



L'automazione dell'intuizione

Proof Assistant e AI nella Pratica Matematica Moderna

Marino Miculan

DMIF, Università di Udine

3º convegno PRIMI - 9 gennaio 2026



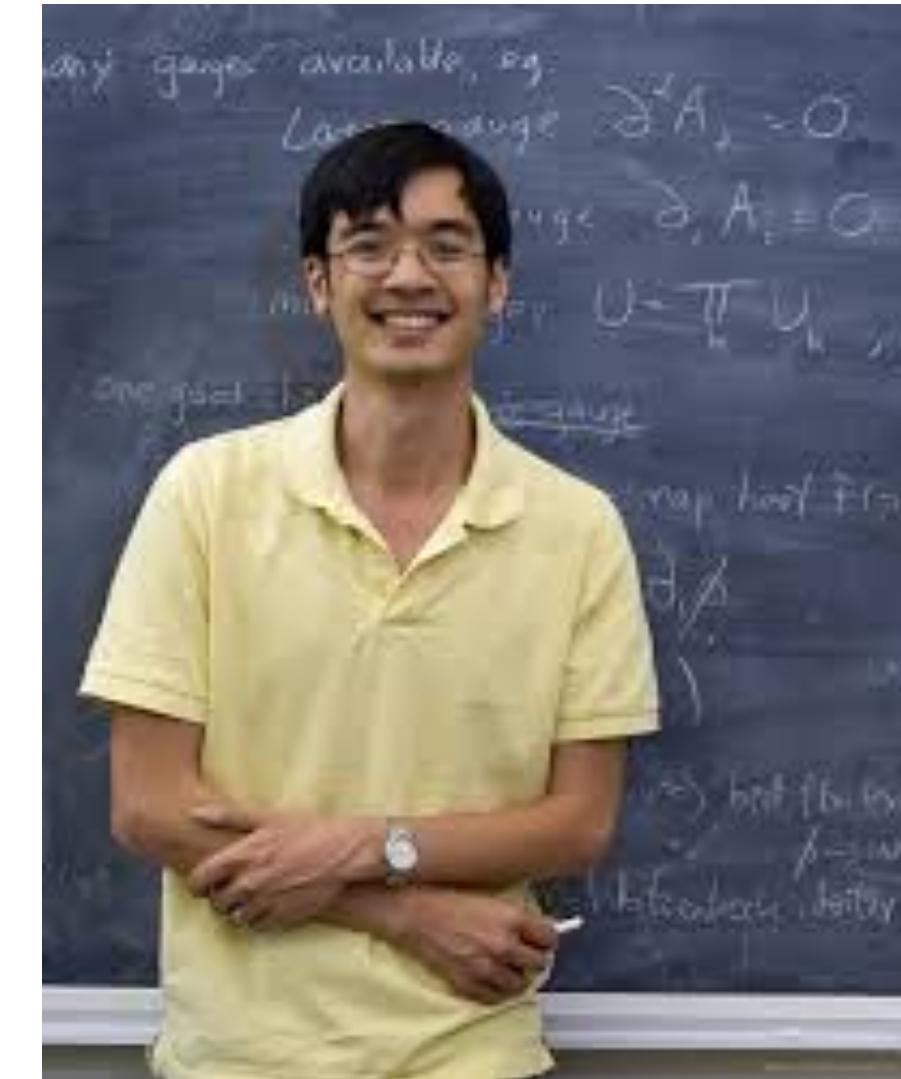
La Crisi della Complessità

- 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)	unit type
logical constant \perp (falsehood)	empty type
implication	function type
conjunction	product type
disjunction	sum type
universal quantification	dependent product type
existential quantification	dependent sum type

