

L'automazione dell'intuizione

Proof Assistant e AI nella Pratica Matematica Moderna

Marino Miculan

DMIF, Università di Udine

3° convegno PRIMI - 8 gennaio 2026



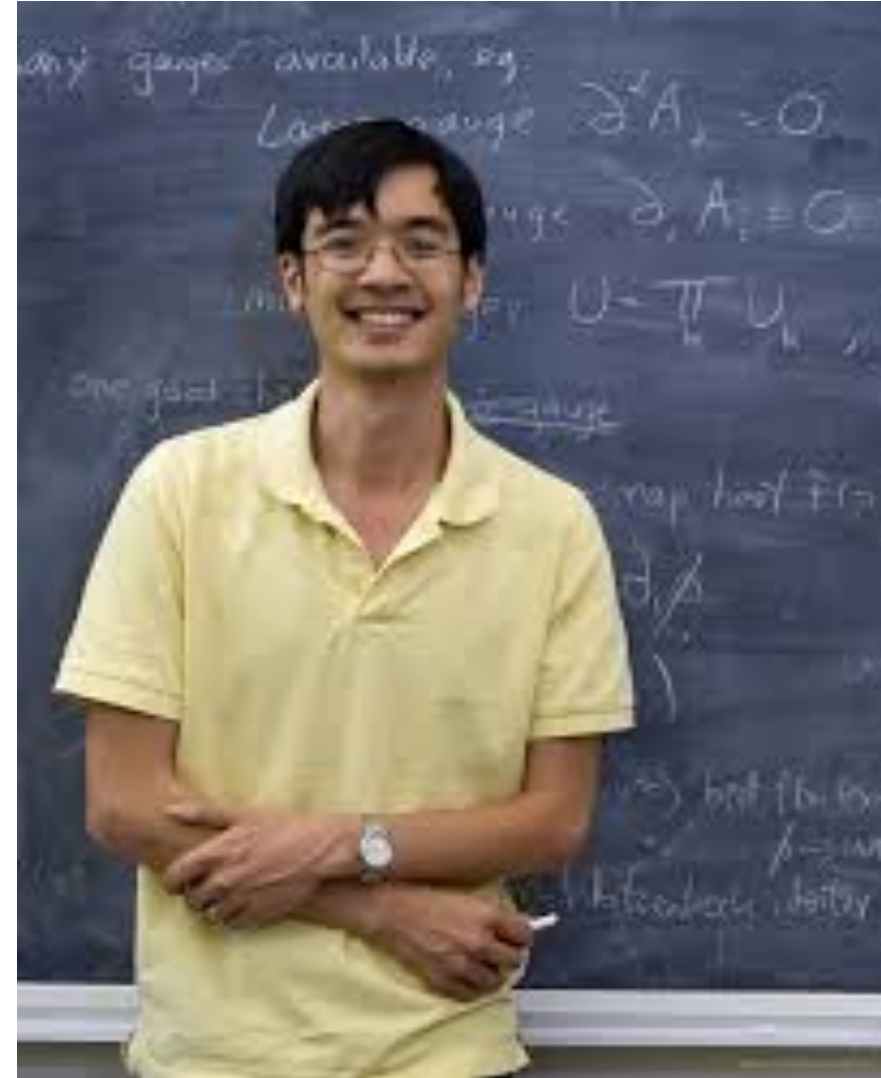
La Crisi della Complessità

- La matematica contemporanea affronta un problema senza precedenti
 - Dimostrazioni che superano le 100 pagine
 - Esempio paradigmatico: Teorema di Classificazione dei Gruppi Semplici Finiti (“The Enormous Theorem”)
 - Il collo di bottiglia della peer review: chi verifica i verificatori?
 - Teoremi che richiedono anni per essere letti e compresi (e forse mai fino in fondo)
- La matematica sta superando la scala della singola mente umana



Il Cambio di Paradigma di Terence Tao

- Da scettico osservatore a pioniere digitale: Tao ha integrato strumenti come Lean e GitHub Copilot nel suo workflow quotidiano.
- "L'intelligenza artificiale cambierà il modo in cui facciamo matematica, non eliminando il matematico, ma potenziandolo."



