Coq

Coq is an interactive theorem prover first released in 1989. It allows for expressing mathematical assertions, mechanically checks proofs of these assertions, helps find formal proofs, and extracts a certified program from the constructive proof of its formal specification. Coq works within the theory of the calculus of inductive constructions, a derivative of the calculus of constructions. Coq is not an automated theorem prover but includes automatic theorem proving tactics (procedures) and various decision procedures.

The Association for Computing Machinery awarded Thierry Coquand, Gérard Huet, Christine Paulin-Mohring, Bruno Barras, Jean-Christophe Filliâtre, Hugo Herbelin, Chetan Murthy, Yves Bertot, and Pierre Castéran with the 2013 ACM Software System Award for Coq.

Coq is named after its principal developer, Thierry Coquand.

Coq (software)

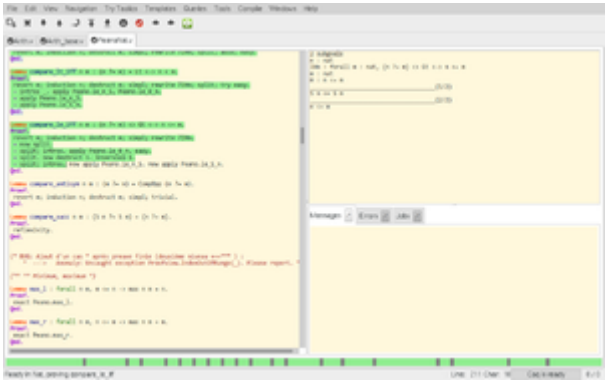
Developer(s)	The Coq development team
Initial release	1 May 1989 (version 4.10)
Stable release	8.13.0[1] / 7 January 2021
Preview release	8.13+beta1[2] / 7 December 2020
Repository	github.com/coq/coq (https://github.com/coq/coq)
Written in	OCaml
Operating system	Cross-platform
Available in	English
Type	Proof assistant
License	LGPLv2.1
Website	coq.inria.fr (https://coq.inria.fr/)

Contents

Overview

When viewed as a programming language, Coq implements a dependently typed functional programming language;[3] when viewed as a logical system, it implements a higher-order type theory. The development of Coq has been supported since 1984 by INRIA, now in collaboration with École Polytechnique, University of Paris-Sud, Paris Diderot University, and CNRS. In the 1990s, ENS Lyon was also part of the project. The development of Coq was initiated by Gérard Huet and Thierry Coquand, and more than 40 people, mainly researchers, have contributed features to the core system since its inception. The implementation team has successively been coordinated by Gérard Huet, Christine Paulin-Mohring, Hugo Herbelin, and Matthieu Sozeau. Coq is mainly implemented in OCaml with a bit of C. The core system can be extended by way of a plug-in mechanism.[4]

The name *coq* means "rooster" in French and stems from a French tradition of naming research development tools after animals.[5] Up until 1991, Coquand was implementing a language called the Calculus of Constructions and it was simply called CoC at this time. In 1991, a new implementation based on the extended Calculus of Inductive Constructions was started and the name was changed from CoC to Coq in an indirect reference to Coquand, who developed the Calculus of Constructions along with Gérard Huet and contributed to the Calculus of Inductive Constructions with Christine Paulin-Mohring.[6]



An interactive proof session in CoqIDE, showing the proof script on the left and the proof state on the right.

Coq provides a specification language called Gallina[7] ("hen" in Latin, Spanish, Italian and Catalan). Programs written in Gallina have the weak normalization property, implying that they always terminate. This is a distinctive property of the language, since infinite loops (non-terminating programs) are common in other programming languages,[8] and is one way to avoid the halting problem.

Four color theorem and SSReflect extension

Georges Gonthier of Microsoft Research in Cambridge, England and Benjamin Werner of INRIA used Coq to create a surveyable proof of the four color theorem, which was completed in 2005.[9] Their work led to the development of the SSReflect ("Small Scale Reflection") package, which was a significant extension to Coq.[10] Despite its name, most of the features added to Coq by SSReflect are general-purpose features and are not limited to the computational reflection style of proof. These features include:

- Additional convenient notations for irrefutable and refutable pattern matching, on inductive types with one or two constructors
- Implicit arguments for functions applied to zero arguments, which is useful when programming with higher-order functions
- Concise anonymous arguments
- An improved `set` tactic with more powerful matching
- Support for reflection

SSReflect 1.11 is freely available, dual-licensed under the open source CeCILL-B or CeCILL-2.0 license, and compatible with Coq 8.11.[11]

Applications

- CompCert: an optimizing compiler for almost all of the C programming language which is largely programmed and proved in Coq.
- Disjoint-set data structure: correctness proof in Coq was published in 2007.[12]
- Feit–Thompson theorem: formal proof using Coq was completed in September 2012.[13]
- Four color theorem: formal proof using Coq was completed in 2005.[9]

