Leçon 6 : Vérification formelle à l'ère de l'IA

Promesses, retournements, questions ouvertes

Pablo Donato

Logique et Fondements de l'Informatique

Année 2026

Du rêve de Leibniz à la vérification formelle

Leibniz, XVII^{e} : *characteristica universalis* + *calculus ratiocinator*

Réalisations :

- seL4 [Kle+09] — noyau OS prouvé
- CompCert [Ler09] — compilateur C certifié
- 4 Couleurs [Gon08], Kepler [Hal+17]

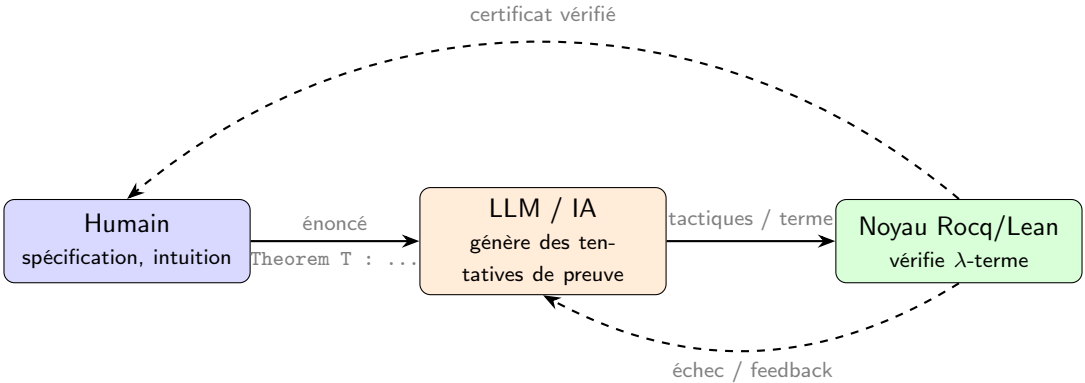
Mais : 20 années-personnes pour seL4, 6 pour CompCert.

Le pari

Garanties *mathématiques* là où l'ingénierie classique n'en offre pas.

Benzmüller [Ben17] : logique d'ordre supérieur comme *meta-logique universelle*.

La nouvelle division du travail : l'IA génère, le noyau vérifie



- **Autoformalisation** : informel → Rocq/Lean [Buz25]
- **Tactic search** : AlphaProof [Dee24]
- **L'IA propose, le noyau dispose**

Mathématiques formalisées : l'appétit de l'IA

Avigad [Avi23] — *tournant formel*

Buzzard [Buz25 ; 25] — après le calcul, le *raisonnement* ?

Jalons :

- Liquid Tensor [al21] / Mathlib
- AlphaProof, IMO 2024 [Dee24]
- **2026** : problèmes d'Erdős [nat26] résolus par Aristotle [Ari26], Aletheia [Goo26]

Severini [Sev26]

$\approx 0{,}05\%$ des maths formalisées.
L'IA a faim de corpus vérifié : la formalisation devient *application phare*.

Massot [Mas21] : outil, pas substitut.

Au-delà des maths : sûreté de l'IA par la preuve

Tegmark & Omohundro [TO23]

Provably Safe Systems — seule voie vers une AGI contrôlable.

Au-delà des maths : sûreté de l'IA par la preuve

Tegmark & Omohundro [TO23]

Provably Safe Systems — seule voie vers une AGI contrôlable.

1. Alignement (RLHF, monitoring) $\rightarrow$ garanties *statistiques*
2. Preuve formelle $\rightarrow$ garantie *mathématique*
3. L'IA rend l'approche *faisable* à grande échelle

Au-delà des maths : sûreté de l'IA par la preuve

Tegmark & Omohundro [TO23]

Provably Safe Systems — seule voie vers une AGI contrôlable.

1. Alignement (RLHF, monitoring) $\rightarrow$ garanties *statistiques*
2. Preuve formelle $\rightarrow$ garantie *mathématique*
3. L'IA rend l'approche *faisable* à grande échelle

La promesse ultime

Vérification + IA = sûreté des systèmes critiques, *y compris* des IA.

Qui vérifie le noyau ?

Si on fonde la sûreté sur la preuve, tout repose sur le vérificateur — commençons par lui.

