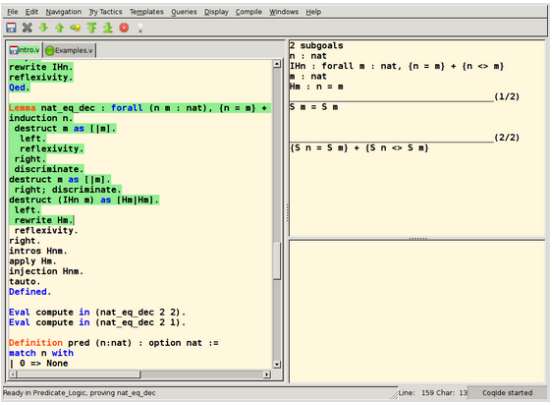
WIKIPEDIA
The Free Encyclopedia

Proof assistant

In computer science and mathematical logic, a **proof assistant** or **interactive theorem prover** is a software tool to assist with the development of formal proofs by human-machine collaboration. This involves some sort of interactive proof editor, or other interface, with which a human can guide the search for proofs, the details of which are stored in, and some steps provided by, a computer.

A recent effort within this field is making these tools use artificial intelligence to automate the formalization of ordinary mathematics.[1]



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

System comparison

Name	Latest version	Developer(s)	Implementation language	Features					
				Higher-order logic	Dependent types	Small kernel	Proof automation	Proof by reflection	Code generation
ACL2	8.3	Matt Kaufmann and J Strother Moore	Common Lisp	No	Untyped	No	Yes	Yes[2]	Already executable
Agda	2.6.3	Ulf Norell, Nils Anders Danielsson, and Andreas Abel (Chalmers and Gothenburg)	Haskell	Yes	Yes	Yes	No	Partial	Already executable
Albatross	0.4	Helmut Brandl	OCaml	Yes	No	Yes	Yes	Unknown	Not yet Implemented
Coq	8.19.0	INRIA	OCaml	Yes	Yes	Yes	Yes	Yes	Yes
F*	repository	Microsoft Research and INRIA	F*	Yes	Yes	No	Yes	Yes[3]	Yes
HOL Light	repository	John Harrison	OCaml	Yes	No	Yes	Yes	No	No
HOL4	Kananaskis-13 (or repo)	Michael Norrish, Konrad Slind, and others	Standard ML	Yes	No	Yes	Yes	No	Yes
Idris	2 0.6.0.	Edwin Brady	Idris	Yes	Yes	Yes	Unknown	Partial	Yes
Isabelle	Isabelle2021 (February 2021)	Larry Paulson (Cambridge), Tobias Nipkow (München) and Makarius Wenzel	Standard ML, Scala	Yes	No	Yes	Yes	Yes	Yes
Lean	v4.7.0[4]	Leonardo de Moura (Microsoft Research)	C++, Lean	Yes	Yes	Yes	Yes	Yes	Yes
LEGO (not affiliated with Lego)	1.3.1	Randy Pollack (Edinburgh)	Standard ML	Yes	Yes	Yes	No	No	No
Metamath	v0.198[5]	Norman Megill	ANSI C						
Mizar	8.1.05	Białystok University	Free Pascal	Partial	Yes	No	No	No	No
Nqthm									
NuPRL	5	Cornell University	Common Lisp	Yes	Yes	Yes	Yes	Unknown	Yes
PVS	6.0	SRI International	Common Lisp	Yes	Yes	No	Yes	No	Unknown
Twelf	1.7.1	Frank Pfenning and Carsten Schürmann	Standard ML	Yes	Yes	Unknown	No	No	Unknown

- ACL2 – a programming language, a first-order logical theory, and a theorem prover (with both interactive and automatic

modes) in the Boyer–Moore tradition.

- Coq – Allows the expression of mathematical assertions, mechanically checks proofs of these assertions, helps to find formal proofs, and extracts a certified program from the constructive proof of its formal specification.
- HOL theorem provers – A family of tools ultimately derived from the LCF theorem prover. In these systems the logical core is a library of their programming language. Theorems represent new elements of the language and can only be introduced via "strategies" which guarantee logical correctness. Strategy composition gives users the ability to produce significant proofs with relatively few interactions with the system. Members of the family include:
  - HOL4 – The "primary descendant", still under active development. Support for both Moscow ML and Poly/ML. Has a BSD-style license.
  - HOL Light – A thriving "minimalist fork". OCaml based.
  - ProofPower – Went proprietary, then returned to open source. Based on Standard ML.
- IMPS, An Interactive Mathematical Proof System.[6]
- Isabelle is an interactive theorem prover, successor of HOL. The main code-base is BSD-licensed, but the Isabelle distribution bundles many add-on tools with different licenses.
- Jape – Java based.
- Lean
- LEGO
- Matita – A light system based on the Calculus of Inductive Constructions.
- MINLOG – A proof assistant based on first-order minimal logic.
- Mizar – A proof assistant based on first-order logic, in a natural deduction style, and Tarski–Grothendieck set theory.
- PhoX – A proof assistant based on higher-order logic which is eXtensible.
- Prototype Verification System (PVS) – a proof language and system based on higher-order logic.
- TPS and ETPS – Interactive theorem provers also based on simply-typed lambda calculus, but based on an independent formulation of the logical theory and independent implementation.

