programming course for non-majors would find it difficult. The explanation of BASIC, in particular, has very little redundancy in it, and would not be suitable for self-study. At the same time, the range of topics covered, the lack of depth of the programming section, and the choice of BASIC as a language would make this book unsuitable for computer science majors.

I noticed that one of the sample programs includes the use of a computed GO TO statement, though this statement is never described in the book. This creates the suspicion that there are other such difficulties. There is a chapter at the end on machine language, compilers, and similar topics. This chapter seems rather badly written and hard to follow; the explanation of compilers and operating systems is so abbreviated as to be virtually worthless.

This book would be appropriate for a class of interested students and an instructor inclined to provide a great deal of extra explanation. For a less interested class (for instance, a required course) or for an instructor who wanted to use a text more intensively, I would not recommend the book.

PAUL ABRAHAMS

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

14 [12].—RALPH E. GRISWOLD, *The Macro Implementation of SNOBOL4*, W. H. Freeman and Co., San Francisco, Calif., 1972, xii + 310 pp., 26 cm. Price $14.95.

A case study of machine-independent software development.

SNOBOL is a computer programming language designed for the manipulation of symbolic, as contrasted to numeric, data, and character strings in particular. It is by far the best and most popular language of its type, and has been implemented on many different machines. SNOBOL originated in 1963, and SNOBOL4 is the present version of it. For a long time there has been interest in the approach used in implementing SNOBOL4, and this book explicates that approach.

As the author points out, there is a scarcity of books describing implementations of programming languages in depth. This book is a fine example of how such a description ought to be written down. It is a hazardous subject, for a writer can easily be bogged down in the details without elucidating the larger themes; or, conversely, a writer may avoid the details only to have the reader forever asking, "How did he *really* do it?" Implementations have a way of becoming less and less coherent as they become encrusted with extensions, and as the implementors seek to compensate for early faults of judgment. Thus the object being described is far more complex and less rational than one would like it to be.

This book is both clear and well-organized, and contains a variety of material. The coherence of the presentation is remarkable; the logical development of concepts and techniques is worthy of a mathematics text. The book is divided into four parts: the SNOBOL4 language, the organization of the system, the language in which the implementation was done, and an overview of the results. The most interesting and challenging section is the one on the system organization; the section on the implementatiốn language is more detailed and of less general interest, though I would not quarrel with the decision to include this material.

The opening section contains some background material about SNOBOL4, a condensed exposition of the language itself, and a brief discussion of those language features that influence the implementation. The exposition of the language manages to be concise without being obscure; someone not familiar with SNOBOL4 but with a good background in programming should be able to learn SNOBOL4 from this exposition without much trouble.

The section on system organization opens with a brief overview and then proceeds to a discussion of the elementary data structures used throughout the system. A discussion of the interpretive mechanism follows, and an understanding of these two issues, data structure and interpretation, is the foundation on which the rest of the book is based. The remainder of the section contains no major surprises, but does cover rather thoroughly the various features of the language and the way they are realized. The discussion of pattern matching is particularly interesting, as the SNOBOL4 pattern is the single most novel and significant concept in the language. For someone with no experience in compiler construction, this section provides not only a presentation of the solution to various problems, but an understanding of how language leads to an implementation design.

The third section discusses the implementation language, which is actually a collection of (more or less) machine-independent macros. The system is written in terms of these macros, and the macros are then defined for each machine on which SNOBOL4 is implemented. There is a discussion of the implementation on two machines: the IBM 360 and the CDC 6000 series, with some comparisons between them. Sample macro definitions are given in machine language. This section probably will not be read closely by anyone other than a prospective SNOBOL4 implementor, though a casual reading is worthwhile even for the nonspecialist.

The concluding section is entitled "Retrospect and Prospect", and includes both a history of the project and an appraisal of the present status of the language. The problems that have been encountered with the implementation are discussed frankly, though the reader may not be made aware of the seriousness of some of these problems. The inefficiencies of SNOBOL4 are severe enough so as to have discouraged many users, and indeed, there is a case reported in the literature where a compiler bootstrap, done with SNOBOL4, had to be abandoned because of the severity of these problems. My personal experience also verifies this difficulty. The section ends with a discussion of the SPITBOL compiler and some thoughts about a SNOBOL machine.

The book is generally free of errors and has been well laid out. A distinct typeface, without proportional spacing, is used for all fragments of computer programs, a practice that ought to be followed more widely than it is. There are interesting exercises scattered throughout the book.

I heartily recommend this book both to the experienced language implementor, for whom it will be a valuable reference work, and to the student of compilers and programming languages, for whom it will provide a clear and palatable introduction to the area.

PAUL ABRAHAMS