Formalised Mathematics: Obstacles and Achievements

Lawrence Paulson

UNIVERSITY OF CAMBRIDGE

Zhejiang University, 15 September 2021

Supported by the ERC Advanced Grant ALEXANDRIA (742178).

Formalised Mathematics

Why do maths by machine?

To *validate* dubious proofs

To *reveal* hidden assumptions

To *codify* mathematical knowledge

But the main reason is…

Mathematicians are fallible

The footnotes on a **single page** (118)
of Jech's *The Axiom of Choice*

[1] The result of Problem 11 contradicts the results announced by Levy [1963b]. Unfortunately, the construction presented there cannot be completed.
[2] The transfer to ZF was also claimed by Marek [1966] but the outlined method appears to be unsatisfactory and has not been published.
[3] A contradicting result was announced and later withdrawn by Truss [1970].
[4] The example in Problem 22 is a counterexample to another condition of Mostowski, who conjectured its sufficiency and singled out this example as a test case.
[5] The independence result contradicts the claim of Felgner [1969] that the Cofinality Principle implies the Axiom of Choice. An error has been found by Morris (see Felgner's corrections to [1969]).

Mathematicians are fallible, II

"When the Germans were planning to publish Hilbert's collected papers …, they realized that they could not publish the papers in their original versions because they were full of errors, some of them quite serious. Thereupon they hired a young unemployed mathematician, Olga Taussky-Todd, to go over Hilbert's papers and correct all mistakes."

[Gian-Carlo Rota, *Indiscrete Thoughts*, p. 201]

"Olga laboured for three years."

``And who would ensure that I did not forget something and did not make a mistake, if even the mistakes in much more simple arguments take years to uncover?" — Vladimir Voevodsky

Obstacles to Formalisation

Is formalised mathematics possible?

Whitehead and Russell needed 362 pages to prove 1+1=2!

We have better formal systems than *Principia*.

Gödel proved that all reasonable formal systems must be incomplete!

But mathematicians also work from axioms!

Church proved that first-order logic is undecidable!

We want to **assist** people, not to **replace** them.

And yet there **really are** *serious difficulties…*

Definedness, or what is 1/0?

- *Don't care*: all terms denote *something*, and $1/0 = 1/0$. [HOL, Isabelle]

- *Dependent types*: to use x/y, must prove $y \neq 0$ (but does the value of x/y depend on this proof?) [Coq, Lean]

- *Free logic*: a formalism where defined[x/y] can be expressed. So $x/0 = x/0$ is false. But is $x/0 \neq x/0$ true?

Syntax, or the legibility problem

Mathematical notation is elegant, but highly ambiguous

$$f(x) \quad f(X) \quad f^{-1}[X]$$

$$x^{-1}y \quad f^{-1}(x) \quad \sin^{-1}(x) \quad \sin^2(x)$$

$$xy \quad x \cdot y \quad \frac{d^2 f}{dx}$$

`Machine notations are unreadable`

though it's possible to try harder!

Ugly proofs (this is HOL Light)

```
let SIMPLE_PATH_SHIFTPATH = prove
 (`!g a. simple_path g /\ pathfinish g = pathstart g /\
         a IN interval[vec 0,vec 1]
         ==> simple_path(shiftpath a g)`,
  REPEAT GEN_TAC THEN REWRITE_TAC[simple_path] THEN
  MATCH_MP_TAC(TAUT
   `(a /\ c /\ d ==> e) /\ (b /\ c /\ d ==> f)
    ==>  (a /\ b) /\ c /\ d ==> e /\ f`) THEN
CONJ_TAC THENL [MESON_TAC[PATH_SHIFTPATH]; ALL_TAC] THEN
REWRITE_TAC[simple_path; shiftpath; IN_INTERVAL_1; DROP_VEC;
            DROP_ADD; DROP_SUB] THEN
REPEAT GEN_TAC THEN DISCH_THEN(CONJUNCTS_THEN2 MP_TAC ASSUME_TAC) THEN
ONCE_REWRITE_TAC[TAUT `a /\ b /\ c ==> d <=> c ==> a /\ b ==> d`] THEN
STRIP_TAC THEN REPEAT GEN_TAC THEN
REPEAT(COND_CASES_TAC THEN ASM_REWRITE_TAC[]) THEN
DISCH_THEN(fun th -> FIRST_X_ASSUM(MP_TAC o C MATCH_MP th)) THEN
REPEAT(POP_ASSUM MP_TAC) THEN
REWRITE_TAC[DROP_ADD; DROP_SUB; DROP_VEC; GSYM DROP_EQ] THEN
REAL_ARITH_TAC);;
```

Some proofs are 50× longer than this one!