Che cos'è un Proof Assistant?

- Software che verifica la derivazione logica riga per riga, trasformando la dimostrazione in un **programma compilabile**
- Se il sistema riesce a “compilare” il programma, la dimostrazione è corretta oltre ogni ragionevole dubbio
- Molti tool sviluppati negli ultimi 50 anni. Alcuni esempi:
 - Lean: moderno e con una comunità molto attiva
 - Coq (ora Rocq): robusto e consolidato
 - Isabelle: potente e flessibile (basato su HOL classica)

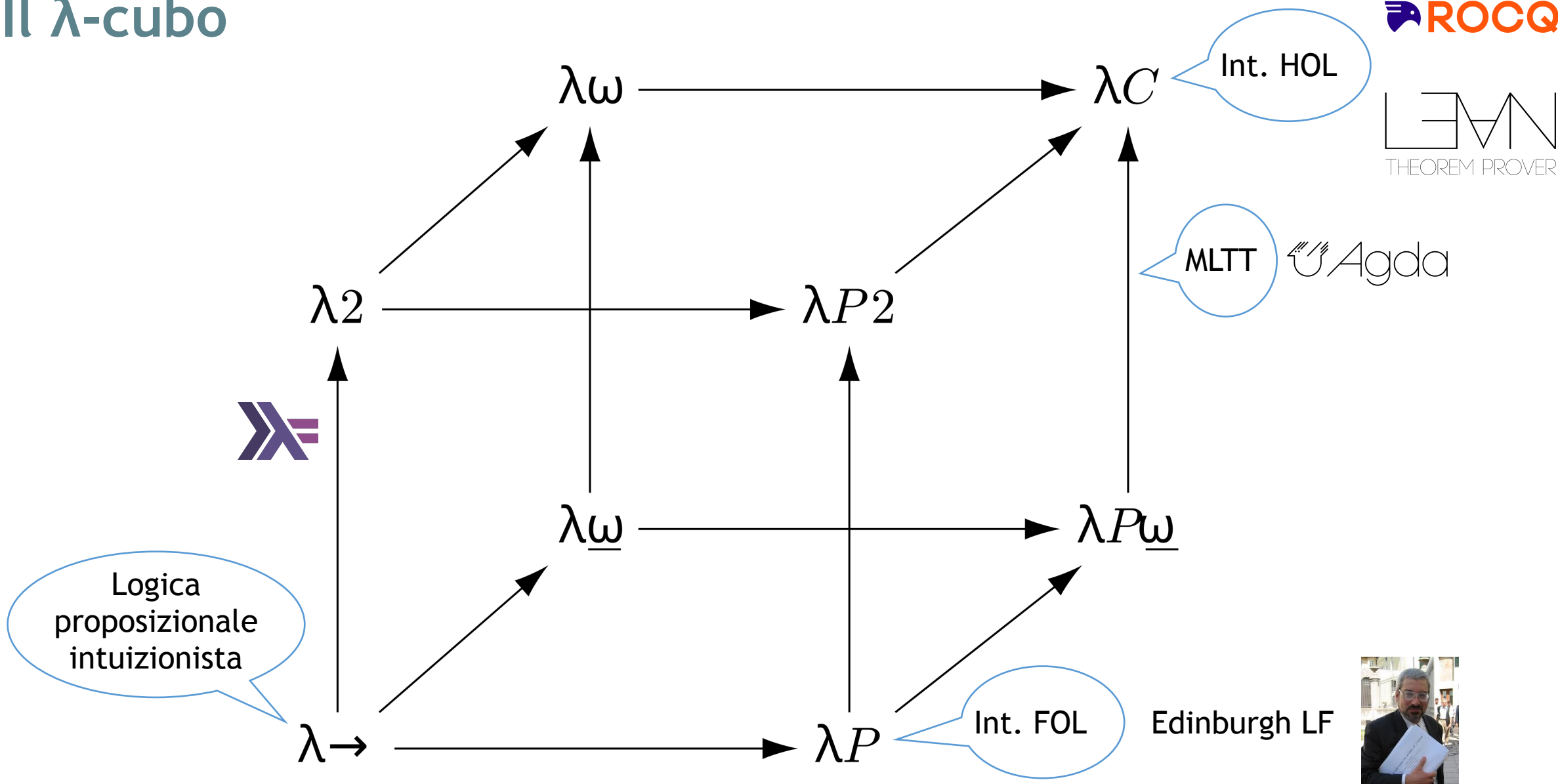
Isomorfismo di Curry-Howard

Logic side	Programming side
formula	type
proof	term
formula is true	type has an element
formula is false	type does not have an element
logical constant \top (truth)	<u>unit type</u>
logical constant \perp (falsehood)	<u>empty type</u>
<u>implication</u>	<u>function type</u>
<u>conjunction</u>	<u>product type</u>
<u>disjunction</u>	<u>sum type</u>
<u>universal quantification</u>	<u>dependent product type</u>
<u>existential quantification</u>	<u>dependent sum type</u>

Isomorfismo di Curry-Howard

Hilbert-style intuitionistic implicative logic	Typed combinatory logic
$\frac{\alpha \in \Gamma}{\Gamma \vdash \alpha} \quad \text{Assum}$	$\frac{x : \alpha \in \Gamma}{\Gamma \vdash x : \alpha}$
$\overline{\Gamma \vdash \alpha \rightarrow (\beta \rightarrow \alpha)} \quad \text{Ax}_K$	$\overline{\Gamma \vdash K : \alpha \rightarrow (\beta \rightarrow \alpha)}$
