Contents

The QED Workshop

held at Argonne National Laboratory
May 18–20, 1994

Gail W. Pieper, synthesizer

Abstract

On May 18–20, 1994, Argonne National Laboratory hosted the QED Workshop. The workshop was supported by special funding from the Office of Naval Research. The purpose of the workshop was to assemble a group of researchers to consider whether it is desirable and feasible to build a proof-checked encyclopedia of mathematics, with an associated facility for theorem proving and proof checking. Among the projects represented were Coq, Eves, HOL, ILF, Imps, MathPert, Mizar, NQTHM, NuPrl, OTTER, Proof Pad, Qu-Prolog, and RRL.

Although the content of the QED project is highly technical—rigorously proof-checked mathematics of all sorts—the discussions at the workshop were rarely technical. No prepared talks or papers were given. Instead, the discussions focused primarily on such political, sociological, practical, and aesthetic questions, such as Why do it? Who are the customers? How can we get mathematicians interested? What sort of interfaces are desirable?

The most important conclusion of the workshop was that QED is an idea worthy pursuing, a statement with which virtually all the participants agreed.

In this document, we capture some of the discussions and outline suggestions for the start of a QED scientific community.

1 Introduction

On May 18–20, 1994, Argone National Laboratory hosted the QED Workshop. The workshop was organized by R. Boyer of the University of Texas at Austin and by E. Lusk of the Mathematics and Computer Science Division at Argonne.

The workshop gathered approximately thirty researchers, representing ongoing worldwide efforts in theorem proving and mathematics. Among the projects represented were the Coq, Eves, HOL, ILF, Imps, MathPert, Mizar, NQTHM, NuPrl, OTTER, Proof Pad, Qu-Prolog, and RRL.

Many of those attending had previously contributed to a document known as the *QED Manifesto*—an anonymously authored document that discusses the desirability and feasibility of organizing a public-domain database of a substantial part of mathematical knowledge, including proofs suitable for machine checking.

The focus of the workshop was on mathematics and automated deduction. The objective was to consider the desirability and feasibility of building a proof-checked encyclopedia of mathematics, with an associated facility for theorem proving and proof checking. Such a project could be used in university mathematics education, graduate research, mathematics research, and K-12 school education.

The structure of the workshop was intentionally kept informal; no formal presentations were given. Moreover, the discussions themselves were focused on nontechnical issues—potential customers, philosophy, linkages with other symbolic and numerical systems. The result was lively discussion, often sharp disagreement, about a variety of political, sociological, and aesthetic questions involved in organizing such a major undertaking as the QED project.

In the remainder of this report, we summarize the activities of the QED Workshop. First, we review the background to the QED project. We then list the participants and present summaries of the pertinent issues raised. We conclude with an evaluation of the workshop and suggestions for future work.

2 Background

Over the past several years, the field of theorem proving has made tremendous strides. Numerous theorem-proving systems have been developed, many formal (and informal) proofs produced, and significant questions in mathematics and logic solved. Nevertheless, many disciplines that heavily use mathematics remain unfamiliar with automated deduction systems.

In particular, professional research mathematicians are only beginning to show interest in using theorem proving to attack hard problems. A recent article in the *New York Times* illustrates the level of skepticism manifest by many mathematicians about the feasibility of automated theorem provers [2]. Compounding the problem is the difficulty of encoding mathematical results and their proofs in a single database that can be easily used by students not only at the graduate level but also at the K-12 level. A principal objective of the QED Workshop, then, was to discuss the feasibility of constructing an "intellectual infrastructure"—an organized framework encapsulating theorem provers and checkers, proofs, and the basic ideas of mathematics.

The groundwork for discussion at the QED Workshop was a document known as the *QED Manifesto* [3]. Many of the participants had contributed to this document, directly or indirectly, by comments via electronic mail. Moreover, several of the participants are already working on mathematics-related projects in theorem proving. One notable example is the Mizar system [4], which has a substantial body of formally checked mathematics. The objective at the QED Workshop, however, was not to identify specific programs but to understand *how* and *whether* such programs might be organized, together with a library of proofs and results, into a multilayered QED system.

2.1 Participants

The workshop comprised approximately 30 participants from a wide variety of areas related to theorem proving, logic, and mathematics. The list of participants is given below.

