Mechanizing Mathematical Proof

Why do it?

What is it like to do?

How are we getting on with it?

Rob Arthan

Lemma 1 Ltd. / Queen Mary University of London

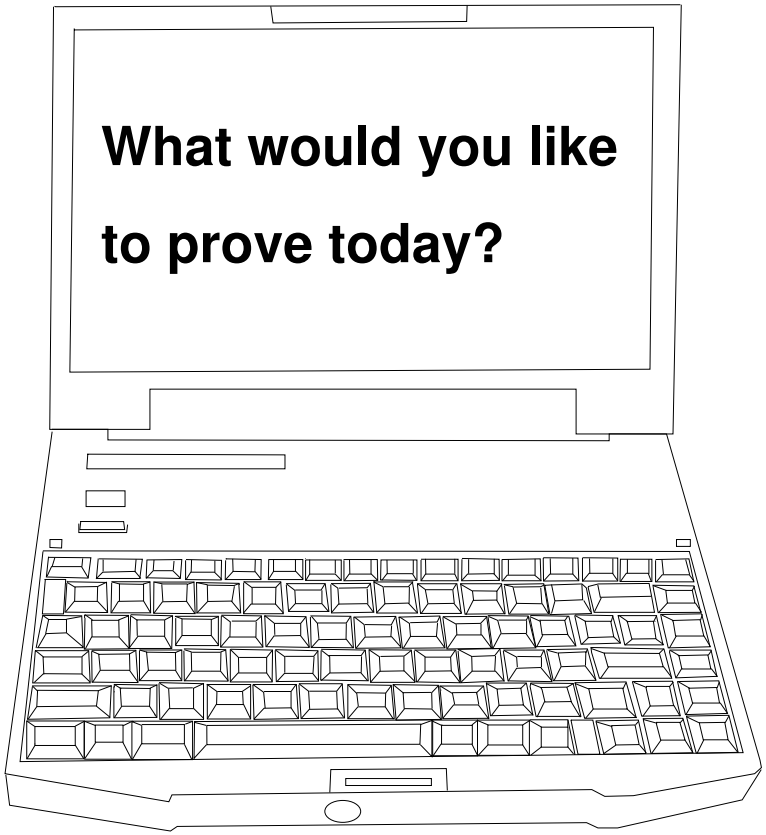
Overview of the Talk

Mechanizing Mathematical Proof

1. Why do it?
   - Thoughts and some opinions (ca. 2000)
   - Perspectives: mathematician, engineer, computer scientist
   - The Kepler Conjecture
2. What is it like to do?
   - The LCF paradigm via a toy example
   - LCF as realised in ProofPower
   - Some mathematical case studies in ProofPower-HOL:
     – some calculus, a little finite combinatorics
3. How are we getting on with it?
   - Some opinions (ca. 2025)
   - Some significant achievements
   - Final remarks

A Hypothetical Question

What would Newton or the Bernoullis or Gauss or Delaunay or Hilbert or . . . your list goes here . . . have done with one of these:



?

Formalising Mathematics ca. 1910: $1 + 1 =$ hmmm?

"The properties of $\dot{2}$ *are largely analogous to those of* 1*, while the properties of* 2_r *are more analogous to those of* 2*."*

A.N. Whitehead and B. Russell *Principia Mathematica *56* **p. 375** 1910

What about properties of H_n or e or π or $\pi_4(S^3)$ or . . . ?

- The immense symbolic processing tasks defy human capabilities.
- But 100 years on we have machines to do all that. . . . However
  - Machines don't know what symbols to process.
  - Synergy between human and computer is what's called for.

So I have to do some work? Is it worth it?

An A – Z of Mathematical Opinions (ca. 2000)

"I will say nothing, for example, about the great events in the area between logic and computing"

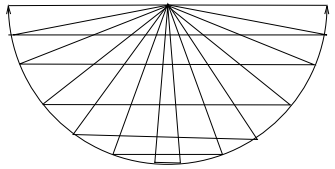
Sir Michael Atiyah. *Mathematics in the 20th Century.*

⋮

"But the MOST significant contribution by 20th-Century human mathematicians and computer scientists was the creation of COMPUTER ALGEBRA."

