Towards a broad spectrum proof certificate

Dale Miller

INRIA-Saclay & LIX, École Polytechnique
Palaiseau, France

Journées Nationales du GDR 2012, 26 January 2012

Can we standardize, communicate, and trust formal proofs?

Based on the ERC Advanced Investigator Grant: ProofCert
(five years: 2012 - 2016)

We must first narrow our topic

- Proofs are *documents* that are used to *communicate trust* within a *community of agents*.

- In general, agents can be machines or humans.

- Our focus: publishing and checking *formal proofs* by computer *agents*

- Not our focus (yet): reading and learning from proofs, interacting with proofs, computing with proofs.

Provers: computer agents that produce proofs

There is a wide range of provers.

- automated and interactive theorem provers
- model checkers, SAT solvers
- type inference, static analysis
- testers

There is a wide range of "evidence" of proof.

- proof scripts that steer a theorem prover to a proof
- resolution refutations, natural deduction, tableaux, etc
- winning strategies, simulations

It is the exception when one prover's evidence is shared with another prover.

A (familiar) revolution is needed in formal methods

Sun Microsystems (1984): The network *is* the computer

The formal methods community uses many isolated provers technologies: proof assistants (Coq, Isabelle, HOL, PVS, etc), model checkers, SAT solvers, etc.

Goal: Permit the formal methods community to become a network of communicating and trusting provers.

We shall use the term "proof certificate" for those documents denoting proofs that are circulated and checked.

Four desiderata for proof certificates

D1: A simple checker can, in principle, check if a proof certificate denotes a proof.

The *de Bruijn's principle:* provers should output proofs that can be checked by *simple* checkers. Here "simple" might mean that the checker can be independently validated (eg, by hand).

"Everything should be made as simple as possible,
but not one bit simpler."
-Albert Einstein

Almost certainly, proof certificates will themselves be programs and a checker will be an interpreter for such programs.

D2: The proof certificate format supports a broad spectrum of proof systems.

One should not need to radically transform your system's proof evidence in order to output a proof certificate.

Clearly, there is a tension between **D1** and **D2**.

Consider the following consequences of these two desiderata.

Marketplaces for proofs

The ACME company needs a formal proof for its next generation of controllers for airplanes, electric cars, medical equipment, etc.

ACME submits to the "proof marketplace" a proposed theorem as a proof certificate with a "hole" for its actual proof.

The contract: You get paid if you can fill the hole in such a way that ACME can check it.

This marketplace could be wide open: anyone using any combination of deduction engines would be able to compete.

Marketplaces for proofs

The ACME company needs a formal proof for its next generation of controllers for airplanes, electric cars, medical equipment, etc.

ACME submits to the “proof marketplace” a proposed theorem as a proof certificate with a “hole” for its actual proof.

The contract: You get paid if you can fill the hole in such a way that ACME can check it.

This marketplace could be wide open: anyone using any combination of deduction engines would be able to compete.

Providing a *partial proof* or a *counter-example* should also have some economic value. The general setting of “proof certificates” should allow for these.

Libraries of proofs

Proof certificates can be archived, searched, and retrieved.

One should be able to browse, apply, and transform them.

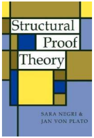
One might *trust* the authority behind the library.

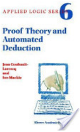
Libraries can invest in significant computing power, thus expanding the proof certificates that they can check.

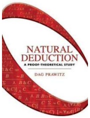
A library has strong motivations to be careful: accepting a non-proof puts their entire library and accumulative trust at risk.

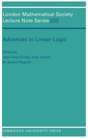
D3: A proof certificate is intended to denote a proof in the sense of structural proof theory.

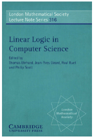
Structural proof theory is a mature field that deals with deep aspects of proofs and their properties.









For example: given certificates for $\forall x(A(x) \supset \exists y\ B(x,y))$ and $A(10)$, can we extract from them a t such that $B(10,t)$ holds?

Such proofs can also be considered **immortal**.

D4: A proof certificate can simply leave out details of the intended proof.

Formal proofs are often huge. All means to reduce their size need to be available.

- Introductions of abstractions and lemma (cut introductions).
- Separate *computation* from *deduction* and leave computation traces out of the certificate.
- Allow trade-offs between *proof size* and *proof reconstruction*: (bounded) proof search maybe need to fill in holes.

D4 leads to challenging demands on proof certificates.

- What bound on search is sensible?
- How to ensure that such search is sensibly directed?

Which logic?

First-order or higher-order?

Which logic?

First-order or higher-order? Both!

Higher-order (à la Church 1940) seems a good choice since it includes propositional and first-order.

Which logic?

First-order or higher-order? Both!

Higher-order (à la Church 1940) seems a good choice since it includes propositional and first-order.

Classical or intuitionistic logic?

Which logic?

First-order or higher-order? Both!

