

24, 30, 40, 60, 120, ∞ , $n_1 \leq n_2$. An ALGOL code (in English) is given for the calculation of f , $\min \alpha$ and $\max \alpha$, as well as for the $\alpha/2$ -percentiles of the Student t -distribution.

STANLEY SAWYER

Brown University
Department of Mathematics
Providence, R. I. 02912

15[9].—JOSEPH B. MUSKAT & ALBERT L. WHITEMAN, *The Cyclotomic Numbers of Order Twenty*, University of Pittsburgh, Pittsburgh, Pennsylvania, and University of Southern California, Los Angeles, California, 40 computer sheets deposited in the UMT file.

This table presents formulas for the cyclotomic numbers of order 20. The derivation and computation of these formulas are described in [1].

The 400 cyclotomic numbers (h, k) , $0 \leq h, k \leq 19$, can be grouped into 77 sets. There is a formula for each set, a linear combination of the prime p , a constant, and sixteen variables associated with Jacobi sums. The formulas depend, however, on $\text{ind } 2 \pmod{10}$ et al., so that there are forty different cases. All forty cases are given, one per sheet. Considerably fewer are necessary, for some cases can be derived from others merely by changing the primitive root used in generating the cyclotomic numbers.

AUTHORS' SUMMARY

1. JOSEPH B. MUSKAT & ALBERT L. WHITEMAN, "The cyclotomic numbers of order twenty," *Acta Arithmetica*, v. 17, no. 2, (to appear).

16[12].—R. E. GRISWOLD, J. F. POAGE & I. P. POLONSKY, *The SNOBOL 4 Programming Language*, Prentice-Hall, Inc., Englewood Cliffs, N. J., x + 221 pp., 28 cm. Price \$6.50 (paperbound).

SNOBOL 4 is a general-purpose string manipulation language and includes many novel features. Wider use has been hampered by the low availability of information about SNOBOL 4, except for photocopied journal extracts. This book clearly and cleanly fills this gap. It includes descriptions and examples of all currently implemented facilities. Many common problems of SNOBOL 4 users are resolved. Also included are seven complete working programs, although none seem to be real solutions of real problems. The book is aimed at advanced students and those with some programming experience and problems which may be solved by SNOBOL 4, and should hit this target well. It should be read by all with any possible interest in SNOBOL 4.

HENRY MULLISH
PAUL STEIER

Courant Institute of Mathematical Sciences
New York University
New York, New York 10012

17[12].—JOHN A. N. LEE, *The Anatomy of a Compiler*, Reinhold Publishing Corp., New York, 1967, xi + 275 pp., 24 cm. Price \$13.75.

This discursive book, far more readable than anything previously available in its subject field, surveys compiler writing, touching in an introductory way on many principal compiler issues. It is quite suitable for classroom use in an introductory course, and also as a guide for the experienced programmer wishing to learn something of the inner workings of compilers. Its point of view is principally shaped by experience with FORTRAN compilers.

The layout of topics in this book is as follows. A first chapter discusses, in general terms, such basic terms as symbolic language, interpreter, compiler, bootstrapping, syntax-oriented translation. Chapter 2 introduces BNF as a mechanism for the definition of languages, and describes various possible additions to the basic BNF apparatus, that is, additions potentially useful in shortening syntactic descriptions. Chapter 3 is a broad introduction to the parsing problem, outlining top-down, bottom-up, and catch-as-catch-can approaches to compiling. After three additional chapters devoted to semantic issues, this discussion of parsing is continued in Chapter 7, which describes precedence parsing in its application to algebraic expressions and the use of precedence methods for translation from ordinary algebraic infix notation to Polish strings.

The remaining chapters of the book are concerned with the semantic portions of compilers, i.e., with the symbol table manipulating and code generating routines which compilers contain. Chapter 4 describes symbol tables in general, outlines the hash schemes by which they may be addressed, and surveys the lexical scan processes used to enter items into such tables. The same chapter goes on to discuss some of the basic object-code issues arising in the assignment of addresses to symbol table items: layout of arrays, analysis of equivalence declarations, treatment of COMMON blocks. Chapter 5 describes target code styles for the treatment of control statements, emphasising techniques available for use in "single-pass" compilers. Chapter 6 discusses some of the special issues arising in connection with FORMAT-controlled I/O statements, describing the structure of a FORMAT interpreter, and the way in which links between a program, its I/O subroutines, and an operating system may be constructed. Chapter 9 gives a general discussion of target code questions connected with subroutine linkages, indicating the manner in which these linkages may be compiled, describing the treatment of arrays when they occur as subroutine parameters, and discussing the special issues which arise when subroutine names are to be transmitted as parameters. Chapter 8 describes the code generation process in additional detail, indicating the manner in which straightforward code may be generated either from pre-compiled Polish strings or directly from algebraic formulae during a precedence parse and the manner in which simple local optimization may be included in this process. The same chapter also details code generation both for the addressing of indexed variables and for the invocation of separately compiled functions.

J. T. SCHWARTZ

Courant Institute of Mathematical Sciences
New York University
New York, New York 10012

18[12, 13.35].—F. GENUYS, Editor, *Programming Languages*, Academic Press, New York, 1968, x + 295 pp., 24 cm. Price \$15.00.