User interfaces

A popular front-end for proof assistants is the Emacs-based Proof General, developed at the University of Edinburgh.

Coq includes CoqIDE, which is based on OCaml/Gtk. Isabelle includes Isabelle/jEdit, which is based on jEdit and the Isabelle/Scala infrastructure for document-oriented proof processing. More recently, Visual Studio Code extensions have been developed for Isabelle by Makarius Wenzel,[7] and for Lean 4 by the leanprover developers.[8]

Formalization extent

Freek Wiedijk has been keeping a ranking of proof assistants by the amount of formalized theorems out of a list of 100 well-known theorems. As of September 2023, only five systems have formalized proofs of more than 70% of the theorems, namely Isabelle, HOL Light, Coq, Lean and Metamath.[9][10]

Notable formalized proofs

The following is a list of notable proofs that have been formalized within proof assistants.

Theorem	Proof assistant	Year
Four color theorem[11]	Coq	2005
Feit–Thompson theorem[12]	Coq	2012
Fundamental group of the circle[13]	Coq	2013
Erdős–Graham problem[14][15]	Lean	2022
Polynomial Freiman-Ruzsa conjecture over $\mathbb{F}_2$[16]	Lean	2023

See also

- Automated theorem proving – Subfield of automated reasoning and mathematical logic
- Computer-assisted proof – Mathematical proof at least partially generated by computer
- Formal verification – Proving or disproving the correctness of certain intended algorithms
- QED manifesto – Proposal for a computer-based database of all mathematical knowledge
- Satisfiability modulo theories – Logical problem studied in computer science

- Prover9 – is an automated theorem prover for first-order and equational logic

Notes

1. Ornes, Stephen (August 27, 2020). "Quanta Magazine – How Close Are Computers to Automating Mathematical Reasoning?" (https://www.quantamagazine.org/how-close-are-computers-to-automating-mathematical-reasoning-20200827/).
2. Hunt, Warren; Matt Kaufmann; Robert Bellarmine Krug; J Moore; Eric W. Smith (2005). "Meta Reasoning in ACL2" (http://www.cs.utexas.edu/~moore/publications/meta-05.pdf) (PDF). *Theorem Proving in Higher Order Logics*. Lecture Notes in Computer Science. Vol. 3603. pp. 163–178. doi:10.1007/11541868_11 (https://doi.org/10.1007%2F11541868_11). ISBN 978-3-540-28372-0.
3. Search for "proofs by reflection": arXiv:1803.06547
4. "Lean 4 Releases Page" (https://github.com/leanprover/lean4/releases). *GitHub*. Retrieved 15 October 2023.
5. "Release v0.198 · metamath/Metamath-exe" (https://github.com/metamath/metamath-exe/releases/tag/v0.198). *GitHub*.
6. Farmer, William M.; Guttman, Joshua D.; Thayer, F. Javier (1993). "IMPS: An interactive mathematical proof system" (https://core.ac.uk/display/23376340). *Journal of Automated Reasoning*. **11** (2): 213–248. doi:10.1007/BF00881906 (https://doi.org/10.1007%2FBF00881906). S2CID 3084322 (https://api.semanticscholar.org/CorpusID:3084322). Retrieved 22 January 2020.
7. Wenzel, Makarius. "Isabelle" (https://marketplace.visualstudio.com/items?itemName=makarius.isabelle). Retrieved 2 November 2019.
8. "VS Code Lean 4" (https://github.com/leanprover/vscode-lean4). *GitHub*. Retrieved 15 October 2023.
9. Wiedijk, Freek (15 September 2023). "Formalizing 100 Theorems" (https://www.cs.ru.nl/~freek/100/).
10. Geuvers, Herman (February 2009). "Proof assistants: History, ideas and future" (https://www.ias.ac.in/article/fulltext/sadh/034/01/0003-0025). *Sādhanā*. **34** (1): 3–25. doi:10.1007/s12046-009-0001-5 (https://doi.org/10.1007%2Fs12046-009-0001-5). hdl:2066/75958 (https://hdl.handle.net/2066%2F75958). S2CID 14827467 (https://api.semanticscholar.org/CorpusID:14827467).
11. 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): 1382–1393, MR 2463991 (https://mathscinet.ams.org/mathscinet-getitem?mr=2463991), archived (https://web.archive.org/web/20110805094909/http://www.ams.org/notices/200811/tx081101382p.pdf) (PDF) from the original on 2011-08-05
12. "Feit thomson proved in coq - Microsoft Research Inria Joint Centre" (https://web.archive.org/web/20161119094854/http://www.msr-inria.fr/news/feit-thomson-proved-in-coq/). 2016-11-19. Archived from the original (http://www.msr-inria.fr/news/feit-thomson-proved-in-coq/) on 2016-11-19. Retrieved 2023-12-07.
13. Licata, Daniel R.; Shulman, Michael (2013). "Calculating the Fundamental Group of the Circle in Homotopy Type Theory". *2013 28th Annual ACM/IEEE Symposium on Logic in Computer Science* (https://ieeexplore.ieee.org/document/6571554). pp. 223–232. arXiv:1301.3443 (https://arxiv.org/abs/1301.3443). doi:10.1109/lics.2013.28 (https://doi.org/10.1109%2Flics.2013.28). ISBN 978-1-4799-0413-6. S2CID 5661377 (https://api.semanticscholar.org/CorpusID:5661377). Retrieved 2023-12-07.
14. "Math Problem 3,500 Years In The Making Finally Gets A Solution" (https://www.iflscience.com/math-problem-3500-years-in-the-making-finally-gets-a-solution-62925). *IFLScience*. 2022-03-11. Retrieved 2024-02-09.
15. Avigad, Jeremy (2023). "Mathematics and the formal turn". arXiv:2311.00007 (https://arxiv.org/abs/2311.00007) [math.HO (https://arxiv.org/archive/math.HO)].
16. Sloman, Leila (2023-12-06). " 'A-Team' of Math Proves a Critical Link Between Addition and Sets" (https://www.quantamagazine.org/a-team-of-math-proves-a-critical-link-between-addition-and-sets-20231206/). *Quanta Magazine*. Retrieved 2023-12-07.

