Formally Correct Deduction Methods for Computational Logic

PhD Project Description

From, Asta Halkjær

Publication date:
2020

Document Version
Publisher's PDF, also known as Version of record

PhD Project Description:
Formally Correct Deduction Methods for Computational Logic

Asta Halkjær From, DTU Compute, Denmark

Background

Deductive reasoning is the logical process of reaching a conclusion from one or more premises. As a society, we are increasingly trusting computers to do this reasoning and surrounding ourselves with ones that do so. The following is how Lawrence Paulson opens his Introduction to Computational Logic [1]:

> *Computers control everything from heart pacemakers to aircraft, putting everyone at risk when they fail.*

Self-driving cars, for instance, make deductions at split-second intervals with possibly severe consequences for a faulty inference. To trust these cars we clearly need to trust the correctness of their deduction methods. More generally, electronic circuits themselves are often computer-verified to behave according to specification using deduction methods. But to have faith in this computer-verification we need to trust the reasoning behind it; reasoning that is typically only checked by humans. That is, we engage with computational logic both directly, as with self-driving cars, and indirectly, as with verified hardware. This ubiquity of computational logic motivates our focus on it.

> *I now do my mathematics with a proof assistant and do not have to worry all the time about mistakes in my arguments or about how to convince others that my arguments are correct.*

The above quote by the late Fields Medal recipient Voevodsky [2] is our motivation for ensuring formal correctness of deduction methods using a proof assistant. A proof assistant is a computer program used to conduct computer-checked proofs based on formal logic, mathematics, and computer science. We can use proof assistants to ensure that logics, and algorithms based on them, satisfy requirements of safety, security, soundness, completeness, and termination. State-of-the-art proof assistants such as Isabelle/HOL [3] or Coq [4] give unprecedented possibilities for developing new logics and algorithms with verifiable properties of correctness. Coq won the ACM Software System Award in 2013 [5].

Proof Assistants Isabelle is well-suited for verification of algorithms. The Verified Algorithm Analysis project at TU Munich has produced several results, for instance a formalization of Gröbner bases that includes the correctness of Buchberger's algorithm and a possible translation of the verified algorithm into executable code [6]. Another example is a formalization of the Edmonds-Karp algorithm for network flow which the authors claim is accessible even to Isabelle novices [7]. Both of these examples are also to be found in the Archive of Formal Proof [8], a project that contains more than 500 entries about topics in computer science, like AVL trees; logic, like Gödel's incompleteness proofs; physics, e.g. no observer can travel faster than the speed of light; and mathematics, like the results mentioned. All entries are checked for correctness by the Isabelle proof assistant.

Another application of Isabelle, among other tools for formal verification, is in the seL4 project, an operating system microkernel programmed in C with an end-to-end proof of implementation correctness and security enforcement [9]. The formal correctness proof guarantees that a number of classical issues from C programming, like buffer overflows or memory leaks, cannot occur.

Jensen et al. have verified the soundness of a declarative prover in Isabelle [10] and Villadsen et al. [11] have verified the soundness and completeness of an automatic prover for a fragment of first-order logic.

Applied Logics Description logic and hybrid logic can model knowledge bases used in systems like IBM's Watson, where information has to be stored and reasoned about. Paraconsistent logic is one way of reasoning in spite of conflicting data to prevent, for instance, a misdiagnosis [12].

Natural logic is applicable to the domain of natural language processing, used in products like Apple's Siri or Amazon's Alexa and an important part of making robots natural to communicate with.

Epistemic logic, which has a verified calculus in Isabelle due to the applicant [13], is used to reason about the knowledge of others. For social robots like those used insides hospitials, a theory of mind is important to not convey needless information or get in the way physically.

Doxastic logic is suited for making agents reason from beliefs instead of certain knowledge and nonmonotic logic deals with belief revision: having a robot change its mind when discovering new informa-

tion. These features are important parts of making robots fit in social situations. Baltag, Gierasimczuk, Özgün, Sandoval and Smets have developed a dynamic epistemic logic that incorporates learning [14].

