

- HOWARD, G. M. (1967). Degrees of Freedom for Unsteady-State Distillation Processes, *Industrial and Engineering Chemistry Fundamentals*, Vol. 6, No. 1.
- MAH, R. S. H., MICHAELSON, S., and SARGENT, R. W. H. (1962). Dynamic behaviour of multi-component multi-stage systems. Numerical methods for the solution, *Chemical Engineering Science*, Vol. 17.
- MIURA, T., and IWATA, J. (1965). *Study on time-shared analogue computation technique*, Hitachi Central Research Laboratory, Tokyo, Japan.
- PEISER, A. M., and GROVER, S. S. (1962). Dynamic Simulation of a Distillation Tower, *Chemical Engineering Progress*, Vol. 58, No. 9, September 1962.
- ROSENBROCK, H. H., and STOREY, C. (1966). *Computational Techniques for Chemical Engineers*, Pergamon Press.
- RUSZKAY, R., and MITCHELL, E. E. L. (1966). Hybrid Simulation of a Reacting Distillation Column, *Proc. SJCC*.
- SARGENT, R. W. H., and MURTAGH, B. A. (1969). The Design of Plate Distillation Columns For Multicomponent Mixtures, *Trans. ICE*, Vol. 47.
- STOJAK, P. F. (1973). *Simulation of Multicomponent Distillation Dynamics*, Ph.D. Thesis, Control Engineering Group, University of Cambridge, April 1973.
- WHITE, G. W. T. (1967). Digital Simulation Languages for the Solution of Process Control Problems, Proc. IBM Scientific Computing Symposium, Digital Simulation of Continuous Systems, New York, 1967.
- WHITE, G. W. T. (1971). Methods of Modelling and Simulating Dynamic Systems, *Measurement and Control*, Vol. 4.
- WILKINSON, W. L., and ARMSTRONG, W. D. (1957). An approximate method of predicting composition response of a fractionating column, *Chemical Engineering Science*, Vol. 7.

Book reviews

The Macro Implementation of SNOBOL 4, by Ralph E. Griswold, 1972, 310 pages. (W. H. Freeman and Company, San Francisco, £7-00)

SNOBOL 4 is a programming language for manipulating character strings. The original SNOBOL language was developed in 1962 at Bell Telephone Laboratories, Holmdel, purely for internal use. It soon attracted attention outside, and in 1963/4 it was made available to several universities, research laboratories and other organisations. I was one of the first university users, and well remember the excitement of suddenly being able to give an undergraduate course on compiler construction. At that time the folklore said that it took 40 man years of programming effort to implement a compiler, and now that time could be reduced to a few hours. Today, the compiler course based on the use of SNOBOL is a cornerstone of university courses throughout the world. SNOBOL was also quickly adopted for a variety of unusual programming tasks ranging from language translation to musical composition. Bell Telephone Laboratories then allowed the authors of the system to continue its development; and, in the fine tradition of service to the community which has characterised this laboratory, subsequent versions of SNOBOL were made freely available to outside users. It is a great pleasure for me to be able to acknowledge here the friendly help which, like many others, I so often received.

Later versions of SNOBOL were more than a refinement of string manipulation facilities. Two main themes were pursued: first the fundamental basis of string manipulation was investigated. From this investigation the semantic concept of a 'pattern' emerged. Patterns can be constructed and manipulated in a wide variety of ways, and for example, can be returned from procedures. In a recent paper in the *Communications of the ACM*, (Vol. 16, No. 2, February 1973) J. F. Gimpel has pointed out that the power of the SNOBOL pattern analysis facilities is still beyond easy explanation by present day programming language theory. Second, the authors aimed at a translator/interpreter system which would be truly portable between different machines. This was achieved by basing the implementation on macros; by using these macros the system has actually been transferred to machines differing as widely as the CDC 6600 and the Atlas 2; this is surely an achievement which has been matched by few other software systems.

The present book is not a detailed manual for SNOBOL 4 implementors, it is an account of the development of SNOBOL 4, with a description of the fundamental problems, as the authors saw them, followed by a discussion of the philosophical basis for their solution. It is thus a glimpse into the minds of the implementors and is therefore much more informative than any manual could be. Because of this, the book is going to become a prime reference for historians of programming, and could well be a standard text in graduate courses on software engineering. Several chapters actually have exercises in them, but these are somewhat restricted to technical details,

and most teachers will, in my opinion, want to follow up the general discussion in each chapter.

Part of the popularity of SNOBOL is the result of the original flavour of the language; problems have been thought through in a new way, and very little has been borrowed. This tone of originality comes through in the book, and readers will find themselves alternately delighted and irritated by the views expressed, for example, the remark

'Declarations are largely a concession to implementation . . . Programmers become used to such requirements and may even think of them as useful programming tools'

is likely to provoke a strong reaction from language designers and implementors. To sum up, the book is obviously required reading for the knowledgeable SNOBOL enthusiast, but will be just as valuable to anyone interested in the broader aspects of language and software design, and will be particularly valuable to advanced students.

J. J. FLORENTIN (London)

Elements of BASIC, 1973; 84 pages. (National Computing Centre, £1-80)

This is an introductory book on BASIC which the authors claim is suitable for use in schools or as a self study book. To overcome the problems of machine dependence, loose cards are included in a flap at the back to show variations in a number of manufacturers implementations. On the whole I found this book unsatisfactory. Any text for a programming language should ensure that the programs given as examples are exemplary. Bearing in mind that the examples are likely to be fairly simple the particular point which should be emphasised is program legibility. This should manifest itself through good layout, adequate commentary, and in the case of BASIC, careful layout of the data statements and good run-time documentation. Few of the programs exhibit these features and some of the necessary facilities such as blank lines and comments following an apostrophe are not introduced. One of the programs only works by the skin of its teeth but is totally incomprehensible. The authors also have or impart a few misconceptions about BASIC, particularly concerning string arrays and the MAT statements. Concepts such as the current file and when one may omit quotes around strings and jargon such as real-time are introduced with no explanation and the description of some of the algorithms are rather brief. Finally the book is an odd shape, meaning 12" wide by 8½" high which makes it difficult to store in a bookshelf.

On the positive side it seems fairly error free with a clear layout. The flowcharts are good and I applaud the use of teletype output for program listings and results. There is a good selection of examples from the simple to the sophisticated. At £1-80 for a hardback it is fairly cheap.

G. M. BULL (Hatfield)