Reliably Reproducing Machine-Checked Proofs with the Coq Platform

Karl Palmskog[1] Enrico Tassi[2]
Théo Zimmermann[3]

[1]KTH Royal Institute of Technology, Sweden
[2]Univ. Côte d'Azur, Inria, France
[3]Inria, Univ. Paris Cité, CNRS, IRIF, France

The Coq proof assistant

- Software to formally verify *mathematical proofs* and *programs*
- Created in 1984 by Gérard Huet and Thierry Coquand
- Publicly released in 1989
- Continuously developed at Inria with an open source community
- Implemented mainly in the OCaml language
- ACM Software System Award 2013
- Open Science Award for Open Source Research Software 2022

Applications of Coq

- CompCert verified C compiler
- VST verification toolchain for C code
- Formal proof of the Four Color Theorem
- Formal proof of the Odd Order Theorem
- Iris separation logic framework

Anatomy of Coq

- Surface language, *vernacular*
- Elaborator of vernacular into *formalism* of terms and types
- Kernel which checks that terms have a given type

Running Coq Example

alternate.v

```
From Coq Require Import List Permutation Omega.
From Equations Require Import Equations.

Hint Resolve Permutation_app_swap.

Equations alternate (A:Type) (l1 l2:list A) :
 list A by wf (length (l1 ++ l2)) lt :=
  alternate A nil l2         := l2;
  alternate A (h1 :: t1) l2 := h1 :: alternate A l2 t1.
Next Obligation.
  assert (Hp: Permutation (l2 ++ t1) (t1 ++ l2)) by auto.
  rewrite (Permutation_length Hp); omega.
Qed.
```

Checking Coq formal proofs

- Install Coq (including its standard library)
- Install all necessary third-party libraries and plugins
- Elaborate and check the vernacular code (w/ `make`/`dune`)
- Inspect the definitions and theorems (optional)
- Write and run additional validation (optional)

Dependency hell

- Some unknown combination of Coq version and library version may be required for checking
- Change of Coq version or library version may break code
- Necessary plugins may not be ported to recent Coq

```
$ coqc alternate.v
File "./alternate.v", line 1, characters 41-46:
Error: Unable to locate library Omega with prefix Coq.
```

Coq's code compatibility policy

- Introduced in 2015
- Makes it *possible* to be compatible with two successive versions
- Breaking changes documented in release notes, with recommendations
- Deprecation warnings required before removal

Examples of deprecation warnings

```
File "./alternate.v", line 4, characters 0-34:
Warning: Adding and removing hints in the core database
implicitly is deprecated. Please specify a hint database.
[implicit-core-hint-db,deprecated]

File "./alternate.v", line 12, characters 2-41:
Warning: omega is deprecated since 8.12; use
"lia" instead.
[omega-is-deprecated,deprecated]
```

The Coq Platform

- Distribution of Coq with curated selection of:
  - libraries
  - plugins
  - tools
- Release 2022.01.0 contains up to 50 packages for 4 Coq versions
- Allows reproducing the Four Color Theorem (among others)
- Partial solution to dependency hell, reproducibility, and maintainability in Coq projects

Using the Platform

- Binary installers (Windows, macOS, Snap)
  - fast
  - simple
  - not customizable
- Interactive scripts (Windows, macOS, Unixes)
  - slower
  - lower level
  - customizable
- Warning: Customization may lead back to dependency hell

Coq Contribs

- Third-party Coq libraries distributed from 1993
- Maintained in version control repository by Coq developers
- Minimally updated as Coq evolved
- Installable manually by users

Coq opam archive

- OCaml Package Manager (opam)
- Distributes third-party Coq packages since 2014
- Package sources under full control of authors
- Packages published on Coq Package Index

Coq Windows binary installer

- Historically modest OCaml support for Windows
- Each Coq release needs binary Windows installer
- Starting in 2017, the installer was extended with libraries and plugins
- Well received by users, in particular in industry

Platform project and charter

- Inspiration from extended Windows installer
- Expansion to macOS and Linux
- Project and charter announced by Michael Soegtrop in 2019
- Platform scripts generated 8.13 installers in 2020

