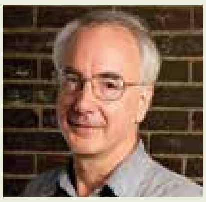
Impact Objectives

- Build a foundation that will allow a wide variety of formal methods, ranging from automatic model checkers to interactive theorem provers, to work together to establish formal properties of computer systems
- Design and validate a framework for defining the semantics of a wide range of proof evidence

The hunt for a broad-spectrum proof certificate

Dale Miller is Director of Research at both Inria Saclay Île-de-France and the Laboratoire d'Informatique, near Paris. Below, he discusses how his work on the recently completed ProofCert project will support a wide range of formal methods to work together to establish formal properties of computer systems



In what ways is it hoped that ProofCert will make an impact on computer security systems?

Formal proofs offer the highest degree of trust, one based on reproducibility and not on reputation. With high degrees of trust should come higher degrees of interoperability: it should be easier for one prover to trust results from other provers. It should also allow a prover to trust a proof whose origin is unknown. ProofCert should allow the communities working on the verification of computer systems and development of formalised mathematics to establish a high degree of trust in their results. Since most security flaws in current computer systems result from programming errors, progress on proving, at least some aspects, of programs correct should also lead to code that is more secure and less subjected to hacking.

Who will make best use of the outcomes from this work?

Further impacts of this work should be an enabling of both marketplaces and libraries for proofs. One can imagine that the manufacturer of some safety critical system, such as a fly-by-wire control system or a controller for a pacemaker, might need to have certain subsystems formally proved correct in order to pass government certification. In such cases, the manufacturer could, in principle, create an empty proof certificate that simply contains the formula to be proved. Such an empty certificate could then be placed into a marketplace where people could attempt to find a proof to fill that certificate using whatever proof technology they like. The only requirement of those tools is that they output their proofs in a checkable format. The Foundational Proof Certificate (FPC) framework, developed by myself and my colleagues Zakaria Chihani and Fabien Renaud, would then allow for the checking of many kinds of proof evidence. Similarly, proof libraries could be built using proof certificates from many different provers.

Can you talk about the current state of play when it comes to formal proofs?

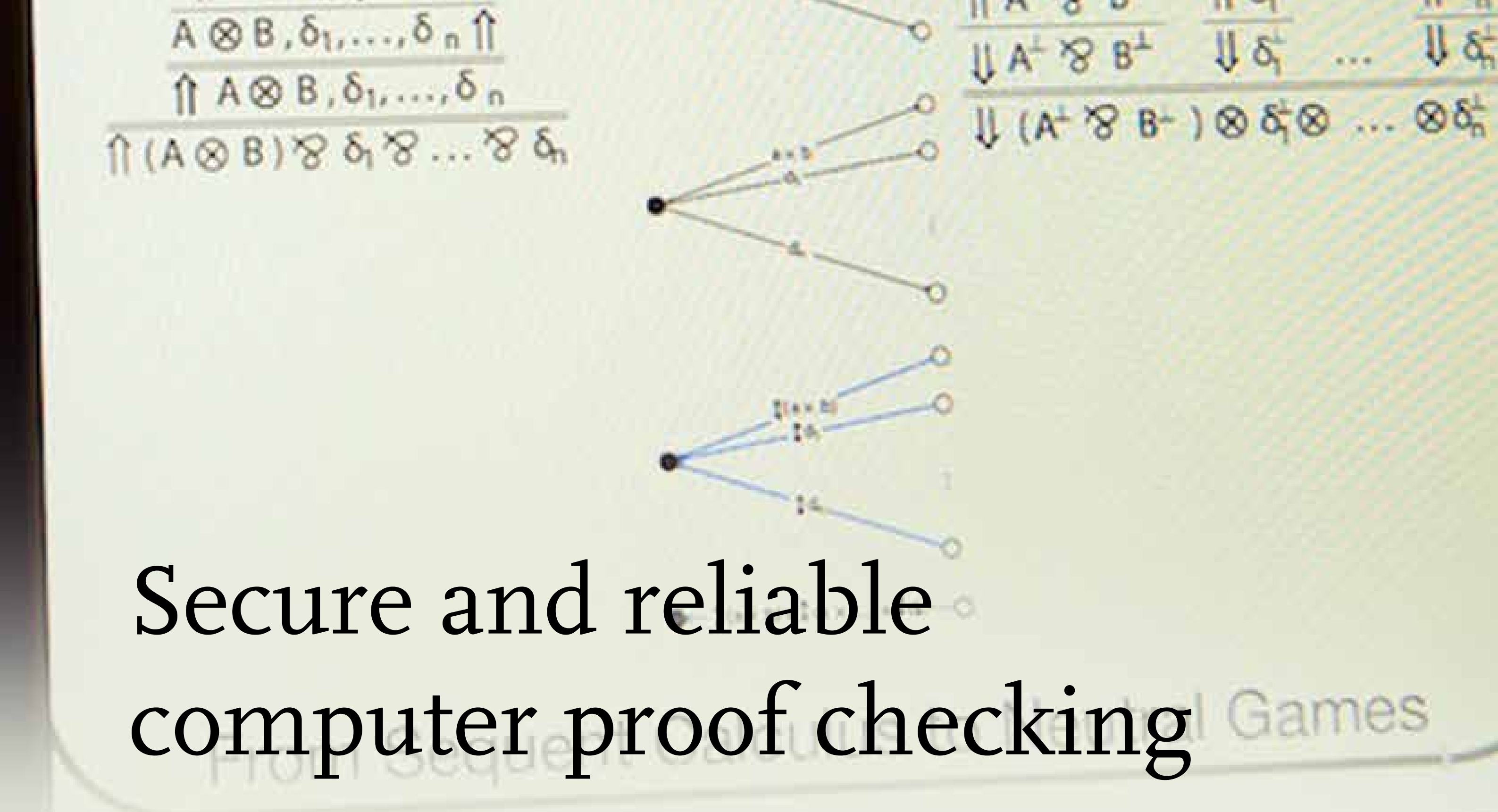
The mathematical topic of formal proofs is rich and diverse. Most computerised proof systems use ad hoc structures to represent proofs: typically, such structures are not meaningful to other proof systems and maybe not to later versions of itself. Thus, libraries of theorems and their proofs are generally tied to one particular proof assistant. Work in any other proof assistant usually must be redone: sharing is generally the exception. The theorem-proving community is aware of this problem and there are several efforts to bridge gaps between small collections of provers. ProofCert's approach is to provide a theoretical framework for these efforts.

