What is...autoformalization?

Or: An old ambition meets modern AI

The old ambition: mathematics a machine can check

From: N.G. de Bruijn (1918–2012) and his road to Automath, the earliest proof checker

Figure 1. The last theorem of Edmund Landau's book 'Grundlagen der Analysis' is entered. From left to right: Bram Kornaat, Bert van Benthem Jutting, Ids Zandleven, Roel de Vrijer, N.G. de Bruijn. Source: The Automath Archive [2].

- Late 1960s Automath makes formal proof checking concrete
- 1970s The completion of the first full machine-checked mathematics textbook
- The bottleneck Someone must translate the mathematics first

Can we automate the translation?

From https://naproche-net.github.io/downloads/2007-The_language_of_proofs.pdf

Structure of the curt system:

Input text ("`Every man dies`")

↕ Parser, Tokenizer (readline.pl)

Tokenized format (`[all, man, die]`)

↕ NLP (natural language processing)

Internal representation (...)

↕

First-order logic format (`all(A, imp(man(A), die(A)))`)

↕ (fol2otter.pl, fol2mace.pl)

Input format for theorem prover Otter / model builder Mace (...)

- Autoformalization Automatically translate prose into formal mathematics
- 2000s Translate and check a controlled mathematical language
- Around 2020 The ambition: learn to translate ordinary mathematical text

2022: language models make the bridge practical

From: Autoformalization with Large Language Models

Case Study 2 Question:
Natural Language version: "*When all the girls at Madeline's school line up in rows of eight, there are seven left over. If instead they line up in rows of four, how many are left over? The final answer is 3.*" Translate the natural language version to an Isabelle version:

PaLM output:

```
theorem
  fixes n::nat
  assumes "n mod 8 = 7"
  shows "n mod 4 = 3"
```

Case Study 3 Question:
Natural language version: "*Let f be a linear function for which $f(6) - f(2) = 12$. What is $f(12) - f(2)$? The final answer is 30.*" Translate the natural language version to an Isabelle version:

Codex output:

```
theorem
  fixes f :: "real \<Rightarrow> real"
  assumes "linear f"
          "f 6 - f 2 = 12"
  shows "f 12 - f 2 = 30"
```

Figure 3: Autoformalizations from natural language to Isabelle code. **Left:** Case study 2 – perfect formalization by PaLM. **Right:** Case study 3 – incorrect formalization by Codex.

- A concrete success Translate competition problems into formal statements
- The change Learn from examples instead of prescribing every phrase
- The payoff The translated statements help train a better theorem prover

2024: autoformalization feeds AlphaProof

IMO 2021 Shortlist, Problem A5

Let $n \geq 2$ be an integer and let $a_1, a_2, \ldots, a_n$ be positive real numbers with sum 1. Prove that

$$\sum_{k=1}^{n} \frac{a_k}{1 - a_k}(a_1 + a_2 + \cdots + a_{k-1})^2 < \frac{1}{3}.$$

From: Olympiad-level formal mathematical reasoning with reinforcement learning

Formalization system

```
theorem imo_shortlist_2021_a5
    (n : ℕ) (h₀ : 2 ≤ n) (a : ℕ → ℝ) (hapos : ∀ i, 0 < a i)
    (hasum : ∑ i in Finset.Icc 1 n, a i = 1) :
    ∑ k in Finset.Icc 1 n, a k / (1 - a k) * (∑ i in Finset.Icc 1 (k-1), a i) ^ 2 < 1 / 3
```

- The supply: Automatically translate informal problems into formal ones
- The loop: Search for proofs, check them, learn from verified results
- Why it matters: Translation supplies problems; checking supplies feedback