Michael Beeson	San Jose State University
Robert Boyer	University of Texas at Austin
Bernd Dahn	Humboldt-University at Berlin
Masami Hagiya	University of Tokyo
John Harrison	University of Cambridge
Joan Hart	University of Wisconsin
M. Randall Holmes	Boise State University
Paul Jackson	Cornell University
Thomas Jech	Pennsylvania State University
Deepak Kapur	State University of New York at Albany
Kenneth Kunen	University of Wisconsin
Ewing Lusk	Argonne National Laboratory
William McCune	Argonne National Laboratory
Chet Murthy	INRIA-Rocquencourt
Ross Overbeek	Argonne National Laboratory
William Pase	Odyssey Research Associates, Inc.
Piotr Rudnicki	University of Alberta
John Staples	The University of Queensland
Rick Stevens	Argonne National Laboratory
F. Javier Thayer	MITRE Corp.
Andrzej Trybulec	Warsaw University
Tomas Uribe	Stanford University
Ralph Wachter	ONR
Richard Waldinger	SRI International
Toby Walsh	IRST
Larry Wos	Argonne National Laboratory

2.2 Organization

The meeting was deliberately informal: the seating was arranged in a circle, and participants were encouraged to voice their opinions freely. No formal presentations were scheduled; indeed, the "agenda" was more a dynamic than a static document, modified each morning and afternoon and adhered to only loosely.

As one might expect among groups who have worked independently (in most cases, for decades), the discussion was lively. The participants did agree that QED must necessarily differ from earlier projects in both scope and strategy. In contrast to the Automath project of the 1960s, for example, QED must be a large project, must be widely available (via Internet), and must have vast numbers of people working on it.

3 Issues Discussed

The participants also agreed on a number of broad issues involving QED—the potential users (those in education, verification, and theorem proving), the need for involving mathematicians, the importance of modularity and ease of use.

In the remainder of this report, we present some of the discussions related to these and other, more controversial issues.

3.1 Objectives of QED

The topic of immediate concern was what QED should be. Several possibilities were raised and discussed briefly:

- An archive and reference source of mathematical results and their proofs, to be invoked by mathematicians and students. ("Referencing" covers a wide range of applications, for example, encyclopaedic reference, a database of software and systems).
- A database of the world's best theorem-proving systems, accessible to both naive users and theorem-proving researchers.
- A facility for checking, storing, and communicating new formal proofs of results.
- An electronic journal of new formal proofs.
- A facility for producing machine-checked mathematical textbooks.
- A component library of machine-checked software and hardware and a facility for composing them to form new verified systems.
- A repository of courses on all areas of mathematics. (Although there are many introductions to mathematics based on mechanical proof-checking, these have typically not been shared.)

One point on which the participants readily agreed was that entries in the QED system should, at least ultimately, be checked with the *highest standard of rigor possible*.

Bourbaki was cited as the best example so far of mathematics organized into a coherent framework. According to André Weil, "Perhaps the most important contribution of Bourbaki was to carry out a famous proposal made by the great German mathematician David Hilbert in 1900 that mathematics be placed on a more secure foundation." He noted: "Hilbert just said so, and Bourbaki did it" [1]. Yet several participants believe that Bourbaki is what QED should *not* be, in particular because Bourbaki is highly praised, but rarely used.

Wide use by research communities was deemed vital. Three research communities appear the most obvious users: (1) computer scientists (e.g., for automated deduction and developing verified systems), (2) logicians, and (3) mathematicians (e.g., for proof-checking, teaching, and publication). Most participants agreed that the idea of merely archiving proofs would be boring to many mathematicians and that if this is the key objective of the QED project, mathematicians probably will not contribute. Mathematicians will be interested only to the extent that we can help them do new mathematics.

Clarifying the benefits of QED to the wider community appears to be the best approach to attracting potential users (as well as the support of sponsors). Several such benefits were identified:

- More attractive and productive mathematics teaching, leading to increased mathematical literacy and higer average levels of mathematical skill.
- More efficient professional use of mathematical reasoning, leading to reduced costs and higher levels of quality assurance.
- Increased industrial reuse of mathematical reasoning, for example, in the development of verified systems, leading to increased productivity.

