Part II, which makes up about 60 percent of this short volume, consists of three chapters, each of which describes a different autocode. The word "autocode" as used in England corresponds roughly to "compiler language" in the United States. The authors apparently consider an autocode to be machine-dependent, in contrast to a "universal computing language" like Algol, which is machine independent. The three autocodes discussed are for the Pegasus-Sirius, the Elliott 803, and the Ferranti Mercury. The machines themselves are not described here in any detail. They are all rather small machines, and are not of very great general interest. Unfortunately, the same is true of their autocodes. The Mercury autocode is treated at greatest length and in greatest detail. It is an interesting system, but its interest is now mostly historical, illustrating some of the early work of Brooker and his colleagues at Manchester. Most of the material in this book will be of interest only to the devoted specialist and perhaps to the historian in the field of computer languages.

A final section of the book presents a 14-page discussion of Algol. It is a good but very brief resumé of the language.

Saul Rosen

Purdue University
Lafayette, Indiana

28[Z].—Irving Allen Dodes, *IBM 1620 Programming for Science and Mathematics*, Haden Publishing Company, Inc., New York, 1963, ix + 276 p., 24 cm. Price $9.95.

This is a very thorough and carefully written text on programming the IBM 1620. The author is chairman of the mathematics department of the Bronx High School of Science, where a course in numerical analysis has been given successfully to seniors. The course includes learning to program the 1620. This text is an outgrowth of a manual used in that course. The general style is obviously influenced by the high-school audience for which it was first intended, but this should not be construed to mean that the book is limited to such an audience. Rather, one could recommend it as a text for any audience unfamiliar with the programming of modern computers and wishing to learn something about this by using the 1620 as a specific machine.

There are four parts. Part 1 is a somewhat elementary treatment of number systems and numerical methods. Part 2 is an extensive discussion of 1620 machine programming. Part 3 describes the symbolic programming system (SPS), and part 4 treats Fortran with Format.

There are numerous exercises and illustrative examples. The material is well-organized and presented with great care.

E. K. Blum

Wesleyan University
Middletown, Connecticut

29[Z].—Seymour Ginsburg, *An Introduction to Mathematical Machine Theory*, Addison-Wesley Publishing Co., Inc., Reading, Mass., 1962, ix + 148 p., 23 cm. Price $8.75.

It is surprising that a book a couple of years old in the rapidly developing area of theory of machines (or theory of automata) is still up to date. The author has accomplished this by undertaking a modest task. He has not attempted a comprehensive development of the subject, but has rather restricted himsel for the most part to a few of those areas in which he and his colleagues have actively worked. In this respect, the title may be misleading.

The first chapter deals with *complete sequential machines* from a point of view initiated by E. F. Moore in "Gedanken Experiments on Sequential Machines." This point of view is, roughly speaking, what can one learn about the contents of a "black box" capable of assuming a finite number of states by performing "experiments" (not to be confused with the technical sense of "experiment" used by the author) which utilize only the box terminals. A typical result of this character is the following. Two complete sequential machines S_i, $i = 1, 2$, each with a fixed initial state will for each input sequence give the same output sequence provided they do so for all input sequences of length $n_1 + n_2 - 1$, where n_i is the number of states of machine S_i; moreover, this number cannot in general be reduced (Theorem 1.2). An interesting problem of this kind for which a definitive solution has been found is given in Theorem 1.5. Given a machine, how long an input sequence does it take so that the corresponding output sequences uniquely determine the final state of the machine started from an unknown initial state? Find a procedure for determining such an input sequence (while observing the corresponding output sequence).

Chapter 4 deals with "recognition devices." The point of view is that initiated by S. C. Kleene (inspired in part by work of McCulloch and Pitts) in "Representation of Events in Nerve Nets and Finite Automata." Here the emphasis is on the structure of the class of "tasks" that finite automata perform rather than on the structure of the finite automata themselves. The "task" is to recognize a set of strings of symbols (called tapes). The sets recognizable by finite automata are shown to be equivalent to the regular (in the sense of Kleene) sets and other basic properties of this class of sets are developed. The two-way and two-tape automata introduced by Rabin and Scott are also discussed here.

Chapter 3 is entitled "Abstract Machines." It incorporates work which is due almost solely to the author. It attempts to answer the question "To what extent can the problems treated in Chapter 1 be given meaning in the context of an arbitrary input and output semi-group (rather than the free ones) and to what extent do their solutions generalize?"

Chapter 2 is entitled "Incomplete Sequential Machines." Its origin is more pragmatic and its development somewhat cumbersome. An incomplete sequential machine is construed as a vehicle for specifying input-output conditions which a given machine may or may not *satisfy*. The chief problem treated is that of finding a minimum state machine which satisfies a given incompletely specified machine.

The treatment is rigorous but unmotivated, concise and sometimes terse. The most serious defect in the development is the lack of uniformity and perspective. The chapters are so loosely knit that the book comes close to being a collection of four papers.

Calvin C. Elgot

Thomas J. Watson Research Center
International Business Machines Corporation
Yorktown Heights, New York