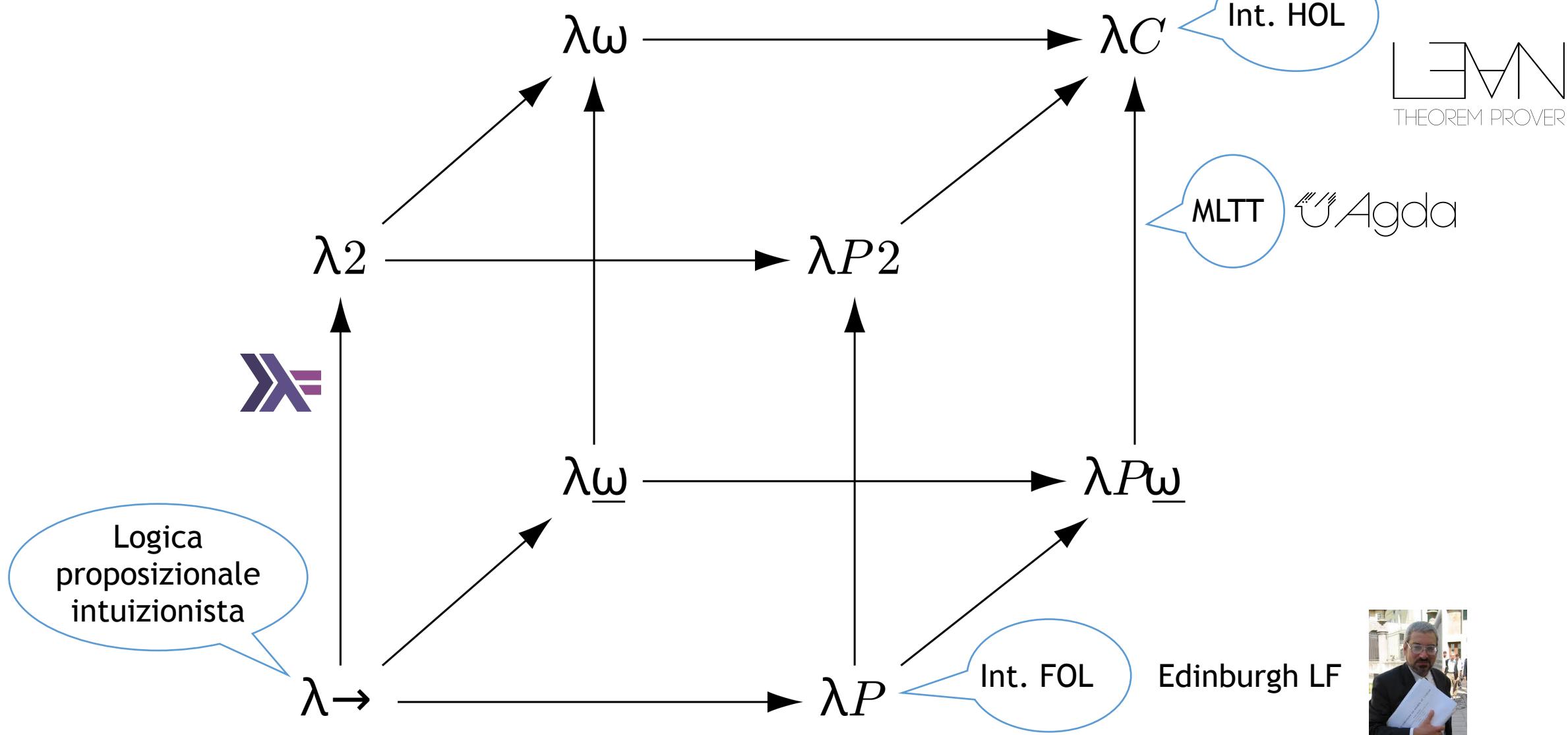


Isomorfismo di Curry-Howard

Hilbert-style intuitionistic implicational logic	Typed combinatory logic
$\frac{\alpha \in \Gamma}{\Gamma \vdash \alpha}$ Assum	$\frac{x : \alpha \in \Gamma}{\Gamma \vdash x : \alpha}$
$\frac{}{\Gamma \vdash \alpha \rightarrow (\beta \rightarrow \alpha)}$ Ax_K	$\frac{}{\Gamma \vdash K : \alpha \rightarrow (\beta \rightarrow \alpha)}$
$\frac{}{\Gamma \vdash (\alpha \rightarrow (\beta \rightarrow \gamma)) \rightarrow ((\alpha \rightarrow \beta) \rightarrow (\alpha \rightarrow \gamma))}$ Ax_S	$\frac{}{\Gamma \vdash S : (\alpha \rightarrow (\beta \rightarrow \gamma)) \rightarrow ((\alpha \rightarrow \beta) \rightarrow (\alpha \rightarrow \gamma))}$
$\frac{\Gamma \vdash \alpha \rightarrow \beta \quad \Gamma \vdash \alpha}{\Gamma \vdash \beta}$ 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



$\text{supp } v \subset K$ compact, $q < \infty$, and

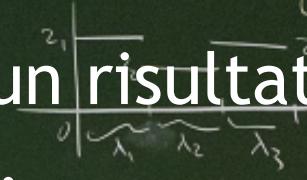
La sfida di Peter Scholze (2020): Verificare un risultato fondamentale ma estremamente complesso.