Structured proofs are clearer

```
lemma simple_path_shiftpath:
  assumes "simple_path g" "pathfinish g = pathstart g" and a: "0 ≤ a" "a ≤ 1"
    shows "simple_path (shiftpath a g)"
  unfolding simple_path_def
proof (intro conjI impI ballI)
  show "path (shiftpath a g)"
    by (simp add: assms path_shiftpath simple_path_imp_path)
  have *: "⋀x y. ⟦g x = g y; x ∈ {0..1}; y ∈ {0..1}⟧ ⟹ x = y ∨ x = 0 ∧ y = 1 ∨ x = 1 ∧ y = 0"
    using assms by (simp add:  simple_path_def)
  show "x = y ∨ x = 0 ∧ y = 1 ∨ x = 1 ∧ y = 0"
    if "x ∈ {0..1}" "y ∈ {0..1}" "shiftpath a g x = shiftpath a g y" for x y
    using that a unfolding shiftpath_def
    apply (simp add: split: if_split_asm)
      apply (drule *; auto)+
    done
qed
```

This is the same proof in Isabelle/HOL

Mathematics in Isabelle/HOL

Jordan curve theorem

Central limit theorem

Residue theorem

Prime number theorem

Algebraic closure of a field

Nash-Williams partition theorem

Matrix theory, e.g. Perron–Frobenius

Analytic number theory, eg Hermite–Lindemann

Topological spaces and homology theory

Complex roots via Sturm sequences

Measure, integration and probability theory

Some notable past formalisations

- *Flyspeck project*: verifying the proof of the Kepler Conjecture by Ferguson and Hales (1998), using HOL Light and Isabelle.
- *Four Colour Theorem*: the 1976 proof relied on code, which was finally verified in Coq by Georges Gonthier.
- *Odd order theorem* (Gonthier et al.) [Coq]
- *Gödel's constructible universe* and (both) *incompleteness theorems* (Paulson)