Doron Zeilberger. *Opinion 47.*

Mathematician: Why bother?



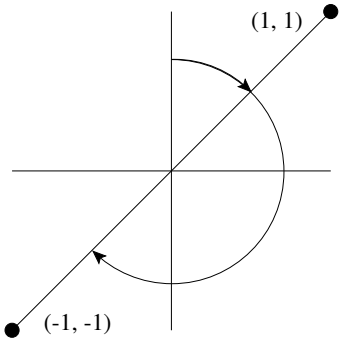
- Mathematical arguments do sometimes contain errors!
- How many in the classification of finite simple groups?
- Even Aigner & Ziegler's *"Proofs from THE BOOK"* (ed. 1).

  The area of a triangle is monotone in the side lengths (that is, if two side lengths stay the same and the third one increases, then the area increases).

- Avoid controversies about use of computers:
  - 4 colour theorem
  - Kepler sphere packing conjecture (of which, more later)
- Potential to deal automatically with circle-squarers, cube-doublers, angle-trisectors and misguided claims about $\mathsf{P} \neq \mathsf{NP}$!

Engineer: Why bother?

A simple example: calculating bearings



- Specification: take $\beta_{(x,y)}$ chosen in the appropriate quadrant where:

$$\mathsf{tan}\ \beta_{(x,y)} = \frac{x}{y}$$

- Design: calculate the bearing using: $\beta = \mathsf{tan}^{-1}(x/y)$
- Implementation: `beta := atan(x/y);`
- BUT: $\beta_{(1,1)} = \pi/4 \neq 5\pi/4 = \beta_{(-1,-1)}$

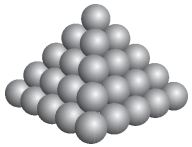
Testing tends not to find subtle errors.

Computer Scientist: Why bother?

Computer scientists...

- ...prove theorems
- ...specify, design and implement hardware and software

So see under Mathematician and Engineer above!



The Kepler Conjecture

> The density of a packing of congruent spheres in three dimensions is never greater than $\pi/\sqrt{18}$.

- Asserted by Kepler in 1611.
- 1998 proof by Hales reduces a nonlinear optimisation problem in an infinite number of variables to a problem in a finite number of variables (≈ 150).
- The finite problem then further reduces to:
  - Enumerating a large set of so-called "tame" graphs describing possible local geometry.
  - A large set of linear programming problems.
  - A large set of non-linear inequalities.
- These subproblems are then solved by extensive computation.

Referees' Verdict

Published subject to a disclaimer.

"The news from the referees is bad .... They have not been able to certify ... the proof, and will not be able to certify it ..., because they have run out of energy to devote to the problem."

"[The chief of the 12 referees] thinks that this situation will occur more and more often in mathematics. He says it is similar to the situation in experimental science — other scientists acting as referees can't certify the correctness of an experiment, they can only subject the paper to consistency checks. He thinks that the mathematical community will have to get used to this state of affairs."

Robert MacPherson. Editor of *Annals of Mathematics*

At least one member of the community disagreed ...

Hales's Flyspeck Project

- Goal: a complete machine-checked formal proof. (Achieved 2015).
- Effort: about 20 – 25 person/years from 2003 to 2015.
- Blueprint of the proof. Hales.
- Enumeration of tame planar graphs verified in Isabelle-HOL. Nipkow, Bauer.
- Transformation of non-linear inequalities into linear inequalities.

  Bauer, Zumkeller.
- Tools to handle the linear programming problems and linear inequalities.

  Obua, Bauer, Zumkeller, Solovyev.
- Main reduction in HOL Light.

  Hales, Harrison and others including a team of MSc. students in Hanoi proving lemmas for cash bounties under Mark Adams as project manager.

But where does the assurance come from?

