Benchmarks for AI-assisted Formal Verification

Lacramioara Astefanoaei*, Sergiu Bursuc*, Oliver Butterley*, Markus Ferdinand Dablander*, Quinn Dougherty*, Theodore Ehrenborg*, Jure Kukovec*, Shaowei Lin*, Alok Singh*, Adem Bizid, Ionel Emilian Chiosa, Alessandro D'Angelo, Debojoti Das Soumya, Máté Kovács, Zhang Liao, S M A Nahian, Max Tan, Teimurazi Toloraia, Hoang Le Truong, Leo Yao, Miranda Zhao, Max Tegmark

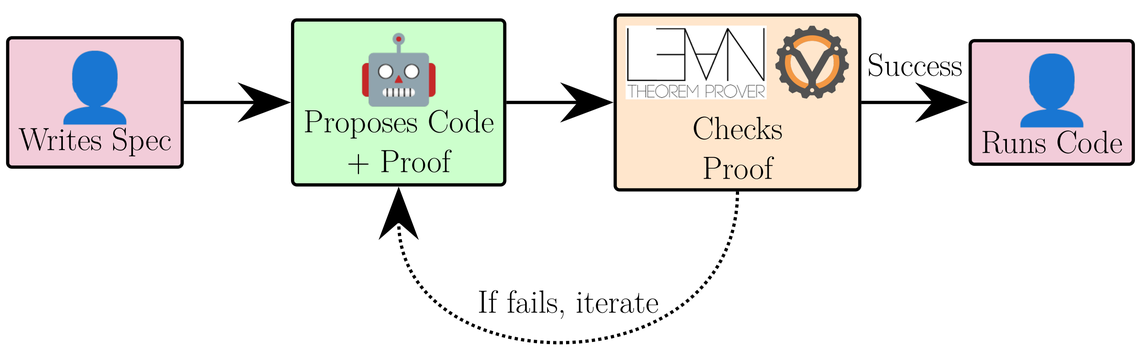
*Primary contributor (alphabetic order)

BENEFICIAL AI FOUNDATION

arXiv:2509.22908

Introduction

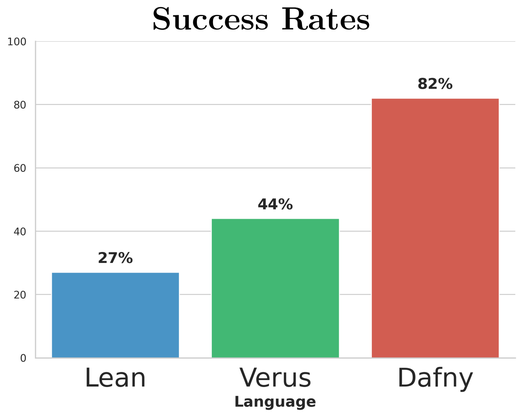
- Formal proofs can be automatically verified even if the writer of the proof is untrusted
- The most impressive LLM-generated proofs have been for mathematical theorems (e.g. [4, 7]), not formal verification (FV) of software
- Better benchmarks for AI-assisted FV will speed up development of AI tools for FV
- End goal is that when AI writes software, it must prove that the code meets a human-written formal specification



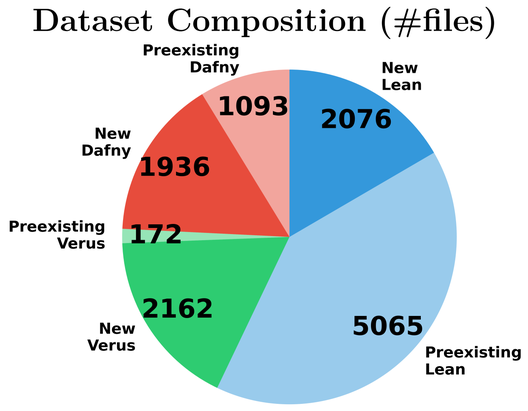
Benchmark 1: Large Multi-Language Dataset

- Standardized format across source benchmarks, see `https://huggingface.co/datasets/beneficial-ai-foundation/vericoding`
- Augmented data with LLM translators and performed quality assurance
- DafnyBench [6] found success rate of 68% with Opus 3.1, but now 89% with Opus 4.1

Success Rates



Dataset Composition (#files)



Discussion

- Future work: Finetune Goedel-Prover [5] and other open-source models
- FV works best for compilers, cryptography, kernels, and parsers [2]
- FV can model the APPS examples from the original AI control paper [3], but not the examples with side effects from BashBench2 [9]

References

[1] Verus Contributors. *Verus Logo*. 2025. URL: `https://verus-lang.github.io/verus/verus/logo.html`.