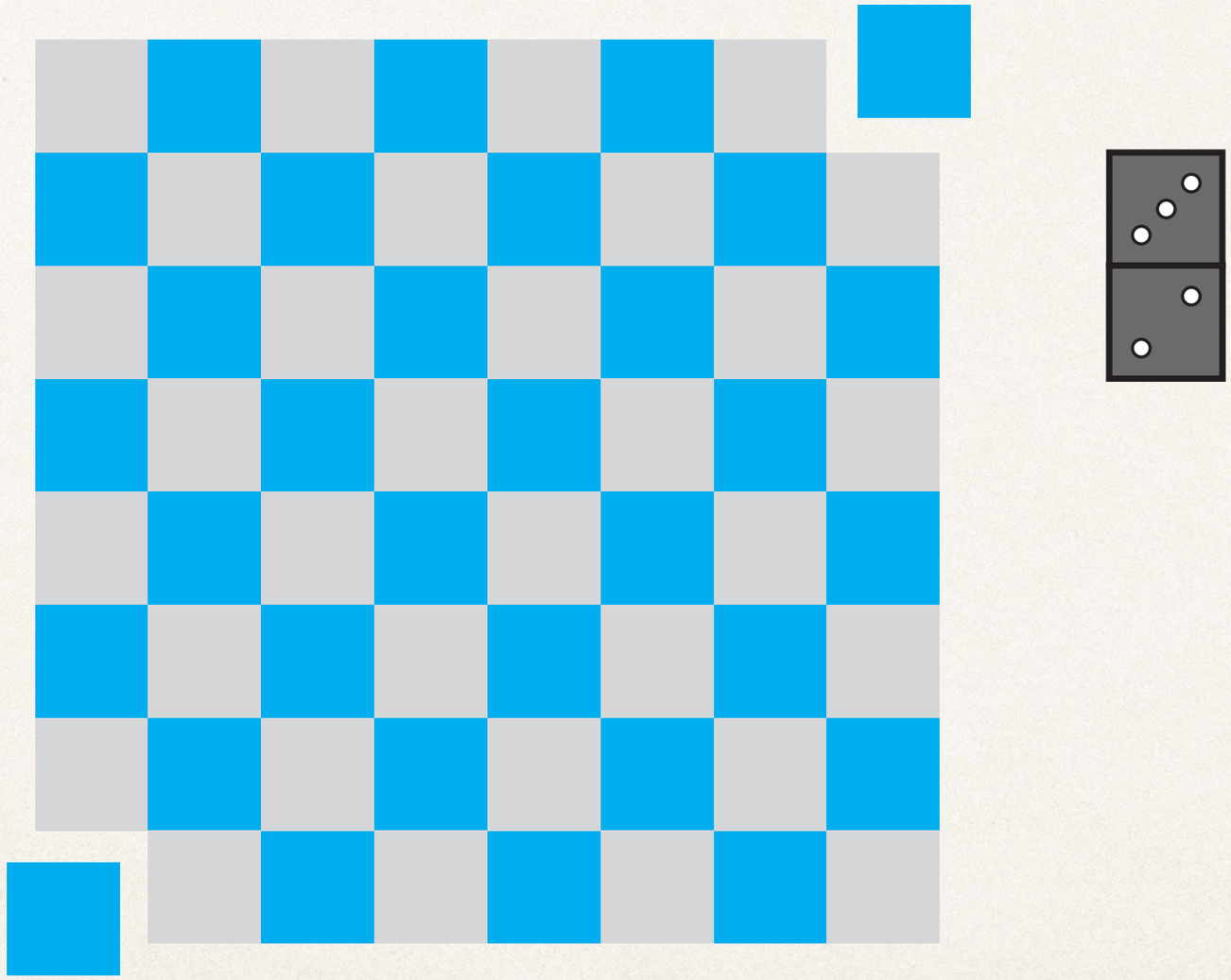
None of which is good enough…

- mostly 19th Century… elementary… boring
- mathematics today involves things like *perfectoid spaces*:

> "Perfectoid spaces are sophisticated objects in arithmetic geometry …. We formalised enough definitions and theorems in topology, algebra and geometry to define perfectoid spaces in the Lean theorem prover. This experiment confirms that a proof assistant can handle complexity in that direction, *which is rather different from formalising a long proof about simple objects*." — Kevin Buzzard

Proving the Obvious

The canonical obvious fact



- combinatorics: “take n coloured balls from the bag”
- chasing commuting diagrams
- “the winding number is obviously 1”
- cancellation of poles in complex analysis (Eberl, 2019)

A trivial fact about derivatives

Suppose f is differentiable at z

Let σ_n be a series of nonzero reals with $\sigma_n \to 0$

$$\text{Then } f'(z) = \lim_{\sigma_n \to 0} \frac{f(z+\sigma_n) - f(z)}{\sigma_n}$$

The formal proof took 32 lines and $\epsilon - \delta$ arguments

Theorem 1.5. *If an elliptic function f has no zeros in some period parallelogram, then f is constant.*

PROOF. Apply Theorem 1.4 to the reciprocal $1/f$. □

Over 60 lines of dense calculations

Theorem 1.6. *The contour integral of an elliptic function taken along the boundary of any cell is zero.*

Three trivial proofs

PROOF. The integrals along parallel edges cancel because of periodicity. □

Over 100 lines

Theorem 1.7. *The sum of the residues of an elliptic function at its poles in any period parallelogram is zero.*

PROOF. Apply Cauchy's residue theorem to a cell and use Theorem 1.6. □

Nearly 200 lines

Some Recent Achievements

Ordinal partition calculus: $\alpha \longrightarrow (\beta, \gamma)^n$

$[A]^n$ denotes the set of unordered n-element sets of elements of A

if $[\alpha]^n$ is partitioned ("coloured") into two parts (0, 1) then either

- $\exists\, B \subseteq \alpha$ of order type β whose n-sets are all coloured by 0
- $\exists\, C \subseteq \alpha$ of order type γ whose n-sets are all coloured by 1

Infinite Ramsey theorem: $\omega \longrightarrow (\omega, \omega)^n$

Erdős' problem (for 2-element sets)

$\alpha \longrightarrow (\alpha,2)$ is trivial $\qquad$ $\alpha \longrightarrow (|\alpha| + 1, \omega)$ **fails** for $\alpha > \omega$

So which countable ordinals α satisfy $\alpha \longrightarrow (\alpha,3)$?

