AI for formalization of category theory, an experience report

Or, AI for the working formalizer

Fernando Chu

April 10, 2026 - DutchCATS

Recent formalization milestones

We are seeing a rapid increase in AI's capabilities.

Recent formalization milestones

We are seeing a rapid increase in AI's capabilities.

07-2024 **Silver level IMO** (Google, Alphaproof+AlphaGeometry)
Mixture of neural networks with symbolic system.

Recent formalization milestones

We are seeing a rapid increase in AI's capabilities.

07-2024 **Silver level IMO** (Google, Alphaproof+AlphaGeometry)
Mixture of neural networks with symbolic system.

09-2025 **Strong Prime Number Theorem** (Math Inc.)
LLM supervised and managed by humans. 25k loc

Recent formalization milestones

We are seeing a rapid increase in AI's capabilities.

07-2024 **Silver level IMO** (Google, Alphaproof+AlphaGeometry)
Mixture of neural networks with symbolic system.

09-2025 **Strong Prime Number Theorem** (Math Inc.)
LLM supervised and managed by humans. 25k loc

02-2026 **Sphere Packing in 24 dimensions** (Math Inc.)
Unsupervised LLM. 500k loc

Recent formalization milestones

We are seeing a rapid increase in AI's capabilities.

07-2024 **Silver level IMO** (Google, Alphaproof+AlphaGeometry)
Mixture of neural networks with symbolic system.

09-2025 **Strong Prime Number Theorem** (Math Inc.)
LLM supervised and managed by humans. 25k loc

02-2026 **Sphere Packing in 24 dimensions** (Math Inc.)
Unsupervised LLM. 500k loc

However...

AI's code is not the highest quality

This is a lemma formalized by Claude for my project.

```
/-- Given `f : p(e) → a` and a morphism `β` in `B(p(e))` ending at
the fiber component of `q(e)`, this transports `β` to a morphism
in `B(a)` via `B.map f`, adjusting by `eqToHom` from the cartesian
condition `hObj : q(fe') = transport(q(e), f)`. -/
def fiberRestriction.transportMor
    {e : C} {a : A} (f : (q ⋙ Grothendieck.forget B).obj e → a)
    {fe' : C} (hProj : (q ⋙ Grothendieck.forget B).obj fe' = a)
    (hObj : q.obj fe' = (q.obj e).transport f)
    {b : (B.obj ((q ⋙ Grothendieck.forget B).obj e))}
    (β : b → (fiberRestriction B q _).obj ⟨e, rfl⟩) :
    (B.map f).toFunctor.obj b →
    (fiberRestriction B q a).obj ⟨fe', hProj⟩ :=
  (B.map f).toFunctor.map β ≫ eqToHom (by
    simp only [fiberRestriction_obj]
    exact (grothendieck_fiber_eq hObj).symm)
```

Can AI be useful for *good* formalization?

A step back

What is wrong with AI generated code?

A step back

What is wrong with AI generated code?

- It produces a term of the right type.

A step back

What is wrong with AI generated code?

- It produces a term of the right type.
- This term is in fact used as part of the proof.

A step back

What is wrong with AI generated code?

- It produces a term of the right type.
- This term is in fact used as part of the proof.
- The definition is well documented.

A step back

What is wrong with AI generated code?

- It produces a term of the right type.
- This term is in fact used as part of the proof.
- The definition is well documented.
- The code follows all style guidelines.

A step back

What is wrong with AI generated code?

- It produces a term of the right type.
- This term is in fact used as part of the proof.
- The definition is well documented.
- The code follows all style guidelines.

My answer: the code is not easily understandable to humans.

A step back

What is wrong with AI generated code?

- It produces a term of the right type.
- This term is in fact used as part of the proof.
- The definition is well documented.
- The code follows all style guidelines.

My answer: the code is not easily understandable to humans.

Main Problem: Mismatch of levels of abstraction.

The HITL Agentic Workflow

Fancy words for the common AI workflow:

1. Make a precise stepwise plan.
   - Human review, feedback and iteration.
2. Execute step 1 of the plan
   - Human review, feedback and iteration.
3. Execute step 2 of the plan
   - Human review, feedback and iteration.
4. ...