2026: from exercises to mathematical literature

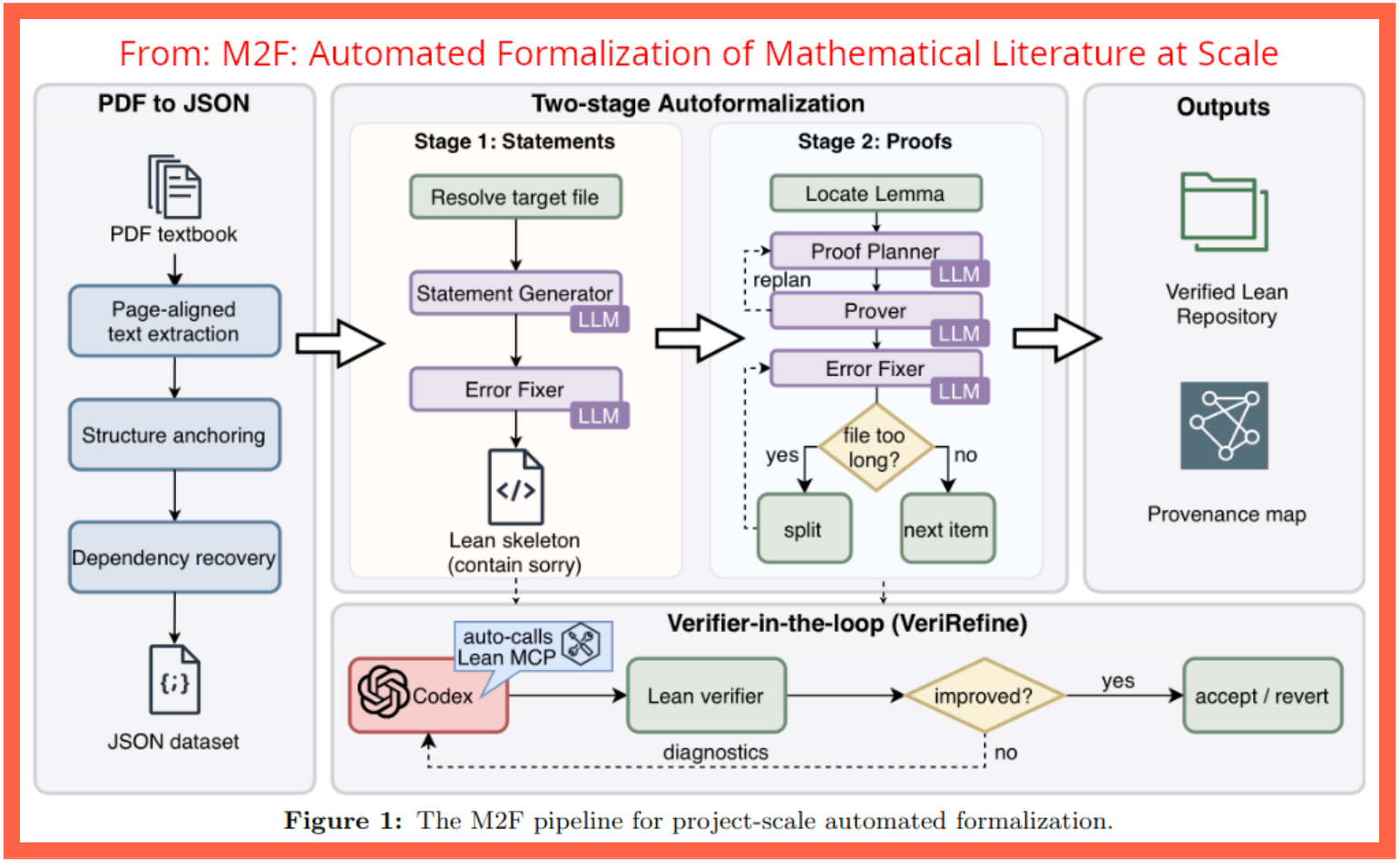

Figure 1: The M2F pipeline for project-scale automated formalization.

- M2F Reports textbook-scale autoformalization in Lean
- MathAtlas Tests graduate mathematics with its web of dependencies
- The frontier Preserve meaning while scaling to whole theories

Thank you for your attention!

Next time: AI-assisted mathematical discovery