Question: $\exists (u_k) \subset \mathbb{R}^n$ bounded and flat b.y b.y

$$\int \varphi(u_k(x)) \varphi(x) dx \rightarrow \iint \varphi(z) d\nu_x(z) \varphi(x) dx ?$$

$$v = \sum_{i=1}^N \lambda_i \delta_{z_i}$$

$$\lambda_i > 0, \sum \lambda_i = 1$$



u + extend periodically
ad set $u_k = u(\lambda_k)$
+ use Problem 2

Assume next:

v is a general prob. measure.
choose v_j discrete s.t. $v_j \xrightarrow{*} v$.

For each j have generating sequence $(u_{k(j)})_k \xrightarrow{*} v_j$

Diagonal argument: $u_{k(j)} \xrightarrow{*} v$ 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.”



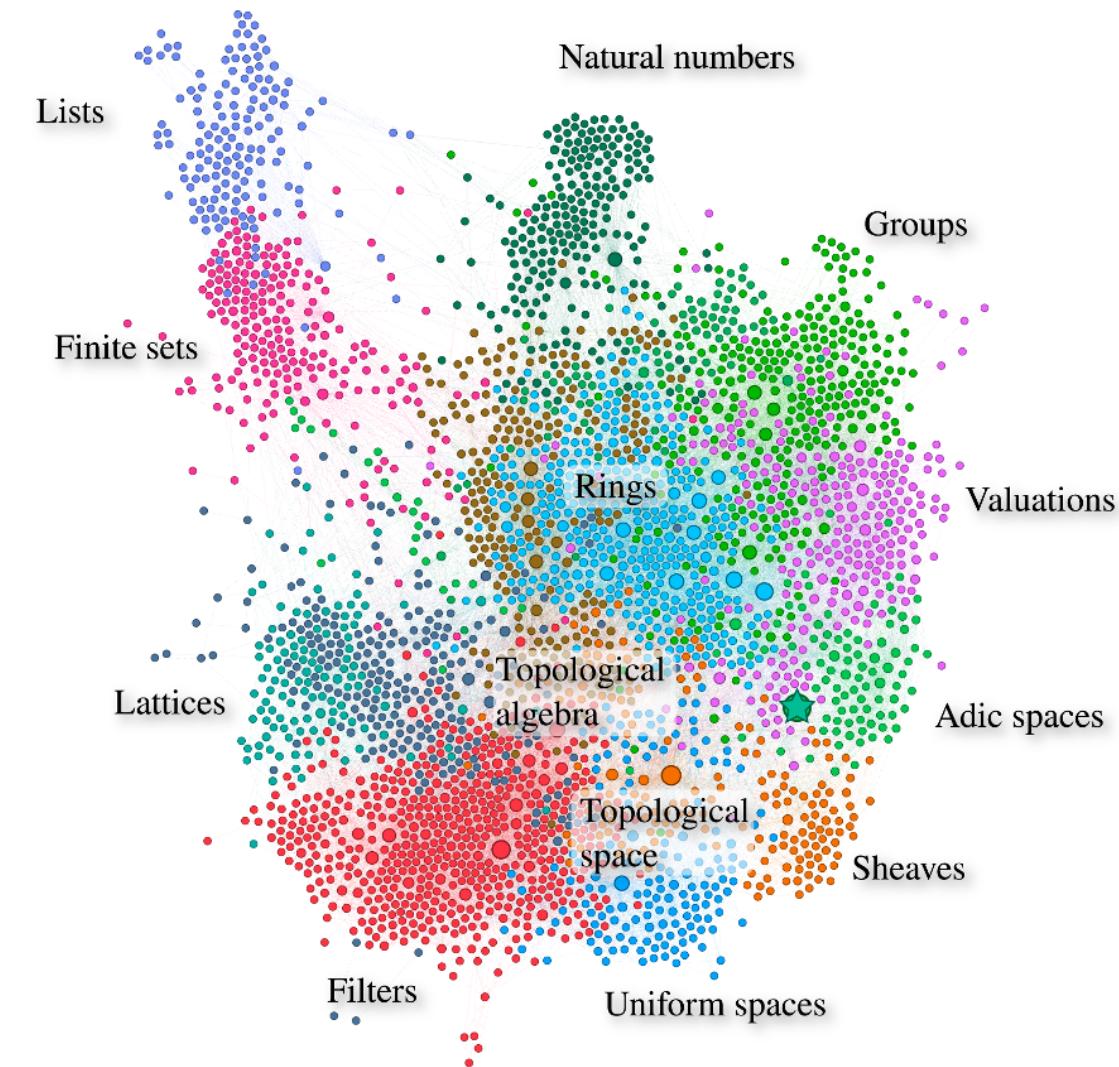
Il "Liquid Tensor Experiment"