Higher-order (à la Church 1940) seems a good choice since it includes propositional and first-order.

Classical or intuitionistic logic? Both!

There are efforts to put these two logics together in one larger logic: Gentzen (LK/LJ), Girard (LU) and, recently, Liang & M.

Which logic?

First-order or higher-order? Both!

Higher-order (à la Church 1940) seems a good choice since it includes propositional and first-order.

Classical or intuitionistic logic? Both!

There are efforts to put these two logics together in one larger logic: Gentzen (LK/LJ), Girard (LU) and, recently, Liang & M.

Modal, temporal, spatial?

Leave these out for now: there is likely to always be a frontier that does not fit. (However, the syntax and semantics of many modal operators fit well with Church's logic.)

Which proof system?

There are numerous, well studied proof systems: *natural deduction*, *sequent*, *tableaux*, *resolution*, etc.

Many others are clearly proof-like: *tables* (in model checking), *winning strategies* (in game playing), etc.

Other: *certificates for primality*, etc.

We wish to capture all of these proof objects.

How can a proof checker for so many formats be “simple?”

Atoms and molecules of inference

About seven years of *basic research* into proof theory suggests that all these desiderata can be based on the following principles.

There are **atoms of inference**.

- Gentzen's **sequent calculus** first uncovered these: introduction and structural rules.
- Girard's **linear logic** refined our understanding of these further.
- **Fixed points** and **equality** account for first-order structures.

There is a **chemistry** that provides rules for assembling atoms into molecules of inference (following *focused proof systems*).

One can build such **molecules of inference** to match a great range of proof structures.

Satisfying the desiderata

D1: Simple checkers.

Only the atoms of inference and the rules of chemistry (both small and closed sets) need to be implemented in the checker.

D2: Certificates supports a wide range of proof systems.

The molecules of inference can be engineered into a wide range of existing inference rules. (Computation can be placed inside rules.)

D3: Certificates are based on proof theory.

Immediate by design.

D4: Details can be elided.

Proof search in the space of atoms can match proof search in the space of molecules. (The checker does not invent new molecules.)

Some technical bits: Focused proof systems

Example: A focused proof systems for classical logic

Two *invertible* introduction inference rules:

$$\frac{\vdash \Delta, B_1, B_2}{\vdash \Delta, B_1 \vee B_2} \qquad \frac{\vdash \Delta, B[y/x]}{\vdash \Delta, \forall x B}$$

The inference rules for their de Morgan duals are not invertible:

$$\frac{\vdash \Delta, B[t/x]}{\vdash \Delta, \exists x B} \qquad \frac{\vdash \Delta_1, B_1 \qquad \vdash \Delta_2, B_2}{\vdash \Delta_1, \Delta_2, B_1 \wedge B_2}$$

Focused proofs are built in *two phases*:

- the "up arrow" $\Uparrow$ phase where one only has invertible rules
- the "down arrow" $\Downarrow$ phase where one has (not-necessarily) invertible rules

There are two different ways to treat t, $\wedge$, f, $\vee$. Instead of choosing between them, we allow both treatments.

LKF : (multi)focused proof systems for classical logic

$$\frac{}{\vdash \Theta \Uparrow \Gamma, t^-} \qquad \frac{\vdash \Theta \Uparrow \Gamma, A \qquad \vdash \Theta \Uparrow \Gamma, B}{\vdash \Theta \Uparrow \Gamma, A \wedge^- B} \qquad \frac{\vdash \Theta \Uparrow \Gamma}{\vdash \Theta \Uparrow \Gamma, f^-} \qquad \frac{\vdash \Theta \Uparrow \Gamma, A, B}{\vdash \Theta \Uparrow \Gamma, A \vee^- B}$$

$$\frac{}{\vdash \Theta \Downarrow t^+} \qquad \frac{\vdash \Theta \Downarrow \Gamma_1, B_1 \qquad \vdash \Theta \Downarrow \Gamma_2, B_2}{\vdash \Theta \Downarrow \Gamma_1, \Gamma_2, B_1 \wedge^+ B_2} \qquad \frac{\vdash \Theta \Downarrow \Gamma, B_i}{\vdash \Theta \Downarrow \Gamma, B_1 \vee^+ B_2}$$

$$\overset{\text{Init}}{\frac{}{\vdash \neg P_a, \Theta \Downarrow P_a}} \qquad \overset{\text{Store}}{\frac{\vdash \Theta, C \Uparrow \Gamma}{\vdash \Theta \Uparrow \Gamma, C}} \qquad \overset{\text{Release}}{\frac{\vdash \Theta \Uparrow \mathcal{N}}{\vdash \Theta \Downarrow \mathcal{N}}} \qquad \overset{\text{Decide}}{\frac{\vdash \mathcal{P}, \Theta \Downarrow \mathcal{P}}{\vdash \mathcal{P}, \Theta \Uparrow \cdot}}$$

$\mathcal{P}$ multiset of positives; $\mathcal{N}$ multiset of negatives;
P_a positive literal; C positive formula or negative literal

Results about LKF

Let B be a propositional logic formula and let $\hat{B}$ result from B by placing $+$ or $-$ on t, f, $\wedge$, and $\vee$ (there are exponentially many such placements).

