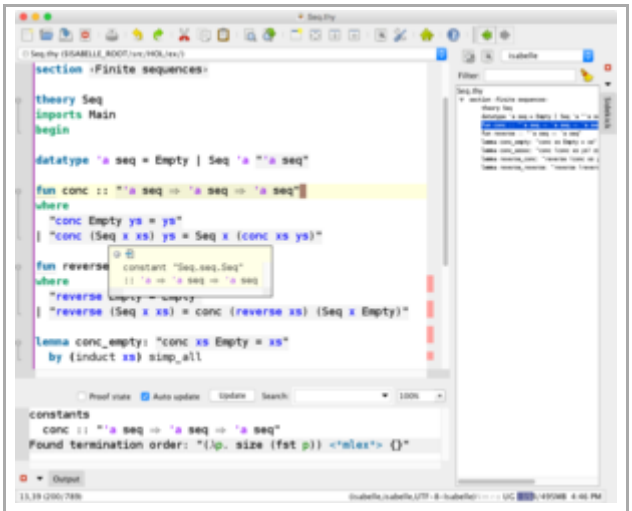
Isabelle (proof assistant)

The **Isabelle**[a] automated theorem prover is an interactive theorem prover, a higher order logic (HOL) theorem prover. It is an LCF-style theorem prover (written in Standard ML). It is thus based on a small logical core (kernel) to increase the trustworthiness of proofs without requiring (yet supporting) explicit proof objects.

Contents

Isabelle



Isabelle/jEdit running on macOS

Original author(s)	Lawrence Paulson
Developer(s)	University of Cambridge and Technical University of Munich et al.
Initial release	1986[1]
Stable release	Isabelle2020 / April 2020
Written in	Standard ML and Scala
Operating system	Linux, Windows, Mac OS X
Type	Mathematics
License	BSD license
Website	isabelle.in.tum.de (https://isabelle.in.tum.de/)

Features

Isabelle is generic: it provides a meta-logic (a weak type theory), which is used to encode object logics like first-order logic (FOL), higher-order logic (HOL) or Zermelo–Fraenkel set theory (ZFC). The most widely used object logic is Isabelle/HOL, although significant set theory developments were completed in Isabelle/ZF. Isabelle's main proof method is a higher-order version of resolution, based on higher-order unification.

Though interactive, Isabelle features efficient automatic reasoning tools, such as a term rewriting engine and a tableaux prover, various decision procedures, and, through the **Sledgehammer** proof-automation interface, external satisfiability modulo theories (SMT) solvers (including CVC4) and resolution-based automated theorem provers (ATPs), including E and SPASS (the **Metis**[b] proof method reconstructs resolution proofs generated by these ATPs).[2] It also features two model finders (counterexample generators): **Nitpick**[3] and **Nunchaku**.[4]

Isabelle features **locales** which are modules that structure large proofs. A locale fixes types, constants, and assumptions within a specified scope[3] so that they do not have to be repeated for every lemma.

Isar ("**intelligible semi-automated reasoning**") is Isabelle's formal proof language. It is inspired by the Mizar system.[3]

Isabelle has been used to formalize numerous theorems from mathematics and computer science, like Gödel's completeness theorem, Gödel's theorem about the consistency of the axiom of choice, the prime number theorem, correctness of security protocols, and properties of programming language semantics. Many of the formal proofs are maintained in the Archive of Formal Proofs, which contains (as of 2019) at least 500 articles with over 2 million lines of proof in total.[5]

The Isabelle theorem prover is free software, released under the revised BSD license.

Isabelle was named by Lawrence Paulson after Gérard Huet's daughter.[6]

Example proof

Isabelle allows proofs to be written in two different styles, the procedural and the declarative. Procedural proofs specify a series of tactics (theorem proving functions/procedures) to apply; while reflecting the procedure that a human mathematician might apply to proving a result, they are typically hard to read as they do not describe the outcome of the outcome of these steps. Declarative proofs (supported by Isabelle's proof language, Isar), on the other hand, specify the actual mathematical operations to be performed, and are therefore more easily read and checked by humans.

The procedural style has been deprecated in recent versions of Isabelle.

For example, a declarative proof by contradiction in Isar that the square root of two is not rational can be written as follows.

```
theorem sqrt2_not_rational:
  "sqrt 2 ∉ ℚ"
proof
  let ?x = "sqrt 2"
  assume "?x ∈ ℚ"
  then obtain m n :: nat where
    sqrt_rat: "¦?x¦ = m / n" and lowest_terms: "coprime m n"
    by (rule Rats_abs_nat_div_natE)
  hence "m^2 = ?x^2 * n^2" by (auto simp add: power2_eq_square)
  hence eq: "m^2 = 2 * n^2" using of_nat_eq_iff power2_eq_square by fastforce
  hence "2 dvd m^2" by simp
  hence "2 dvd m" by simp
  have "2 dvd n" proof -
    from ‹2 dvd m› obtain k where "m = 2 * k" ..
    with eq have "2 * n^2 = 2^2 * k^2" by simp
    hence "2 dvd n^2" by simp
    thus "2 dvd n" by simp
  qed
  with ‹2 dvd m› have "2 dvd gcd m n" by (rule gcd_greatest)
  with lowest_terms have "2 dvd 1" by simp
  thus False using odd_one by blast
qed
```

Applications

Isabelle has been used to aid formal methods for the specification, development and verification of software and hardware systems.