What are the major steps or challenges to be overcome in the ProofCert programme?

Most of the effort in ProofCert has been focused on theory and design. Our prototype systems have helped to validate both that theory and design work and they can work as reference checkers. The next step will be to undertake a serious engineering and experimentation effort. Many in the theorem-proving and formal methods communities are interested in communicating their proofs for other systems to trust. The FPC framework should provide them with flexible means to do exactly that.

From your perspective what are the next steps to overcome these challenges?

There are several important next steps in this effort. One of these is to try to reinsert humans into the picture. The core of the ProofCert effort involved having computers generate proofs that can be transmitted to other computers for checking, using and archiving. Of course, this is all done so humans can trust their computer systems and their mathematics. We should also be able to find ways for humans to learn from proofs as well via browsing, interacting or transforming proofs. Another step to take is to standardise our formats and to build tools around those standards. This step is critical for getting our colleagues to see the practical value this project could have in their efforts.



Secure and reliable computer proof checking

The recently completed European Research Agency-funded ***ProofCert*** *project worked for five years to deliver a framework to validate computer systems in order to improve the safety and security of modern lives*

Computer systems are everywhere in our society and their integration with all parts of our lives is constantly increasing. Along with the global use of computing systems comes an increasing need to deal with the accuracy of these systems. There are a host of computer systems, such as those in cars, airplanes, missiles and hospital equipment, where the precision of software is paramount.

THE NEED FOR FORMAL PROOFS

Big changes in the attitude towards accuracy are also taking place in consumer electronics. In the past, establishing the correctness of desktop PCs, music players and telephones, for example, was not urgent since rebooting their systems to recover from errors or living without a feature due to bugs were mostly nuisances and not life-threatening. But today, these same devices are now tightly integrated into networks that need to ensure the security of information and the anonymity of users while remaining safe from attacks from malicious software, which almost always exploit bugs within software.

The ability to provide at least some formal guarantees about software systems is directly related to the ability to deploy new functionality and services. Formal proofs can play an important role in making software more secure and reliable. A formal proof is a mathematically defined object that describes in specific detail why a particular statement must be true. Since these details can be trusted, the existence of a checked formal proof should mean people can be sure of its accuracy.