$\overline{\Gamma \vdash (\alpha \rightarrow (\beta \rightarrow \gamma)) \rightarrow ((\alpha \rightarrow \beta) \rightarrow (\alpha \rightarrow \gamma))} \quad \text{Ax}_S$	$\overline{\Gamma \vdash S : (\alpha \rightarrow (\beta \rightarrow \gamma)) \rightarrow ((\alpha \rightarrow \beta) \rightarrow (\alpha \rightarrow \gamma))}$
$\frac{\Gamma \vdash \alpha \rightarrow \beta \quad \Gamma \vdash \alpha}{\Gamma \vdash \beta} \quad \text{Modus Ponens}$	$\frac{\Gamma \vdash E_1 : \alpha \rightarrow \beta \quad \Gamma \vdash E_2 : \alpha}{\Gamma \vdash E_1 E_2 : \beta}$

Il λ -cubo



La sfida di Peter Scholze (2020): Verificare un risultato fondamentale ma estremamente complesso.

supp $\nu_x \subset K$ compact, a.e. x , and
 $x \mapsto \int \varphi(y) d\nu_x(y)$ measurable for every
 Question: $\exists \varphi_k \in L^1$ bounded and that $\varphi_k \rightarrow \varphi$
 $\int \varphi(\varphi_k(x)) \varphi(x) dx \rightarrow \int \int \varphi(z) d\nu_x(z) \varphi(x) dx$?

$\nu = \sum_{i=1}^N \lambda_i \delta_{x_i}$
 $\lambda_i > 0, \sum \lambda_i = 1$
 Assume next: ν is a general prob. measure.
 Choose ν_j discrete s.t. $\nu_j \rightarrow \nu$
 For each j have generating sequence $(\varphi_k^j)_k \rightarrow \nu_j$
 Diagonal argument: $\varphi_{k(j)}^j \rightarrow \nu$ if $k(j)$ is large enough.



“A volte mi svegliavo di notte pensando: 'C'è un errore fondamentale in tutto questo?'. Avevo bisogno di un computer che mi dicesse che il Teorema 9.1 era corretto, perché la mia mente non riusciva più a contenerne l'intera complessità senza dubbi.”