It turns out that α must be a power of ω

In 1987, Erdős offered a $1000 prize for a full solution

Partition calculus work recently formalised

$\omega^2 \longrightarrow (\omega^2, m)$ (Specker)

$\omega^{1+\alpha n} \longrightarrow (\omega^{1+\alpha}, 2^n)$ (Erdős and Milner)

$\omega^\omega \longrightarrow (\omega^\omega, m)$ (Milner, Larson)

Plus background theories: Cantor normal form for ordinals; facts about order types; the Nash-Williams partition theorem

(With Mirna Džamonja and Angeliki Koutsoukou-Argyraki)

Other formalisations at Cambridge

- Transcendence of Certain Infinite Series (criteria by Hančl and Rucki)
- Irrationality Criteria for Series by Erdős and Straus
- Irrational Rapidly Convergent Series (a theorem by J. Hančl)
- Counting Complex Roots
- Budan–Fourier Theorem and Counting Real Roots
- Localization of a Commutative Ring
- Projective Geometry
- Quantum Computation and Information
- **Grothendieck Schemes**

Brief remarks on Grothendieck Schemes

- Build-up of mainstream structures in algebraic geometry: presheaves and sheaves of rings, locally ringed spaces, affine schemes
- the *spectrum of a ring* is a locally ringed space, hence an affine scheme
- any *affine scheme* is a scheme
- *They said it couldn't be done* in simple type theory.
- But we did it faster and with less manpower than the Lean guys.
- One key technique: a structuring mechanism known as *locales*.*
- *led by Anthony Bordg*

Locales in Isabelle/HOL

- Part of the **proof language**, not the logic

- A locale is an abbreviation for a *predicate*

- … and denotes a *proof context* of local variables, assumptions and types.

- Locale inheritance and instantiation checked automatically

Hierarchies of locales for algebraic structures

```
locale cover_of_subset =
  fixes X:: "'a set" and U:: "'a set"
    and index:: "real set" and cover:: "real ⇒ 'a set"
  assumes "U ⊆ X"
    and "⋀i. i ∈ index ⟹ cover i ⊆ X"
  and "U ⊆ (⋃i∈index. cover i)"
```

```
locale open_cover_of_subset = topological_space X is_open
          + cover_of_subset X U I C
  for X and is_open and U and I and C +
  assumes "⋀i. i∈I ⟹ is_open (C i)"
```

```
locale open_cover_of_open_subset
              = open_cover_of_subset X is_open U I C
  for X and is_open and U and I and C +
  assumes "is_open U"
```

… continuing all the way to schemes!

What Do Mathematicians Want?

- Harvey Friedman: *set theoretic foundations* with "soft typing", traditional mathematical notation and **undefined terms**

- NG de Bruijn: **NO** to set theory! His AUTOMATH formalised maths using *dependent types* and classical logic.

- Tim Gowers: *automatic* theorem proving, no search, definitions and proofs written in *natural language*