Critère de de Bruijn [Bru80]

Noyau *minimal* : vérificateur de λ-termes, auditable par un humain.

- Pas de confiance dans les tactiques
- Confiance dans le **noyau** ($\sim$ quelques milliers de lignes)
- Noyau correct $\Rightarrow$ toute preuve acceptée est correcte

Qui vérifie le noyau ?

Si on fonde la sûreté sur la preuve, tout repose sur le vérificateur — commençons par lui.

Critère de de Bruijn [Bru80]

Noyau *minimal* : vérificateur de λ-termes, auditable par un humain.

- Pas de confiance dans les tactiques
- Confiance dans le **noyau** ($\sim$ quelques milliers de lignes)
- Noyau correct $\Rightarrow$ toute preuve acceptée est correcte

Régression

Noyau correct ? Compilateur du noyau ? Matériel ?

Thompson 1984 — *Reflections on Trusting Trust*

Ken Thompson, Turing Award [Tho84].

Compilateur C malveillant :

1. porte dérobée injectée dans `login`
2. s'auto-reproduit à la compilation d'un compilateur

Source *propre* $\rightarrow$ binaire *compromis*.

Thèse

« You can't trust code that you did not totally create yourself. »

Conséquence

La vérification *déplace* la confiance — elle ne l'élimine pas.

Le retournement 2026 : l'IA casse le cadre

Mythos [Ant26]

Anthropic, avril 2026

- 0-days dans OpenBSD, Linux, FreeBSD
- Milliers de vulnérabilités
- Source reconstruite depuis binaires

L'IA comme attaquant de classe nouvelle.

`False` dans Rocq [Roc26 ; Sté26]

Mars 2026

- Sept preuves de `False` via bugs indépendants
- Guard checker, modules
- IA + humain, avec l'équipe Rocq

Le noyau minimal n'est pas infaillible.

Watchers [Gop26]

`lean-zip`, **avril 2026**

- Décodeur ZIP prouvé en Lean 4
- Parseur non vérifié : overflow
- Un 2^{e} bug hors-spec
- Trouvés par Claude Code

Preuve OK, runtime cassé, spec incomplète.

Trois niveaux, trois fractures — toutes mises au jour par l'IA.

Trois niveaux de confiance, ré-examinés

1. Spécification — *Formaliser les valeurs ?*

2. Noyau — *`False` (Opus 4.6) [Roc26]*

3. Runtime / compilateur / matériel — *Overflow Lean [Gop26]*

À retenir

Pas de garantie absolue. Garanties *relatives* à des hypothèses explicites.

Bilan — six semaines

Savoir-faire :

- Habiter un type
- Déduction naturelle
- Rocq : `Inductive`, `Fixpoint`, `induction`

Philosophiquement :

- **Curry-Howard** : preuves = programmes
- **Constructivisme**
- **Vérificationnisme**
- **Mécanisation** du travail épistémique
- **Confiance** déplacée, pas supprimée

Pour la suite

DM — rev(rev(l)) = l — **Deadline 30 avril**.
Calculemus, mais lucidement.

Bibliographie I

[25] *Professor Kevin Buzzard : What Is Formalization and Why Does It Matter ?* 1er oct. 2025. URL : https://www.youtube.com/watch?v=0fy8VURvwac (visité le 17/03/2026).

[al21] Peter Scholze et al. "Liquid Tensor Experiment". In : (2021). url : https://leanprover-community.github.io/liquid/.

[Ant26] Anthropic. *Claude Mythos Preview*. 2026. url : https://red.anthropic.com/2026/mythos-preview/ (visité le 16/04/2026).

[Ari26] Aristotle team (Harmonic) and Neel Somani. *Resolution of Erdős Problem #728 : a writeup of Aristotle's Lean proof*. 2026. arXiv : 2601.07421. url : https://arxiv.org/abs/2601.07421 (visité le 21/04/2026).

[Avi23] Jeremy Avigad. *Mathematics and the Formal Turn*. 4 nov. 2023. doi : 10.48550/arXiv.2311.00007. arXiv : 2311.00007 [math]. url : http://arxiv.org/abs/2311.00007 (visité le 17/03/2026). Prépubl.

