BULLETIN (New Series) OF THE
AMERICAN MATHEMATICAL SOCIETY
Volume 2, Number 1, January 1980

0002-9904/80/0000-0009/$01.75

BOOK REVIEWS

An introduction to the general theory of algorithms, by Michael Machtey and Paul Young, North-Holland, New York, Oxford, Shannon, 1978, vii + 264 pp.

In order to discuss this book properly, a brief recounting of some of the history of the theory of algorithms is in order.

Although the notion of mathematical algorithm had been used for centuries, the present rigorous definition was not accomplished until Gödel, Church, Turing and Kleene gave their formulations in the 1930's. That all these formulations, and several later ones, proved to be equivalent is a kind of event that is indeed rare in mathematics. The assertion that these formulations coincide with the intuitive notion of mathematical algorithm is known as Church's Thesis. In the 1930's Gödel also demonstrated the existence of algorithmically unsolvable problems, particularly the unsolvability of the decision problem for arithmetic (which dashed the hopes of Hilbert's Program, and at the same time was the first application of Cantor's diagonalization technique in this area).

In the 1940's the construction of computers that could implement algorithms had a profound effect on the development of the theory. Von Neumann explored the program-data dichotomy (one man's program is another man's data) and produced the notion of a stored program computer and the manipulation of one program by another. Finally, in the 1960's, after a decade of programming experience, the realization of the importance of an algorithm's complexity lead to the replacement of the notion of algorithmic unsolvability by algorithmic intractability. That is: while certain problems may be theoretically algorithmically solvable, their intrinsic computational complexity prevents any algorithm from running to completion.

As in most texts in this area, the invariance of the notion of algorithmic computability is demonstrated by introducing several models of computation and showing them to be equivalent. Unlike most texts, however, the present one accomplishes this in the first fifty pages, within the first chapter. While many details of this equivalence are omitted and left to the reader to supply, anyone with formal mathematical experience or reasonable computer programming experience can readily follow the outline of the simulations, even if he cannot completely fill the gaps. The other significant feature of the first chapter is that measures of computational complexity for the various models considered are introduced there as well.

Not only are the usual models of computation equivalent, they are effectively intertranslatable. The properties common to the standard models that are responsible for such intertranslatability can be abstracted axiomatically through what are called acceptable programming systems (or acceptable Gödel numberings). An acceptable (universal) programming system (§3.1) is one in which there exists (1) a universal program, which, given any program and any input, computes the output of that program on that input, and (2) an