Book reviews

DAVID SKILLIKORN
Foundations of Parallel Programming. Cambridge University Press. 1994. ISBN 0-521-45511-1. £25.00. 197 pp. hardbound.

While this book contains, as the word 'foundations' in the title may suggest, much of interest to theoreticians, it aims nevertheless to provide a methodology which can be used to develop efficient parallel programs and, moreover, to do this in a context independent of particular hardware architectures. One of the main claims of the book is that it is possible to decouple parallel software from parallel hardware so that the software can be moved from one parallel architecture to another.

Based on categorical data types which provide a structured way to search for algorithms and structured programs to implement them, the key idea is that of distributing computation over a data structure so that calculations can be carried out on the components of the structure in parallel. This is done by the use of homomorphisms over the structured type constructors:

$$h(a \bowtie b) = h(a) \otimes h(b)$$

If such a homomorphism can be found, h can be distributed over a structure built by using the $\bowtie$ constructor and h can then be executed in parallel at all the leaves of the data structure. As well as developing a general theory based on these ideas, the book looks at how they apply to particular data types such as lists, trees and arrays.

The categorical data type model also facilitates the development of formal methods for traditional software engineering purposes such as reasoning about complexity, correctness and cost measures. Two chapters are devoted to software development by transformation using equational reasoning. One of the main advantages of the categorical data type approach over traditional abstract data types is claimed to be that the former guarantees that the set of transformation rules derived by the method is complete.

As well as the development of categorical data types described above, the book contains useful chapters on other approaches to parallel software, to the main parallel styles of machines and to their architectures, and to development of a cost calculus for lists. An appendix gives a brief historical background.

The book is by no means a lightweight read but careful study and rereading is rewarding. As the author readily admits, much of the material is still at a research stage, but there is also plenty in the way of software methodology which is immediately useful.

A. DAVIE
University of St Andrews

IAN PARBERRY
Circuit Complexity and Neural Networks. The MIT Press. 1994. ISBN 0-262-16148-6. £40.50. 270 pp. hardbound.

This book is aimed at researchers and practitioners working in the area of Artificial Neural Networks, one of its primary purposes being to consider this model within the framework of classical computational complexity theory. The opening chapter introduces the ideas underlying computation by neural networks and discusses the issue of scalability: quantifying how the complexity of a neural network increases with respect to increases in the size of the problem being solved. Chapter 2 reviews classical models of computation and computational complexity theory. In this chapter the use of Boolean combinational circuits as the vehicle for comparing neural network complexity with an established model of decision problem complexity is justified and a lucid discussion of the problems of uniform and non-uniform models is presented. Having established the foundations for the comparative study the remaining chapters describe increasingly sophisticated circuit models—*alternating circuits, threshold circuits, probabilistic models, Boltzmann machines*—and relate these to the issue of scalability in the neural network model.

This book is of interest not only to those working in the field of neural networks but should also be found to be of value to researchers in the area of computational complexity theory in general and the complexity theory of Boolean circuits in particular. In the latter area, Chapters 5–7 of the book provide a valuable summary of recent results in an important subfield of Boolean complexity theory. While a certain degree of mathematical awareness is required, the results and ideas presented are clearly described and easily accessible.

PAUL E. DUNNE
University of Liverpool

RANDY M. KAPLAN
Constructing Language Processors for Little Languages. John Wiley. 1994. ISBN 0-471-59754-6. £41.50. 452 pp. softbound.

This book takes a refreshingly new look at the problem of developing software and provides a neat, formal, proven solution. Language processing technology is often left to the world of systems programming to the extent that the application world would like to have none of it. Contrary to the popular opinion, it can make a world of difference even in developing large applications due to its sound mathematical basis. Technologies like parsing and lexical analysis, to name but a few, have proven tools available which make the task of a