BUILDING TRUST

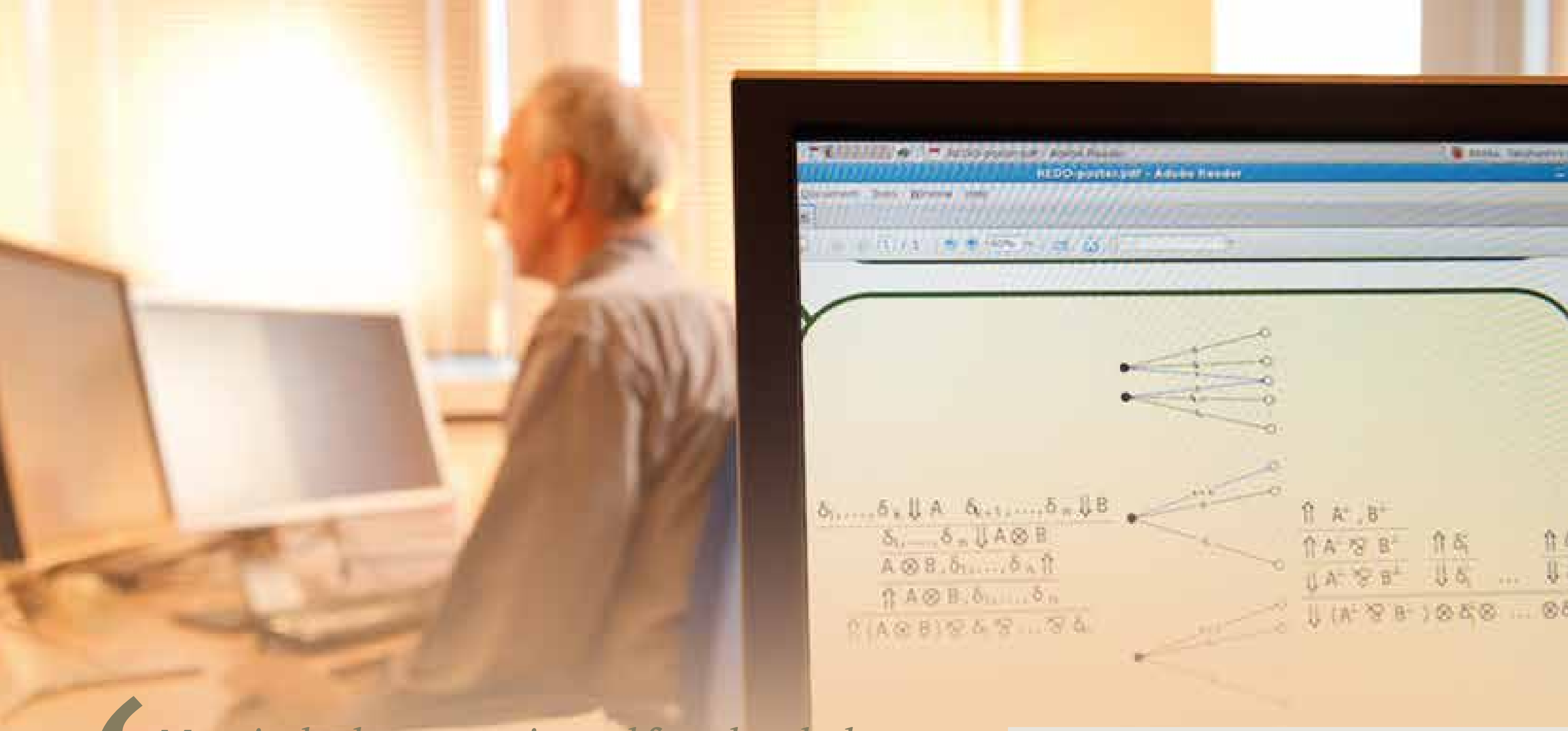
The recently completed ProofCert project was funded by the European Research Agency and was tasked with addressing many of these issues around computer security. Project Coordinator Professor Dale Miller observes that in the modern world, trust in computer systems is important and becoming critical. He notes that the work on establishing formal methods attempts to ensure the accuracy of software and hardware systems with mathematical precision and certainty. Trust is often based on either reputation or reproducibility. The existence of formal proofs and proof checking means that the origin of a proof is not important for trusting it.