3.2 QED Components

Complementing the discussion about what QED should be was a discussion about how QED should be organized. A modular system seems most desirable, comprising the following levels:

-
- Library of results. Couched in high-level language and standard mathematics notation, this library would be organized by field, with references to related results.
- Library of proofs. One should be able to examine the proof of each theorem at different levels of complexity. The decision as to how much to display at each level should be largely under human control, but it might be somewhat subject to automation.
- Interaction with theorem provers. A naive user should be able to encounter a prover superficially (talking in high-level language); he should be able to enter a tutorial which will teach him to be a sophisticated user.
- Mechanism for database entries. It should be possible to submit results proven on the system to "editors" at various levels, from high-school and hobbyist projects through master's level development of existing theories not yet implemented through genuine new research results.

Mizar may be considered a trial run of part of QED. It has the high-level language and the proof checker in preliminary versions, and it operates on something close to proof objects. Nevertheless, Mizar lacks powerful automated reasoning techniques and the sophisticated low-level language with reflection projected for QED. If Mizar were to be used as a model of a basis of QED, then, one would wish to add (at least) a database browser, more automatic proving facilities, and a facility for organizing lemmas more systematically. Of the greatest problems facing the Mizar user (and probably the user of any QED-like system of significant size) is how to find out whether and where routine facts have already been established.

3.3 Interfaces

Interface issues raised considerable discussion. The term itself is ambiguous. It can refer to something as apparently superficial as the choice of a window system, or it can include the definition of the proof language itself.

Some participants view interface matters as unimportant so long as deep technical issues remain to be solved. Yet, interface issues determine 50% of the final code of most systems.

Three areas of interfacing were identified as significant:

- Networking technologies. QED must be interfaced with emerging network systems such as Mosaic and the World Wide Web.
- Advanced visualization. The interface should include facilities for graphing and the use of diagrams. We should also explore the possibility of virtual reality (for example, for improving one's understanding of non-Euclidean geometries).
- Language. An effective user interface requires the development of a high-level language (something that looks rather like natural language, though perhaps more closely resembling a "pidgin English"). It would be desirable to have the interface structure designed so that it could be converted painlessly from stilted formal English to stilted formal French, German, Chinese, etc., without meddling with its fundamental structure. The objective here is to enable naive users (including computer-naive mathematicians and mathematically naive computer scientists) to talk to the system, read theorems, and enter proofs into the system without knowing very much about the underlying system.

Interfaces with symbolic and numerical systems would also be useful, particularly in moving the QED philosophy into other disciplines that heavily use mathematics. Admittedly, this task task would not be easy, however, since many disciplines do not think in terms of an axiomatic approach. Moreover, some symbolic manipulation systems are very unsound and hence might lead to corrupting the reliability of the QED structure.

3.4 Object Language and Basic Theory

Some attention was given to the topic of object languages and proofs in QED.

Some participants felt that a standard object language was needed (although it should allow for the evolution of dialects of the standard object language and coexistence with alien object languages such as constructive systems).

Others noted that one should distinguish between the object language as a set of well-formed formulas and the basic theory formulated in the object language. For example, first-order logic can be used to reason in different basic theories (e.g., Peano arithmetic, MacLane set theory). The question thus becomes, Should one declare a single theory to be basic?

It was argued that this might create obstacles to QED research: for example, restricting the quantifiers by types could force users to verify these constraints throughout a proof, even in field where the types have nothing to do with the mathematical contents. The alternative would be to implement mechanisms to do the type checking automatically or to prove formally metatheorems that show that type checking is not necessary. Both are ambitious tasks.

Alternatively, one might demand that each accepted proof in QED state explicitly the basic theory and the calculus it employs. Then, the connection between the systems could be made by (1) proving the axioms of the basic theory of the first system in the second, and (2) proving that provability in the first calculus implies provability in the second calculus.

3.5 Meta-Theory

A research issue that raised considerable controversy was that of relating various theorem-proving systems.

The group was split on the importance of having a *common theory,* say, T, and translations from the object theories of the individual systems into T that could provably transform proofs of the individual systems into proofs in T. At first T was taken to be some variant of set theory (Mizar's brand). Subsequently, a different proposal emerged, in which the proof system of each system was formalized in primitive recursive arithmetic, or PRA (Boyer-Moore like). The meta-logic equivalent to PRA would be a predicative type theory, with a theory of syntactical objects at the base (theory of syntactical objects roughly equivalent to a theory of finite tree structures). It would, in effect, be a programming language (perhaps with polymorphism in the type system) as well as a logical theory; this strategy would facilitate the "bootstrapping" of results about proofs in PRA to procedures allowing one to skip many steps in later proofs in PRA (verification of derived inference rules by reflection). One could write a version of the proof checker in the algorithmic part of the theory.