LCF — A Paradigm for High Assurance Proof Tools

- Robin Milner's LCF is an approach that offers:
  - high assurance ($1 \neq 2$)
  - extensibility and programmability
  - access to a growing library of existing results
  - ability to carry out highly assured specific calculations
- In the next two parts of the talk I hope to:
  - Show how the LCF paradigm offers high assurance
  - Give a flavour of what it is like to develop a theory like the calculus in an LCF-style system

LCF stands for **L**ogic of **C**omputable **F**unctions, a specific logic for program verification. But forget that: the approach works for any logic.

Characteristics of the LCF Approach

- *"All computer programs have bugs!"*
  — How can we mitigate the risk of proving $1 = 2$?
- Robin Milner's idea: build system on top of a logical kernel that is small and feasible to check.
- Use a strongly typed programming language that prevents code bypassing the kernel.
- Untrusted code may fail to produce desired output but can't produce unsound results.
- Allows safe coding of very complex proof-finding algorithms.

A Deductive System

Judgments: $m \mathsf{D} n$, where $m, n \in \mathbb{Z}$

(intended meaning: m divides n)

Inference rules:

$$\frac{}{m \mathsf{D} m} : \text{axiom}, m \neq 0 \qquad \frac{m \mathsf{D} n}{m \mathsf{D} -n} : - \qquad \frac{m \mathsf{D} n_1 \quad m \mathsf{D} n_2}{m \mathsf{D} n_1 + n_2} : +$$

A deduction:

$$\dfrac{\dfrac{}{1 \mathsf{D} 1} : \text{axiom} \quad \dfrac{\dfrac{}{1 \mathsf{D} 1} : \text{axiom}}{1 \mathsf{D} -1} : -}{1 \mathsf{D} 0} : +$$

- LCF implements the deductive system as an abstract data type.
- ADT values can only be constructed by applying inference rules.

The Deductive System in ML

Demo 1

```
local
  datatype THEOREM = D of (int * int);
in
  type THEOREM = THEOREM;
  infix D; infix ++;
  exception NotAllowed;

  fun axiom m = if m <> 0 then m D m else raise NotAllowed;
  fun -- (m D n) = m D ~n;
  fun (m1 D n1) ++ (m2 D n2) =
    if m1 = m2 then m1 D (n1 + n2) else raise NotAllowed;
end;
```

A Decision Procedure

Demo 1 concluded.

Given `m` and `n`, the function `decide` tries to prove that `m` divides `n`:

```
fun decide m n =
  if n < 0 then --(decide m (~n))
  else if n <= m then axiom m
  else decide m (n-m) ++ axiom m;
```

Logical kernel will not allow invalid deductions:

```
- decide 2 6;
val it = 2 D 6 : THEOREM
- decide 2 7;
val it = 2 D 8 : THEOREM
- decide 0 0;
Exception- NotAllowed raised
- decide ~1 1; (* Doesn't terminate *)
```

A Real System: ProofPower-HOL

Demo 2

- Member of the HOL family implemented to meet some specific requirements for industrial use (primarily support for multiple object languages via semantic embedding into HOL). Jones, Arthan *et al.*, 1989 – present
- Cf. Classic HOL (Gordon), HOL IV (Slind, Norrish), HOL Light (Harrison), HOL Zero (Adams), Candle (Kumar, Myreen, Owens).
- Expressions and predicates represented by the type $TERM$, entered using "Quine corners": $\ulcorner 1 \ + \ 2 \urcorner$, $\ulcorner 1 \ = \ 2 \urcorner$
- Abstract data type of theorems is the type THM. Printed with a turnstile: $\vdash \ \neg 1 \ = \ 2$.
- Extensive facilities for programming with syntax.
- Maintains a database of theories containing specifications (i.e., defining properties of types and constants) and theorems.
- Powerful higher-level tools for automated and interactive proof.

Definitions In ProofPower-HOL

Demo 2 continued

- Syntax for defining new constants (including sets, functions, functionals etc.) with an arbitrary defining property:

$$\begin{array}{|l} \textbf{<NAME>} : <TYPE> \\ \hline <DEFINING\ PROPERTY> \end{array}$$

- Proof obligation to verify consistency (often discharged automatically).