References

- Barendregt, Henk; Geuvers, Herman (2001). "18. Proof-assistants using Dependent Type Systems" (https://web.archive.org/web/20070727062855/http://www.ncc.up.pt/~nam/aulas/0506/t_coq/barendregt01proofassistants.pdf) (PDF). In Robinson, Alan J. A.; Voronkov, Andrei (eds.). *Handbook of Automated Reasoning*. Vol. 2. Elsevier. pp. 1149–. ISBN 978-0-444-50812-6. Archived from the original (http://www.ncc.up.pt/~nam/aulas/0506/t_coq/barendregt01proofassistants.pdf) (PDF) on 2007-07-27.
- Pfenning, Frank. "17. Logical frameworks" (https://www.cs.cmu.edu/~fp/papers/handbook01.pdf) (PDF). *Handbook vol 2 2001*. pp. 1065–1148.
- Pfenning, Frank (1996). "The practice of logical frameworks". In Kirchner, H. (ed.). *Trees in Algebra and Programming – CAAP '96*. Lecture Notes in Computer Science. Vol. 1059. Springer. pp. 119–134. doi:10.1007/3-540-61064-2_33 (https://doi.org/10.1007%2F3-540-61064-2_33). ISBN 3-540-61064-2.
- Constable, Robert L. (1998). "X. Types in computer science, philosophy and logic" (https://books.google.com/books?id=MfTMDeCq7ukC&pg=PA683). In Buss, S. R. (ed.). *Handbook of Proof Theory*. Studies in Logic. Vol. 137. Elsevier. pp. 683–786. ISBN 978-0-08-053318-6.
- Wiedijk, Freek (2005). "The Seventeen Provers of the World" (https://www.cs.ru.nl/~freek/comparison/comparison.pdf) (PDF). Radboud University Nijmegen.

External links

- Theorem Prover Museum (https://theoremprover-museum.github.io/)
- "Introduction" (http://adam.chlipala.net/cpdt/html/Intro.html) in *Certified Programming with Dependent Types*.
- Introduction to the Coq Proof Assistant (http://video.ias.edu/univalent/appel) (with a general introduction to interactive theorem proving)
- Interactive Theorem Proving for Agda Users (http://www.cs.swan.ac.uk/~csetzer/lectures/intertheo/07/interactiveTheoremProvingForAgdaUsers.html)
- A list of theorem proving tools (https://github.com/johnyf/tool_lists/blob/master/verification_synthesis.md#theorem-provers)

Catalogues

- Digital Math by Category: Tactic Provers (https://www.cs.ru.nl/~freek/digimath/bycategory.html#tacticprover)
- Automated Deduction Systems and Groups (http://www.mcs.anl.gov/research/projects/AR/others.html)
- Theorem Proving and Automated Reasoning Systems (https://www.cs.cmu.edu/afs/cs/project/ai-repository/ai/areas/reasonng/atp/systems/0.html)
- Database of Existing Mechanized Reasoning Systems (http://www-formal.stanford.edu/clt/ARS/Pages/systems.html)
- NuPRL: Other Systems (http://www.nuprl.org/Intro/others.html)
- "Specific Logical Frameworks and Implementations" (https://web.archive.org/web/20220410151836/https://www.cs.cmu.edu/~fp/lfs-impl.html). Archived from the original (https://www.cs.cmu.edu/~fp/lfs-impl.html) on 10 April 2022. Retrieved 15 February 2024. (By Frank Pfenning).
- DMOZ: Science: Math: Logic and Foundations: Computational Logic: Logical Frameworks (http://www.dmoz.org/Science/Math/Logic_and_Foundations/Computational_Logic/Logical_Frameworks/)
