coding begins, a stage that beginning programmers too often bypass in their rush to get something "on the machine".

The major weakness of the text (a weakness that unfortunately overshadows the strengths mentioned above) is the extremely narrow and rather dated view of computer programming. In this respect, the text might well have been written ten years ago. Subprograms are mentioned only briefly toward the end of the book, and almost no hint of the central role of subprograms in programming is given. The sort of problem analysis and program design suggested to precede coding is entirely concerned with questions of *run-time efficiency*—no emphasis is given to good program structure, readability, documentation (in spite of statements to the contrary in the text), or modifiability. The text is sprinkled with suggestions for writing programs, but most are of debatable value and in conflict with current ideas, e.g., "use the faster forms whenever feasible: ... $C * C * C * C$ is faster than $C ** 4$ which in turn is much faster than $C ** 4.0$" (p. 422), "make a simple case run first, then make it fancy" (p. 184). The beginning programmer studying this book is far too likely to get the impression that "efficiency" is not just the primary, but almost the only, criterion by which program design is to be judged.

In addition, *assembly language* is presented as the way to get more power and versatility, should FORTRAN prove too restricted; there is only the merest mention of other high-level languages. The book has a number of other oddities: A six-page section on "The Computer in our Society" included in a chapter on "Subscripts and DO Loops", 45 pages devoted to the details of assembly language programming on the IBM 1130 (a small pre-IBM 360 machine), and the mixing of FORTRAN language details, job control language statements, particular hardware restrictions (six-digit accuracy is taken as standard), and other trivia.

In sum, an instructor might glean some interesting problems and examples from this text, but it is not recommended as a primary text for student use. Basically, the contents reflect practices, languages and computer systems of 1964 rather than 1974. A beginning student would be better served by a text incorporating more recent practice in programming.

T. W. PRATT

Computer Science Department
University of Texas at Austin
Austin, Texas 78712

36 [12].—FREDERICK W. WEINGARTEN, *Translation of Computer Languages,* Holden-Day, Inc., San Francisco, Calif., 1973, xi + 180 pp., 24 cm. Price $9.95.

The title of this book might lead one to infer that it treats compiler construction rather generally. The title is deceptive, however, in that the book is devoted almost exclusively to an expository presentation of different parsing methods and to the essential aspects of the theory underlying them. Such matters as run-time organization are not even mentioned, and code generation is given only the most cursory treatment. Thus, the book is not really a satisfactory text for a compiler course, although that would seem to be its intent.

After some preliminary introduction to relevant mathematical notation, the book discusses some of the early methods used for translating arithmetic expressions. It then goes on to a more general discussion of general formal grammars, context-free grammars, and the structure of translation trees. The author has chosen to represent such trees in a purely binary form, claiming that this approach makes parsing easier. However, I found that, as a result, his algorithms were complicated and not intuitively convincing. He then discusses different parsing methods, treating, in sequence, the top-down parse, the bottom-up parse, the general left-to-right parse (using Earley's nodal span method), and parsing based on restricted grammars, specifically, $LR(k)$ grammars, bounded-context grammars, and precedence grammars. Unfortunately, the interesting special case of operator precedence is not treated.

The author has a clear and engaging expository style; but he is unfortunately fighting an uphill battle against a poor choice of algorithms and data representations. For instance, the discussion of the top-down parse occupies two chapters and requires about five pages of flow-charts; with proper choice of representation and algorithm, the top-down parse becomes exceedingly simple—in fact, the simplest of all methods. Furthermore, there are numerous minor errors which make the discussion hard to follow. After several hours of attempting to understand the discussion of the nodal span parse, I realized that my difficulties were due to a number of different misplaced or missing arrows in one diagram (figure 8.6)! Among other errors that I noticed were the use of $\bar{y}$ rather than $\bar{x}$ on line 10, page 47, the use of $\underline{m}$ rather than $\underline{i}$ on the last line of page 114, and the omission of $\bar{z}$ following ξ in the statement of the first part of the closure rule on the same page.

The advantages of this book are its excellent organization and fine expository style. I found these outweighed, however, by the disadvantages of cumbersome algorithms and confusing errors. The restriction of the subject matter to parsing is a mistake, from my point of view; but that is a matter of taste.

PAUL ABRAHAMS

Courant Institute of Mathematical Sciences
New York University
251 Mercer Street
New York, New York 10012

37 [13, 25].—H. MELVIN LIEBERSTEIN, *Mathematical Physiology, Blood Flow and Electrically Active Cells,* American Elsevier Publishing Co., Inc., 1973, New York, xiv + 377 pp., 24 cm. Price $19.50

This book is a collection of the specific contributions of its author to the growing field of mathematical physiology, and the reader would be well advised not to form an overall impression of this field on the basis of this book.

The section on blood flow is based on a power series expansion for the velocity profile in pulsatile blood flow. The first term of this series is the parabolic profile of steady flow, and the subsequent terms contain successively higher time derivatives of the driving force. One would expect this series to converge rapidly only in the smaller arteries where the flow is, in fact, quasi-steady.

The section on electrophysiology is based on a modification of the Hodgkin-Huxley equations, which is referred to by the author as a "reformulation." The equations for nerve conduction, as stated by Hodgkin and Huxley [1], have the form:

$$ri = -v_x, \tag{1}$$

$$cv_t + I = -i_x, \tag{2}$$

where $I = I(v, s_1 \cdots s_N)$ is the ionic current through the membrane, and where the membrane parameters s_k obey ordinary differential equations of the form