est faite par le *noyau* de l'assistant de preuve

Les deux premières parties sont très complexes, le but étant d'écrire les maths comme « au tableau »

La vérification est faite par le *noyau* de l'assistant de preuve

Il s'agit d'un logiciel simple

Les deux premières parties sont très complexes, le but étant d'écrire les maths comme « au tableau »

La vérification est faite par le *noyau* de l'assistant de preuve

Il s'agit d'un logiciel simple, facile à vérifier (par des humains)

Les deux premières parties sont très complexes, le but étant d'écrire les maths comme « au tableau »

La vérification est faite par le *noyau* de l'assistant de preuve

Il s'agit d'un logiciel simple, facile à vérifier (par des humains) et il y en existe plusieurs versions indépendantes

Les deux premières parties sont très complexes, le but étant d'écrire les maths comme « au tableau »

La vérification est faite par le *noyau* de l'assistant de preuve

Il s'agit d'un logiciel simple, facile à vérifier (par des humains) et il y en existe plusieurs versions indépendantes

Une erreur dans le kernel est très improbable

Un exemple

Démontrons qu'il y a une infinité de nombres premiers en utilisant des résultats de *Mathlib*

Un exemple

Démontrons qu'il y a une infinité de nombres premiers en utilisant des résultats de *Mathlib*

Théorème

Pour tout $n \in \mathbb{N}$ il existe $p \in \mathbb{N}$ tel que p est premier et $p > n$.

Un exemple

Démontrons qu'il y a une infinité de nombres premiers en utilisant des résultats de *Mathlib*

Théorème

Pour tout $n \in \mathbb{N}$ il existe $p \in \mathbb{N}$ tel que p est premier et $p > n$.

Démonstration.

Soit n un naturel. On pose p égal à un facteur premier de $n! + 1$. Montrons que p marche.

- Par définition p est premier.
- Supposons $p \leq n$. On a que p divise $n!$ et donc il divise 1, ce qui est absurde.

Pourquoi formaliser

Formaliser permet de vérifier l'exactitude d'une démonstration

Pourquoi formaliser

Formaliser permet de vérifier l'exactitude d'une démonstration

La formalisation est un processus stimulant

Pourquoi formaliser

Formaliser permet de vérifier l'exactitude d'une démonstration

La formalisation est un processus stimulant. On est amené à repenser certains concepts mathématiques fondamentaux

Pourquoi formaliser

Formaliser permet de vérifier l'exactitude d'une démonstration

La formalisation est un processus stimulant. On est amené à repenser certains concepts mathématiques fondamentaux

Le lecteur (et pas l'auteur) décide le niveau de détails qu'il veut voir

Pourquoi formaliser

Formaliser permet de vérifier l'exactitude d'une démonstration

La formalisation est un processus stimulant. On est amené à repenser certains concepts mathématiques fondamentaux

Le lecteur (et pas l'auteur) décide le niveau de détails qu'il veut voir

La formalisation peut nous donner une meilleure compréhension de certains concepts mathématiques

Le *Liquid Tensor Experiment*

Le *Liquid Tensor Experiment*

Il y a des démonstration trop longues même pour les experts

Le *Liquid Tensor Experiment*

Il y a des démonstration trop longues même pour les experts
En décembre 2020 Peter Scholze à lancé un défi : la formalisation du théorème suivant

Le *Liquid Tensor Experiment*

Il y a des démonstration trop longues même pour les experts
En décembre 2020 Peter Scholze à lancé un défi : la formalisation du théorème suivant

Théorème (Clausen-Scholze)

Soient $0 < p' < p \leq 1$ *des nombres réels,* S *un ensemble profini et* V *un p-espace de Banach réel. Alors*

$$\mathrm{Ext}^1_{\mathrm{Cond(Ab)}}(\mathcal{M}_{p'}(S), V) = 0.$$

Le *Liquid Tensor Experiment*

Il y a des démonstration trop longues même pour les experts
En décembre 2020 Peter Scholze à lancé un défi : la formalisation du théorème suivant

Théorème (Clausen-Scholze)

Soient $0 < p' < p \leq 1$ *des nombres réels,* S *un ensemble profini et* V *un p-espace de Banach réel. Alors*

$$\mathrm{Ext}^1_{\mathrm{Cond(Ab)}}(\mathcal{M}_{p'}(S), V) = 0.$$

Sans rentrer dans les détails, il s'agit d'un théorème récent et important, dont la démonstration est très difficile

Voici Scholze sur le blog *Xena project* (le blog de Kevin Buzzard) :

Voici Scholze sur le blog *Xena project* (le blog de Kevin Buzzard) :

Why do I want a formalization?

- *... I think the theorem is of utmost foundational importance, so being 99.9 % sure is not enough*
- *... As it will be used as a black box, a mistake in this proof could remain uncaught*
- *... In the end, we were able to get an argument pinned down on paper, but I think nobody else has dared to look at the details of this, and so I still have some small lingering doubts*
- *... It is the kind of argument that needs to be closely inspected*
- *While I was very happy to see many study groups on condensed mathematics throughout the world, to my knowledge all of them have stopped short of this proof. (Yes, this proof is not much fun...)*

- *From what I hear, it sounds like the goal is not completely out of reach. ... If achieved, it would be a strong signal that a computer verification of current research in very abstract mathematics has become possible. I'll certainly be excited to watch any progress*
- *I think this may be my most important theorem to date*
- *I didn't think I'd have the mental capacity to rebuild this in my head again*