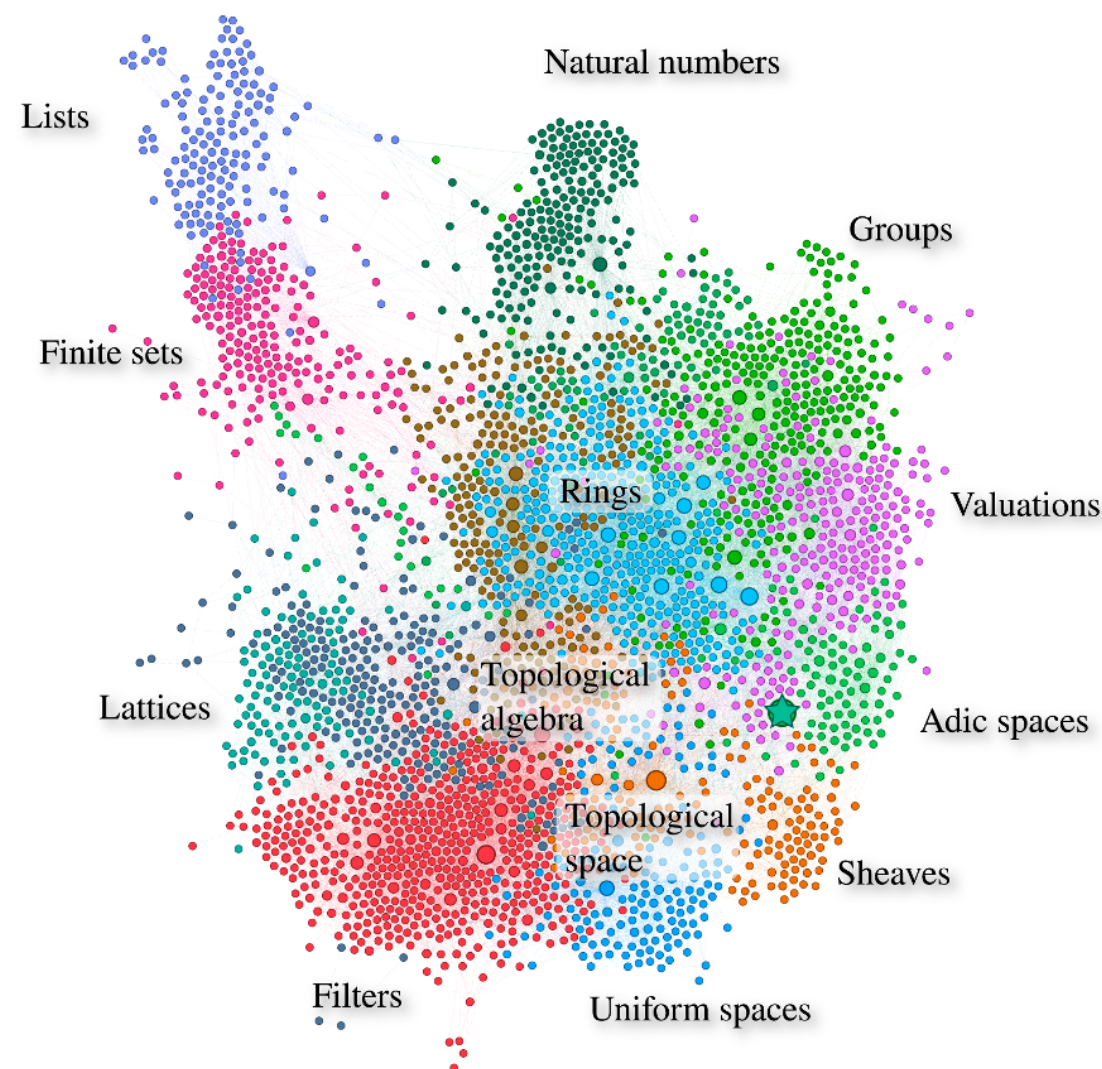
Il "Liquid Tensor Experiment"

- La sfida di Peter Scholze (2020): Verificare un risultato fondamentale ma estremamente complesso.
- Il successo della comunità Lean (2021): La dimostrazione è stata formalizzata e verificata in sei mesi.
- Risultato: Una maggiore fiducia nella matematica d'avanguardia.