Theorem. B is a tautology if and only if $\hat{B}$ has an LKF proof. [Liang & M, TCS 2009]

Thus the different polarizations do not change *provability* but can radically change the *proofs*.

Notice that:

- Only positive formulas are contracted (in the Decide rule).
- Negative (non-atomic) formulas are treated linearly (never weakened nor contracted).

An example

Assume that Θ contains the formula $a \wedge^+ b \wedge^+ \neg c$ and that we have a derivation that Decides on this formula.

$$
\dfrac{\dfrac{}{\vdash \Theta \Downarrow a}\ \mathit{Init} \quad \dfrac{}{\vdash \Theta \Downarrow b}\ \mathit{Init} \quad \dfrac{\dfrac{\vdash \Theta, \neg c \Uparrow \cdot}{\vdash \Theta \Uparrow \neg c}\ \mathit{Store}}{\vdash \Theta \Downarrow \neg c}\ \mathit{Release}}{\dfrac{\vdash \Theta \Downarrow a \wedge^+ b \wedge^+ \neg c}{\vdash \Theta \Uparrow \cdot}\ \mathit{Decide}}\ \wedge^+
$$

This derivation is possible iff Θ is of the form $\neg a, \neg b, \Theta'$. Thus, the "macro-rule" is

$$
\frac{\vdash \neg a, \neg b, \neg c, \Theta' \Uparrow \cdot}{\vdash \neg a, \neg b, \Theta' \Uparrow \cdot}
$$

A certificate for propositional logic: compute CNF

Use $\wedge^-$ and $\vee^-$. Their introduction rules are invertible. The initial "macro-rule" is huge, having all the clauses in the conjunctive normal form of B as premises.

$$\cfrac{\cdots \quad \cfrac{\overline{\vdash L_1, \ldots, L_n \Downarrow L_i}\ \textit{Init}}{\vdash L_1, \ldots, L_n \Uparrow \cdot}\ \textit{Decide} \quad \cdots}{\cfrac{\vdots}{\vdash \cdot \Uparrow \hat{B}}}$$

The proof certificate can specify the complementary literals for each premise or it can ask the checker to *search* for them.

Proof certificates can be tiny but require exponential time for checking.

Positive connectives allow for inserting information

Let B have several alternations of conjunction and disjunction.

Using positive polarities with the tautology $C = (p \vee^{+} B^{+}) \vee^{+} \neg p$ allows for a more clever proof then the previous one.

$$
\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{}{\vdash C, \neg p \Downarrow p}}{\vdash C, \neg p \Downarrow C}\ *}{\vdash C, \neg p \Uparrow \cdot}\ \mathit{Decide}}{\vdash C \Uparrow \neg p}}{\vdash C \Downarrow \neg p}}{\vdash C \Downarrow C}\ *}{\vdash C \Uparrow \cdot}\ \mathit{Decide}}{\vdash \cdot \Uparrow C}
$$

Clever choices $*$ are injected twice. The subformula B is avoided.

Focused proofs system more generally

Focused sequent calculus proof systems are available for:

- **Linear Logic:** provided by Andreoli 1992 as the first comprehensive focused proof system
- **Intuitionistic Logic:** LFJ [Liang & M, TCS 2009] accounts for all other focused intuitionistic proof system: uniform proofs, LJT, LJQ, λRCC, etc.

First order quantification, equality, and least and greatest fixed points have also been accounted for in focused sequent systems.

Fixed points permit

- non-deterministic computations within inference rules, and
- a framework for combining model checking and theorem proving.

Engineering proof systems

A number of proof systems and certificates have been defined on top of either LKF or LJF.

- natural deduction and tableaux
- dependently typed λ-terms
- resolution refutation
- winning strategies / bisimulations
- Pratt primality certificates
- expansion trees, etc.

The work on Deduction-modulo [Dowek, Hardin, & Kirchner] and Dedukti [Boespflug] is related.

- Reduce proofs in the "λ-cube" to $\lambda\Pi$ using functional computations.

Future work

- Finish the work on merging intuitionistic and classical logic into a single, focused sequent calculus.
- Lay the proof theoretic foundation for partial proofs and counter-examples.
- Proof reconstruction for logics
  - without equality and fixed points is given by well-known logic programming techniques: unification and back-tracking search.
  - with equality and fixed points is currently not solved.
- Engineering and infra-structure
  - Can proof checkers remain simple enough while being optimized for performance?
  - Will the theorem proving community agree that the benefits of sharing proofs out ways the cost of supplying them.

Thank You