- In 2009, the L4.verified project at NICTA produced the first formal proof of functional correctness of a general-purpose operating system kernel:[7] the seL4 (secure embedded L4) microkernel. The proof is constructed and checked in Isabelle/HOL and comprises over 200,000 lines of proof script to verify 7,500 lines of C. The verification covers code, design, and implementation, and the main theorem states that the C code correctly implements the formal

specification of the kernel. The proof uncovered 144 bugs in an early version of the C code of the seL4 kernel, and about 150 issues in each of design and specification.

- The definition of the programming language Lightweight Java was proven type-sound in Isabelle.[8]

Larry Paulson keeps a list of research projects (https://isabelle.in.tum.de/community/Projects) that use Isabelle.

Alternatives

Several proof assistants provide similar functionality to Isabelle, including:

- Coq, similar system written in OCaml
- HOL, similar to Isabelle's HOL implementation
- Lean, similar system written in C++
- Mizar system
- Metamath
- Prover9

Notes

a. /ˌɪzəˈbɛl/
b. /ˈmiːtɪs/

References

1. Paulson, L. C. (1986). "Natural deduction as higher-order resolution". *The Journal of Logic Programming*. **3** (3): 237. arXiv:cs/9301104 (https://arxiv.org/abs/cs/9301104). doi:10.1016/0743-1066(86)90015-4 (https://doi.org/10.1016%2F0743-1066%2886%2990015-4).
2. Jasmin Christian Blanchette, Lukas Bulwahn, Tobias Nipkow, "Automatic Proof and Disproof in Isabelle/HOL" (https://people.mpi-inf.mpg.de/~jblanche/frocos2011-dis-proof.pdf), in: Cesare Tinelli, Viorica Sofronie-Stokkermans (eds.), *International Symposium on Frontiers of Combining Systems – FroCoS 2011* (https://books.google.com/books?id=TT18o_HohVwC&dq=), Springer, 2011.
3. Jasmin Christian Blanchette, Mathias Fleury, Peter Lammich & Christoph Weidenbach, "A Verified SAT Solver Framework with Learn, Forget, Restart, and Incrementality" (https://www.cs.vu.nl/~jbe248/sat.pdf), *Journal of Automated Reasoning* **61**:333–365 (2018).
4. Andrew Reynolds, Jasmin Christian Blanchette, Simon Cruanes, Cesare Tinelli, "Model Finding for Recursive Functions in SMT" (http://homepage.divms.uiowa.edu/~ajreynol/ijcar16a.pdf), in: Nicola Olivetti, Ashish Tiwari (eds.), *8th International Joint Conference on Automated Reasoning* (https://books.google.com/books?id=HxFkDAAAQBAJ&dq=), Springer, 2016.
5. Eberl, Manuel; Klein, Gerwin; Nipkow, Tobias; Paulson, Larry; Thiemann, René. "Archive of Formal Proofs" (https://www.isa-afp.org/). Retrieved 22 October 2019.
6. Gordon, Mike (1994-11-16). "1.2 History" (http://www.cl.cam.ac.uk/~mjcg/Research94/node3.html). *Isabelle and HOL*. Cambridge AR Research (The Automated Reasoning Group). Retrieved 2016-04-28.
7. Klein, Gerwin; Elphinstone, Kevin; Heiser, Gernot; Andronick, June; Cock, David; Derrin, Philip; Elkaduwe, Dhammika; Engelhardt, Kai; Kolanski, Rafal; Norrish, Michael; Sewell, Thomas; Tuch, Harvey; Winwood, Simon (October 2009). "seL4: Formal verification of an OS kernel" (htt

p://www.sigops.org/sosp/sosp09/papers/klein-sosp09.pdf) (PDF). *22nd ACM Symposium on Operating System Principles*. Big Sky, Montana, US. pp. 207–200.

8. Strniša, Rok; Parkinson, Matthew (2011-02-07). "Lightweight Java" (https://www.isa-afp.org/entries/LightweightJava.html). *Archive of Formal Proofs* (Feb 2011 ed.). ISSN 2150-914X (https://www.worldcat.org/issn/2150-914X). Retrieved 2019-11-25.

Further reading

- Lawrence C. Paulson, "The Foundation of a Generic Theorem Prover" (https://arxiv.org/abs/cs/9301105), *Journal of Automated Reasoning*, Volume 5, Issue 3 (September 1989), pages: 363–397, ISSN 0168-7433 (https://www.worldcat.org/search?fq=x0:jrnl&q=n2:0168-7433).
- Lawrence C. Paulson and Tobias Nipkow, "Isabelle Tutorial and User's Manual" (https://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-189.pdf), 1990.
- M. A. Ozols, K. A. Eastaughffe, and A. Cant, "DOVE: Design Oriented Verification and Evaluation" (https://link.springer.com/chapter/10.1007/BFb0000502), *Proceedings of AMAST 97*, M. Johnson, editor, Sydney, Australia. Lecture Notes in Computer Science (LNCS) Vol. 1349, Springer Verlag, 1997.
- Tobias Nipkow, Lawrence C. Paulson, Markus Wenzel, "Isabelle/HOL – A Proof Assistant for Higher-Order Logic" (https://isabelle.in.tum.de/doc/tutorial.pdf), 2020.

External links

- Isabelle website (https://isabelle.in.tum.de/)
- Isabelle on Stack Overflow (https://stackoverflow.com/tags/isabelle/)
- The Archive of Formal Proofs (https://www.isa-afp.org/)
- IsarMathLib (https://savannah.nongnu.org/projects/isarmathlib)