Bibliographie II

[Ben17] Christoph Benzmüller. *Universal Reasoning, Rational Argumentation and Human-Machine Interaction*. 28 mars 2017. doi : 10.48550/arXiv.1703.09620. arXiv : 1703.09620 [cs]. url : http://arxiv.org/abs/1703.09620 (visité le 17/03/2026). Prépubl.

[Bru80] N. G. de Bruijn. "The Mathematical Language Automath, Its Usage, and Some of Its Extensions". In : *Synthese* 43.1 (1980), p. 1-29. url : https://doi.org/10.1007/BF00485013.

[Buz25] Kevin Buzzard. *Mathematical Reasoning and the Computer*. 11 fév. 2025. doi : 10.48550/arXiv.2502.07850. arXiv : 2502.07850 [cs]. url : http://arxiv.org/abs/2502.07850 (visité le 17/03/2026). Prépubl.

[Dee24] Google DeepMind. "AlphaProof and AlphaGeometry 2 solve IMO problems". In : (2024). url : https://deepmind.google/discover/blog/ai-solves-imo-problems-at-silver-medal-level/.

Bibliographie III

[Gon08] Georges Gonthier. "Formal Proof—The Four Color Theorem". In : *Notices of the AMS* (2008). url : https://www.ams.org/notices/200811/tx081101382p.pdf.

[Goo26] Google DeepMind. *Aletheia : Autonomous Mathematical Discovery with Gemini 3 Deep Think*. 2026. url : https://deepmind.google/blog/accelerating-mathematical-and-scientific-discovery-with-gemini-deep-think/ (visité le 21/04/2026).

[Gop26] Kiran Gopinathan. *Who Watches the Watchers ? Fuzzing a Formally-Verified ZIP Parser*. 2026. url : https://kirancodes.me/posts/log-who-watches-the-watchers.html (visité le 16/04/2026).

[Hal+17] Thomas Hales et al. "A Formal Proof of the Kepler Conjecture". In : *Forum of Mathematics, Pi* (2017). url : https://doi.org/10.1017/fmp.2017.1.

Bibliographie IV

[Kle+09] Gerwin Klein et al. “seL4 : Formal verification of an OS kernel”. In : *SOSP*. 2009. url : https://dl.acm.org/doi/10.1145/1629575.1629596.

[Ler09] Xavier Leroy. “Formal verification of a realistic compiler”. In : *Communications of the ACM* 52.7 (2009), p. 107-115. url : https://dl.acm.org/doi/10.1145/1538788.1538814.

[Mas21] Patrick Massot. *Why Formalize Mathematics ?* 5 déc. 2021. url : https://www.imo.universite-paris-saclay.fr/~patrick.massot/files/exposition/why_formalize.pdf.

[nat26] natsothanaphan. “AI contributions to Erdős problems”. In : (2026). url : https://github.com/teorth/erdosproblems/wiki/AI-contributions-to-Erd%5C%C5%5C%91s-problems/.

Bibliographie V

[Roc26] Rocq Zulip community. *Proof of False Found by Opus 4.6 and mxdys (bbchallenge)*. 2026. url : https://rocq-prover.zulipchat.com/#narrow/channel/237977-Rocq-users/topic/Proof.20of.20false.20found.20by.20Opus.204.2E6.20and.20mxdys.20.28bbchallenge.29/with/580830257 (visité le 16/04/2026).

[Sev26] Simone Severini. *Infrastructure for Mathematics*. 2026. url : https://simoneseverini.github.io/infrastructure-for-mathematics.html (visité le 16/04/2026).

[Sté26] Tristan Stérin. *In Search of Falsehood : AI-Assisted Hunt for Inconsistencies in Rocq*. 2026. url : https://tristan.st/blog/in_search_of_falsehood (visité le 20/04/2026).

[Tho84] Ken Thompson. "Reflections on Trusting Trust". In : *Communications of the ACM* 27.8 (1984), p. 761-763. url : https://dl.acm.org/doi/10.1145/358198.358210.

Bibliographie VI

[TO23] Max Tegmark et Steve Omohundro. *Provably Safe Systems : The Only Path to Controllable AGI*. 5 sept. 2023. doi : 10.48550/arXiv.2309.01933. arXiv : 2309.01933 [cs]. url : http://arxiv.org/abs/2309.01933 (visité le 16/04/2026).