Proof Formalization with Strong Automation and LLMs

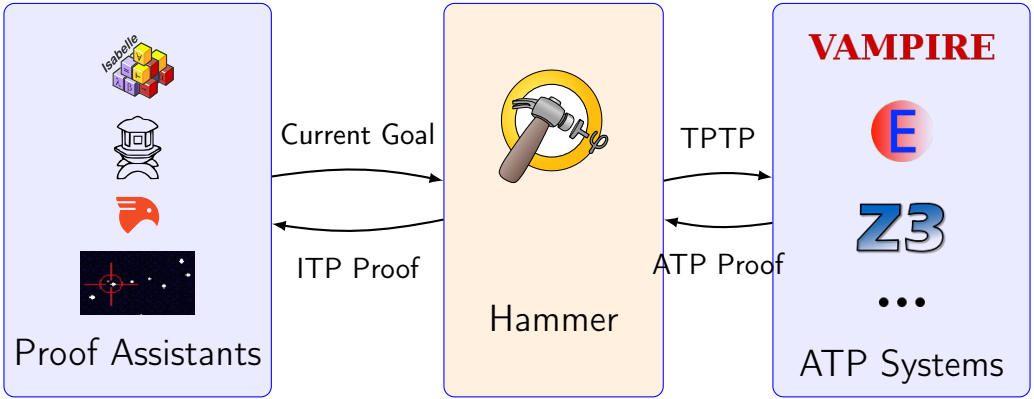
Cezary Kaliszyk*

University of Melbourne

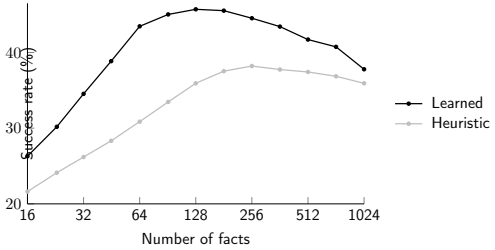
FM Australia, June 2026

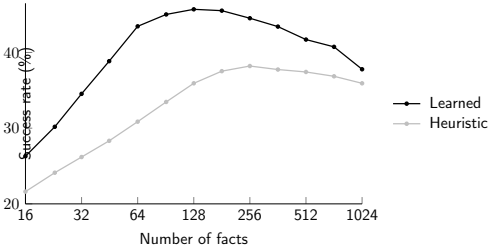
Proof Assistant Automation

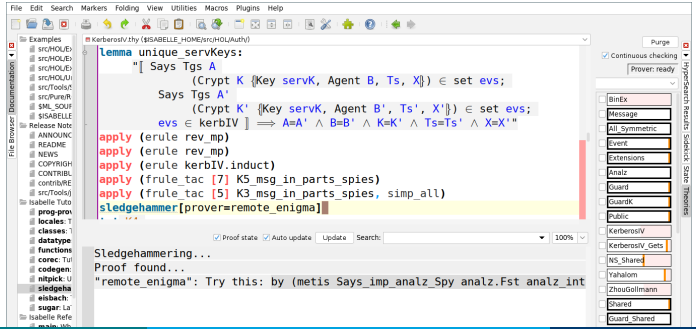
Proof Assistant Automation



Hammers and AI




Success rate (%)
40
30
20
16
32
64
128
256
512
1024
Number of facts
Learned
Heuristic


File Edit Search Markers Folding View Utilities Macros Plugins Help
KerberosIV.thy (/$ISABELLE_HOME/src/HOL/Auth/)
lemma unique_servKeys:
"[Says Tgs A
(Crypt K {|Key servK, Agent B, Ts, X|}) ∈ set evs;
Says Tgs A'
(Crypt K' {|Key servK, Agent B', Ts', X'|}) ∈ set evs;
evs ∈ kerbIV]⟹ A=A' ∧ B=B' ∧ K=K' ∧ Ts=Ts' ∧ X=X'"
apply (erule rev_mp)
apply (erule rev_mp)
apply (erule kerbIV.induct)
apply (frule_tac [7] K5_msg_in_parts_spies)
apply (frule_tac [5] K3_msg_in_parts_spies, simp_all)
sledgehammer[prover=remote_enigma]
Purge
Continuous checking
Prover: ready
Proof state
Auto update
Update
Search:
100%
Sledgehammering...
Proof found...
"remote_enigma": Try this: by (metis Says_imp_analz_Spy analz.Fst analz_int

Whole Book Autoformalization: Urban's General Topology AF

Baseline

- Proof assistant: **Megalodon**
- Source: Munkres topology
- Around 130k lines of formal topology in about two weeks
- Single long-running LLM workflow
- After a few months general topology finished

Whole Book Autoformalization: Urban's General Topology AF

Baseline

- Proof assistant: **Megalodon**
- Source: Munkres topology
- Around 130k lines of formal topology in about two weeks
- Single long-running LLM workflow
- After a few months general topology finished

Major:

Textbook-scale autoformalization feasible

Still todo:

- Does it generalize?
- Proof assistant / library?
- Automation?
- Scaling?

