DOI:10.1145/1629175.1629176

Moshe Y. Vardi

More Debate, Please!

In the May 1979 issue of *Communications*, a powerfully written article by Richard A. De Millo, Richard J. Lipton, and Alan J. Perlis entitled “Social Processes and Proofs

of Theorems and Programs,” argued that formal verification of programs is “difficult to justify and manage.” The article created the perception, in the minds of many computer scientists, that formal verification is a futile area of computing research.

That article did not cite a 1977 paper by Amir Pnueli entitled “The Temporal Logic of Programs.” His paper had attracted little attention by 1979, but by 1997 it would be described as a “landmark paper” in the citation that accompanied Pnueli’s 1996 ACM A.M. Turing Award. In his paper, Pnueli, whose sudden and unexpected death on Nov. 2, 2009 shocked the computer science community, laid the foundation for formal verification of concurrent and reactive programs. (An article describing Pnueli’s scientific legacy appears on page 22.) The paper also laid the foundation for the development of model checking, an automated formal-verification technique for which Edmund A. Clarke, E. Allen Emerson, and Joseph Sifakis received the 2007 ACM Turing Award.

With hindsight of 30 years, it seems that De Millo, Lipton, and Perlis’ article has proven to be rather misguided. In fact, it is interesting to read it now and see how arguments that seemed so compelling in 1979 seem so off the mark today. Should we infer that *Communications* erred in publishing that article? My answer is a resounding “no!”

My basic education included exposure to Talmudic scholarship. Jewish scholars in the first half of the first millennium believed that truth will emerge

from vigorous debate. The Talmud, a monumental work of Jewish scholarship concluded circa 500 CE, is in essence a compendium of legal debates. Vigorous debate, I believe, exposes all sides of an issue—their strengths and weaknesses. It helps us to reach more knowledgeable conclusions. To quote Benjamin Franklin: “When Truth and Error have fair Play, the former is always an overmatch for the latter.” In my opinion, however, the editors of *Communications* in 1979 did err in publishing an article that can fairly be described as tendentious without publishing a counterpoint article in the same issue. Indeed, the article instigated so many reader responses, the editors published 10 pages of letters in the November 1979 Forum section of *Communications*, calling the work everything from “marvelous” to “humorous.”

In 2007, when I met with various focus groups to discuss the relaunching of *Communications*, I was encouraged to keep this publication engaged in controversial topics. “Let blood spill over the pages of *Communications*,” said one discussant jokingly. At the same time, however, participants believed that the magazine should represent all points of view fairly. This sentiment led to the establishment of the Point-Counterpoint feature, in which both sides of an issue are represented by opposing articles. Quoting Franklin again: “when Men differ in Opinion, both Sides ought equally to have the Advantage of being heard by the Publick.”

Since the relaunch in July 2008, we have published several Point-Counter-

point pairs: on computing curricula, e-voting, Net neutrality, and the direction of CS education in the U.S. At this point, however, the pipeline for such articles is dry. I had assumed that both members of the editorial board and readers would propose topics for Point-Counterpoint articles, but that does not seem to be the case. It is almost as if people believe there is something improper about engaging in direct debate. In fact, several authors whom I invited to participate in Point-Counterpoint debates have declined in order to avoid head-on confrontation. The truth is, however, that there are many issues in computing that inspire differing opinions. We would be better off highlighting the differences rather than pretending they do not exist.

In this issue of *Communications* we have a debate that is quite a rarity in computing research: a *technical* debate. MapReduce (MR) is a software framework to support distributed computing on large data sets on computer clusters. It was introduced by J. Dean and S. Ghemawat of Google in a highly influential 2004 article, and featured as a Research Highlight paper in the January 2008 issue of *Communications*. The success of MapReduce led some to claim that the extreme scalability of MR will “relegate relational database management systems (RDBMS) to the status of legacy technology.” A pair of Contributed Articles in this issue—Dean and Ghemawat on one side and Stonebraker et al. on the other—debate the relative merits of MR and RDBMS beginning on page 64. As parallel computation is one of the hottest topics in computing today, I have no doubt that our readers will find this technical debate highly instructive.

If you have topics that you think should be debated on the pages of *Communications*, please contact me. More debate, please!

Moshe Y. Vardi, EDITOR-IN-CHIEF