Book Reviews

MARK LAMOURINE

The Logician and the Engineer: How George Boole and Claude Shannon Created the Information Age
Paul J. Nahin
Princeton University Press, 2013, 228 pages
ISBN 978-0-691-15100-7

When I started reading this book, I didn't know that I would finish it about the time of George Boole's 200th birthday, but it was a nice note. Claude Shannon worked during my lifetime and probably most of yours. The two of them don't get the attention that other luminaries of computing do, but their contributions rank with Charles Babbage, Ada Lovelace, Alan Turing, and John von Neumann. In some ways their work is more significant because of its cross-over application to physics as well as computation.

Nahin is a fan of both Boole and Shannon, and in this book he shows how Shannon's work built on Boole's to bring us the fundamental logical basis for modern computing hardware. He also wants to help the reader understand the formal results of their work. In that, he can only be as successful as the reader is dedicated. He is clear in the introduction that a certain level of mathematical background will be needed and that the reader will need to take care to follow along to get the most from the book.

In the first third of the book Nahin offers a brief biography of both Boole and Shannon. He's clearly not happy with at least Shannon's treatment by some popular modern writers. He takes a shot at James Gleick in the opening paragraph of the first chapter, quoting a somewhat disparaging comment about Shannon's sense of humor. Nahin also goes to some lengths later to highlight some of what would today be known as geeky humor. Neither biography is particularly deep or insightful, but they do give a sense of the time and influences on the men.

The middle section considers Boole's contribution to computing, the algebra of two-value logic. All of this should be familiar to anyone who's studied programming in any formal way. For a non-programmer, the discussion of De Morgan's Theorem and Karnaugh maps will give some sense of how to combine Boolean operators. Sometimes I think some programmers should remember how to reduce logical operators.

The last and largest section shows how Shannon picked up where Boole left off. Electrical relays didn't exist in Boole's time, and transistors were new during Shannon's career. Nahin explains how Shannon discovered the way in which Boole's logic could be expressed in terms of relays. It was adapted naturally to electronic circuits. This is something I did learn in college as part of a computer science course. I'm not sure whether this is commonly taught as a core course anymore, but if not, this would be a great section for a curious coder or admin to read. But, while Shannon's own work was in computer engineering, the implications didn't end there.

Shannon also analyzed the theory of signaling, describing what it meant to "send a message" in the most fundamental terms. His goal was to understand the limits of logical expression in his circuits, where mechanical relays often fail. In the process he created the field of information theory, which brings together mathematics, physics, and computation. In combination with Boole's binary logic, he produced a way of understanding the logic of quantum mechanics.

Here I agree with Nahin, that Shannon's work is underappreciated. While he didn't set out to found a new mathematical field, his straightforward inquiry has had an outsized influence, in both theory and practice, on numerous fields. Nahin's book on the contributions of Shannon and Boole is both timely and overdue.