See also

- Nuprl
- Agda
- Idris
- Calculus of constructions
- Curry–Howard correspondence
- Isabelle (proof assistant) – similar/competing software
- Intuitionistic type theory
- HOL (proof assistant)

References

1. "Coq 8.12.2 is out" (https://coq.inria.fr/news/coq-8-12-2-is-out.html). 2020-12-11.
2. "Coq 8.13+beta1 is out" (https://coq.inria.fr/news/coq-8-13beta1-is-out.html). 2020-12-07.
3. A short introduction to Coq (http://coq.inria.fr/a-short-introduction-to-coq)
4. Avigad, Jeremy; Mahboubi, Assia (3 July 2018). *Interactive Theorem Proving: 9th International Conference, ITP 2018, Held as ...* (https://books.google.com/books?id=I-tiDwAAQBAJ&q=%22coq%22,%22plugins%22,%22extended%22&pg=PA21) ISBN 9783319948218. Retrieved 21 October 2018.
5. "Frequently Asked Questions" (https://github.com/coq/coq/wiki/Presentation). Retrieved 2019-05-08.
6. "Introduction to the Calculus of Inductive Constructions" (https://hal.inria.fr/hal-01094195/document). Retrieved 21 May 2019.
7. Adam Chlipala. "Certified Programming with Dependent Types": "Library Universes" (http://adam.chlipala.net/cpdt/html/Universes.html).
8. Adam Chlipala. "Certified Programming with Dependent Types": "Library GeneralRec" (http://adam.chlipala.net/cpdt/html/GeneralRec.html). "Library InductiveTypes" (http://adam.chlipala.net/cpdt/html/InductiveTypes.html).
9. Gonthier, Georges (2008), "Formal Proof—The Four-Color Theorem" (https://www.ams.org/notices/200811/tx081101382p.pdf) (PDF), *Notices of the American Mathematical Society*, **55** (11), pp. 1382–1393, MR 2463991 (https://www.ams.org/mathscinet-getitem?mr=2463991)
10. Georges Gonthier, Assia Mahboubi. "An introduction to small scale reflection in Coq": "Journal of Formalized Reasoning" (http://jfr.unibo.it/article/view/1979).
11. "The Mathematical Components Library 1.11.0" (https://github.com/math-comp/math-comp/releases/tag/mathcomp-1.11.0).
12. Conchon, Sylvain; Filliâtre, Jean-Christophe (October 2007), "A Persistent Union-Find Data Structure", *ACM SIGPLAN Workshop on ML* (https://www.lri.fr/~filliatr/puf/), Freiburg, Germany
13. "Feit-Thompson theorem has been totally checked in Coq" (https://web.archive.org/web/20161119094854/http://www.msr-inria.fr/news/feit-thomson-proved-in-coq/). Msr-inria.inria.fr. 2012-09-20. Archived from the original (http://www.msr-inria.fr/news/feit-thomson-proved-in-coq/) on 2016-11-19. Retrieved 2012-09-25.

External links

- The Coq proof assistant (http://coq.inria.fr/) – the official English website
- coq/coq (https://github.com/coq/coq) – the project's source code repository on GitHub
- JsCoq Interactive Online System (https://x80.org/rhino-coq/) – allows Coq to be run in a web browser, without the need for any software installation

- Alectryon (https://plv.csail.mit.edu/blog/alectryon.html) – a library to process Coq snippets embedded in documents, showing goals and messages for each Coq sentence
- Coq Wiki (https://github.com/coq/coq/wiki)
- Mathematical Components library (https://math-comp.github.io/math-comp/) – widely used library of mathematical structures, part of which is the SSReflect proof language
- Constructive Coq Repository at Nijmegen (http://corn.cs.ru.nl/)
- Math Classes (https://math-classes.github.io/)
- Coq (https://www.openhub.net/p/coq) at Open Hub

Textbooks

- The Coq'Art (http://www.labri.fr/perso/casteran/CoqArt/index.html) – a book on Coq by Yves Bertot and Pierre Castéran
- Certified Programming with Dependent Types (http://adam.chlipala.net/cpdt/) – online and printed textbook by Adam Chlipala
- Software Foundations (http://www.cis.upenn.edu/~bcpierce/sf/) – online textbook by Benjamin C. Pierce et al.
- An introduction to small scale reflection in Coq (http://jfr.unibo.it/article/view/1979) – a tutorial on SSReflect by Georges Gonthier and Assia Mahboubi

Tutorials

- Introduction to the Coq Proof Assistant (http://video.ias.edu/univalent/appel) – video lecture by Andrew Appel at Institute for Advanced Study
- Video tutorials for the Coq proof assistant (http://math.andrej.com/2011/02/22/video-tutorials-for-the-coq-proof-assistant/) by Andrej Bauer.