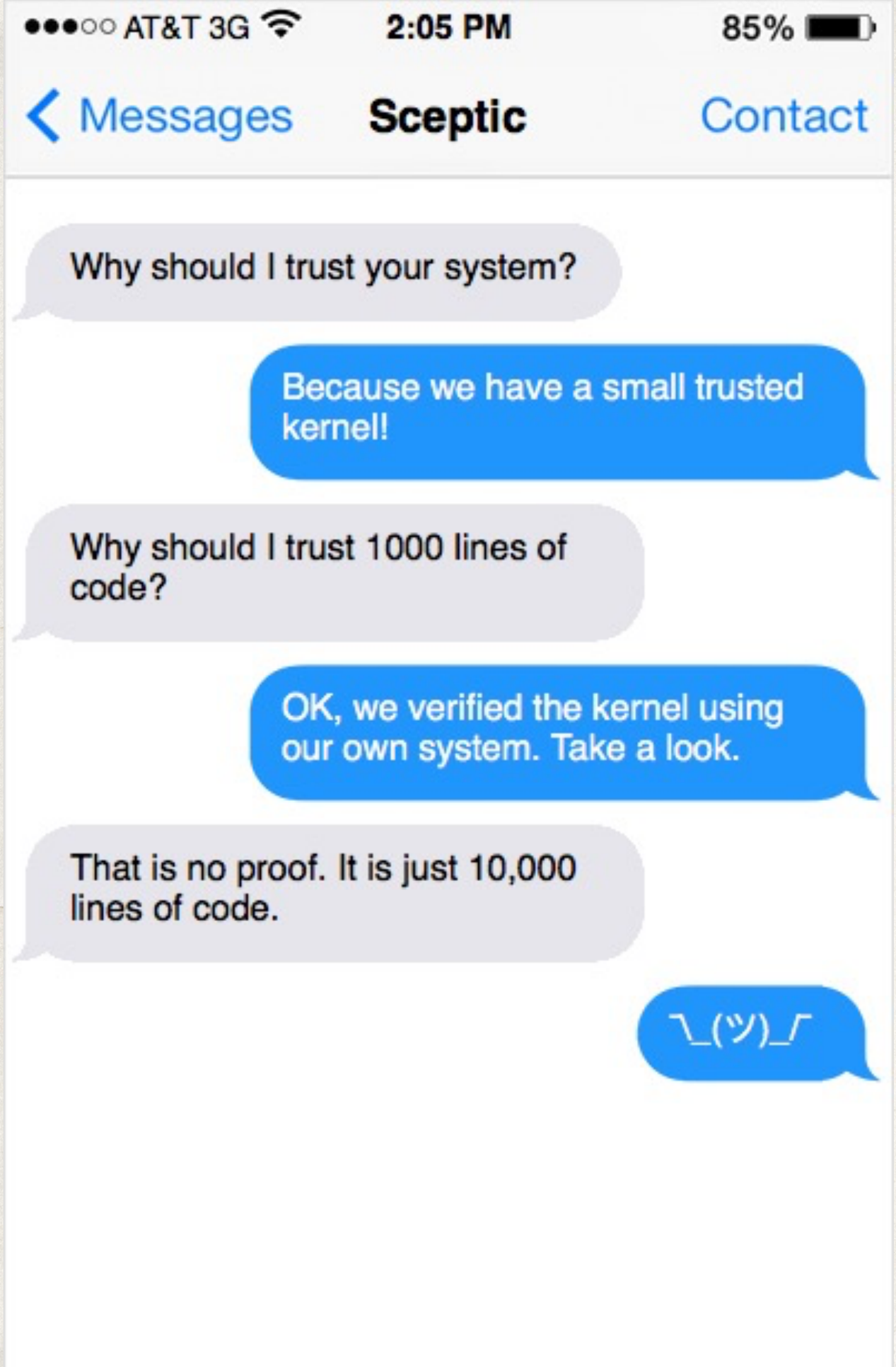
- Kevin Buzzard and students: making great strides using Lean (dependent types, similar to Coq).