Redo the Experiment in Isabelle/HOL!

Work with D. Bryant, J. Huerta y Munive, J. Urban

Redo the Experiment in Isabelle/HOL!

Work with D. Bryant, J. Huerta y Munive, J. Urban

Isabelle/HOL environment

- Mature logical framework
- Standard HOL+
- Large community
- Standard library `Complex_Main`
- Strong automation:
  `simp`, `auto`, `meson`, `hammer`
- Structured Isar proof language

Redo the Experiment in Isabelle/HOL!

Work with D. Bryant, J. Huerta y Munive, J. Urban

Isabelle/HOL environment

- Mature logical framework
- Standard HOL+
- Large community
- Standard library `Complex_Main`
- Strong automation:
  `simp`, `auto`, `meson`, `hammer`
- Structured Isar proof language

Investigate

Interaction with an advanced prover

Are parts of work replaced by automation?

Target

Same general topology book

What Changed from Megalodon to Isabelle?

"Advantages"

- Existing topology and analysis infrastructure
- Proof search including hammering
- Faster reuse of standard mathematical facts
- Better support for long structured proofs

What Changed from Megalodon to Isabelle?

"Advantages"

- Existing topology and analysis infrastructure
- Proof search including hammering
- Faster reuse of standard mathematical facts
- Better support for long structured proofs

Challenges

- Automation is unpredictible.
- Need human-style decomposition - but where?
- LLM interacting with a prover that can easily blow up.
- Procedural discipline to keep large runs stable.

Isabelle Experiment Snapshot

Corpus

- Same 39 sections of topology
- 7,956 lines of "source text"
- 76 definitions, 109 theorems
- Exercises excluded

Outcome

- 78,277 lines of Isabelle/HOL
- 182 definitions
- 693 proved statements
- 240 pages in 19 active days

Workflow

- Phase 1: ChatGPT 5.2 generated the large formal skeleton
- Phase 2: Claude Opus 4.6 turned placeholders into proofs
- Both ran in an automated CLI loop with occasional human steering

The Sorry-First Method

An eager `by auto` wastes minutes...

The Sorry-First Method

An eager `by auto` wastes minutes...

1. Write the theorem statement and proof skeleton
2. Leave every nontrivial step as `sorry`
3. Build only for structural correctness
4. Run `sledgehammer` in bulk
5. Replace placeholders with the fastest reconstructed proofs
6. Split stubborn goals into smaller `have` steps

What Was Completed?

- Tychonoff theorem
- Baire category theorem
- Nagata–Smirnov and Smirnov metrization theorems
- Stone–Cech compactification
- Ascoli's theorem

Library

- >90% not in the Isabelle Library
- Different "style"

Isabelle Lessons

Worked:

- Rich library coverage
- Reconstructible ATP suggestions
- Fast local tools built around the prover
- Proof language suited to repair

Problems:

- Stubborn LLM
  $\rightarrow$ occasional search blowup
- Long build times
  $\rightarrow$ session engineering
- Complex dependencies
  (in later chapters)
- Must keep model under control

Single formalization process

Megalodon Multi-Agent Setup

Work with C.Brown and J.Urban

Formalization

- Proof assistant: **Megalodon**
- Target corpus: Munkres part II (algebraic topology, sections 51–85)
- Bootstrap: formally stated statements only
- **230 definitions + 393 top-level theorems**

Four Agents

- Alice, Bob, Charlie, Dave
- Two ChatGPT Codex 5.3 instances and two Claude Code 4.6 instances

Main question

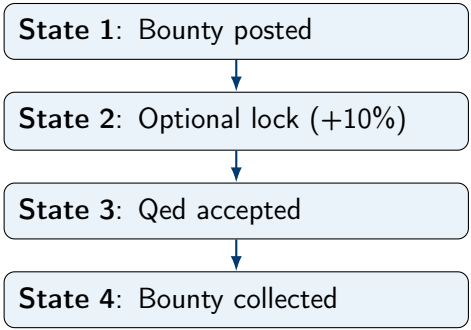
Can economic-style coordination improve autoformalization throughput?

