Automated reasoning

In computer science, in particular in knowledge representation and reasoning and metalogic, the area of **automated reasoning** is dedicated to understanding different aspects of reasoning. The study of automated reasoning helps produce computer programs that allow computers to reason completely, or nearly completely, automatically. Although automated reasoning is considered a sub-field of artificial intelligence, it also has connections with theoretical computer science and philosophy.

The most developed subareas of automated reasoning are automated theorem proving (and the less automated but more pragmatic subfield of interactive theorem proving) and automated proof checking (viewed as guaranteed correct reasoning under fixed assumptions). Extensive work has also been done in reasoning by analogy using induction and abduction.[1]

Other important topics include reasoning under uncertainty and non-monotonic reasoning. An important part of the uncertainty field is that of argumentation, where further constraints of minimality and consistency are applied on top of the more standard automated deduction. John Pollock's OSCAR system[2] is an example of an automated argumentation system that is more specific than being just an automated theorem prover.

Tools and techniques of automated reasoning include the classical logics and calculi, fuzzy logic, Bayesian inference, reasoning with maximal entropy and many less formal *ad hoc* techniques.

Early years

The development of formal logic played a big role in the field of automated reasoning, which itself led to the development of artificial intelligence. A formal proof is a proof in which every logical inference has been checked back to the fundamental axioms of mathematics. All the intermediate logical steps are supplied, without exception. No appeal is made to intuition, even if the translation from intuition to logic is routine. Thus, a formal proof is less intuitive and less susceptible to logical errors.[3]

Some consider the Cornell Summer meeting of 1957, which brought together many logicians and computer scientists, as the origin of automated reasoning, or automated deduction.[4] Others say that it began before that with the 1955 Logic Theorist program of Newell, Shaw and Simon, or with Martin Davis' 1954 implementation of Presburger's decision procedure (which proved that the sum of two even numbers is even).[5]

Automated reasoning, although a significant and popular area of research, went through an "AI winter" in the eighties and early nineties. The field subsequently revived, however. For example, in 2005, Microsoft started using verification technology in many of their internal projects and is planning to include a logical specification and checking language in their 2012 version of Visual C.[4]

Significant contributions

Principia Mathematica was a milestone work in formal logic written by Alfred North Whitehead and Bertrand Russell. Principia Mathematica - also meaning Principles of Mathematics - was written with a purpose to derive all or some of the mathematical expressions, in terms of symbolic logic. Principia Mathematica was initially published in three volumes in 1910, 1912 and 1913.[6]

Logic Theorist (LT) was the first ever program developed in 1956 by Allen Newell, Cliff Shaw and Herbert A. Simon to "mimic human reasoning" in proving theorems and was demonstrated on fifty-two theorems from chapter two of Principia Mathematica, proving thirty-eight of them.[7] In addition to proving the theorems, the program found a proof for one of the theorems that was more elegant than the one provided by Whitehead and Russell. After an unsuccessful attempt at publishing their results, Newell, Shaw, and Herbert reported in their publication in 1958, *The Next Advance in Operation Research*:

> *"There are now in the world machines that think, that learn and that create. Moreover, their ability to do these things is going to increase rapidly until (in a visible future) the range of problems they can handle will be co- extensive with the range to which the human mind has been applied."*[8]

Examples of Formal Proofs

Year	Theorem	Proof System	Formalizer	Traditional Proof
1986	First Incompleteness	Boyer-Moore	Shankar[9]	Gödel
1990	Quadratic Reciprocity	Boyer-Moore	Russinoff[10]	Eisenstein
1996	Fundamental- of Calculus	HOL Light	Harrison	Henstock
2000	Fundamental- of Algebra	Mizar	Milewski	Brynski
2000	Fundamental- of Algebra	Coq	Geuvers et al.	Kneser
2004	Four Color	Coq	Gonthier	Robertson et al.
2004	Prime Number	Isabelle	Avigad et al.	Selberg-Erdős
2005	Jordan Curve	HOL Light	Hales	Thomassen
2005	Brouwer Fixed Point	HOL Light	Harrison	Kuhn
2006	Flyspeck 1	Isabelle	Bauer- Nipkow	Hales
2007	Cauchy Residue	HOL Light	Harrison	Classical
2008	Prime Number	HOL Light	Harrison	Analytic proof
2012	Feit-Thompson	Coq	Gonthier et al.[11]	Bender, Glauberman and Peterfalvi
2016	Boolean Pythagorean triples problem	Formalized as SAT	Heule et al.[12]	None

Proof systems

Boyer-Moore Theorem Prover (NQTHM)

The design of NQTHM was influenced by John McCarthy and Woody Bledsoe. Started in 1971 at Edinburgh, Scotland, this was a fully automatic theorem prover built using Pure Lisp. The main aspects of NQTHM were:

1. the use of Lisp as a working logic.
2. the reliance on a principle of definition for total recursive functions.
3. the extensive use of rewriting and "symbolic evaluation".
4. an induction heuristic based the failure of symbolic evaluation.[13][14]

HOL Light
Written in OCaml, HOL Light is designed to have a simple and clean logical foundation and an uncluttered implementation. It is essentially another proof assistant for classical higher order logic.[15]

Coq
Developed in France, Coq is another automated proof assistant, which can automatically extract executable programs from specifications, as either Objective CAML or Haskell source code. Properties, programs and proofs are formalized in the same language called the Calculus of Inductive Constructions (CIC).[16]

