BOOK REVIEW


This book is undoubtedly one of the best introductions available today to the machine-independent programming language ALGOL 60, and is addressed to all future scientists with access to computers who are willing to understand it. ALGOL is here championed among programming languages as the most natural, the nearest to the usual forms of mathematical expression, the best structured, and the least encumbered by arbitrary constraints.

The book arose from a series of lectures and is notable for its thoroughness of explanation, both of elementary fundamentals and of the most advanced concepts, all being amply illustrated by examples. After an introductory chapter on the notion of an algorithm and of algorithmic language the reader is offered a choice of a detailed introduction in Part I, or a graded series of 36 examples in Part II ("ALGOL par l'usage") which is intended to be largely self explanatory by virtue of embedded comments in the text. Cross-reference between the two is recommended. Finally there is a résumé of the rules of syntax which contains an account of J. W. Backus' metalanguage.

The introduction takes as a first example the ordinary process for the greatest common divisor of two integers (Euclid's algorithm), expressed in ordinary mathematical terms, and shows how naturally this may be transcribed into ALGOL. By this means a small repertoire of ALGOL terms is grasped, on which to build. Each new idea is skillfully approached through examples which in this way throw up quite naturally the need for new means of expression in the language, and these are then shown to be available. While a lucidly informal style is maintained, there is no sacrifice of rigor, and happily few misprints. The concepts of syntax, semantics, occurrence and use of identifiers, bound and free variables, local and nonlocal variables, syntactic equivalence of two algorithms, and syntactic level of a declaration are all admirably explained. French words are used in place of the ALGOL basic symbols with English mnemonic significance, which is well justified in view of the ease of learning and transcription of the dozen or so commonly used symbols.

Part I starts by developing in three chapters a sound understanding of ALGOL without the facilities of general Boolean expressions, switches, functions, or procedures. These are then treated in the following two chapters as a "second niveau syntaxique." Integration of a function by the trapezoidal rule serves well to introduce switches, procedures, and procedure identifiers as parameters.

A healthily pragmatic attitude is taken towards the defining Report on ALGOL 60. Rather than reproduce this the authors have deliberately tried to express its basic principles independently, and to accommodate accurately the net effect of the original formulation in carefully presented direct definitions. In this they are
most successful, and will disappoint only those who happen to have decided a
doubtful detail of interpretation in a different sense.

They choose to eliminate side effects of function designators by forbidding the
use, within the bodies of the corresponding procedure declarations, of variables
which are nonlocal and of assignments to formal parameters called by name. They
evidently intend also to exclude calls of nonlocal procedures, which could produce
such effects through assignments to variables nonlocal to their bodies, but this
seems to have been overlooked (p. 142, section 7.4). Since the main use of function
designators is as constituents of expressions where side effects are unnatural, and
ordinary procedures are available for the general case, this is a reasonable de-
cision. Function designators may not stand alone as procedure statements. The
authors permit a parameter specified as a "function" procedure to be called by
value, in which case its occurrences in the procedure body and its corresponding
actual parameters must be parameterless function designators.

As in the ALGOL Report, the treatment of input and output is separate from
the rest of the language. Simple "pseudo instructions" are used to assign a single
value from input and to transfer one to an output buffer. The output buffer is
assumed to unload onto an external medium when full and also on receipt of a
third kind of pseudo instruction, effectively "newline." Format is not discussed,
nor is precision of arithmetic.

The high standard of detail prompts the reviewer to mention a few minor
blemishes. Footnote (53), p. 68, is mistaken because a negative integer exponent
must give a real result. The array bounds are missing from the declaration for
tabv in the examples on pp. 138 and 139. On p. 145, the variable "z" in the body
of the declaration for \( f \) should be "u," and the value of \( x \) after the second block
should be 17. The expressions "(1)" on pp. 146–148 should remain as "n." The
parameter \( F \) in Algorithms 31 and 36 should be specified \( \text{rèel procédure} \), and the
description of interpretation of nonintegral array bounds in Algorithm 12 omits
the rounding off operation.

The authors are preparing a treatise on the theory and practise of compilation,
principally of ALGOL, intended for specialists in programming. This will show
how a language may be constructed by adjunction, to a minimal nucleus, of suc-
cessive means of expression which are in their nature not reducible to it. It should
be worth waiting for.

M. Woodger
Ministry of Technology,
National Physical Laboratory,
Teddington, Middlesex.

**Failure-Tolerant Computer Design.** By William H. Pierce. Academic

This book presents, in a clear and readable style, some results about increasing
the reliability of digital circuits by adding redundancy. The author begins by
proving theorems on asymptotic relations between the reliability and the amount