

Computer Assisted Proofs

Andrés Sicard-Ramírez

16th December 2022

Universidad EAFIT

Introduction

A big spectrum

pencil and paper proofs

⋮

formally verified proofs

own computer programs

⋮

automatic theorems provers

⋮

interactive theorems provers

Some families in the spectrum

- Automatic theorems provers for first-order logic
- Satisfiability modulo theories solvers
- Inductive theorem provers
- Interactive proof assistants
- Tools for the verification of programs

Automatic Theorems Provers for First-Order Logic

Automatic Theorems Provers for First-Order Logic (ATPs)

Some features

- The TPTP World

'A well know and established infrastructure that supports research, development, and deployment of automated theorem proving systems for classical logics.' [Sutcliffe 2010, p. 1.]

See <http://www.tptp.org/>.

Automatic Theorems Provers for First-Order Logic (ATPs)

Some features

- The TPTP World

'A well know and established infrastructure that supports research, development, and deployment of automated theorem proving systems for classical logics.' [Sutcliffe 2010, p. 1.]

See <http://www.tptp.org/>.

- The CADE ATP System Competition

The World Championship for Automated Theorem Proving!

Automatic Theorems Provers for First-Order Logic (ATPs)

Some features

- The TPTP World

'A well know and established infrastructure that supports research, development, and deployment of automated theorem proving systems for classical logics.' [Sutcliffe 2010, p. 1.]

See <http://www.tptp.org/>.

- The CADE ATP System Competition

The World Championship for Automated Theorem Proving!

- Some ATPs for (classic) first-order logic: E, Equinox, Metis, SPASS and Vampire.

The TPTP Languages

- **FOF** (Full First-Order Form): Full first-order logic [Sutcliffe 2009].
- **CNF** (Clause Normal Form): First-order logic in conjunctive clausal form [Sutcliffe 2009].
- **TFF** (Typed First-Order Form): **TF0** (monomorphic) [Sutcliffe, Schulz, Claessen and Baumgartner 2012] and **TF1** (rank-1 polymorphism) [Blanchette and Paskevich 2013].
- **THF** (Typed Higher-Order Form): **TH0** (monomorphic) [Sutcliffe and Benz Müller 2010] and **TH1** (rank-1 polymorphism) TODO.
- **TXF** (Typed Extended First-Order Form): **TX0** (monomorphic) and **TX1** (rank-1 polymorphism) TODO.

FOF Annotated Formulae and Problems

`fof(name, role, formula)` where:

`name` identifies the formula,

`formula` is a FOL-formula and

`role` \in { axiom, definition, hypothesis, conjecture }.

Each TPTP problem contains one or more annotated formulae.

Logical constant	Symbol
True	\$true
False	\$false
Conjunction	&
Disjunction	
Negation	~
Conditional	=>
Biconditional	<=>
Universal quantifier	![X] : p(X)
Existential quantifier	?[X] : p(X)

Satisfiability Modulo Theories Solvers

Satisfiability Modulo Theories

- The study of automatic methods for checking the satisfiability of first-order formulae with respect to some **background** theory is called Satisfiability Modulo Theories (SMT) [Barret, Sebastiani, Seshia and Tinelli [2009](#)].

Satisfiability Modulo Theories Solvers (SMT Solvers)

Satisfiability Modulo Theories

- The study of automatic methods for checking the satisfiability of first-order formulae with respect to some **background** theory is called Satisfiability Modulo Theories (SMT) [Barret, Sebastiani, Seshia and Tinelli 2009].

- Examples of background theories:

Theories of real numbers, integers, arrays, lists, records, bit-vectors, . . .

Satisfiability Modulo Theories Solvers (SMT Solvers)

Satisfiability Modulo Theories

- The study of automatic methods for checking the satisfiability of first-order formulae with respect to some **background** theory is called Satisfiability Modulo Theories (SMT) [Barret, Sebastiani, Seshia and Tinelli 2009].

- Examples of background theories:

Theories of real numbers, integers, arrays, lists, records, bit-vectors, . . .

SMT-LIB

The SMT library and standard.

Satisfiability Modulo Theories Solvers (SMT Solvers)

Some SMT solvers

- Z3 (Microsoft), CVC4 and veriT.

Satisfiability Modulo Theories Solvers (SMT Solvers)

Some SMT solvers

- Z3 (Microsoft), CVC4 and veriT.
- See http://en.wikipedia.org/wiki/Satisfiability_Modulo_Theories.