Applications

Automated reasoning has been most commonly used to build automated theorem provers. Oftentimes, however, theorem provers require some human guidance to be effective and so more generally qualify as proof assistants. In some cases such provers have come up with new approaches to proving a theorem. Logic Theorist is a good example of this. The program came up with a proof for one of the theorems in Principia Mathematica that was more efficient (requiring fewer steps) than the proof provided by Whitehead and Russell. Automated reasoning programs are being applied to solve a growing number of problems in formal logic, mathematics and computer science, logic programming, software and hardware verification, circuit design, and many others. The TPTP (Sutcliffe and Suttner 1998) is a library of such problems that is updated on a regular basis. There is also a competition among automated theorem provers held regularly at the CADE conference (Pelletier, Sutcliffe and Suttner 2002); the problems for the competition are selected from the TPTP library.[17]

See also

- Automated machine learning (AutoML)
- Automated theorem proving
- Reasoning system
- Semantic reasoner
- Program analysis (computer science)
- Applications of artificial intelligence
- Outline of artificial intelligence
- Casuistry • Case-based reasoning
- Abductive reasoning
- Inference engine

- Commonsense reasoning

Conferences and workshops

- International Joint Conference on Automated Reasoning (IJCAR)
- Conference on Automated Deduction (CADE)
- International Conference on Automated Reasoning with Analytic Tableaux and Related Methods

Journals

- *Journal of Automated Reasoning*

Communities

- Association for Automated Reasoning (AAR)

References

1. Defourneaux, Gilles, and Nicolas Peltier. "Analogy and abduction in automated deduction (http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.81.501&rep=rep1&type=pdf)." IJCAI (1). 1997.
2. John L. Pollock
3. C. Hales, Thomas "Formal Proof" (https://www.ams.org/notices/200811/tx081101370p.pdf), University of Pittsburgh. Retrieved on 2010-10-19
4. "Automated Deduction (AD)" (http://www.cs.cornell.edu/info/projects/nuprl/Intro/AutoDeduction/autoded.html), *[The Nature of PRL Project]*. Retrieved on 2010-10-19
5. Martin Davis (1983). "The Prehistory and Early History of Automated Deduction". In Jörg Siekmann; G. Wrightson (eds.). *Automation of Reasoning (1) — Classical Papers on Computational Logic 1957–1966* (https://www.springer.com/gp/book/9783642819544). Heidelberg: Springer. pp. 1–28. ISBN 978-3-642-81954-4. Here: p.15
6. "Principia Mathematica" (http://plato.stanford.edu/entries/principia-mathematica/), at Stanford University. Retrieved 2010-10-19
7. "The Logic Theorist and its Children" (http://www.cs.swarthmore.edu/~eroberts/cs91/projects/ethics-of-ai/sec1_2.html). Retrieved 2010-10-18
8. Shankar, Natarajan *Little Engines of Proof (http://www.csl.sri.com/~shankar/)*, Computer Science Laboratory, SRI International. Retrieved 2010-10-19
9. Shankar, N. (1994), *Metamathematics, Machines, and Gödel's Proof* (https://books.google.com/books?id=JmEXH9TllNcC), Cambridge, UK: Cambridge University Press, ISBN 9780521585330
10. Russinoff, David M. (1992), "A Mechanical Proof of Quadratic Reciprocity", *J. Autom. Reason.*, **8** (1): 3–21, doi:10.1007/BF00263446 (https://doi.org/10.1007%2FBF00263446), S2CID 14824949 (https://api.semanticscholar.org/CorpusID:14824949)

11. Gonthier, G.; et al. (2013), "A Machine-Checked Proof of the Odd Order Theorem" (http://www.cs.unibo.it/~asperti/PAPERS/odd_order.pdf) (PDF), in Blazy, S.; Paulin-Mohring, C.; Pichardie, D. (eds.), *Interactive Theorem Proving*, Lecture Notes in Computer Science, vol. 7998, pp. 163–179, CiteSeerX 10.1.1.651.7964 (https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.651.7964), doi:10.1007/978-3-642-39634-2_14 (https://doi.org/10.1007%2F978-3-642-39634-2_14), ISBN 978-3-642-39633-5, S2CID 1855636 (https://api.semanticscholar.org/CorpusID:1855636)
12. Heule, Marijn J. H.; Kullmann, Oliver; Marek, Victor W. (2016). "Solving and Verifying the Boolean Pythagorean Triples Problem via Cube-and-Conquer". *Theory and Applications of Satisfiability Testing – SAT 2016*. Lecture Notes in Computer Science. Vol. 9710. pp. 228–245. arXiv:1605.00723 (https://arxiv.org/abs/1605.00723). doi:10.1007/978-3-319-40970-2_15 (https://doi.org/10.1007%2F978-3-319-40970-2_15). ISBN 978-3-319-40969-6. S2CID 7912943 (https://api.semanticscholar.org/CorpusID:7912943).
13. *The Boyer-Moore Theorem Prover (http://www.cs.utexas.edu/~moore/best-ideas/nqthm/index.html)* Retrieved on 2010-10-23
14. Boyer, Robert S. and Moore, J Strother and Passmore, Grant Olney *The PLTP Archive (https://www.cs.utexas.edu/users/moore/best-ideas/pltp/index.html)*. Retrieved on 2023-07-27
15. Harrison, John *HOL Light: an overview (http://www.cl.cam.ac.uk/~jrh13/slides/tphols-18aug09/slides.pdf)*. Retrieved 2010-10-23
16. *Introduction to Coq (http://coq.inria.fr/a-short-introduction-to-coq)*. Retrieved 2010-10-23
17. *Automated Reasoning (http://plato.stanford.edu/entries/reasoning-automated/)*, Stanford Encyclopedia. Retrieved 2010-10-10

External links

- International Workshop on the Implementation of Logics (https://web.archive.org/web/20090310081227/http://www.csc.liv.ac.uk/~konev/iwil2008/)
- Workshop Series on Empirically Successful Topics in Automated Reasoning (https://web.archive.org/web/20100206221517/http://www.eprover.org/EVENTS/es_series.html)