factors. My personal attitude is that the second approach is going to appear progressively more attractive as:

(1) the capability of machines increases with respect to cost (that is, the capability per dollar increases);
(2) the desirability of minimizing overall time from problem statement to executed solution becomes increasingly important;
(3) improved techniques of translating programs into machine code are developed.

The development of larger and cheaper memories seems virtually assured. We can expect as a consequence that the storage capacities of machines will be progressively less limiting for a given cost.

In order to decrease preparation time of a program one can put more programmers on the job, but a law of diminishing returns prevents program preparation time from being made arbitrarily short. If we wish to decrease the preparation time still further, some of the programmer's work must be shifted to a facility which can carry it out more speedily than the programmer, namely, a "translating program" (this is a poor name for it) on a machine.

The reason we currently rely on human beings to produce efficiently coded programs is that our techniques used in translating programs produce results having noticeably poorer efficiency. There is no reason to believe, however, that the capability of translating programs will not be continuously improved to the point that translating programs rival in operating efficiency the present products of programmers.

I do not pretend to have analysed here all of the elements of programming nor treated all of the things which make programming hard. I have presented one major facet of programming complication and suggested an approach to alleviate it.

Grateful acknowledgement is made to Thomas L. Connors of The MITRE Corporation whose searching questions and comments led to the formulation of the ideas expressed here.

References

ARDEN, B., GALLER, B., and GRAHAM, R. (1960). *The Michigan Algorithm Decoder.* Ann Arbor, Michigan: Univ. of Michigan.
CONFERENCE ON DATA SYSTEMS LANGUAGES (1960). *Initial Specifications for a Common Business-Oriented Language (COBOL).* Washington, D.C.: U.S. Dept. of Defense.
McCARTHY, J., *et al.* (1960). *LISP I Programmer's Manual.* Cambridge, Mass.: M.I.T. Computation Center and Research Lab. of Electronics.
NAUR, P., *et al.* (1960). "Report on the Algorithmic Language ALGOL 60," *Communications of the A.C.M.,* Vol. 3, p. 299.
WOODGER, M. (1960). "An Introduction to ALGOL 60," *The Computer Journal,* Vol. 3, p. 67.
WOODWARD, P. M., and JENKINS, D. P. (1961). "Atoms and Lists," *The Computer Journal,* Vol. 4, p. 47.

Book Review

Digital Computer and Control Engineering, by R. S. LEDLEY, 1960; 835 pages. (London: *McGraw-Hill Publishing Company Ltd.,* 112s. 6d.)

This book forms a comprehensive introduction to digital system engineering, and many of its 23 chapters are authoritative and well written. The book is divided into five main sections under the following titles:

1. Introduction to Digital Programmed Systems.
2. Functional Approach to Systems Design.
3. Foundations for the Logical Design of Digital Circuitry.
4. Logical Design of Digital Circuitry.
5. Electronic Design of Digital Circuits.

The general tone of the book is well suited to the needs of the advanced student of engineering. The first section introduces the main topics of computer engineering, and prepares the reader for subsequent sections of the book. It contains in addition two chapters devoted to programming, the first presenting elementary concepts, the second covering more advanced topics, ending with a brief description of automatic-programming techniques including ALGOL. Most of this introductory material is presented remarkably clearly. However, floating-point representation is dismissed in one and a half pages leaving, I fancy, the student unaware of its importance, and complementary representations of negative numbers are not mentioned at all!

Section 2 is a brief introduction to systems design which concludes with the introduction of PEDAGAC, a simple general-purpose computer whose design is used throughout the remainder of the book to provide the "thread of continuity" between the various topics. Sections 3, 4, and 5 deal with the central theme of the book, logical and electronic design.

I thought the author seemed more at home with logical design techniques than with electronic design. In particular, I enjoyed his treatment of logical design under constraints, in Chapter 12. A good deal of this is original work and is presented as a complete treatise for the first time. Earlier chapters lead naturally from first principles of Boolean algebra to the manipulation of Boolean matrices and the design of arithmetic and control circuits.

The section on electronic design, particularly Chapter 20 which deals with semiconductor circuits, is less successful. I found three incorrect and several confusing descriptions of circuit behaviour. Nevertheless, a great deal of ground is covered, including comparatively recent developments such as tunnel diodes, microwave logic, and cryotrons.

On the whole, the book must rank as one of the most successful introductions to computer and control engineering.

N. E. WISEMAN.