HOL Constant

$$\begin{array}{|l} \textbf{Even} : \mathbb{N}\ SET \\ \hline \forall n \bullet\ n \in Even \Leftrightarrow (\exists m \bullet n = 2*m) \end{array}$$

HOL Constant

$$\begin{array}{|l} \textbf{Odd} : \mathbb{N}\ SET \\ \hline \forall n \bullet\ \neg\ 2*n \in Odd \wedge 2*n+1 \in Odd \end{array}$$

Calculus In ProofPower-HOL 1

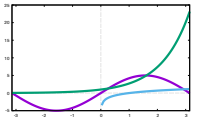
- Write $(f\ \mathit{Deriv}\ c)\ x$ to mean function f has derivative c at x (i.e., $df/dx = c$ or $f'(x) = c$ in the vernacular).

$$\$\boldsymbol{Deriv} : (\mathbb{R} \to \mathbb{R}) \to \mathbb{R} \to \mathbb{R} \to \mathit{BOOL}$$

$$\forall f\ c\ x \bullet (f\ \mathit{Deriv}\ c)\ x$$
$$\Leftrightarrow\ \ \forall e \bullet\ 0. < e \Rightarrow \exists d \bullet\ 0. < d \wedge$$
$$\forall y \bullet\ \mathit{Abs}(y - x) < d \wedge \neg y = x \Rightarrow \mathit{Abs}((f\ y - f\ x)/(y - x) - c) < e$$

- This definition is trivially consistent.
- All the usual theorems: product rule, chain rule, Rolle, IVT, MVT, ...

Calculus In ProofPower-HOL 2



- Define the exponential function by the differential equation:

$$\boldsymbol{Exp} : \mathbb{R} \to \mathbb{R}$$

$$\vdash \quad Exp\ 0. = 1. \wedge (\forall x \bullet (Exp\ Deriv\ Exp\ x)\ x)$$

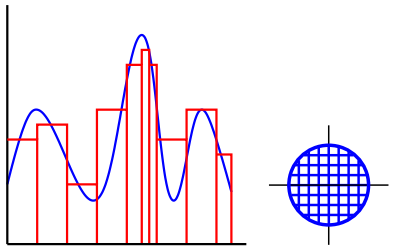
- Prove the consistency via theory of power series and differentiation of limits.
- Logarithm defined as left inverse of exponential.
- All the usual basic theorems, e.g.,

$$\vdash \forall\ x \bullet\ 0. < x \Rightarrow (Log\ Deriv\ x\ ^{-1})\ x : THM$$

- Similar treatment of trigonometric functions.

Calculus In ProofPower-HOL 3

Demo 2 continued



- Integration via the Kurzweil-Henstock gauge integral.
  - Fundamental theorem of calculus

$$\vdash \forall\ a\ b\ sf\ f \bullet$$
$$a < b \wedge (\forall\ x \bullet\ a < x \wedge x < b \Rightarrow (sf\ \ Deriv\ f\ x)\ x)$$
$$\wedge\ sf\ \ Cts\ a \wedge sf\ \ Cts\ b\ (*\ Don't\ need\ to\ assume\ f\ is\ integrable\ *)$$
$$\Rightarrow (f\ \ Int\ (sf\ \ b - sf\ \ a))\ (ClosedInterval\ a\ b):$$

  - Areas, etc.

- A theorem from antiquity:

$$\vdash \forall\ r \bullet\ 0. < r \Rightarrow \{(x, y) | Sqrt\ (x\ \hat{}\ 2 + y\ \hat{}\ 2) \leq r\}\ Area\ \pi * r\ \hat{}\ 2$$

- de Bruijn factor: $\dfrac{\text{size of formal proof}}{\text{size of informal proof}} \sim 0.5 \ldots 5.$

Example: A Combinatorics Problem 1

- $(m+n)^m$ should give the number of ways of drawing m samples with replacement out of a set of $m+n$ elements.
- $DistinctSamples\ n\ m$ should give the number of ways of drawing m samples without replacement out of a set of $m+n$ elements.

$$
\begin{array}{|l}
\boldsymbol{DistinctSamples} : \mathbb{N} \to \mathbb{N} \to \mathbb{N} \\
\hline
\\
\quad (\forall n \bullet\ DistinctSamples\ n\ 0 = 1) \\
\land\ (\forall m\ n \bullet \\
\quad DistinctSamples\ n\ (m+1) = (n+m+1) * DistinctSamples\ n\ m)
\end{array}
$$

