

Big Proof

Jeremy Avigad (Carnegie Mellon University), Georges Gonthier (Microsoft Research),
Ursula Martin (Oxford University), J Strother Moore (University of Texas at Austin, Emeritus),
Lawrence Paulson (Cambridge University), Andrew Pitts (Cambridge University),
and Natarajan Shankar (SRI International)

Proofs as constructive demonstrations of mathematical validity have been at the heart of mathematics since antiquity. Formal proof systems capture the definitions, statements, and proofs of mathematical discourse using precisely defined formal languages and rules of inference. Formal proofs have enabled mathematicians to rigorously explore foundational issues of expressiveness, consistency, independence, completeness, computability, and decidability. The formalization of proof facilitates the representation and manipulation of mathematical knowledge with modern digital computers. During the last sixty years, the digitization of formal mathematics has yielded satisfiability solvers, rewriting engines, computer algebra systems, automated theorem provers, and interactive proof assistants. Proof technology can be used to perform large calculations reliably, solve systems of constraints, discover and visualize examples and counterexamples, simplify expressions, explore hypotheses, navigate large libraries of mathematical knowledge, capture abstractions and patterns of reasoning, and interactively construct proofs. The scale and sophistication of proof technology is approaching a point where it can effectively aid human mathematical creativity at all levels of expertise. Modern satisfiability solvers can efficiently solve problems with millions of Boolean constraints in hundreds of thousands of variables. Automated theorem provers have discovered proofs of open problems. Interactive proof assistants have been used to check complicated mathematical proofs such as those for the Kepler's conjecture and the Feit-Thompson odd order theorem. Such systems have also been applied to the verification of practical artifacts such as central processing units (CPUs), compilers, operating system kernels, file systems, and air traffic control systems. Several high-level programming languages employ logical inference as a basic computation step.

The six week program on Big Proof at the Isaac Newton Institute is directed at the challenges of bringing proof technology into mainstream mathematical practice. The specific challenges addressed in the Isaac Newton Institute Big Proof program include

1. Novel pragmatic foundations for representing mathematical knowledge and vernacular inspired by set theory, category theory, and type theory.
2. Large-scale formal mathematical libraries that capture background knowledge spanning a range of domains.
3. Algorithmic and engineering issues in building and integrating large-scale inference engines.
4. The social exploration and curation of formalized mathematical and scientific knowledge.
5. Educational proof technology in support of collaborative learning.

The Big Proof program brings together mathematicians interested in employing proof technology in their research, logicians exploring pragmatic and foundational issues in the formalization of mathematics, and computer scientists engaged in developing and applying proof technology. The program includes a week-long workshop exploring foundational, theoretical, and practical challenges in exploiting proof technology to transform mathematical practice across a range of scientific and engineering disciplines. A key expected output from the Big Proof program is a concrete, long-term research agenda for making computational inference a basic technology for formalizing, creating, curating, and disseminating mathematical knowledge in digital form.