Aim and Scope

The goal of the project is to strengthen the logical foundations of computer reasoning by ensuring correctness of the deductive methods involved, using state-of-the-art tools like the Isabelle proof assistant.

The motivating idea is to bring the state and understanding of reasoning systems up to date by documenting and developing the logics applied in Artificial Intelligence for mutual benefit. On the one hand, formally verifying properties like soundness, completeness or computational complexity is part of the way towards responsible and explanable AI. This formal verification is not standard practice, leading us to base our systems on definitions that may contain undiscovered ambiguities or proofs that are only human-verified sketches with unpublished details. The application of a proof assistant will not only allow us to fully trust these logics but produce fully-spelled out, non-ambiguous formalizations to learn from in a format that makes them easy for others to build on or modify for new purposes. On the other hand, it may be necessary to develop new proof-assisting techniques for this project, contributing not only to other disciplines but to automated or interactive theorem proving itself. Such tools would be of great importance to researchers.

This project takes up the challenge of developing and verifying logics for computer reasoning as well as algorithms based on these logics. The Isabelle proof assistant allows exporting verified algorithms into programming languages like Haskell, OCaml and Scala, thus enabling us to produce running systems.

References

[1] Computational Logic: Its Origins and Applications. Lawrence C. Paulson. Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences, 474(2210), 2018.

[2] Univalent Foundations. Vladimir Voevodsky. Institute for Advanced Study, Princeton, NJ, 2014.

[3] Isabelle/HOL — A Proof Assistant for Higher-Order Logic. Tobias Nipkow, Lawrence C. Paulson & Markus Wenzel. Springer, 2002.

[4] Interactive Theorem Proving and Program Development — Coq'Art: The Calculus of Inductive Constructions. Yves Bertot & Pierre Castéran. Springer, 2004.

[5] ACM Software System Award: Institutions/individuals who developed software systems with lasting influence on computing. Association for Computing Machinery. (Accessed 2019-10-07.) `https://awards.acm.org/software-system/award-winners?year=2013`

[6] Alexander Maletzky & Fabian Immler. Gröbner Bases of Modules and Faugère's F_4 Algorithm in Isabelle/HOL. International Conference on Intelligent Computer Mathematics (pp. 178–193). Springer, 2018.

[7] Formalizing the Edmonds-Karp Algorithm. Peter Lammich & S. Reza Sefidgar. International Conference on Interactive Theorem Proving (pp. 219–234). Springer, 2016.

[8] Archive of Formal Proofs. `https://www.isa-afp.org`

[9] seL4: Formal Verification of an Operating-System Kernel. Gerwin Klein et al. Commun. ACM, 53(6):107–115, June 2010.

[10] Programming and Verifying a Declarative First-Order Prover in Isabelle/HOL. Alexander Birch Jensen, John Bruntse Larsen, Anders Schlichtkrull & Jørgen Villadsen. AI Communications 31:281–299 2018.

[11] A Verified Simple Prover for First-Order Logic. Jørgen Villadsen, Anders Schlichtkrull & Andreas Halkjær From. Proceedings of the 6th Workshop on Practical Aspects of Automated Reasoning (PAAR) — CEUR-WS 2162:88–104 2018.

[12] Formalizing a Paraconsistent Logic in the Isabelle Proof Assistant. Jørgen Villadsen & Anders Schlichtkrull. Transactions on Large-Scale Data- and Knowledge-Centered Systems 34:92–122 2017.

[13] Epistemic Logic. Andreas Halkjær From. Archive of Formal Proofs, October 2018.

[14] A Dynamic Logic for Learning Theory. Alexandru Baltag, Nina Gierasimczuk, Aybüke Özgün, Ana Lucia Vargas Sandoval, & Sonja Smets. Journal of Logical and Algebraic Methods in Programming Volume 109, December 2019.