Mathlib: La Biblioteca Globale della Matematica

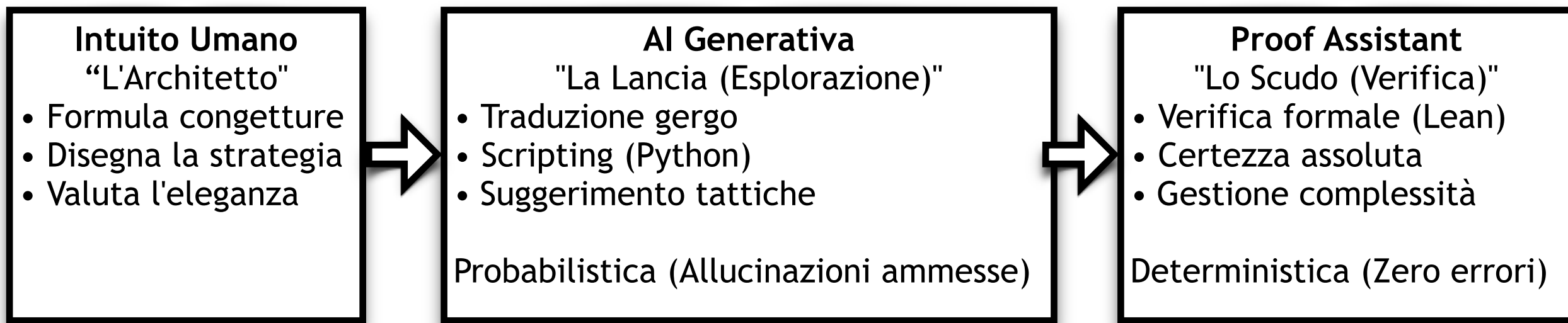
- Un deposito centralizzato di matematica formalizzata.
- Ogni nuovo teorema si poggia su migliaia di lemmi già verificati.
- L'interconnessione automatica tra settori diversi della matematica.
- Ma è enorme...



L'AI come Co-Pilota (L'Intuito)

- LLM (ChatGPT, Claude, Gemini): Ottimi per riassumere teorie, suggerire analogie e scrivere script Python.
- Limiti: L'allucinazione: l'AI inventa teoremi falsi.
- **Soluzione:** Usare l'AI per generare bozze, poi verificarle rigorosamente con i Proof Assistant.

L'AI come Co-Pilota (L'Intuito)



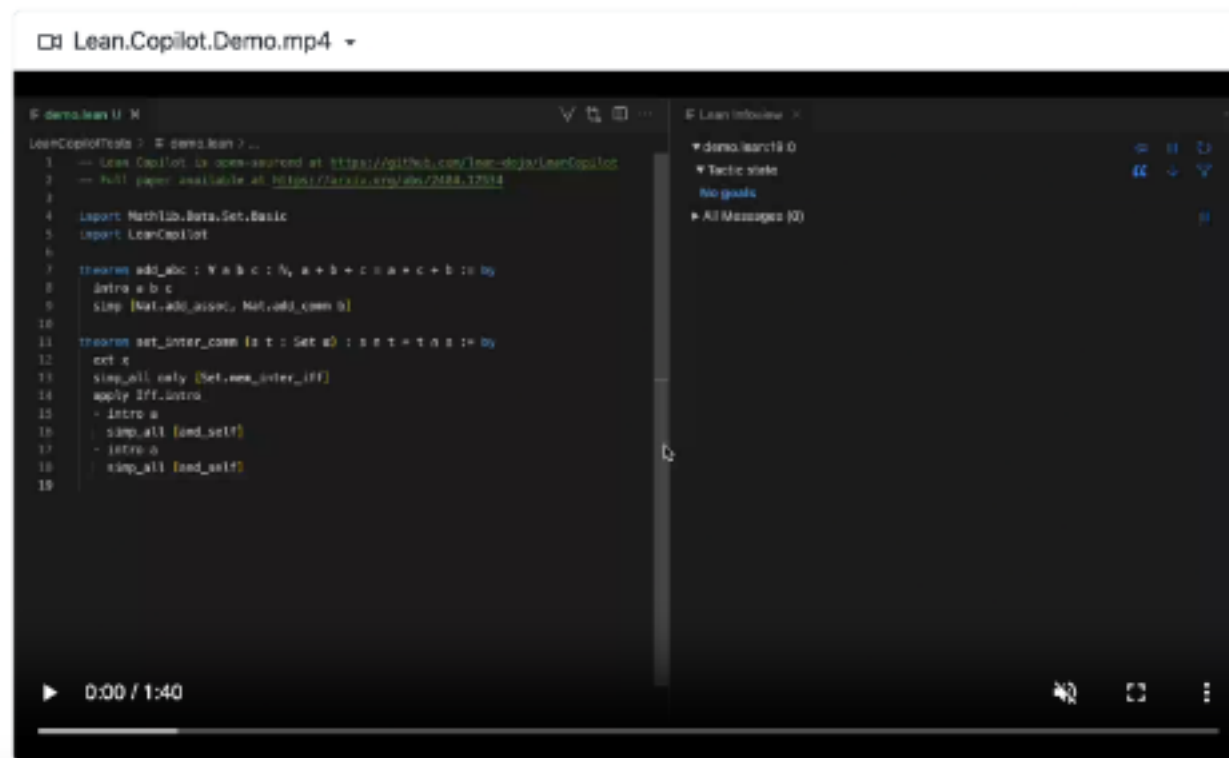
L'umano immagina, l'AI propone, il proof assistant dispone

