Quick & Dirty refereeing of proofs?

Madhu Sudan*

June 30, 2003

On her first day on the job as an editor of a math journal, Alice opens her mailbox to be overwhelmed by ten submissions claiming to prove the Riemann Hypothesis, each a 100 pages long! Presumably, most are buggy proofs, but is there an easy way to find the errors? Conventional wisdom has it that this is an impossible task. However a surprising (even shocking?) result developed by theoretical computer scientists gives Alice a way out, at least in theory. They have shown how it is possible to specify a format for writing proofs that makes error-detection extremely easy. More accurately, they give a probabilistic procedure to verify proofs that only looks at three bits of a purported proof to verify it, and accepts correct proofs (provided they are in the new format) while rejecting any purported proof of a false theorem with probability at least half. If Alice doesn't like this high probability of accepting incorrect proofs, well, she just has to read three more bits and the probability of accepting incorrect proofs goes down to a 25%, and so on. The new format for writing proofs does make the proofs somewhat longer, but not unreasonably so (at least in a theoretical sense).

So what do theorems and proofs have to do with computer science? It turns out that both notions, and in particular, the distinction between them, are *computational*: Proofs are by definition *easy* to verify, while theorems are in general *hard* to prove, where *easiness* and *hardness* qualify the computational effort required to perform the corresponding tasks. The distinction between the two lies at the heart of the famed "P=NP?" question; and the computational perspective has been exploited for years in computer science, logic, and optimization. For example, it is possible to cast theorem-proving as an instance of the Travelling Salesman Problem (TSP) (cf. [1]). Given a purported theorem statement T and an estimate n on the length of the proof, one can write down the inter-city distance chart of a hypothetical universe with N cities (where N is only polynomially larger than n), and a bound B such that the following is true: There exists a tour of the N cities returning back to the starting point of length at most B, if and only if the theorem T has a proof of length at most n. Thus fundamental tasks of mathematics reduce to mere combinatorics in this interpretation. More usefully, the simple combinatorial task of finding optimal TSP tours is seen to be as hard as proving "general mathematical theorems.

The new probabilistically checkable proofs (PCPs) are based on a transformation that is similar in spirit, though much more involved technically. They rely on techniques and insights from a multitude of areas including the theory of error-correcting codes, algebra, Fourier analysis, extremal combinatorics. The simplest of these transformations uses algebraic ideas employed earlier in the theory of error-correcting codes, the area of mathematics developed to deal with the problem of errors that occur when one is communicating information over noisy media. In this transformation, one views every string (be it the theorem or the proof) as a multivariate polynomial. So a theorem statement T becomes a polynomial τ; and the search for a proof P becomes a search for a polynomial

*MIT CS & AI Laboratory, 200 Technology Square, NE43-307, Cambridge, MA 02139, USA. Supported in part by NSF award ITR 0312575.

π; and the consistency relation between T and P as required by the logical system is expressed by a operator R that maps two polynomials in a new one. The goal of proving theorems is thus rephrased in the following way: Given a polynomial τ, does there exist a polynomial π of a pre-specified degree such that $R(\tau, \phi)$ is the zero polynomial? (To see that such problems can get quite complex to solve, consider the simple operator R that maps the pair (τ, π) to the polynomial $\phi(x, y) = \tau(x, y) - \pi(x)\pi(y)\pi(x + y)$: Solving such a relation is already quite complex; and gets even more so as the complexity of R increases.)

The advantage with algebraic interpretations of logical requirements comes from the following well-known fact: If a polynomial is not identically zero, then it is almost always non-zero. At a high level, interpreting this theorem in the logical setting suggests that if we propose a proof π for the theorem τ, then the proof is either always valid; or rarely so (since $R(\tau.\pi)$ is either identically zero, or rarely so). Utilizing this theorem requires many technical conditions to enable $R(\tau, \pi)$ to be easily evaluated given access to τ, π in appropriate form; and methods to ensure that the format used to specify the polynomial π does not allow the prover to cheat by giving functions that are not of the specified degree. We wont get into all this, but the interested reader is directed to survey articles (cf. [2]) for descriptions of the earlier work and to [3, 4] for some of the more recent works.