Example: A Combinatorics Problem 2

Demo 2 concluded

- Plan: give high assurance solution to a combinatorics problem by:
  - 1) Proving that $(m+n)^m$ and $DistinctSamples\ n\ m$ do give the desired results.
  - 2) Symbolically executing them inside the theorem prover.
- For 1) we prove:

$$\begin{array}{ll}
\vdash \forall\ a\ n \bullet & a \in Finite \\
\quad \Rightarrow & \{L | Elems\ L \subseteq a \wedge \#\ L = n\} \in Finite \\
\quad \wedge & \#\{L | Elems\ L \subseteq a \wedge \#\ L = n\} = \#\ a\ \widehat{}\ n \\
\vdash \forall\ a\ n \bullet & a \in Finite \wedge n \leq \#\ a \\
\quad \Rightarrow & \{L | Elems\ L \subseteq a \wedge \#\ L = n \wedge L \in Distinct\} \in Finite \\
\quad \wedge & \#\{L | Elems\ L \subseteq a \wedge \#\ L = n \wedge L \in Distinct\} \\
 & \quad = DistinctSamples\ (\#\ a - n)\ n
\end{array}$$

- For 2), let's try a calculation.

Example: A Combinatorics Problem 3

- What we have done is to calculate a discrete probablity:

$$\begin{aligned}
&\vdash \mathit{let}\ S = \{L \mid \mathit{Elems}\ L \subseteq \{i | 1 \leq i \wedge i \leq 365\} \wedge \#\ L = 23\} \\
&\quad \mathit{in}\ \mathit{let}\ X = \{L \mid L \in S \wedge \neg\ L \in \mathit{Distinct}\} \\
&\quad \mathit{in}\ S \in \mathit{Finite} \wedge \neg\ \#S = 0 \wedge X \subseteq S \wedge \#X/\#S > 1/2
\end{aligned}$$

  Does anyone recognise that one?

- And we can calculate probabilities over continuous spaces too:

$$\begin{aligned}
&\vdash \mathit{let}\ S = \{(\theta,\ d) \mid \theta \in \mathit{ClosedInterval}\ 0.\ \pi \wedge d \in \mathit{ClosedInterval}\ 0.\ 1.\} \\
&\quad \mathit{in}\ \mathit{let}\ x_axis = \{(x,\ y) \mid y = 0.\} \\
&\quad \mathit{in}\ \mathit{let}\ needle\ (\theta,\ d) = \{(x,\ y) \mid \exists\ t\bullet\ t \in \mathit{ClosedInterval}\ 0.\ 1. \\
&\qquad\qquad \wedge\ x = t * \mathit{Cos}\ \theta \wedge y = d - t * \mathit{Sin}\ \theta\} \\
&\quad \mathit{in}\ \mathit{let}\ X = \{(\theta,\ d) \mid (\theta,\ d) \in S \wedge \neg\ needle\ (\theta,\ d) \cap x_axis = \{\}\} \\
&\quad \mathit{in}\ X \subseteq S \wedge (\exists x\ s\bullet \neg\ s = 0. \wedge X\ \mathit{Area}\ x \wedge S\ \mathit{Area}\ s \wedge x/s = 2./\pi)
\end{aligned}$$

  That's the solution to Buffon's needle problem.

Other Real Systems

- Many systems around, LCF style and other approaches.
- Deductive system usually mathematical foundation system, e.g.,
  - Set theory – Mizar, Isabelle-ZF, Metamath
  - Polymorphic type theory — HOL family, Coq, PVS, Lean
  - Higher order set theory – Megalodon
  - Theory of recursive functions – NQTHM, ACL2
- Many systems inspired by computer science applications, e.g., program verification, but turn out to be general purpose.
- Most systems now have very extensive mathematical libraries.

But can you prove real theorems?

An A – Z of Mathematical Opinions (ca. 2025)

"If AI is the problem, then mechanized mathematics is the solution."