'Building formal proofs of non-trivial software is difficult and relies on various software systems, such as interactive proof assistants and model checkers,' explains Miller. 'Proofs of software are generally not deep and elegant. Instead they can be tedious, shallow and large. In order to discover and manipulate such proofs, a number of software tools are needed.' Since systems such as interactive proof assistants are complex and evolving, formal verification of these is difficult and a major undertaking. Furthermore, formally validating provers is, in many ways, not desirable since verification usually means freezing the evolution of a software system. It would also lead to a siloed world where trust centred on the few proof assistants that had been formally validated.

A FRAMEWORK FOR PROOF CHECKING

The primary goal of ProofCert was to develop the Foundational Proof Certificate (FPC) framework. This, explains Miller, is where a wide range of proof systems can be formally defined. If a proof is then paired with the definition of its proof systems as an FPC, that pair can be checked by proof checkers both now and in the future. 'The existence and use of the FPC framework also means that checking proofs becomes reproducible in the sense that at any time in the future it is possible to confirm that a certain large and tedious document is, in fact, a proof,' Miller adds.

A secondary goal of ProofCert has been the construction of reference proof checkers. Miller outlines that the checkers built are based on the logic programming paradigm, in particular the use of the λProlog programming language (developed by Miller and Gopalan Nadathur from the University of Minnesota) has allowed the building of simple, flexible and trustworthy checkers for a wide range of FPC definitions. 'The ambition with the ProofCert effort,' says Miller, 'is to shift the centre of the formal verification effort from individual proof assistants to a network of sharing and mutually trusting provers and proof libraries.'



> Many in the theorem-proving and formal methods communities are interested in communicating their proofs for other systems to trust. The FPC framework should provide them with flexible means to do exactly that

At the very core of this work is the mathematical topic of structural proof theory. 'Since early work by Frege, Hilbert and Gentzen in the first half of the 20th century, proofs in logic have been studied and their properties and structures have been established', notes Miller. In that literature, various kinds of proof systems, with names like 'sequent calculus', 'natural deduction' and 'resolution refutations', have been defined and their formal mathematical properties have been proved. Ultimately, this means that programmers should be able to use the same format to write proof-checking software.

Unfortunately, the many theorem-proving systems that are in use today generally output proofs that are completely dependent on the technology that produced them. This means that if a version number on a prover changes, its previous proofs may not be proofs anymore. This is what makes the work the ProofCert project has been involved with so important, because it is focused on moving the machine-based formal proofs away from technology and embedding them as universal and permanent.

FUTURE EFFORTS

The ultimate goal is to make it possible for someone 50 years from now to recheck a proof certificate with absolute certainty. However there are a number of challenges to this. For example, if a proof certificate is based on one particular application which in 50 years has disappeared, that certificate may no longer be checkable. Also, the field of proofs is complex and so it is highly unlikely that one piece of software will be able to effectively check large proofs in a number of different domains. Multiple checkers will most likely need to be available so that they can be optimised for different tasks.

The project team will now be looking at ways to standardise the formats and build tools around those standards which Miller says is essential to ensure users can see the benefits the project can bring. Working also with Tomer Libal, Marco Volpe and Sonia Marin, the effort is growing to incorporate various forms of modal and temporal logics that are in common use in the analysis of computer systems.

With the ProofCert project now completed the team is looking at ways to communicate their results. Working with Roberto Blanco and Quentin Heath, Miller is hoping to employ proof certifications into several aspects of the operation of Abella, an existing interactive theorem prover being developed by the Inria-based team and their colleagues from the University of Minnesota. They are also considering other less conventional approaches to the use of proof certificates, such as making it possible for theorem-proving competitions to check that competitors are actually providing complete correct proofs.

Project Insights

FUNDING

- European Research Council
- This project has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement number 291592

CONTACT

Dale Miller
Project Coordinator

T: +33 177578053
E: dale.miller@inria.fr
W: http://cordis.europa.eu/project/rcn/102755_en.html

PROJECT COORDINATOR BIO

Professor Dale Miller has been a Professor at the University of Pennsylvania and the École Polytechnique in France, and a Department Head at the Pennsylvania State University. Miller was a two-term editor-in-chief of the *ACM Transactions on Computational Logic* and is on the editorial board of the *Journal of Automated Reasoning*. He was awarded the LICS Test-of-Time award in 2011 and 2014 for papers written in 1991 and 1994, respectively.

ProofCert

Inria
INVENTEURS DU MONDE NUMÉRIQUE

erc
European Research Council
Established by the European Commission

This project is funded by
the European Union