# Formal proof and trust

Dale Miller

Inria Saclay & LIX, École Polytechnique
Palaiseau, France

Upscale meeting, 9 October 2018

# A web of distrust

A great triumph of the World Wide Web (W3) is the ease at which anyone can access a great deal of diverse information.

A glaring flaw of the W3 is the lack of tools to help consumers of information actually trust the assertions made in documents

Trusting is important since trust leads to actions

- if I trust an particular engineering company, I fly their planes
- if I trust Microsoft, I do my taxes on their computers
- if I trust that ConjectureA is really a theorem, I will spend my next months on trying to prove ConjectureB.

# State of the art: WWW

Digital signatures determine authorship of signed information, but few techniques are available to provide trust in what is actually claimed.

Blockchains and Merkle trees can help establish provenance and dependency.

The web has changed from a *cooperative* to an *adversarial* environment. Formal proofs can provide winning strategies against bad guys.

Changes in WWW are enabled by selecting the right *frameworks* on which new behaviors *emerge*.

# First anchor of permanent trust: Formal Proof

Formal proofs have helped to establish trust during two different epochs.

In the late 1800s and early 1900s, there were various crises in mathematics.

- The uses of infinity and infinitesimals was questionable.
- Foundations were naive.

# First anchor of permanent trust: Formal Proof

Formal proofs have helped to establish trust during two different epochs.

In the late 1800s and early 1900s, there were various crises in mathematics.

- The uses of infinity and infinitesimals was questionable.
- Foundations were naive.

In the late 1900s and early 2000s, there have been numerous crises in our digital infrastructure.

- The application of buggy computer systems for operating and controlling our infrastructure is questionable.
- Foundations for correctness, security, and privacy are often overlooked.

Formal proofs have improved the situation in both settings.

# State of the art: Formal proof

- Specialized proof certificates: DRAT/DRUP, CPF, primality certificates, etc
- General-purpose proof certificates: Dedukti, FPC
- Proof-Carrying Code (PCC)
- Frameworks for logics and proofs: logical frameworks, mathematical knowledge management

One can now imagine a proposal such as the following for imposing formal proof on the web.

# The world wide web of documents and proof

| | **WWW of documents** | **WWW of proof** |
|---|---|---|
| *Standards and Infrastructure* | | |
| Documents | Files in various formats | Proofs in various formats |
| Standards | SGML, HTML, etc | FPC, Dedukti, CPF, RUP, etc |
| Naming | URI, DOI | Content addressable storage |
| Transport | HTTP, FTP, torrents | In addition: IPFS |
| Trust | certificate authorities, public logs, encryption, open source, etc | *Reputation* (eg, proved by Coq 8.1) & *Reproducibility* (rechecking proof evidence) |
| *Emergent structures* | | |
| Access | browsers, JavaScript | interacting with proofs, proof browsers |
| Curation | Wikipedia, etc | proof libraries, textbooks |

Not discussed more here. A serious issues appears here.

# Proof checking is a physical process

One can make the argument that formal proofs of significant theorems do not exist without computers, since it is computers that create and consume (check/transform) them.

Despite de Bruijn's pleas for weak frameworks, proof checkers are complex computational systems containing

- printers and parsers
- interpreters, compilers
- garbage collectors
- hardware processors

All of these can have flaws. We have good grounds to be skeptical of proof checkers.

It is the *reputation* of a theorem prover, kernel, or proof checker in which we place our trust. Unfortunately, reputation has limitations.

# Second anchor of permanent trust: reproducibility


Sir Francis Bacon's introduction of the scientific method—with its focus on reproducible results—was seen by the academics at that time as a way out of the political and social chaos that arose from the English Civil War (1642-1651).

Bacon's thoughts were enshrined in the Royal Society's creed "Nullius in verba" (take no one's word for it): that is, before trusting something, check it for yourself.

# Who checks the proof checkers?

The familiar and ancient conundrum "Quis custodiet ipsos custodes?" (Who will guard the guards?).

# Who checks the proof checkers?

The familiar and ancient conundrum "Quis custodiet ipsos custodes?" (Who will guard the guards?).

There is a modern approach to solving this problem: make it possible for anyone and everyone to monitor and audit the guards (proof checkers, in our case).

# Who checks the proof checkers?

The familiar and ancient conundrum "Quis custodiet ipsos custodes?" (Who will guard the guards?).

There is a modern approach to solving this problem: make it possible for anyone and everyone to monitor and audit the guards (proof checkers, in our case).

In 50 years, skeptics should be able to write their own checkers in order to re-check a formal proof.

Thus the format and semantics of documents containing formal proofs must be neither proprietary nor technology-based: this is possible if the format has a well defined mathematical semantics.