- *From what I hear, it sounds like the goal is not completely out of reach. ... If achieved, it would be a strong signal that a computer verification of current research in very abstract mathematics has become possible. I'll certainly be excited to watch any progress*
- *I think this may be my most important theorem to date*
- *I didn't think I'd have the mental capacity to rebuild this in my head again*

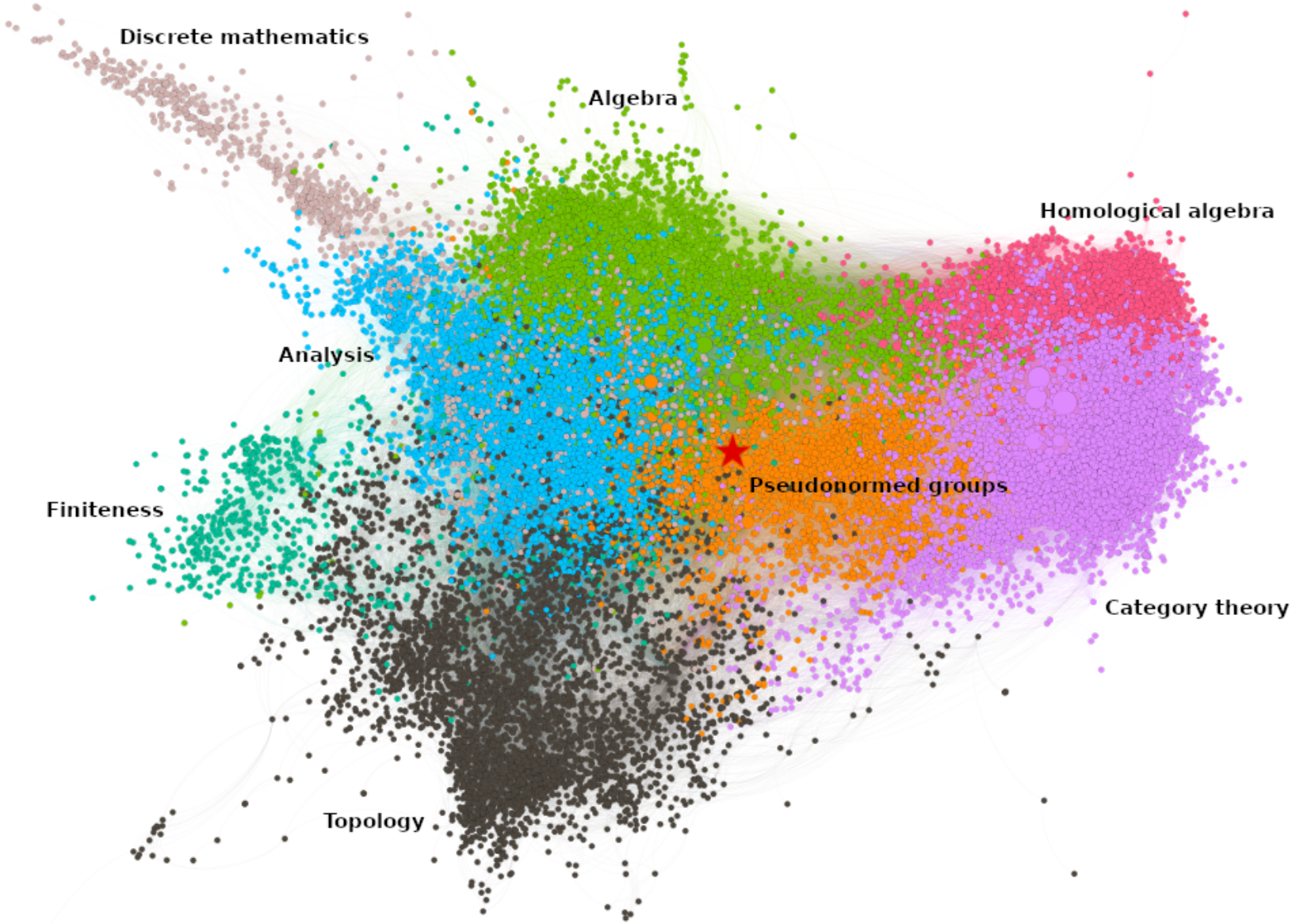
En juillet 2022 on a terminé la formalisation du théorème

- *... I am excited to announce that the Experiment has verified the entire part of the argument that I was unsure about. I find it absolutely insane that interactive proof assistants are now at the level that within a very reasonable time span they can formally verify difficult original research*

- *... I am excited to announce that the Experiment has verified the entire part of the argument that I was unsure about. I find it absolutely insane that interactive proof assistants are now at the level that within a very reasonable time span they can formally verify difficult original research*

- *... When I wrote the blog post [...], I did not understand why the argument worked*

- *... I am excited to announce that the Experiment has verified the entire part of the argument that I was unsure about. I find it absolutely insane that interactive proof assistants are now at the level that within a very reasonable time span they can formally verify difficult original research*
- *... When I wrote the blog post [...], I did not understand why the argument worked*
- *The Lean Proof Assistant was really that: An assistant in navigating through the thick jungle that this proof is. Really, one key problem I had when I was trying to find this proof was that I was essentially unable to keep all the objects in my "RAM" ... So I think here we have witnessed an experiment where the proof assistant has actually assisted in understanding the proof*


Discrete mathematics
Algebra
Homological algebra
Analysis
Pseudonormed groups
Finiteness
Category theory
Topology

Merci pour l'attention !

Si vous voulez jouer avec *Lean*, essayez le *natural number game* !

https://adam.math.hhu.de/