induce an artificial recovery at either the current or outer level of the stacked definitions.

Acknowledgements

Too many people have made contributions to the design of the director for them all to be individually acknowledged. Particular thanks are, however, due to S. T. Hayes, H. Whitfield and A. S. Wight. Thanks are also due to F. Barratt and his staff at the Edinburgh Regional Computing Centre for their valuable help in the elucidation of the multiplexor hardware. S. Michaelson has always provided encouragement and support. H. Whitfield implemented the signal mechanism.

References

Dijkstra, E. W. (1968). Co-operating sequential processes, in *Programming Languages*, F. Genuys (Ed.).

Millard, G. E., Rees, D. J., and Whitfield, H. (1975). The Standard EMAS Subsystem. *The Computer Journal*, to be published.

Stephens, P. D. (1974). The IMP Language and Compiler, *The Computer Journal*, Vol. 17, No. 3, pp. 216-223.

Whitfield, H., and Wight, A. S. (1973). EMAS—The Edinburgh Multi-Access System, *The Computer Journal*, Vol. 16, No. 4, pp. 331-346.

Wight, A. S. (1975). The EMAS archiving program, *The Computer Journal*, Vol. 18, No. 1, pp. 131-134.

Book reviews

Functional Analysis of Information Networks: A Structured Approach to the Data Communications Environment. by H. B. Becker, 1974; 281 pages. (*John Wiley*, $8.40)

In his preface, the author indicates that failures in implementing information networks (IN's) are usually due, not to faulty or inadequate design, nor to misunderstandings of the problem, but to one cause only—the lack of a DEFINITION of the problem.

In his book, Hal Becker has attempted, and achieved, this difficult definition task. Network components and functions are presented in a structured format, independent of reference to any proprietary devices. IN components are reduced to a basic set of simple functions, which are related by a hierarchical tree structure. Examples are then produced which evaluate designs from the top of this tree down to the applications level, rather than the traditional 'bottom-up' approach.

As an introduction to the subject, the evolution and definition of IN's are discussed, and their levels are then defined. Network functions in the form of the physical network are analysed to depth, and this is followed by a description of control functions in the form of the logical network. The main text concludes with an excellent IN example based on the ARPA network.

The appendix is worth mentioning, for it includes a full IN tree, a good but brief section on related reading, and a useful summary of the standardisation work taking place in this field.

Finally, it is appropriate to mention that the book is one of a series of 20-odd volumes intended for systems analysts, programmers, and DP managers.

I. G. Dewis (Teddington)

Macro Processors and Techniques for Portable Software, by P. J. Brown, 1974; 244 pages. (*John Wiley*, £5·25)

This is one of the most useful books which I have seen for a long time. It is almost two books in one, as the title suggests.

The first 'book within a book' deals with the whole field of macro-processors. Starting from a simple example of writing standard letters to visiting speakers by means of macros, the basic concepts are introduced, and the facilities which may be needed are described. Distinctions such as general purpose versus special purpose, free mode versus warning mode etc. are illustrated with actual macro-processors such as the IBM OS Macro-assembler, PL/I preprocessing facilities, GPM, TRAC, ML/I and STAGE2. These are not examined thoroughly, but provide examples of various features. Several more chapters fill out the ramifications of these features, so that the subject is examined from all angles, always in a lucid readable style, with well-chosen illustrations.

The second 'book within a book' describes the technique of using macro-processors to implement a programming language in a portable fashion. An attempt is made to quantify (roughly) the expenditure of effort needed to write software and to transfer it to another machine by various methods. The method which is recommended is to code the software by means of a DLIMP (Descriptive Language for Implementing Macro-Processors). The DLIMP is low enough in level to be mapped on to most assemblers by means of suitable macros. As an example, almost a hundred pages are devoted to a detailed explanation (and full listings of the programs) of the implementation of ALGEBRA by means of the DLIMP called LOWL.

Throughout, the temptation to mount the hobby horse of ML/I (the author's own macro-processor) is staunchly resisted. The result is a wide-ranging description of the state of the art which never loses sight of practical objectives. Despite the price and a fair sprinkling of misprints, it is highly recommended reading.

A. C. Day (London)

Graph Theory with Applications to Engineering and Computer Science, by Narsingh Deo, 1974; 478 pages. (*Prentice-Hall*, $17·05)

In his preface the author states that the book grew out of a number of courses. He attempts to recommend the sections relevant to particular subject areas. Even so, it has grown farther than the likely needs of most degree courses. It is rather a reference work on Graph Theory to which the author has confined himself most expertly and comprehensively. For all that, the text is greatly enhanced by constant reference to and example of practical application. The book maintains rigour and gives careful proof of most assertions yet particular attention is paid to the danger of losing contact with the reader. Contact is maintained when the going is heavy by taking time off to summarise progress in orderly and numbered notes and, most refreshingly, by simple numerical illustration. Copious exercises with useful hints are provided and a complete and fairly up-to-date bibliography will be of great benefit to anyone embarking on research in the subject.

The first ten chapters, almost exactly half the book, cover the theory of graphs, though, even here, one meets a mass of illustrative applications from chemistry to feeding dogs. Chapter eleven presents a number of basic algorithms, all flow charted and some programmed in a mixture of APL and FORTRAN. There is a useful list of graph oriented languages and software packages. The remaining four longish chapters are devoted to applications including switching and coding, electrical networks, operations research and a survey of other applications. Two theorems have, rather curiously, been relegated to two short appendices.

The book itself is nicely produced and a pleasure to handle. It should be on the bookshelves of every combinatorial mathematician and referred to by many more.

R. J. Ord-Smith (Bradford)