Step 0. Set up a plan

- Make a plan (or let AI do it).
- E.g.:

  I want to formalize internal categories in this Lean project. Make a precise plan to formalize the 2-categories of internal categories on a category C...

- Review the plan and modify it accordingly.
  - Bad definitions, names
  - Lack of intermediate lemmas
  - Missing API

Step i. Execute step i from the plan

- Tell the AI to execute step i.
- Do not be overly permissive.
- Teach it from its mistakes.
- Keep the plan up to date.

Step i. Execute step i from the plan

- Tell the AI to execute step i.
- Do not be overly permissive.
- Teach it from its mistakes.
- Keep the plan up to date.
- ...That's it.

Some results

1. Formalizing D2SFibs.
   - Formalization of split (op)fibrations, wip D2SFibs.
   - $\sim$30 euros.
   - $\sim$1200 loc.
   - $\sim$70% of the code is where I want to be at.
2. Formalizing the bicategory of internal categories.
   - Formalization of naive internal categories.
   - $<$20 euros.
   - $\sim$800 loc.
   - $\sim$80% of the code is where I want to be at.

Each took $<$30 minutes of my time.

What is left to do?

- Coding. AI can help.

What is left to do?

- Coding. AI can help.
- Documentation. AI can help.

What is left to do?

- Coding. AI can help.
- Documentation. AI can help.
- Refactoring: AI can do it.

What is left to do?

- Coding. AI can help.
- Documentation. AI can help.
- Refactoring: AI can do it.
- Reviewing: AI can help.

What is left to do?

- Coding. AI can help.
- Documentation. AI can help.
- Refactoring: AI can do it.
- Reviewing: AI can help.

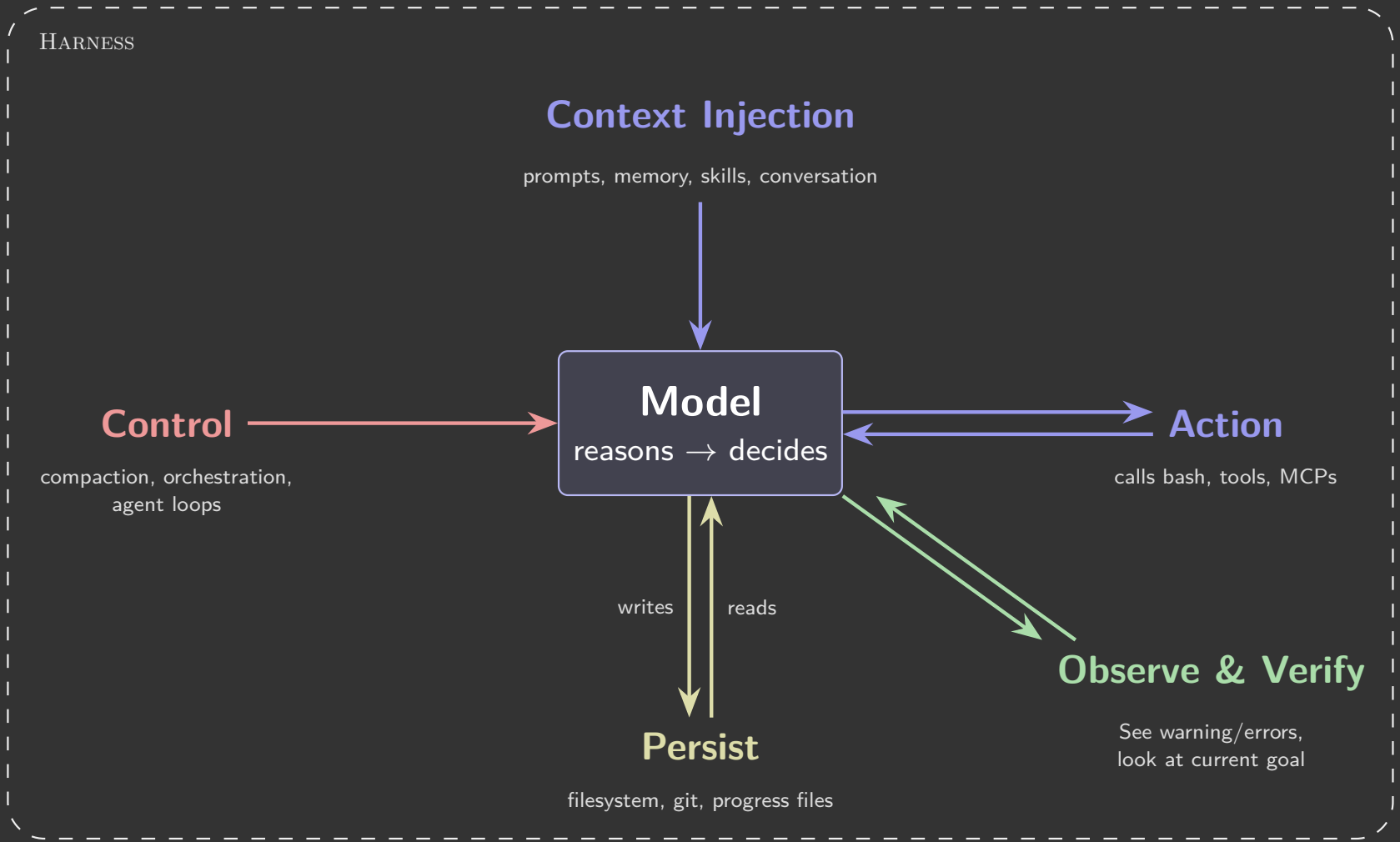
The cost of a decent/good formalization is significantly lower.

