Foreword

Manfred Kerber · Christoph Lange · Colin Rowat

Published online: 19 April 2014

There is a long-term vision of making information systems dependable. Indeed formal systems have reached a strength that would allow to use them for this purpose. However, in many application domains they are not actually employed since their usage still requires a lot of expertise and also a major commitment in resources. In particular, there is a high threshold, which has to be overcome to get started, and only after a significant initial investment there is a benefit from their usage. This means that often the systems are used only in areas where failure is prohibitively expensive, such as aircraft control.

Contributions to this special issue have been solicited in order to address this and related issues. This issue originates in the symposium "Enabling Domain Experts to use Formalized Reasoning" (see http://cs.bham.ac.uk/research/projects/formare/events/aisb2013), which we organized in 2013 in order to bring researchers from the formal methods/theorem proving community and different application domains together.

There are some examples from engineering where costly accidents could have been prevented if formal methods had been applied. There are areas where the application of such methods is now standard, and others where there is still a lot of scope for a formal analysis and the usage of corresponding systems. In economics, the subtlety of issues involved in good auction design may have led to low revenues in auctions of public goods such as the 3G radio spectra. Similarly, banks' value-at-risk (VaR) models—the leading method of financial risk measurement—are too

The editors would like to thank the EPSRC, (grant EP/J007498/1, http://cs.bham.ac.uk/research/projects/formare/) for its support in the organization of a symposium which resulted in this special issue.

M. Kerber · C. Lange (✉)
School of Computer Science, University of Birmingham, Birmingham, UK
e-mail: math.semantic.web@gmail.com

M. Kerber
e-mail: M.Kerber@cs.bham.ac.uk

C. Lange
Institute for Applied Computer Science, University of Bonn, Bonn, Germany

C. Lange
Fraunhofer IAIS, Sankt Augustin, Germany

C. Rowat
Department of Economics, University of Birmingham, Birmingham, UK
e-mail: C.Rowat@bham.ac.uk

large and change too quickly to be thoroughly vetted by hand. Verifying properties of a model requires formally specifying them; for VaR models, any work would have to start with this most basic step, as regulators' current desiderata are subjective and ambiguous.

We believe that progress can be made by representing the knowledge underlying such models and mechanisms in a formal, explicit, and machine-verifiable way. Contemporary computer science offers a wide choice of knowledge representation languages well supported by verification tools. Such tools have been successfully applied, e.g., for verifying software that controls commuter rail or payment systems. Still, domain experts without a strong computer science background find it challenging to choose the right tools and to use them. This special issue aims at investigating ways to support them. Some problems can be addressed now, others will bring new challenges to computer science.

For this special issue we have invited economists to review the formalization efforts of Arrow's theorem. These are special in two ways, firstly since Arrow's theorem plays an important role in economic theory and secondly, since its formal proofs are among the first attempts (if not the first ones) to apply formal methods to economics.

Furthermore we have, in an open call, invited submissions from attendees of the 2013 symposium and others. Of these submissions we have chosen another four for publication in this special issue. They describe approaches of practical applications of reasoning systems in physical environments (railway systems, hybrid systems in engineering, and optical systems) and a machine learning approach to lower the threshold of applying a formal system by domain experts rather than by experts in formal methods.

In the rest of this foreword we give a brief introduction to the different articles. We hope that the reader enjoys reading the articles.

Automated Reasoning in Social Choice

One of the most celebrated results of 20th century economic theory is Arrow's theorem on the impossibility of aggregating individual preferences into a social welfare function that respects a number of apparently innocuous axioms. As a classic result, it has inspired a variety of proofs, including the three one page proofs of Geanakoplos [1]. Interest in what could be learned from the relative ease with which these proofs could be formalized led to formalization's of some of these proofs in higher-order logic [2,3].

These efforts, in turn, inspired Tang and Lin [4] to develop a new proof of Arrow's theorem, using a SAT solver on a small base case, and extending the impossibility found there to the full case by a manually proved induction result. Inspecting the output of their SAT solver led Tang and Lin to identify a generalization of Arrow's theorem. Their techniques were then applied by Geist and Endriss [5] to a related problem, and used to produce 84 impossibility theorems—a lifetime's manual work.

With the exception of Geanakoplos' work, though, all of the papers cited above have been written by members of the mechanized reasoning community, and published in their journals. Thus, economists have remained almost completely unaware of their work and their methods. An immediate consequence of this is that there has been no detailed comment from the economics community on the significance of the results of Tang and Lin, or Geist and Endriss. The informal discussion has largely been sceptical, expressing a preference for general principles and gaining conceptual insights over lists of new theorems. Thus, the fruitfulness of this interface between computer science and economics has remained largely unexplored.

Siddhart Chatterjee's and Arunava Sen's contribution to this volume is therefore a welcome first step towards bridging the gap between these disciplines. Seeking to draw both into the discussion, they provide an introduction to both the Tang and Lin and Geist and Endriss techniques as well as to social choice theory. They agree with the informal commentary, that "new proofs of existing results are not the most interesting aspects of the papers", with existing manual proofs often being more insightful. However, they find some of the newly generated results "quite striking", concluding that Tang and Lin's generalization of Arrow's theorem is the only such result known to them that uses only the IIA (Independence of Irrelevant Alternatives) axiom. They believe that, without Tang and Lin's exhaustive computational search, this demonstration of the power of IIA would not have been discovered.

Noting that almost any result in social choice theory can be phrased as an impossibility result, Chatterjee and Sen believe that the scope for applying these techniques is considerable. Their subsequent discussion of possibilities defines a promising and feasible agenda for future research.

