DOI:10.1145/3144590

Moshe Y. Vardi

Would Turing Have Won the Turing Award?

In 2017, we celebrated 50 years of the ACM A.M. Turing Award, known simply as the Turing Award. The list of Turing Award winners (http://amturing.acm.org), starting from Alan Perlis in 1966, "for his influence in the area of advanced computer programming techniques and compiler construction," to Sir Tim Berners-Lee in 2016, "for inventing the World Wide Web, the first Web browser, and the fundamental protocols and algorithms allowing the Web to scale," offers a bird-eye view of the highlights of computing science and technology over the past 50 years. Justifiably, the Turing Award is often accompanied by the tagline "The Nobel Prize in Computing." How did this prestigious award come to be?

The early history of the Turing Award is somewhat murky. The minutes of meetings of ACM Council from the mid-1960s shed some, but not complete light on this history. The Turing Award was not originally created as a "big prize," but rather a lecture given at the annual ACM meeting. In August 1965, ACM Council considered and tabled a proposal that "the National ACM Lecture be named the Allen [sic] M. Turing Lecture." In December 1965, ACM Council adopted the motion that "A.M. Turing be the name of the National Lectureship series." In a 1966 meeting, ACM Council voted to name Alan Perlis as first lecturer. The minutes shed no light on *why* the lectureship was named after Alan Turing. The historical record is also not clear on how a lectureship turned into a major award. Perhaps there is a lesson here for ACM to keep better minutes of its Council's meetings!

From today's perspective, however, we can wonder whether ACM Council was justified in 1966 in naming its National Lecture after Turing. Today, Turing is widely regarded as one of the most outstanding scientists of the 20th century, but that was not the case in 1966. The question, therefore, can be posed as follows: Had Turing been alive in 1966 (he died in 1954), would he have been selected for ACM's first National Lecture?

A debate about Turing's accomplishments has been going on for quite a while. In 1997, in an after-dinner speech in Cambridge, U.K., Maurice Wilkes, the 1967 Turing Award winner (for designing and building the EDSAC, the first stored-program computer in 1949), offered some biting comments about Turing: "However, on a technical level, of course I did not go along with his ideas about computer architecture, and I thought that the programming system that he introduced at Manchester University was bizarre in the extreme. ... Turing's work was of course a great contribution to the world of mathematics, but there is a question of exactly how it is related to the world of computing." (See Wilkes's complete comments at https://goo.gl/XkjM7n.)

The controversy about Turing's accomplishments flared again over the last few years. In a 2013 *Communications*' editorial (https://goo.gl/SpkhKw) I argued that "The claims that Turing invented the stored-program computer, which typically refers to the uniform handling of programs and data, are simply ahistorical." In response to this editorial, Copeland et al. argued in the 2017 *Turing Guide* (https://goo.gl/DjC8uk) that "Vardi is ignoring the fact that some inventions belong equally to the realm of mathematics and engineering. The Universal Turing Machine was one such, and this is part of its brilliance." So who is right?

When it comes to historical interpretation, the same facts may lead different people to different interpretations, but one should pay attention to the facts! In August 2017, Leo Corry published an article in *Communications* on "Turing's Pre-War Analog Computers: The Fatherhood of the Modern Computer Revisited" (https://goo.gl/M7jCaj) in which he carefully examined the purported connection between the "Universal Turing Machine," as introduced in Turing's 1936 paper and the design and implementation in the mid-1940s of the first stored-program computers. He concluded "There is no straightforward, let alone deterministic, historical path leading from Turing's 1936 ideas on the Universal Machine to the first stored-program electronic computers of the mid-1940s."

But the debate about how much credit Turing should get for the idea of the stored-program computer diminishes, in my opinion, from Turing's actual contributions. The Turing Machine model offered a robust definition of computability that has been studied, refined, and debated since 1936, giving rise in the 1960s to computational complexity theory, a gem of theoretical computer science. Turing's philosophical examination in 1950 of the possibility of machine intelligence is lucid and incisive today as it was then. Finally, we learned in the 1970s about Turing's critical contributions to computing-aided code breaking.

Would Turing have won the Turing Award? My answer is, he should have!

Follow me on Facebook, Google+, and Twitter.

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