Lean Copilot: La Sintesi Perfetta

- Integrazione di modelli linguistici direttamente dentro Lean.
- L'AI suggerisce la "tattica" successiva (il passaggio logico); il matematico approva o scarta il suggerimento.
- Premise Selection: Trova istantaneamente i teoremi rilevanti dentro Mathlib.

Lean Copilot: LLMs as Copilots for Theorem Proving in Lean

Lean Copilot allows large language models (LLMs) to be used in Lean for proof automation, e.g., suggesting tactics/premises and searching for proofs. You can use our built-in models from [LeanDojo](https://lean-dojo.org/LeanDojo) or bring your own models that run either locally (w/ or w/o GPUs) or on the cloud.



Il Nuovo Workflow di Tao

1. **Apprendimento** (“Rubber duck”): usa l'AI per spiegare a se stesso concetti di branche matematiche non familiari.
2. **Esplorazione**: Usare Python/AI per testare esempi numerici di congetture.
3. **Architettura**: Disegnare la struttura della prova in linguaggio naturale, ad alto livello.
4. **Formalizzazione**: Tradurre in Lean con l'aiuto dell'AI Copilot.
5. **Verifica**: Il computer conferma immediatamente la correttezza totale.



L'Industrializzazione della Matematica

- Dalla "Matematica Artigianale" (un solo autore, un solo articolo)...
- Alla "Matematica Industriale" (grandi collaborazioni, librerie di codice, verifica automatizzata).
- **Democratizzazione**
 - Permettere a più persone di contribuire a parti specifiche di un grande progetto.
 - Fine del Principio di Autorità: non conta chi sei, conta se il tuo codice compila
- È il momento in cui la matematica diventa una risorsa 'pronta all'uso' per l'informatica, la fisica e l'ingegneria del futuro.

Sfide e Questioni Etiche

- La formazione dei futuri matematici
 - dobbiamo insegnare a programmare?
 - Come bilanciare il rigore del codice con la libertà del pensiero creativo?
- Il rischio di perdere l'intuizione "profonda" a favore della manipolazione di simboli?
 - Esiste il pericolo di una matematica 'funzionante ma opaca'.
 - Come usare l'AI per *aumentare* la nostra comprensione?
- Accesso alle risorse computazionali
 - l'AI richiede enormi risorse computazionali. Chi controllerà la direzione della ricerca matematica? Le grandi corporation?
 - Come cambierà l'etica della scoperta scientifica?

Conclusione: L'Inizio di un'Era

- La matematica sta cambiando linguaggio
- L'invito di Tao: "Esplorate questi strumenti, non temeteli".
- La macchina fornisce la certezza, l'uomo fornisce la direzione. Insieme, stiamo solo iniziando a vedere la vera forma dell'infinito.