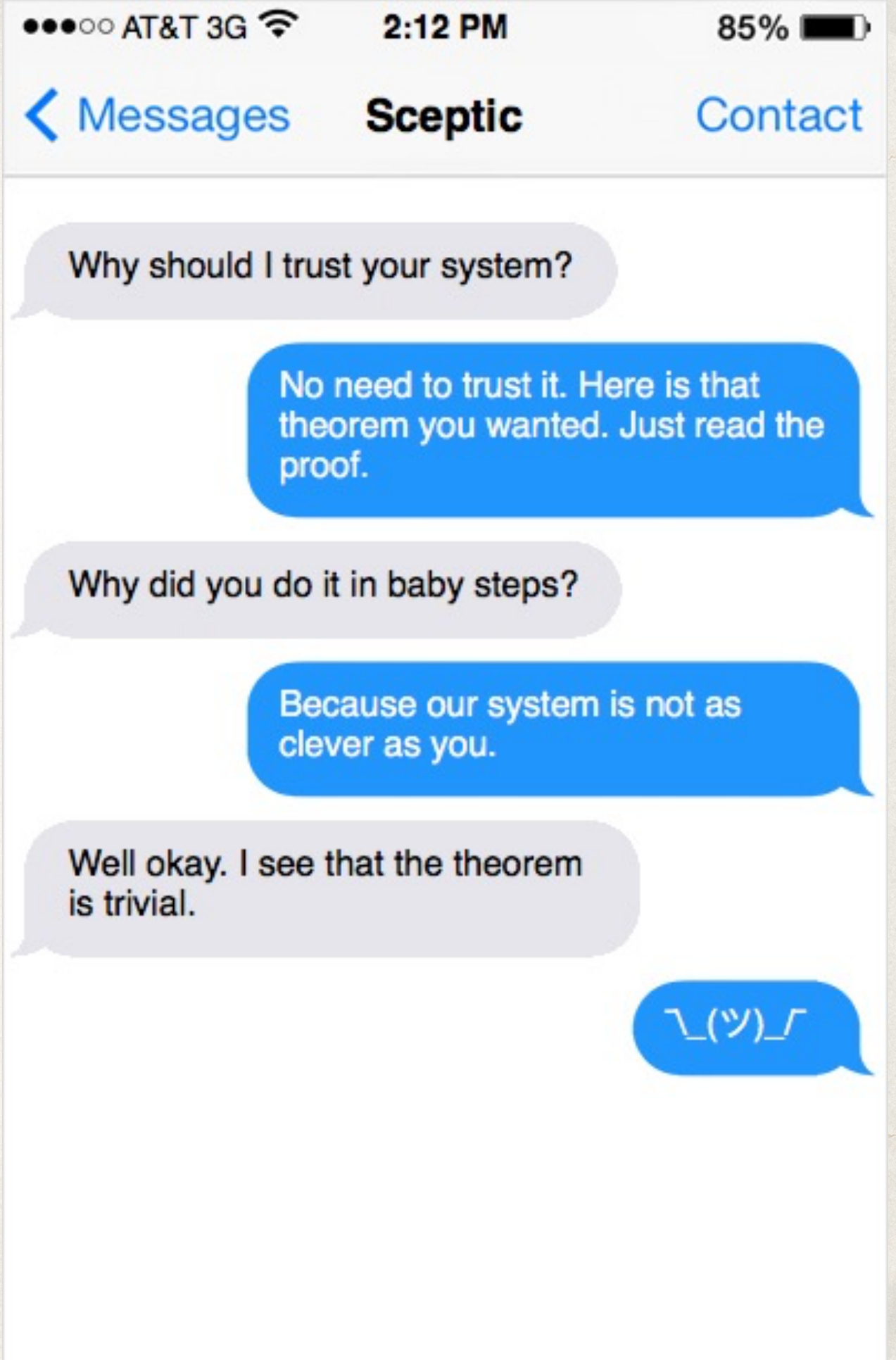
Questions of trust

- Computer algebra systems often give wrong answers. Don't **all** computer systems have bugs?

- Are their logics consistent and faithful? Even if $x/0 = 0$?

But legible structured proofs allow **human** inspection


AT&T 3G
2:05 PM
85%
Messages
Sceptic
Contact
Why should I trust your system?
Because we have a small trusted kernel!
Why should I trust 1000 lines of code?
OK, we verified the kernel using our own system. Take a look.
That is no proof. It is just 10,000 lines of code.
¯_(ツ)_/¯


AT&T 3G
2:12 PM
85%
Messages
Sceptic
Contact
Why should I trust your system?
No need to trust it. Here is that theorem you wanted. Just read the proof.
Why did you do it in baby steps?
Because our system is not as clever as you.
Well okay. I see that the theorem is trivial.
¯_(ツ)_/¯

And the future…?

- Lots of activity, especially using Lean and Isabelle/HOL

- Increasing use of *Machine Learning*, driven by our millions of lines of proofs

- Key questions being settled:

  - Which formalism really is the best?

  - … and what is the best way to use our formalisms?

"Thus we are led to conclude that, although everything mathematical is formalisable, it is nevertheless impossible to formalise all of mathematics in a *single* formal system, a fact that intuitionism has asserted all along."

–Kurt Gödel (1935)