- 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)

Intuito Umano

"L'Architetto"

- Formula congettura
- Disegna la strategia
- Valuta l'eleganza



AI Generativa

"La Lancia (Esplorazione)"

- Traduzione gergo
- Scripting (Python)
- Suggerimento tattiche

Probabilistica (Allucinazioni ammesse)



Proof Assistant

"Lo Scudo (Verifica)"

- Verifica formale (Lean)
- Certezza assoluta
- Gestione complessità

Deterministica (Zero errori)

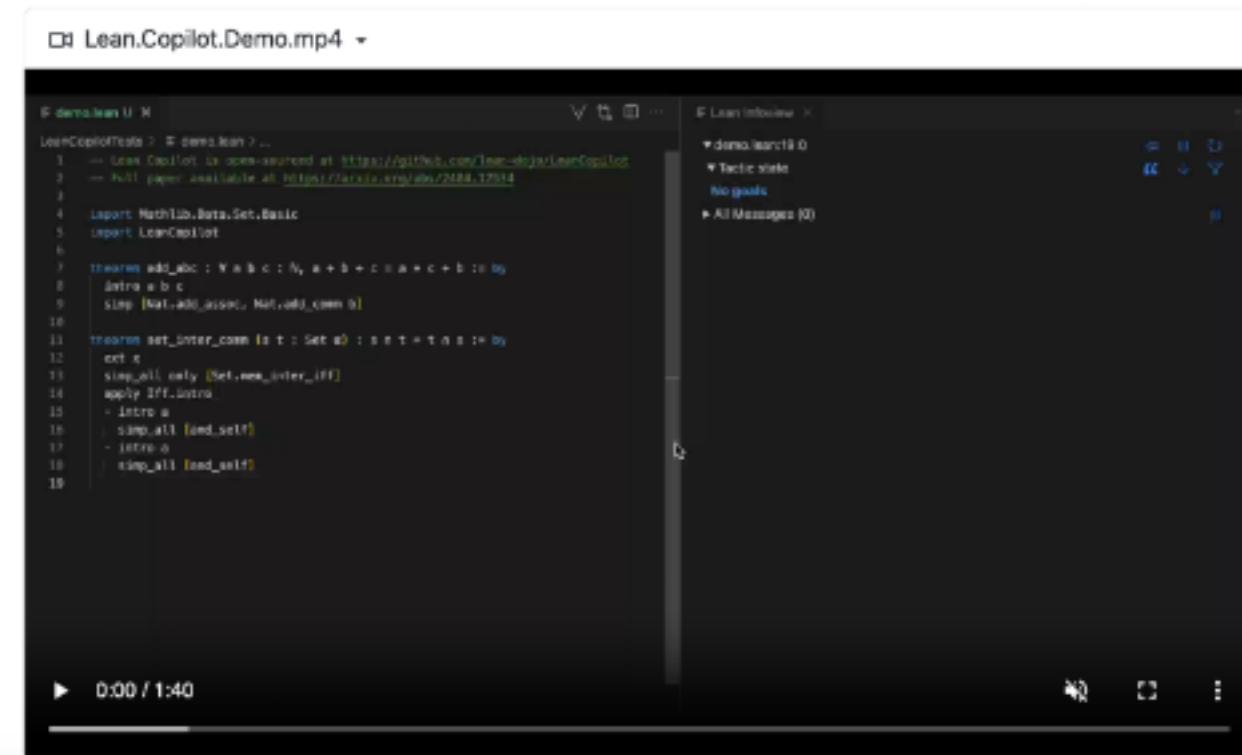
L'umano immagina, l'AI propone, il proof assistant dispone

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](#) 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 Eetiche

- 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.