Satisfiability Modulo Theories Solvers (SMT Solvers)

Some SMT solvers

- **Z3** (Microsoft), **CVC4** and **veriT**.
- See http://en.wikipedia.org/wiki/Satisfiability_Modulo_Theories.

Z3 demo

See **Z3** home page.

Inductive Theorem Provers

Inductive Theorem Provers (ITPs)

- There is not standard

Inductive Theorem Provers (ITPs)

- There is not standard
- ITPs (Google for List of All Inductive Theorem Provers in The World).

Interactive Proof Assistants

*Proof assistants are computer systems that allow a user to do mathematics on a computer, but not so much the computing (numerical or symbolical) aspect of mathematics but the aspects of **proving** and **defining**. So a user can **set up** a mathematical theory, define properties and do logical reasoning with them. [Geuvers 2009, p. 3.]*

Some Proof Assistants

- Agda
- Coq
- Isabelle
- Lean

- Chalmers University of Technology and University of Gothenburg (Sweden)
- Based on Martin-Löf Type Theory
- Direct manipulation of proofs-objects
- Back-ends to **Haskell** (GHC) and **JavaScript**
- Written in **Haskell**

- INRIA (France)
- Based on the Calculus of Inductive Constructions
- Tactic-based
- Extraction of programs to **Haskell**, **OCaml** and **Scheme**
- Written in **OCaml** (with a bit of **C**)

- University of Cambridge (England) and Technical University of Munich (German)
- Based on HOL (Higher Order Logic)
- Tactic-based
- Extraction of programs to **Haskell**, **OCaml**, **Scala** and **SML**
- Written in **SML**
- Integration with ATPs and SMT solvers

- Microsoft Research and Carnegie Mellon University (USA)
- Based on dependent type theory
- Declarative style and tactic-bases
- Written in C++
- Integration with ATPs and SMT solvers

Real Applications

1977 PhD thesis [van Benthem Jutting 1977] ([Automath](#)).

One of the largest repository of formalised mathematics (1990–current).

MML includes [Bancerek, Byliński, Grabowski, Kornitowicz, Matuszewski, Naumowicz and Pak [2018](#)]:

- over 12000 definitions
- over 59000 theorems
- over 1290 articles
- over 253 authors

Proof of the Four Colour Theorem

- 1976** Every planar graph is four colorable. Announce in the Bulletin of AMS [K. Appel and Haken [1976](#)].
- 1977** Description of the proof in Scientific American [K. Appel and Haken [1977](#)].
- 1977** Simpler proof (still using a computer) [Robertson, Sanders, Seymour and Thomas [1997](#)].
- 2005** Formalised proof. Microsoft report [Gonthier [2005](#)] ([Coq](#)).
- 2008** Description of the formalised proof in Notices of AMS [Gonthier [2008](#)].

CompCert: A Verified Compiler for a Large Subset of C

2006 Journal (and proceedings) publication: POPL 2006 ([Coq](#)).

2009 Description in CACM [[Leroy 2009](#)].

2009 Proceedings of SOSP 2009 ([Isabelle/HOL](#)).

2010 Description in CACM [Klein, Andronick, Elphinstone, Heiser, Cock, Derrin, Elkaduwe, Engelhardt, Kolanski, Norrish, T. Sewell, Tuch and Winwood [2010](#)].

Proof of the Odd-Order Theorem

- 1963** Solvability of groups of odd-order [Feit and Thompson [1963](#)].
- 2012** Announce of the formalised proof (Gonthier and Théry 2012) ([Coq](#)).
- 2013** Formalised proof. Journal publication [Gonthier, Asperti, Avigad, Bertot, Cohen, Garillot, Le Roux, Mahboubi, O'Connor, Biha, Pasca, Rideau, Solovyev, Tassi and Théry [2013](#)].

Proof of Kepler Conjecture

- 1998** Hales announced proof of Kepler Conjecture.
- 2005** Journal publication 'uncertified' (Hales 2005).
- 2014** Announce of the formalised proof (Hales 2014) ([Isabelle](#) and [Isabelle/HOL](#), [HOL Light](#), [OCaml](#), [CamLP5](#) and [GLPK](#)).
- 2017** Formalised proof. Journal publication [Hales, Adams, Bauer, Dang, Harrison, Hoang, Kaliszyk, Magron, McLaughlin, T. T. Nguyen, Q. T. Nguyen, Nipkow, Steven, Pleso, Rute, Solovyev, Ta, Tran, Trieu, Urban, Vu and Zumkeller [2017](#)].