The workshop participants also disagreed about the amount of "blow-up" that might occur in translating proofs between existing prover projects. Some participants felt that while an approach working mechanically through the meta-logic (or root logic, as it is misleadingly called in the manifesto) might be bad, a heuristic approach using known common features of particular pairs of theories might make proof translations feasible. Further research in this area is clearly warranted, particularly to distinguish between the theoretical problems of exponential growth and the practical problems of translating huge quantities of proofs.

3.6 Libraries and Tools

The QED project can certainly extrapolate from the experiences of the software community in meeting the needs of scientists. Two needs stand out:

- Libraries. The survival of Fortran is based on the existence of good, large libraries. The QED similarly must provide libraries that include standard proofs, theorems, and mathematical definitions ("the basic mathematical domain knowledge"). So too, must the QED project develop methods that will enable researchers to locate and reuse this information. Finally, the QED project should explore the feasibility of libraries of verified software.

- Tools. It can be very expensive to understand how to use an automated reasoning tool. This cost is a major inhibitor to potential users. It often appears easier to start over from scratch on building an automated reasoning system than to learn how to use someone else's. The QED project must develop the tools that will eliminate the need for the user to make himself an expert in the system.

4 Conclusions and Future Work

The most important conclusion drawn at the QED Workshop was that *QED is an idea worthy pursuing*, a statement with which virtually all the participants agreed.

To achieve this objective will require several important changes. First, common terminology is needed. The numerous projects represented at the workshop have been developed in relative isolation so far. In these isolated projects, people tended to name same things differently and to give the same name to different functions. Second, there is also a need to agree on the name for the area into which QED-like activities fall. This will be the starting point of a QED scientific community. Third, the experience collected by separate teams, no matter how impressive, is still too small to extrapolate into the fully fledged QED. More of such small-scale experience is required.

The QED Workshop also drew the following, related conclusions:

1. A variety of systems should be admissible within QED. It should not be a monolithic, single-language system.

2. We should reduce isolation between groups. To achieve this, we need people who have a working knowledge of two systems or more.

3. We should have a common framework such as PRA, but should take care to make the details of the intercommunication between theories as invisible as possible.

4. We should build bridges between the various provers and checkers, so that work done in one framework can be relied upon in another.

5. We should involve mathematicians as best we can, specifically supporting languages and interfaces that appeal to mathematicians (i.e., resemble ordinary mathematical discourse).

6. We should expand the QED project to be a standards body, providing, for example, a standard low-level language for theorems and proofs (PRA) for which a proof checker can easily be written. If the standards body does its job, and the existing automated reasoning projects "plug in," the growth of the database will take care of itself.

We also drew several suggestions about future research directions:

- Some specific exercises are needed. One would be implementing inductive definitions (as in Coq) in the framework of set theory (as in Mizar). For example, the Church-Rosser theorem in (untyped) lambda calculus has been proved in NQTHM and in various systems for constructive type theory. NQTHM has inductive data types and recursive functions on them, and the latter systems usually use inductive definitions for representing various syntactic objects.

- Bilateral projects should be considered for the startup phase. One idea is collaboration between HOL (probably closest to standard object language of the prover projects) and Mizar (closest to the QED core of any project). The Boyer-Moore theorem prover seems to be a prototype tool for working with the very low level of the core missing from Mizar; one might therefore consider a three-tier project. Another idea is to develop a translation from one prover project down through Mizar and up through Mizar to another prover project and vice versa; set up bilateral communication.

Acknowledgments

Many of the participants in the QED Workshop contributed to the writing of this report, either directly (through their own "mini-reports") or indirectly (through written comments which were then incorporated in the final draft).

References

1. Horgan, John, interview with André Weil, *Scientific American*, June 1994, p. 34.

2. Kolata, G. "Computers Still Can't Do Beautiful Mathematics," *New York Times*, Week in Review Section E, July 14, 1991.

3. The "QED Manifesto." Available by anonymous ftp from Internet site `info.mcs.anl.gov` in directory `pub/qed`.

4. Rudnicki, Piotr. "An Overview of the MIZAR Project," preprint, 1994.