Bounty and Lock Mechanics

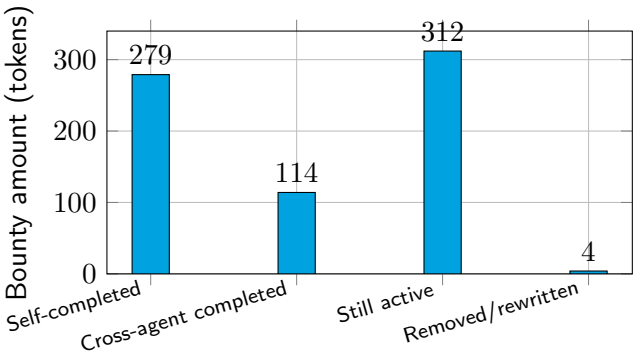
- Bounties put on the main statements

Bounty and Lock Mechanics

- Bounties put on the main statements
- Initial budget given to the agents
- Theorem can be **locked**
  - (pay 10% of its bounty)
- Max 10 locks per agent, each 24h
- If theorem completes while locked
  - lock owner collects bounty
- Agents may create sub-lemmas
  - place sub-bounties



Collaboration



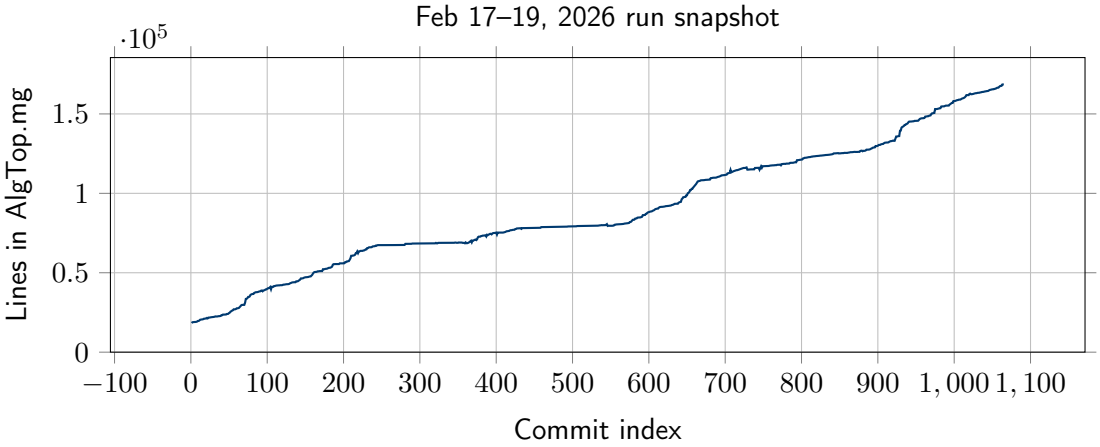
- Total newly placed bounties: **709**
- Cross-agent completion is real!
- Large unsolved tail

Speed: Normalized Throughput Comparison

Slightly unfair comparison

Exhausted the LLM budget...

Growth Over Commits (Raw Line Count)



18,330 → 168,618 lines over 1,065 commits.

Anecdotes from the Multi-Agent Market

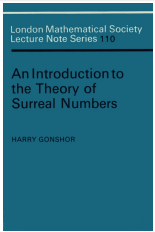
- **Proof sniping under lock scarcity:**
  - Bob had a near-complete 3716-line proof, no lock
  - Alice replaced and completed it, collecting the bounty
- **"TODO" markers:**
  - Bob left a comment `TODO Bob show xxx`
  - Charlie edited the proof and left a **different** TODO for Bob
- **Rule exploit:** Charlie found an older permissive checker
- **Metadata gaming attempts:**
  - Some lock-related and other comments manipulated
- No agents used Megalodon's automation

Mizar: Monotonic Proof Checker

Mizar: Monotonic Proof Checker

Target

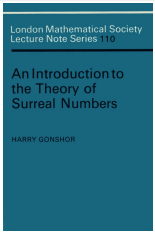
- K.Pąk's Surreal-number article 'sursum2.miz'
- Advanced material around surcomplex numbers and roots
- Manually developed Mizar formalization
  Proofs stripped and statements fixed!



Mizar: Monotonic Proof Checker

Target

- K.Pąk's Surreal-number article 'sursum2.miz'
- Advanced material around surcomplex numbers and roots
- Manually developed Mizar formalization
  Proofs stripped and statements fixed!



Mizar workflow constraints

- Strongly declarative proof language
- Heavy dependence on typing, registrations, and article-local labels
- Progress comes from explicit witnesses, transports, and coercions
- Statements were treated as already correct; the task was proof repair

Mizar Autoformalization Progress

Coarse growth

- Fixed 204 statements, 35 definitions, 56 registrations
- 'Iteration 1': 2,697 lines, 274 unresolved incomplete proofs
- 'Iteration 100': 20,251 lines, first half proved.