So what impact does all this have on the life of the editor of the math. journal? Sadly, none so far. While the methods developed by the theoretical computer scientists are quite ingenious and correct they have had little influence on the way mathematicians write their theorems and proofs. Most centrally this is possibly because the role of a proof goes well beyond the task of merely proving the theorem. Mathematicians look to proofs for providing insight and intuition into the theorem and use it to develop their own theories. So it is unlikely that these modern formats for proofs will replace the traditional formats of theorems and proofs.

However, developments in the field of probabilistically checkable proofs have nevertheless had a resounding impact in computational mathematics. Most directly, they show why many combinatorial problems, such as the travelling salesperson problem (TSP), are hard to solve even near optimally. Recall that the classical theory of NP-completeness showed that finding optimal TSP tours are hard since they lead to automatic methods for proving theorems. A modern analogous result shows that finding near-optimal tours lead to nearly-correct proofs in the PCP format, which in turn suffices to determine if a theorem is true or not. Indirectly, the study of probabilistically checkable proofs has led to a resurgence of interest in the study of computational aspects of error-correcting codes; and to some dramatic improvements in the speed and fraction of errors that we are now able to cope with in adverse settings of errors [5, 6].

Finally, there is one place where one hopes that the fast probabilistic verification of theorems and proofs may actually be practical: a scenario where theorems and proofs are more automated, and one doesnt care much for any intuition given by the proof. This is in the area of verifying correctness of execution of computer programs. Often one would like to audit the execution and verify that it did indeed take place correctly. In such scenarios a computer executing a program can easily produce a proof of correct execution, but it would be inordinately long (as long as the number of steps the program took to execute). Transforming such proofs into their probabilistically checkable formats would provide an efficient way of auditing. Technical drawbacks, in particular the increase in the size of the probabilistically checkable proof relative to the size of the size of the traditional proof, have held back this application. But there has been significant progress on this front recently [7, 8] and this area continues to be a subject of active investigation.

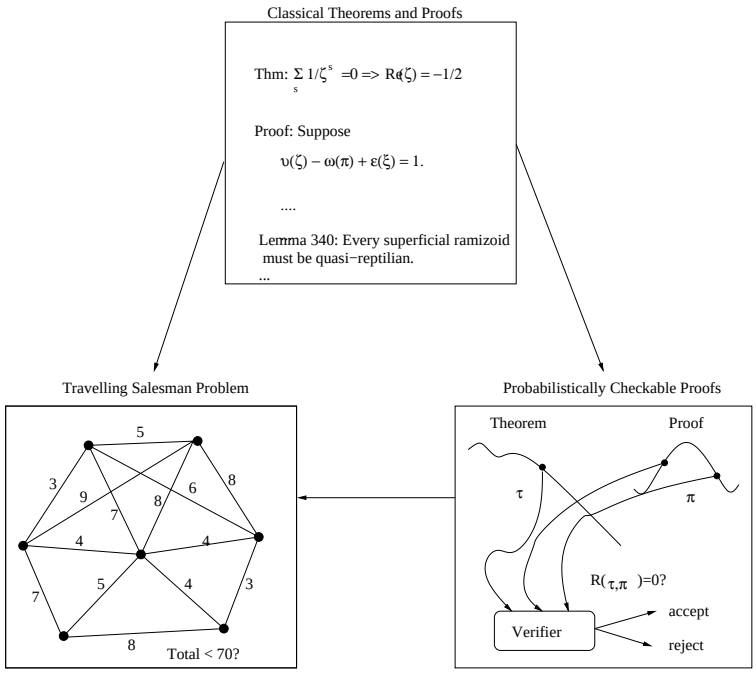

Figure 1: Traditional proofs can be transformed into a new format, leading to efficient probabilistic verification; and new implications for the travelling salesman problem.

References

[1] M. R. Garey and D. S. Johnson. *Computers and Intractability.* W. H. Freeman and Company, New York, 1979.

[2] M. Sudan. Notes on PCPs, 2000. Available from `http://theory.lcs.mit.edu/~madhu/pcp`.

[3] Johan Håstad. *J. ACM*, 2001, pp. 798-859.

[4] I. Dinur and S. Safra. *STOC* 2002, pp. 33-42.

[5] D. Spielman. *IEEE Trans. Inf. Th.*, 1996, pp. 1723–1732.

[6] V. Guruswami and M. Sudan. *IEEE Trans. Inf. Th.*, 1999, pp. 1757–1767.

[7] O. Goldreich and M. Sudan. *FOCS* 2002, pp. 13-22.

[8] E. Ben-Sasson et al. *STOC* 2003, pp. 612-621.