[2] Mike Dodds. *Specifications Don't Exist*. Galois, Inc. June 2025. URL: `https://www.galois.com/articles/specifications-dont-exist`.

[3] Ryan Greenblatt et al. *AI Control: Improving Safety Despite Intentional Subversion*. 2024. arXiv: `2312.06942 [cs.LG]`.

[4] Harmonic. *How Aristotle Achieved its IMO Gold Medal-Level Performance*. Oct. 2025. URL: `https://harmonic.fun/news#blog-post-aristotle-tech-report`.

[5] Yong Lin et al. *Goedel-Prover: A Frontier Model for Open-Source Automated Theorem Proving*. 2025. arXiv: `2502.07640 [cs.LG]`.

[6] Chloe Loughridge et al. *DafnyBench: A Benchmark for Formal Software Verification*. 2024. arXiv: `2406.08467 [cs.SE]`.

[7] Thang Luong and Edward Lockhart. *Advanced version of Gemini with Deep Think officially achieves gold-medal standard at the International Mathematical Olympiad*. July 2025. URL: `https://deepmind.google/blog/advanced-version-of-gemini-with-deep-think-officially-achieves-gold-medal-standard-at-the-international-mathematical-olympiad/`.

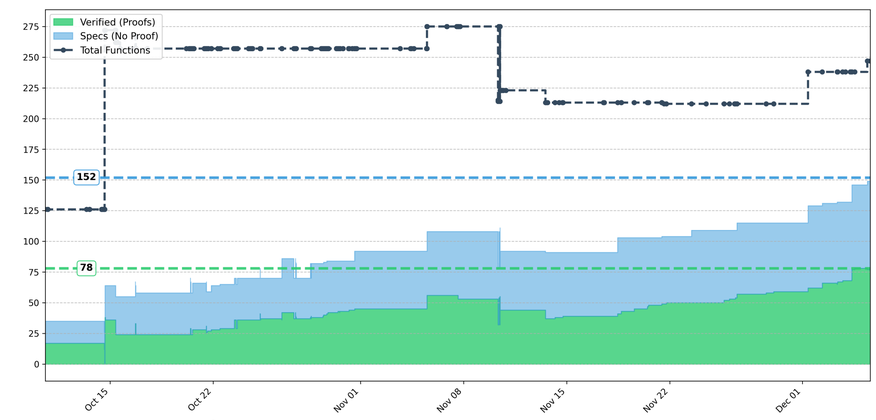
[8] Leonardo de Moura. *Lean 4 Logo*. 2014. URL: `https://github.com/leanprover/lean4/blob/master/images/lean.png`.

[9] UK AI Security Institute. *BashBench2*. ControlArena. 2025. URL: `https://control-arena.aisi.org.uk/settings/bashbench2.html`.

