View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by Online Research Database In Technology

Technical University of Denmark

Formalization of Algorithms and Logical Inference Systems in Proof Assistants

Schlichtkrull, Anders

Published in:
Proceedings of the 13th Scandinavian Conference on Artificial Intelligence (SCAI 2015)

Link to article, DOI:
10.3233/978-1-61499-589-0-188

Publication date:
2015

Document Version
Peer reviewed version

Link back to DTU Orbit

Citation (APA):
Schlichtkrull, A. (2015). Formalization of Algorithms and Logical Inference Systems in Proof Assistants. In S. Nowaczyk (Ed.), Proceedings of the 13th Scandinavian Conference on Artificial Intelligence (SCAI 2015) (pp. 188-190). IOS Press. (Frontiers in Artificial Intelligence and Applications, Vol. 278). DOI: 10.3233/978-1-61499-589-0-188

DTU Library
Technical Information Center of Denmark

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

© 2015 The authors and IOS Press

Thirteenth Scandinavian Conference on Artificial Intelligence – SCAI 2015
S. Nowaczyk (Ed.) pp. 188–190
IOS Press, 2015

Formalization of Algorithms and Logical Inference Systems in Proof Assistants

Anders SCHLICHTKRULL [1]
Technical University of Denmark, Kgs. Lyngby, Denmark

Extended Abstract

The purpose of the project is to use proof assistants, in particular Isabelle, to prove key properties about algorithms and logical inference systems. Algorithms and logic are fundamental concepts of artificial intelligence (AI) and are used to make abstract descriptions of hardware and software that one can reason about. The two disciplines date back to ancient Greece, but have become increasingly important during the last century with the invention and rise of computers.

Logical inference systems are central in the study of logics. A logical inference system consists of axioms and inference rules. The inference rules can then be used to prove the validity of formulas in a formal language from the axioms and other formulas that have already been proved. Some inference systems can be implemented as computer programs called automated theorem provers, and therefore, a computer can be used to automatically prove formulas valid.

A proof assistant is a computer program that helps its user in formalizing programs or mathematical concepts, and in proving theorems about those. In addition, the proof assistant checks that the proofs are correct according to a set of axioms and rules. Many proof assistants can automate smaller or larger proof steps, and contain large libraries of lemmas that the user can take advantage of in her own proofs. Some proof assistants can even export executable code from the formalized programs.

The use of a proof assistant has several advantages compared to doing proofs only in natural language. Since the proof is checked, we know that it only uses arguments that we have agreed are correct – namely the axioms and rules. Therefore, one cannot by mistake make a clever, but wrong, argument in the proof. Historically there have been many examples of mistakes happening in mathematics. For instance, in 1880 a "proof" of the four color theorem was presented, and it was accepted by many mathematicians. However, many years later it was found to contain the mistake of relying on a false assumption [1]. A proof assistant would not have accepted the proof in the first place. Another advantage is the automation which can help the user focus on the big picture without having to worry about the small steps, while at the same time, having the computer check the small steps.

[1]Corresponding Author: Anders Schlichtkrull, DTU Compute, Richard Petersens Plads, Building 324, 2800 Kgs. Lyngby, Denmark; E-mail: s103467@student.dtu.dk. MSc student; PhD student from 15 September 2015.

Despite of the advantages of proof assistants, they are still only used by a small number of mathematicians and computer scientists. An obvious step towards getting them to be more used is to actually use them. In the project this will be done by applying proof assistants to the fields of algorithms and logical inference systems.

The application of proof assistants to these fields is especially advantageous since the correctness of algorithms and logical inference systems have real world implications on our society. For instance, proof systems are used in description logic which has applications in medical science, software engineering, and the semantic web [2]. An inference system implemented as an automated theorem prover has also been used in the verification of the floating point divide code of the AMD5K86 microprocessor. Likewise, an aircraft alerting algorithm has been verified [3,4].

Another aspect is that some of the proof assistants can generate code from the developed formalizations [5]. This means that if an algorithm or a logical inference system is formalized in a deterministic way, then it can be exported as executable code. Thus, one can, from a formalized logical inference system, generate an automated theorem prover and be very confident in its correctness.

Many algorithms, logical inference systems, and proofs about them have yet to be formalized in a proof assistant. The project will contribute with a study and formalization of a subset of those. This is worthwhile because it increases confidence in their correctness. Furthermore, these proofs and a framework around them can then be used and adapted by other researchers to study variants of the systems without having to build up the proofs from the bottom, but still having confidence in their correctness. Additionally, it could also be interesting to try to formalize some newer research or try to make formalizations of results as they are being made.