2014 Journal publication [Marché 2014] (Frama-C).

2014 Journal publication [Kumar, Myreen, Norrish and Owens 2014] ([HOL4](#)).

2015 Journal publication [A. W. Appel 2015] (Coq).

Certifying a File System using Crash Hoare Logic

2015 Proceedings of SOSP 2015 ([Coq](#)).

2017 Description in CACM [Chajed, Chen, Chlipala, Kaashoek, Zeldovich and Ziegler [2017](#)].

Rigorous Test-Oracle Specification and Validation for TCP/IP and the Sockets API

2001 Technical report ([HOL4](#)).

2015 Proceedings publication (2008, 2006, 2005, 2002, 2001).

2017 Journal publication [Bishop, Fairbairn, Mehnert, Norrish, Ridge, P. Sewell, Smith and Wansbrough [2018](#)].

2019 Proceedings of PLDI 2019 [Löow, Kumar, Tan, Myreen, Norrish, Abrahamsson and Fox [2019](#)] ([HOL4](#)).

- 2019** Journal publication [Beeson, Narboux and Wiedijk [2019](#)] ([HOL Light](#) and [Coq](#)).
- 2022** Description in The American Mathematical Monthly [Beeson [2022](#)].

Resolution of Keller's Conjecture

2020 Proceedings of IJCAR 2020 (SMT solver).

2021 Description in CACM [Monroe [2021](#)].

2020 Proceedings of IJCAR 2020 ([Isabelle/HOL](#)).

2022 Journal publication [Eßmann, Nipkow, Robillard and Sulejmani [2022](#)].

Tools for the Verification of Programs

Dedicate Tools and Systems for the Verification of Functional Programs

- **Hip**: The Haskell Inductive Prover

Dedicate Tools and Systems for the Verification of Functional Programs

- **Hip**: The Haskell Inductive Prover
- **Sparkle**: An interactive proof assistant for Clean

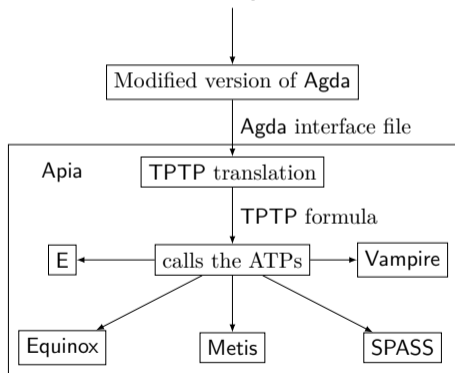
Dedicate Tools and Systems for the Verification of Functional Programs

- **Hip**: The Haskell Inductive Prover
- **Sparkle**: An interactive proof assistant for Clean
- **Zeno**: An automated proof system for Haskell program properties

An Application: Reasoning about Functional Programs

Combining Agda with ATPs*




Agda file + ATP-pragmas + [logical schemata options]







(Un)proven conjecture

References




References

-  Appel, A. W. (2015). Verification of a Cryptographic Primitive: SHA-256. ACM Transactions on Programming Languages and Systems 37.2, 7:1–31. DOI: [10.1145/2701415](https://doi.org/10.1145/2701415) (cit. on p. 39).
-  Appel, K. and Haken, W. (1976). Every Planar Graph is Four Colorable. Bulletin of the American Mathematical Society 8.5, pp. 711–712. DOI: [10.1090/bull/1556](https://doi.org/10.1090/bull/1556) (cit. on p. 32).
-  Appel, K. and Haken, W. (Oct. 1977). The Solution of the Four-Color-Map Problem. Scientific American, pp. 108–121. DOI: [10.1038/scientificamerican1077-108](https://doi.org/10.1038/scientificamerican1077-108) (cit. on p. 32).






References

-  Bancerek, G., Byliński, C., Grabowski, A., Kornilowicz, A., Matuszewski, R., Naumowicz, A. and Pak, K. (2018). The Role of the Mizar Mathematical Library for Interactive Proof Development in Mizar. *Journal of Automated Reasoning* 61.1–4, pp. 9–32. DOI: [10.1007/s10817-017-9440-6](https://doi.org/10.1007/s10817-017-9440-6) (cit. on p. 31).
-  Barret, C., Sebastiani, R., Seshia, S. A. and Tinelli, C. (2009). Satisfiability Module Theories. In: *Handbook of Satisfiability*. Ed. by Biere, A., Heule, M., van Maaren, H. and Walsh, T. IOS Press. Chap. 26 (cit. on pp. 13–15).
-  Beeson, M. (2022). Euclid after Computer Proof-Checking. *The American Mathematical Monthly* 129.7, pp. 623–646. DOI: [10.1080/00029890.2022.2069985](https://doi.org/10.1080/00029890.2022.2069985) (cit. on p. 43).
-  Beeson, M., Narboux, J. and Wiedijk, F. (2019). Proof-Checking Euclid. *Annals of Mathematics and Artificial Intelligence* 85.2–4, pp. 213–257. DOI: [10.1007/s10472-018-9606-x](https://doi.org/10.1007/s10472-018-9606-x) (cit. on p. 43).