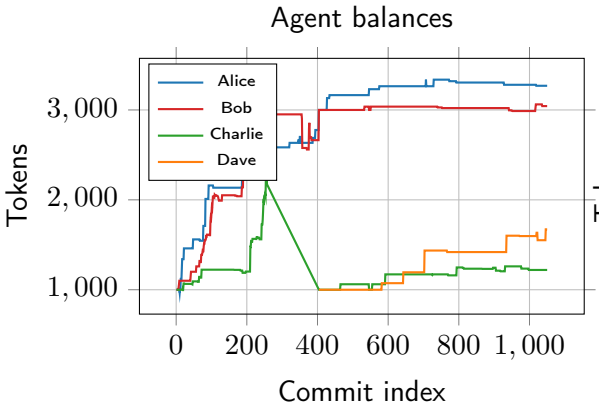
Error-code progression

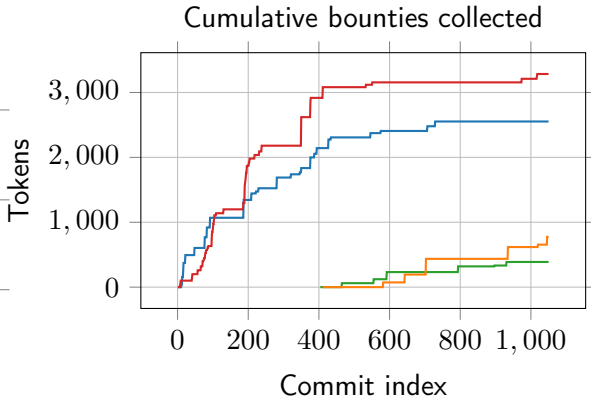
- '70': "something remains to be proved" dominated the early file
- '73' at 'Def21': correctness condition missing
- later '4': inference not accepted in local proof steps
- early side effects also showed '103', '140', '216', '391', '396'

Overview

- Close open proof holes, expose proof-shape, typing failures
- Then pushing the non-accepted inferences
- Completed proofs twice longer than human ones

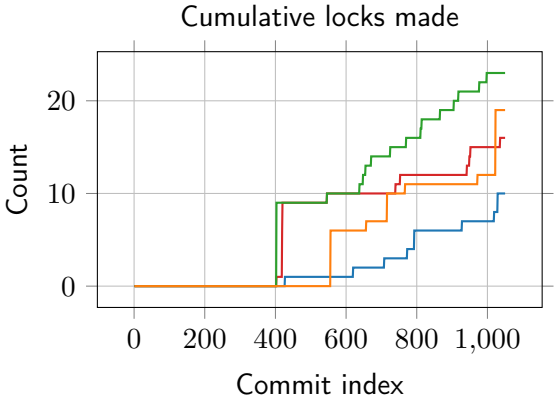
Other Run Graphs: Balances and Collections

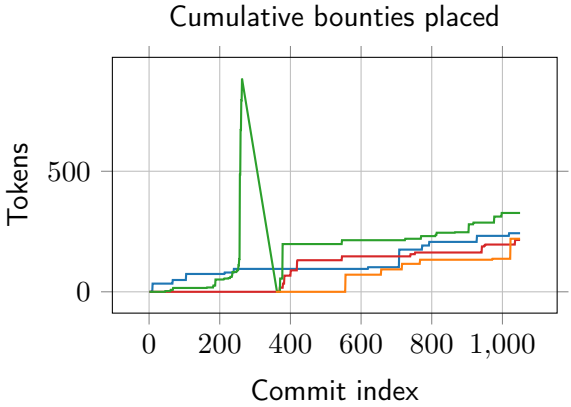




Charlie's balance reset

Other Run Graphs: Locks and New Bounties





“It is rude to show AI output to people”

“It is rude to show AI output to people”

1. Commit-level successful edit:

`https://github.com/mgwiki/alg_top/commit/1a06db28af958cc8d2d544e29cf790778e2540d2`

   - Commit message: `Prove simply_connected_R_standard via explicit loop contraction`
   - Date: 2026-02-20, changes: +622 / -23 lines

2. Rendered formalization browser:

`http://grid01.ciirc.cvut.cz/~cek/demo/AlgTop.html`

Conclusion

AI for Better Automation

LLMs for Autoformalization

- Autoformalization is feasible; automation-first workflow $\rightarrow$ slightly more readable
- Gains from multi-agent coordination

Conclusion

AI for Better Automation

LLMs for Autoformalization

- Autoformalization is feasible; automation-first workflow → slightly more readable
- Gains from multi-agent coordination

Future

- "Ugly" formal proofs
- Autoformalization is slow: Comparing them is really hard
- "Lessons Learned" file: So far did not give improvement
- Improve dependency control for long-running work

Conclusion

AI for Better Automation

LLMs for Autoformalization

- Autoformalization is feasible; automation-first workflow → slightly more readable
- Gains from multi-agent coordination

Future

- "Ugly" formal proofs
- Autoformalization is slow: Comparing them is really hard
- "Lessons Learned" file: So far did not give improvement
- Improve dependency control for long-running work

Ultimately: Autoformalize all of math... Programs?