Both the Dedukti and the ProofCert projects define the semantics of proof evidence using (different) mathematical frameworks.

# The starting point: [Principle says Assertion]

It seems that we are forced to deal with a logic whose atomic statements are taken from "logics for access control" (such as in Abadi, Burrows, Lampson, and Plotkin 1993)

$P$ says String $\qquad$ $P$ says $(\vdash B)$ $\qquad$ $P$ says $(\Xi \vdash B)$

$P$ Speaks-for $Q$: $\forall A.(P \text{ says } A \supset Q \text{ says } A)$

The truth of [$P$ *says* $A$] is given by a cryptographic signing using the private key of $P$ of (the string/file denoting) $A$.

If Coq8.1 says $(\vdash B)$ and HOL6.5 says $(\vdash B)$ then I say $(\vdash B)$.

It is more likely that in 50 years, it will be proof certificates that are rechecked multiply ways.

If Ker12 says $(\Xi \vdash B)$ and Check51 says $(\Xi \vdash B)$ then I say $(\vdash B)$.

# A first attempt at a worthy goal

> *Goal: Construct a large library of formalized mathematics*

*use, Mizar, Coq, Agda, Isabelle, etc.*

This is a commonly stated problem: one articulation of it was the QED manifesto (1994).

# A first attempt at a worthy goal

> *Goal: Construct a large library of formalized mathematics use, Mizar, Coq, Agda, Isabelle, etc.*

This is a commonly stated problem: one articulation of it was the QED manifesto (1994).

The most important reasons offered by Freek Wiedijk for why the QED effort failed to advance

> *"is that only very few people are working on formalization of mathematics."*

There is no compelling application for fully mechanized mathematics among "working mathematicians", the intended target of QED. (An early suggested topic for the QED project was ring theory.)

That is, the audience is too small.

# History doesn't repeat itself but it often rhymes [1]

What if Tim Berners-Lee had built the repository of physics and engineering documents that he thought CERN needed?

[1] Often attributed to Mark Twain.

# History doesn't repeat itself but it often rhymes [1]

What if Tim Berners-Lee had built the repository of physics and engineering documents that he thought CERN needed?

The web could have taken another decade or so to emerge from that Ivory Tower prison.

[1] Often attributed to Mark Twain.

# History doesn't repeat itself but it often rhymes [1]

What if Tim Berners-Lee had built the repository of physics and engineering documents that he thought CERN needed?

The web could have taken another decade or so to emerge from that Ivory Tower prison.

Can we move this approach to trust and formal proof from its own Ivory Tower prison?

[1] Often attributed to Mark Twain.

# Let's try for sometime much more ambitious

- Goal 1: Libraries of formalized mathematics.

# Let's try for sometime much more ambitious

- Goal 1: Libraries of formalized mathematics.
- Goal 2: Reproducibility in science. Formal proofs can capture the collecting of data values, computation on them, statistical inference, etc.

# Let's try for sometime much more ambitious

- Goal 1: Libraries of formalized mathematics.
- Goal 2: Reproducibility in science. Formal proofs can capture the collecting of data values, computation on them, statistical inference, etc.
- Goal 3: Security and correctness of mobile and modular computing platforms. "Is this app safe to put on my mobile phone?" Echos of Proof Carrying Code.

# Let's try for sometime much more ambitious

- Goal 1: Libraries of formalized mathematics.
- Goal 2: Reproducibility in science. Formal proofs can capture the collecting of data values, computation on them, statistical inference, etc.
- Goal 3: Security and correctness of mobile and modular computing platforms. "Is this app safe to put on my mobile phone?" Echos of Proof Carrying Code.
- Goal 4: Journalism and "fake news". While unlikely to use rich logic and proof techniques, journalism could benefit from the infrastructure of transparency and the signing of assertions that accompanies the infrastructure of proofs.

# Some specific challenges for this project

- *Challenge 1: Permanent, signed electronic documents*

Cryptographic hash functions and content-addressable storage (CAS) (e.g., BitTorrent and the Interplanetary File System IPFS) can be used to provide the permanence of such signed documents.

- *Challenge 2: Structuring libraries of theorems and proofs*

Bindings (such as eigenvariables) can be implemented locally via, say, de Bruijn numerals and distributively via nonces.

- *Challenge 3: Interoperatbility of proofs*

Provide tools for moving between implicit and explicit proofs. A proof using a naive approach to foundations might be moved to different formalize foundations.

- *Challenge 4: Replication in experimental sciences*

Link traditional tools (Maple, Sage) and new web-based services (Open Science Framework (`osf.io`), Life Sciences Protocol Repository (`www.protocols.io`) to proof certificates that include computation and inference.

# Thank you

Questions?