References

-  Bishop, S., Fairbairn, M., Mehnert, H., Norrish, M., Ridge, T., Sewell, P., Smith, M. and Wansbrough, K. (2018). Engineering with Logic: Rigorous Test-Oracle Specification and Validation for TCP/IP and the Sockets API. *Journal of the ACM* 66.1, pp. 1–77. DOI: [10.1145/3243650](https://doi.org/10.1145/3243650) (cit. on p. 41).
-  Blanchette, J. C. and Paskevich, A. (2013). TFF1: The TPTP Typed First-Order Form with Rank-1 Polymorphism. In: *Automated Deduction (CADE-24)*. Ed. by Bonacina, M. P. Vol. 7898. *Lecture Notes in Artificial Intelligence*. Springer, pp. 414–420. DOI: [10.1007/978-3-642-38574-2_29](https://doi.org/10.1007/978-3-642-38574-2_29) (cit. on p. 9).
-  Chajed, T., Chen, H., Chlipala, A., Kaashoek, M. F., Zeldovich, N. and Ziegler, D. (2017). Certifying a File System Using Crash Hoare Logic: Correctness in the Presence of Crashes. *Communications of the ACM* 60.4, pp. 75–84. DOI: [10.1145/3051092](https://doi.org/10.1145/3051092) (cit. on p. 40).




References

-  Eßmann, R., Nipkow, T., Robillard, S. and Sulejmani, U. (2022). Verified Approximation Algorithms. Logical Methods in Computer Science 18.1, 36:1–36:21. DOI: [10.46298/lmcs-18\(1:36\)2022](https://doi.org/10.46298/lmcs-18(1:36)2022) (cit. on p. 45).
-  Feit, W. and Thompson, J. G. (1963). Solvability of Groups of Odd Order. Pacific Journal of Mathematics 13.3 (cit. on p. 35).
-  Geuvers, H. (2009). Proof Assistants: History, Ideas and Future. Sadhana 34.1, pp. 3–25. DOI: [10.1007/s12046-009-0001-5](https://doi.org/10.1007/s12046-009-0001-5) (cit. on p. 23).
-  Gonthier, G. (2005). A Computer-Checked Proof of The Four Colour Theorem. Tech. rep. Microsoft report (cit. on p. 32).
-  — (2008). Formal Proof—The Four Colour Theorem. Notices of the AMS 55.11, pp. 1382–1393 (cit. on p. 32).





References

-  Gonthier, G., Asperti, A., Avigad, J., Bertot, Y., Cohen, C., Garillot, F., Le Roux, S., Mahboubi, A., O'Connor, R., Biha, S. O., Pasca, I., Rideau, L., Solovyev, A., Tassi, E. and Théry, L. (2013). A Machine-Checked Proof of the Odd Order Theorem. In: 4th International Conference on Interactive Theorem Proving (ITP 2013). Ed. by Blazy, S., Paulin-Mohring, C. and Pichardie, D. Vol. 7998. Lecture Notes in Computer Science. Springer, pp. 163–179. DOI: [10.1007/978-3-642-39634-2_14](https://doi.org/10.1007/978-3-642-39634-2_14) (cit. on p. 35).
-  Hales, T., Adams, M., Bauer, G., Dang, T. D., Harrison, J., Hoang, L. T., Kaliszyk, C., Magron, V., McLaughlin, S., Nguyen, T. T., Nguyen, Q. T., Nipkow, T., Steven, O., Pleso, J., Rute, J., Solovyev, A., Ta, T. H. A., Tran, N. T., Trieu, T. D., Urban, J., Vu, K. and Zumkeller, R. (2017). A Formal Proof of the Kepler Conjecture. Forum of Mathematics, Pi 5, e2. DOI: [10.1017/fmp.2017.1](https://doi.org/10.1017/fmp.2017.1) (cit. on p. 36).