Verifying Railway Scheme Plans

Transportation has been one of the more traditional domains of applying formal methods; indeed, it has been argued that verification of railway control software poses a grand challenge for computer science. However, the long-standing use of formal methods in the railway domain does not yet mean that they are easy to use for railway engineers.

Phillip James and Markus Roggenbach base their work on a visual, UML-based domain-specific language (DSL) that has already been used in industrial settings. Railway engineers who would like their scheme plans to be verified formally can continue modelling them in a familiar DSL. The scheme plans are automatically translated to the Common Algebraic Specification Language CASL, a language well supported by formal verification tools. In addition, a safety property of the scheme plan, which implies that two trains cannot collide, is generated automatically and can be verified in a push-button manner. This push-button verification, however, requires preparation to become feasible and scalable, by proving intermediate lemmas that follow from domain knowledge.

While the paper presents this result for a simplified "academic" DSL, James and Roggenbach have also applied the same methods and techniques in a long-standing collaboration with industry. Moreover, given the grand challenge nature of the railway domain, they are confident that their approach will be transferable to other domains of industrial engineering.

Formal Analysis of Optical Systems

Most successful applications of formal methods so far have used discrete mathematics. Safety-critical applications in need of formal verification, however, also occur in the continuous domain of optics—think, e.g., of laser surgery. Optical systems have so far been analyzed using error-prone paper proofs, computer simulations, which are approximate, hard to scale, and not even applicable to quantum optics, and computer algebra systems (CASs), which may be inaccurate and not completely reliable.

Higher-order logic (HOL) is a suitable formalism for continuous mathematics; many of the mathematical foundations of optics have already been formalized in the HOL Light system. On top of this, Sanaz Khan-Afshar, Umair Siddique, Mohamed Yousri Mahmoud, Vincent Aravantinos, Ons Seddiki, Osman Hasan and Sofiène Tahar present HOL Light libraries for four different theories of optics—ray, wave, electromagnetic, and quantum optics—and the application of the ray optics library to the Two-Mirror Fabry Pérot Resonator, a component widely used in optical systems. Still, even with this formalization in place, optical system analysis exposes equations with no closed-form solution or problems not covered by formal libraries. While being aware of the loss in trustability, the authors therefore present an initial link, via the OpenMath exchange language, between HOL Light and the Mathematica CAS, which is employed for simplifying expressions and computing values.

Collaborative Verification of Hybrid Systems

Advances in robotics and computer science lead to increasingly autonomous systems being used or being about to become used more widely. For instance, Nevada was the first jurisdiction to license cars that autonomously participate in road traffic (in 2011). Hybrid systems, which have physical and software components, are getting pervasive and we all depend more and more on them. To prevent such systems from doing potentially serious damage it is important that they function to a high level of reliability. For instance, an autonomous vehicle should not cause accidents with cars, pedestrians or other entities in its environment; that is, it should observe a safe behaviour. In particular, this means that an autonomous vehicle has to keep an appropriate distance to other traffic participants and must choose its speed such that under all circumstances it can come to a complete standstill to avoid imminent collision.

In order to increase reliability and acceptability of such hybrid systems, but also in order to minimize the risk of indemnification claims, safety conditions of hybrid systems can be proved formally in a system that allows to reason about its physical properties. In their contribution, Stefan Mitsch, Grant Olney Passmore and André Platzer present a system that enables different experts to collaboratively reason about physical systems (with paths, velocities, and accelerations). They present a heterogeneous tool set that allows to model a hybrid system graphically (in UML) or textually, to exchange and compare models and proofs, and to manage verification tasks. This way it is possible to tackle large-scale verification tasks.

Recycling Proof Patterns

Verification tools such as the interactive theorem prover Coq require users to have specialized knowledge of the proof strategies applied in verification in specific domains. Even when a user understands how the system can be applied in principle, it is often difficult to develop and find proofs in practice. In many practical situations good knowledge of the proof is needed before making a proof attempt. Since systems like Coq contain large libraries of proofs, a lot of material is available from which related proof patterns can be extracted. Jónathan Heras and Ekaterina Komendantskaya apply machine learning techniques in their ML4PG (Machine Learning for Proof General) system to extract related patterns automatically in order to relate them to the new proof via the Proof General interface. The main technique applied is the method of recurrent clustering of Coq terms, types, and proof objects.

In this contribution, the authors report on several case studies they have conducted. They firstly illustrate how to use ML4PG for detecting proof patterns in given proof libraries prior to the start of a new proof development. Secondly, they study how their system can help to indicate to the user that there are no related patterns across different libraries, which means that the user should not waste time searching for such a pattern. Thirdly, they study how a novice user can be supported in finding and reusing proof patterns that have been previously built by experienced users of the system.

References

1. Geanakoplos, J.D.: Three brief proofs of Arrow's impossibility theorem. Econ. Theory **26**(1), 211–215 (2005)
2. Nipkow, T.: Social choice theory in HOL: Arrow and Gibbard-Satterthwaite. J. Automa. Reason **43**(3), 289–304 (2009)
3. Wiedijk, F.: Arrow's impossibility theorem. J. Formaliz. Math. **15**(4), 171–174 (2009)
4. Tang, P., Lin, F.: Computer-aided proofs of Arrow's and other impossibility theorems. Artif. Intell. **173**(11), 1041–1053 (2009)
5. Geist, C., Endriss, U.: Automated search for impossibility theorems in social choice theory: ranking sets of objects. J. Artif. Intell. Res. **40**, 143–174 (2011)