Many new logical inference systems and variants of existing logical inference systems for different logics are developed every year. These are for instance presented at the International Joint Conference on Automated Reasoning (IJCAR) and its subconferences. However, only few of the presented results are formalized in proof assistants. The project therefore has great potential for paving the way for the adaption of proof assistants in this scientific community.

There are already some formalizations of logic and logical inference systems. For instance, the syntax and semantics of first-order logic as well as several important theorems about those have been formalized [6]. Furthermore, a natural deduction system has been proved sound and complete [7]. Another logical inference system has been proven correct and exported to executable code [8]. The famous ground resolution system has also been formalized [9]. Likewise, several algorithms have been formalized, for instance, mergesort [10] and Dijkstra's algorithm [11]. The completeness of logical inference systems, using an abstract property of possibly infinite derivation trees independently of the concrete syntax or inference rules, has recently been established [12].

Algorithms and logical inference systems abound in AI. Several chapters are devoted to first-order logic in the leading AI textbook [13] and on pages 345-357 the resolution calculus is described in details. It provides a sound and complete logical inference system using knowledge bases in conjunctive normal form. I have formalized the resolution calculus in Isabelle and have proved it sound. Completeness is work in progress, but the results so far show that there are a number of imprecisions and even mistakes in the literature. In addition I have contributed to the formalization of natural deduction [14,15], another well-established logical inference system.

References

[1] J.A. Bundy and U.S.R. Murty. *Graph Theory with Applications*. The Macmillian Pres Ltd., 1976.

[2] F. Baader, D. Calvanese, D.L. McGuinness, D. Nardi, and P.F. Patel-Schneider. *The Description Logic Handbook: Theory, Implementation and Applications*. Cambridge University Press, 2010.

[3] J Strother Moore, T. Lynch, and M. Kaufmann. A Mechanically Checked Proof of the Correctness of the Kernel of the AMD5K86 Floating-Point Division Algorithm. *IEEE Transactions on Computers*, 47, 1996.

[4] V. Carreo and C. Muoz. Aircraft Trajectory Modeling and Alerting Algorithm Verification. In Mark Aagaard and John Harrison, editors, *Theorem Proving in Higher Order Logics*, volume 1869 of *Lecture Notes in Computer Science*, pages 90–105. Springer Berlin Heidelberg, 2000.

[5] Isabelle Developers. Isabelle overview. `https://isabelle.in.tum.de/overview.html`, 2014. [Online; accessed 1-April-2015].

[6] J. Harrison. Formalizing Basic First Order Model Theory. In Jim Grundy and Malcolm Newey, editors, *Theorem Proving in Higher Order Logics*, volume 1479 of *Lecture Notes in Computer Science*, pages 153–170. Springer Berlin Heidelberg, 1998.

[7] S. Berghofer. First-Order Logic According to Fitting. *Archive of Formal Proofs*, 2007. `http://afp.sf.net/entries/FOL-Fitting.shtml`, Formal proof development.

[8] T. Ridge. A Mechanically Verified, Efficient, Sound and Complete Theorem Prover For First Order Logic. *Archive of Formal Proofs*.

[9] F. Maric. Formal Verification of Modern SAT Solvers. *Archive of Formal Proofs*.

[10] C. Sternagel. Proof Pearl - A Mechanized Proof of GHC's Mergesort. *Journal of Automated Reasoning*, 51(4):357–370, 2013.

[11] B. Nordhoff and P. Lammich. Dijkstra's Shortest Path Algorithm. *Archive of Formal Proofs*.

[12] J. C. Blanchette, A. Popescu, and D. Traytel. Unified Classical Logic Completeness - A Coinductive Pearl. In *Automated Reasoning - 7th International Joint Conference, IJCAR 2014, Held as Part of the Vienna Summer of Logic, VSL 2014, Vienna, Austria, July 19-22, 2014. Proceedings*, pages 46–60, 2014.

[13] S. J. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 3rd edition, 2009.

[14] J. Villadsen, A. Schlichtkrull, and A. V. Hess. Meta-Logical Reasoning in Higher-Order Logic. Accepted at 29th International Symposium Logica, Hejnice, Czech Republic, 2015.

[15] J. Villadsen, A. B. Jensen, and A. Schlichtkrull. NaDeA: A Natural Deduction Assistant with a Formalization in Isabelle. Pages 253-262 in Proceedings of 4th International Conference on Tools for Teaching Logic, Rennes, France, 2015.