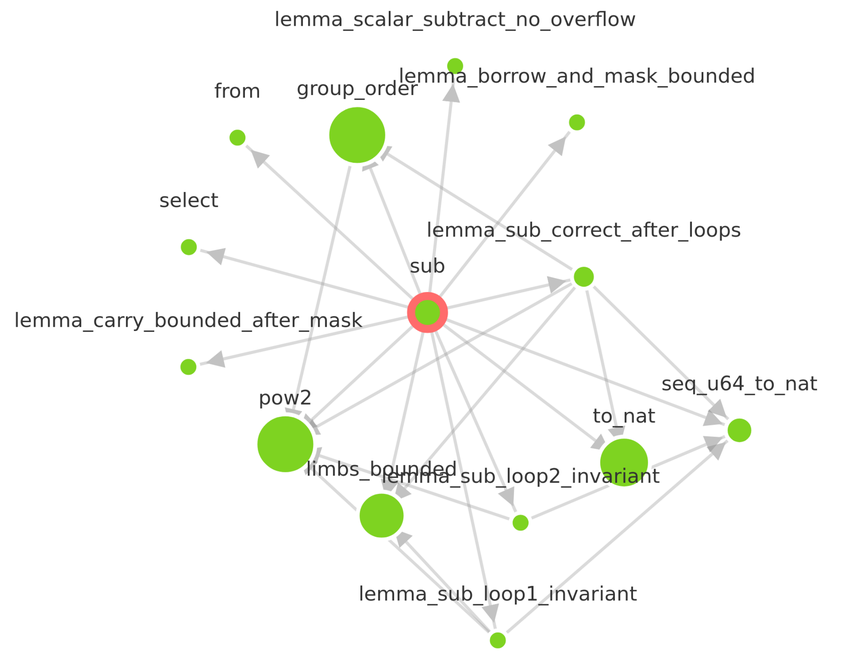
Benchmark 2: Real-World Specifications

Based on `curve25519-dalek` cryptography library, used by Signal app

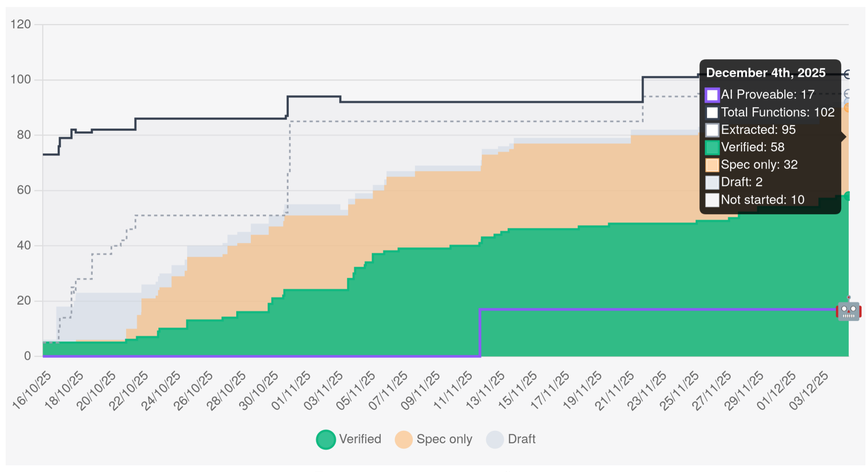
- Gold standard specifications in Lean and Verus, manually proved to match code
- Property-based testing can cheaply disprove specifications
- Some proofs take days for human experts, >1000 lines of code



Verus progress for `curve25519-dalek` `https://beneficial-ai-foundation.github.io/dalek-lite`



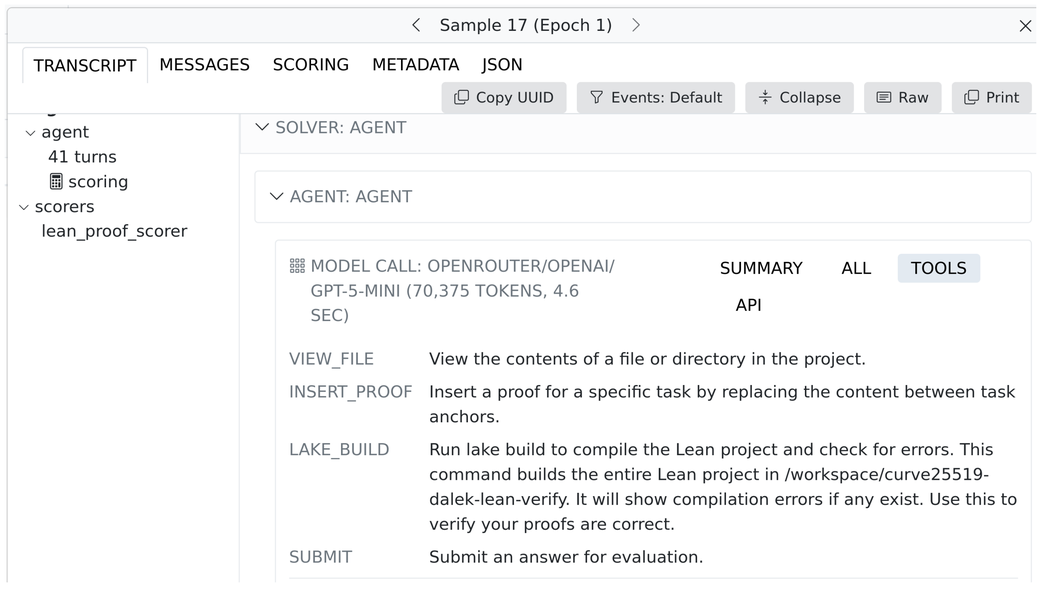
Depth 1 dependency graph of lemmas needed to verify Scalar52 subtraction in `curve25519-dalek`, using Verus. Interactive version:
`https://beneficial-ai-foundation.github.io/scip-callgraph`



Lean progress for `curve25519-dalek`
`https://beneficial-ai-foundation.github.io/curve25519-dalek-lean-verify`

Preliminary Results (Inspect AI with GPT-5-mini):

- Successfully replicated 17/58 Lean proofs (29%)



Tools available to the LLM within the Inspect AI framework. The INSERT_PROOF tool does not allow the LLM to cheat by changing the theorem