Platform goals

Distribution of Coq for developing and teaching with Coq that is:

- Operating-system indendent
- Dependable
- Easy to install
- Comprehensive

Problems addressed

- Release coordination in Coq ecosystem
- Compatibility and availability of packages
- Build automation
- Reproducibility of Coq artifacts

Platform package case study: Equations

- Function definition plugin
- Automation for recursion and termination

```
From Coq Require Import List Permutation Lia.
From Equations Require Import Equations.

#[local] Hint Resolve Permutation_app_swap : core.

Equations alternate (A:Type) (l1 l2:list A) :
 list A by wf (length (l1 ++ l2)) lt :=
  alternate A nil l2         := l2;
  alternate A (h1 :: t1) l2 := h1 :: alternate A l2 t1.
Next Obligation.
  assert (Hp: Permutation (l2 ++ t1) (t1 ++ l2)) by auto.
  rewrite (Permutation_length Hp); lia.
Qed.
```

Release cycle

- Coq releases every 6 months, handled by Coq core team
- Platform releases follow Coq releases with 1-2 month delay
- Platform team handles releases, partially overlaps with Coq core team

Release process

- Coq release manager tags major version release candidate
- Candidate made available for continuous integration testing
- Platform manager asks package maintainers for versions
- Coq release manager tags major version
- Platform manager tags release with package versions

Platform version scheme

- Based on calendar month of release
- Each release contains several Coq versions and package picks
- 2022.01.0 contains Coq 8.12, 8.13, 8.14 and 8.15-beta-pick

Platform repository

- Platform developed on GitHub: github.com/coq/platform
- Releases are tags in the repository
- CI/CD used to build binary installers

Platform CI/CD

- GitHub CI builds Coq and packages
- CI also builds installers:
  - Windows exe
  - macOS dmg
  - Linux Snap
- Useful to generate custom Platform installers

Coq Contribs successors

- Coq Package Index: third-party package distribution
- Coq-community: maintenance of code on GitHub
- Coq's CI suite: compatibility testing and updating by Coq developers

Platform and the ecosystem

- Platform convenient for installing curated packages on Index
- Platform releases incentivize Platform package developers to be compatible
- Platform releases incentivize non-Platform package developers to adapt to new Coq versions

Lean and mathlib

- Lean has similar foundations to Coq
- Lean has single large library, mathlib
- Version management for mathlib almost completely avoided
- Lean porting handled by community tools, if at all

Isabelle and the Archive of Formal Proofs

- Isabelle generic proof assistant, commonly instantiated to Isabelle/HOL
- Isabelle has the Archive of Formal Proofs (AFP) with libraries
- AFP entries more reviewed than Coq opam archive
- AFP maintained by Isabelle developers

Linux distributions

- Provide compatible sets of packages, including Coq
- Coq package often lags behind the official releases
- Often include packages for the TeXLive distribution

Haskell Platform and Stackage

- Haskell Platform included sets of packages, like the Coq Platform
- Haskell Platform deprecated in 2022
- Stackage fills similar function
- Stackage has maintainer social contracts like the Coq Platform

Coq artifact evaluation

- Low current standards for reproducing Coq scientific artifacts
- Platform can help reviewers reproduce artifacts with less effort
- Subset of Platform available in official Docker containers
- Platform curates trustworthy basic specifications

From Platform to artifact evaluation

- Platform provides a **reliable foundation** for Coq evaluations
- Full evaluations require *additional* tools and practices:
  - ANSSI requirements
  - Mahboubi's "Checking machine-checked proofs"
- Artifacts may conglomerate Coq and other tools/languages
  - Lightbulb verified software/hardware system by Erbsen et al.

Ongoing and future work

- Additional Docker containers
- Improve trustworthiness with cryptography (repo signing)
- Additional packages
- Additional task automation

Replication of Coq code

- Possible meaning of *replication*: "make old code compile on recent Coq"
- Under this definition, Platform can replicate many old research results now found in Coq-community:
  - Chapar: verified distributed key-value store (POPL 2016)
  - regexp-Brzozowski: decision procedures for regular expression equivalence (CPP 2011)
  - Apery: proof of the irrationality of zeta(3) (ITP 2014)