What is left to do?

We should add value to AI-assisted formalizations.

1. Add value on the AI part.
2. Add value on the formalization part.

Agent = Model + Harness



Adapted from https://blog.langchain.com/the-anatomy-of-an-agent-harness/

Optimizing the harness

The model is given, we can optimize the harness.

- Optimized harness $\Rightarrow$ 50% better on some benchmarks
- It's speculated that Math Inc. Gauss is just a custom harness for Claude.
- Low barrier to entry, potentially great results.

Some harness optimization examples

- Custom orchestration tool.
- Specific MCPs.
- Specific skills.
- Better memory management, etc.

More value on formalization

- Formalization as a teaching tool
  - Natural language proofs
  - Waterproof (Rocq), Verbose (Lean)
- Formalization as a *readable* archive of proofs
  - No library can be immediately read by a non-expert.
  - Natural language like DSL?
  - Translation from formal into informal?

More value on formalization

Empirical comparison of different approaches.

- Formalization papers *explain* a choice.
- We can now *compare* design choices.
- Some examples I'd like to see:
  - Displayed categories: naive v.s. displayed approach.
  - Internal categories: naive v.s. simplicial objects.
  - Categories: naive v.s. equality-respecting categories.

Autoformalization perspectives

In three weeks Math Inc. did[1]

- **Dead code elimination:** 500k loc → 300k loc
- **Declaration deduplication:** 300k loc → 250k loc
- **Iterative refactorization / golfing:** 250k loc → 180k loc
  - Project structure refactorization
  - Cross-file content refactorization
  - Proof golfing

[1] Talk by Auguste Poiroux at Swiss Mathematical Society Spring Meeting "Formalization and Proof Assistants"

Autoformalization perspectives

In three weeks Math Inc. did[1]

- **Dead code elimination:** 500k loc → 300k loc
- **Declaration deduplication:** 300k loc → 250k loc
- **Iterative refactorization / golfing:** 250k loc → 180k loc
  - Project structure refactorization
  - Cross-file content refactorization
  - Proof golfing

Speculation: Autoformalization will continue to improve.

[1] Talk by Auguste Poiroux at Swiss Mathematical Society Spring Meeting "Formalization and Proof Assistants"

Conclusion

- AI can already be used effectively for formalization.
- We should not ignore these new capabilities.
- We should add value to formalization.