MAROVAC, N. (1973). *The organisation of the data-display structure and the structure of NEDLAN* (*NEtwork Defining LANguage*), to be published.

MAROVAC, N. (1973a). *The syntax of NEDLAN* (NEtwork Defining LANguage), to be published.

MAROVAC, N. (1973b). A single-data display structure: a new view on interactive computer graphics in CAD. *The Computer Journal*, Vol. 16, No. 2, May 1973, pp. 152-156.

RULLY, A. D. (1968). A subroutine package for FORTRAN, *IBM Syst. J.* Nos. 3 and 4, 1968.

SCHWINN, P. M. (1967). A problem-oriented graphic language, *Proc. 22nd Nat. Conf. ACM*, 1967.

SHEARING, G. (1970). VISPLAY. The CSIRO graphical FORTRAN system, International Symposium, *Computer Graphics '70*, Brunel, England.

# Book reviews

*A Mathematical Theory of Global Program Optimization*, by M. Schaefer, 1973; 198 pages. (*Prentice-Hall International*, £5·00.)

The term 'software engineering' is often used to describe the field of software system development, to emphasise that the construction of a compiler or an operating system is not an art but is a process to which the well-known techniques of engineering design can usefully be applied. Engineering is usually based on a sub-structure of science, and this book is a noteworthy contribution to the scientific substructure of compiler writing. It is also noteworthy in that it must be the first book about compiling techniques for which '. . . previous exposure to computer programming is helpful but not necessary'.

The designer of a compiler may try to optimise the generated code at two levels. There is 'within-statement' optimisation that endeavours to exploit the architecture of the target machine in the code generated for individual statements, e.g. making best use of the available registers for intermediate results. Much more difficult is global optimisation which aims to improve efficiency by removing non-varying quantities from loops, recognising patterns in the array elements, and reordering the statements of the program to avoid redundant re-computation of sub-expressions. The gains to be obtained are considerable: the IBM H-level FORTRAN compiler is reported to have produced object programs occupying 25% less space and using 40% less processor time than the programs produced from the same source by the non-optimising G-level compiler. However, the dangers are equally great: after it was released to users, it was discovered that the ICL optimising FORTRAN compiler sometimes generated incorrect code. Thus the designer of an optimising compiler needs to be sure that his transformations retain the meaning of the program, and it is here that science comes in, giving the engineer the assurance that his artefacts will perform as expected.

The book under review provides such a foundation for global optimisation. Since such optimisation involves a detailed analysis of the flow of control in a program, the approach is to express the structure of the program as a directed graph. The transformations required for optimisation are now transformations on the graph, and the mathematics of graph theory can be used to give a rigorous proof of the validity of these transformations. The first seven chapters of the book are devoted to establishing the necessary (and considerable) body of mathematical background: there then follow a series of chapters each dealing with a particular class of optimising transformations. Although these transformations are expressed in abstract mathematical form, each chapter begins with a number of concrete examples to give the reader a feel for what is to be achieved. Finally three appendices cover optimising algorithms (written in APL/360), an overview of the phases of an optimising compiler, and a discussion of the influence of partial recompilation ('incremental compiling') on the global optimising process.

The book is a most valuable contribution to the literature of compilers, though the mathematical standard will make it inaccessible to many workers in the field. Those with the necessary degree of mathematical sophistication to cope will find it hard work, but exceedingly rewarding. For the rest it is a portent of the shape of things to come, and a valuable counter-example to the proposition that the computer scientist need have no mathematical ability.

D. W. BARRON (Southampton)

*Problems of Heuristics*, edited by V. N. Pushkin, translated from Russian by D. Louvish, 1973; 202 pages. (Israel program for Scientific Translations; *England, John Wiley*, £8·45.)

*Problems of Heuristics* is a translation of a collection of papers published in the USSR in 1969. It contains four sections: Theoretical Problems of Heuristics, Experimental Studies of Heuristic Processes, Programming of Components of Human Thought and Investigation of Reflexive Processes.

As well as editing the collection, Pushkin is either an author or has his work referred to in rather more than half of the nineteen papers.

A heuristic process is seen as one which constructs a new action aimed at the achievement of some goal in a situation which is new to the system performing the process. Thus by heuristics is meant the science which studies the laws governing the design of new actions in new situations. In 'Toward a definition of heuristics', Pospelov, Pushkin and Sadovskii deny that a computer program can be heuristic. It is clear that this view is strongly held but they fail to make an adequate case to support it. They argue, for example, that a potential infinity of languages for the formation of a model is needed by a system that is to be heuristic, but it is not obvious to the reviewer that this must be so in a sense that makes it impossible in a computer system.

The main thesis of the book, insofar as there is one, is that the simple maze view of heuristic search is not adequate and that we must look at systems which can both construct internal models and can also radically revise them in the course of problem solving. This leads to the most important achievement of the book, its insistance on a close relationship between theory and psychological fact. For example, one paper puts forward the view that complex tasks can be automated most successfully by first conducting a psychological study of the human methods of solution.

By far the longest of the four sections is Experimental Studies of Heuristic Processes. This contains studies of human problem solving, mostly using protocol analysis, but in one interesting case, by using recordings of the eye movements of chess players whilst they decide upon a move.

The book ends with a short section on reflexive control, i.e. on attempts to win in human and man-machine conflict situations.

*Problems of Heuristics* is a lively collection containing within it a range of differing, and at times contradictory, views. An editorial guide to those various attitudes would have been a useful addition to this interesting book.

E. A. EDMONDS (Leicester)