Jeremy Avigad. *Title of a talk to the US National Museum of Mathematics 2024.*

⋮

"I now clearly understand that software such as Lean is part of the inevitable future of mathematics. . . . and now I never want to go back to pen and paper mathematics — I am beginning to mistrust it."

Kevin Buzzard. *The future of mathematics? Talk to Microsoft Research 2019.*

⋮

"Many generations of students shed tears trying to "prove" . . . propositions, by a step-by-step "deductive" process."

Doron Zeilberger *Opinion 174.*

Some Achievements 1

A personal selection — no warranty offered or implied!

- Mizar Mathematical Library. Trybulec *et al.* 1970s – present
- Calculus. Harrison, Gottliebsen, Arthan, *et al.* 1990s – present
- Nonstandard analysis. Fleuriot, 1996 – present
- Gödel's incompleteness theorem. Shankar, 1994
- Sylow's theorem. Kammüller. 1997.
- Prime Number Theorem. Avigad *et al.*, 2004, Harrison, 2008
- Isabelle Archive of Formal Proofs Nipkow *et al.*, 2004 – present
- Jordan Curve Theorem. Hales, 2005
- Brouwer fixed point theorem. Harrison, 2005
- 4 Colour Theorem Gonthier, 2005

Some Achievements 2

See previous slide for Terms & Conditions.

- CakeML — Kumar, Myreen, Owens *et al.*, 2013 – present
- Flyspeck — Hales *et al.*, 2003 – 2015
- CompCert — Leroy *et al.*, (INRIA) 2005 – present
- Feit-Thompson Odd Order Theorem — Gonthier *et al.*, 2006 – 2012
- seL4 — Klein *et al.* (NICTA/UNSW), 2006 – 2009
- Lean mathlib — Carneiro *et al.*, 2021 – present
- Fermat's Last Theorem (WIP[a]) — Buzzard *et al.*, 2025 – present

[a] It's very early days, but, hey, they got the grant! ⌣̈

Links 1

- Freek Wiedijk's progress report on 100 theorems:
  - http://www.cs.ru.nl/~freek/100/index.html
- ProofPower and ProductHOL
  - Home – http://www.lemma-one.com/ProofPower/index.html
  - Case Studies – https://www.lemma-one.com/ProofPower/examples/examples.html
- Other implementations of HOL:
  - HOL IV – http://hol.sourceforge.net/
  - Isabelle-HOL – http://www.cl.cam.ac.uk/research/hvg/Isabelle/
  - HOL Light – http://www.cl.cam.ac.uk/~jrh13/hol-light/
  - HOL Zero – http://www.proof-technologies.com/holzero/
  - Candle – https://cakeml.org/candle/
- OpenTheory (proof interchange between the different HOLs):
  - https://gilith.com/opentheory/
- Isabelle Archive of Formal Proofs: https://www.isa-afp.org

Links 2

- The Flyspeck project:
  - Source repo – https://github.com/flyspeck/flyspeck
  - Final report – https://arxiv.org/abs/1501.02155
- CakeML: https://cakeml.org
- Some other powerful interactive theorem provers (not LCF-based):
  - Mizar – http://mizar.uwb.edu.pl/
  - PVS – https://pvs.csl.sri.com
  - Metamath – https://us.metamath.org/index.html
  - Rocq (Coq) – https://en.wikipedia.org/wiki/Rocq_(software)
  - Lean – https://lean-lang.org
  - Megalodon – http://grid01.ciirc.cvut.cz/~chad/megalodon/index.php
- Feit-Thompson in Coq: https://github.com/math-comp/odd-order
- CompCert: https://compcert.org/compcert-C.html
- seL4: https://trustworthy.systems/projects/OLD/seL4-verification/
- Lean mathlib: https://github.com/leanprover-community/mathlib4

Final Remarks

- Machine-checked proof has an important rôle to play.
- Technology can help: critical bugs are not inevitable!
- Formalisation need not lead to combinatorial explosion.
- Formalisation can be instructive and even fun.
- We have come a long way since p. 375 of Whitehead & Russell.
- Lots of fascinating work still to be done.

Thank you!