References

-  Klein, G., Andronick, J., Elphinstone, K., Heiser, G., Cock, D., Derrin, P., Elkaduwe, D., Engelhardt, K., Kolanski, R., Norrish, M., Sewell, T., Tuch, H. and Winwood, S. (2010). seL4: Formal Verification of an Operating-system Kernel. *Communications of ACM* 53.6, pp. 107–115. DOI: [10.1145/1743546.1743574](https://doi.org/10.1145/1743546.1743574) (cit. on p. 34).
-  Kumar, R., Myreen, M. O., Norrish, M. and Owens, S. (2014). CakeML: A Verified Implementation of ML. In: *Proceedings of the 41th annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL 2014)*, pp. 179–191. DOI: [10.1145/2535838.2535841](https://doi.org/10.1145/2535838.2535841) (cit. on p. 38).
-  Leroy, X. (2009). Formal Verification of a Realistic Compiler. *Communications of the ACM* 52.7, pp. 107–115. DOI: [10.1145/1538788.1538814](https://doi.org/10.1145/1538788.1538814) (cit. on p. 33).



References

-  Löow, A., Kumar, R., Tan, Y. K., Myreen, M. O., Norrish, M., Abrahamsson, O. r. and Fox, A. (2019). Verified Compilation on a Verified Processor. In: Proceedings of the 40th ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI 2019). ACM, pp. 1041–1053. DOI: [10.1145/3314221.3314622](https://doi.org/10.1145/3314221.3314622) (cit. on p. 42).
-  Marché, C. (2014). Verification of the functional behavior of a floating-point program: An industrial case study. Science of Computer Programming 96, Part 3, pp. 279–296. DOI: [10.1016/j.scico.2014.04.003](https://doi.org/10.1016/j.scico.2014.04.003) (cit. on p. 37).
-  Monroe, D. (2021). A Satisfying Result. Communications of the ACM 64.5, pp. 10–12. DOI: [10.1145/3453650](https://doi.org/10.1145/3453650) (cit. on p. 44).
-  Robertson, N., Sanders, D., Seymour, P. and Thomas, R. (1997). The Four-Colour Theorem. Journal of Combinatorial Theory, Series B 70.1, pp. 2–44. DOI: [10.1006/jctb.1997.1750](https://doi.org/10.1006/jctb.1997.1750) (cit. on p. 32).

References

-  Sicard-Ramírez, A. (2015). Reasoning about Functional Programs by Combining Interactive and Automatic Proofs. PhD thesis. PEDECIBA Informática. Universidad de la República. Uruguay (cit. on p. 50).
-  Sutcliffe, G. (2009). The TPTP Problem Library and Associated Infrastructure. The FOT and CNF Parts, v3.5.0. Journal of Automated Reasoning 43.4, pp. 337–362. DOI: [10.1007/s10817-017-9407-7](https://doi.org/10.1007/s10817-017-9407-7) (cit. on p. 9).
-  — (2010). The TPTP World—Infrastructure for Automated Reasoning. In: Logic for Programming, Artificial Intelligence, and Reasoning (LPAR-16). Ed. by Clarke, E. and Voronkov, A. Vol. 6355. Lecture Notes in Computer Science. Springer, pp. 1–12. DOI: [10.1007/978-3-642-17511-4_1](https://doi.org/10.1007/978-3-642-17511-4_1) (cit. on pp. 6–8).
-  Sutcliffe, G. and Benzmüller, C. (2010). Automated Reasoning in Higher-Order Logic using the TPTP THF Infrastructure. Journal of Formalized Reasoning 3.1, pp. 1–27. DOI: [10.6092/issn.1972-5787/1710](https://doi.org/10.6092/issn.1972-5787/1710) (cit. on p. 9).

References

-  Sutcliffe, G., Schulz, S., Claessen, K. and Baumgartner, P. (2012). The TPTP Typed First-Order Form with Arithmetic. In: Logic for Programming, Artificial Intelligence, and Reasoning (LPAR-18). Ed. by Bjørner, N. and Voronkov, A. Vol. 7180. Lecture Notes in Computer Science. Springer, pp. 406–419. DOI: [10.1007/978-3-642-28717-6_32](https://doi.org/10.1007/978-3-642-28717-6_32) (cit. on p. 9).
-  van Benthem Jutting, L. S. (1977). Checking Landau's "Grundlagen" in the Automath System. PhD thesis. Technische Hogeschool Eindhoven. DOI: [10.6100/IR23183](https://doi.org/10.6100/IR23183) (cit. on p. 30).