Moshe Y. Vardi

DOI:10.1145/3469113

Program Verification: Vision and Reality

IN 1969, TONY HOARE published a classical *Communications'* article, "An Axiomatic Basis for Computer Programming." Hoare's article culminated a sequence of works by Turing, McCarthy, Wirth, Floyd, and Manna, whose essence is an association of a proposition with each point in the program control flow, where the proposition is asserted to hold whenever that point is reached.

Hoare added two important elements to that approach. First, he described a formal logic, now called *Hoare Logic*, for reasoning about programs. Second, he offered a compelling vision for the program-verification project: "When the correctness of a program, its compiler, and the hardware of the computer have all been established with mathematical certainty, it will be possible to place great reliance on the results of the program, and predict their properties with a confidence limited only by the reliability of the electronics."

Hoare's vision came under a scathing attack a decade later in an influential 1979 *Communications'* article, "Social Processes and Proofs of Theorems and Programs," by De Millo, Lipton, and Perlis. They argued that mathematical proofs are accepted through a social process. Program-correctness proofs will not be subject to a similar social process, due to their length and narrowness, so they will not be socially accepted. They concluded that "this makes the formal verification process difficult to justify and manage." Hoare himself retracted, to some extent, his 1969 vision in 1995, writing "It has turned out that the world just does not suffer significantly from the kind of problems that our research was originally intended to solve."

In a parallel development, Amir Pnueli introduced the temporal logic of pro-

grams in 1977. Clarke and Emerson, and independently, Queille and Sifakis, then built on Pnueli's work and developed, in the early 1980s, *model checking*, an algorithmic technique for checking properties of finite-state programs. That led to Pnueli receiving the ACM A.M. Turing Award in 1996, and Clarke-Emerson-Sifakis receiving the award in 2007. By the mid-1990s, several model checkers had been built and adopted for industrial usage by semiconductor and design-automation companies. Industrial temporal logics, such as PSL and SVA, based on Pnueli's work, became industry standards in the early 2000s.

The success of model checking in the semiconductor industry, where post-production error correction is very difficult, points to an important insight that was missing in the early literature on program verification. Program verification is an expensive activity. Navigating the cost-benefit trade-off of program verification is ultimately a business decision. Model checking offered a different price point than full program verification: on one hand, model checking offers less—property checking and not full program verification, on the other hand, model checking costs less, due to higher level of automation.

This cost-benefit trade-off suggests that how much verification should be done is context dependent. An operation-system microkernel is probably a more appropriate target for a major verification effort than a dating app. Such an undertaking was initiated by an Australian team, who verified the seL4 microkernel using the Isabelle proof-assistant tool. Isabelle is based on a small logical core to increase the trustworthiness of proofs. This approach acknowledges that De Millo et al. were right—proofs do require a social process to be accepted—but in

Isabelle (and similar tools) this social process can be confined to the small logical core. In fact, with the help of proof assistants, formal verification today is even bringing a new standard for rigor in mathematics.

The emergence of cloud computing as the major context for much of today's computing shifts the cost-benefit trade-off of verification, due to its large scale. Because different users of the same cloud platform share hardware resources, security and privacy are of paramount interest. The Automated-Reasoning Group at Amazon Web Service (AWS) has been focusing on the development and use of formal-verification tools at AWS to increase the security assurance of its cloud infrastructure and to help customers secure themselves. At the same time, as the Spectre and Meltdown attacks have demonstrated, the large gap between the logical model (ISA) and the underlying microarchitecture of the X86 microprocessor not only provides side channels to attackers but also erects a major barrier to full verification.

In 1969, Hoare wrote about mathematical certainty, great reliance, and confidence. In retrospect, the hope for "mathematical certainty" was idealized, and not fully realistic, I believe. Verification can give us great reliance and confidence, but at a cost that must be justified by the benefits. The deployment of autonomous systems with machine learning-based components brings new urgency and excitement to this important research area.

Follow me on Facebook and Twitter.

Moshe Y. Vardi (vardi@cs.rice.edu) is University Professor and the Karen Ostrum George Distinguished Service Professor in Computational Engineering at Rice University, Houston, TX, USA. He is the former Editor-in-Chief of *Communications*.

Copyright held by author.