of the bibliography to researchers new to the field. Also, although the exercises are plentiful and instructive, some crucial results depend on exercises for which no solutions are given. Despite these shortcomings, this book constitutes a useful starting point for new researchers.

CAROLYN BROWN
University of Sussex

JUAN BICARREGUI *et al.*
Proof in VDM: A Practitioner's Guide. Springer-Verlag. 1994. ISBN 3 540 19813 X. 363 pp. softbound.

Proof is often cited as a technique that may be used with respect to a formal specification but not with an informal one. Using proof, we can not only discharge well-formedness and satisfiability obligations, but we can also prove validation conjectures and/or satisfaction of particular properties. However, to date, few examples of such proofs exist in the literature, and those examples that do exist provide little or no guidance to the novice on how they have been constructed. This book seeks to redress the balance by demonstrating how to go about constructing proofs in formal specifications. The style of presentation clearly illustrates how proofs evolve, from a general outline of the proof, to a detailed formal proof using axioms and inference rules. Although the text is based on specifications written in VDM, the emphasis is on the principles of formal proof, rather than on idiosyncratic properties of VDM.

The book is divided into three sections. The first section describes the logical system used in proofs and presents an axiomatization of the basic data types in VDM. In particular, this section introduces the idea of proof as a means of deriving theorems concerning data types from the axioms for these data types. Each data type is introduced by giving the appropriate axioms of formation, and, where appropriate, direct definitions. Examples of the use of such axioms, definitions and theorems are liberally scattered throughout the text, and in general are thoughtfully described. Some data types also include axioms which describe induction rules, together with corresponding examples. However, the treatment of induction is slightly weaker than the overall standard of the section—more overview of the general idea would be kinder on the novice reader.

The second section, entitled 'Proofs In Practice', presents some applications of the proof theory described in the first section. Three categories of application are considered: proof obligations arising from an arbitrary specification, proof obligations arising from reification and an illustrative case study. The first category describes, at an abstract level, how to prove well-formedness and satisfiability of specifications. The description is largely restricted to the general form that such proofs take, though a few concrete examples are given. The chapter on reification describes data reification and operation modeling, using retrieve functions. An example is presented, which although simple, adequately illustrates the concepts described. The result is a concise, elegant description of data reification that is a pleasure to read.

The case study, however, is the real revelation of the section. It is customary, when reading tutorial texts on formal specification, to expect illustrative case studies to be so simple as to be totally unrealistic. Here, quite the opposite is true: the case study presented, though necessarily simplified to avoid excessive detail, is remarkably realistic. The case study itself concerns the specification of a tool that manages the allocation of aircraft to air-traffic controllers within an air-traffic control region. Following an analysis of the system requirements, the formal state model is presented and its corresponding theory given. Within this theory, the arising proof obligations are formulated and discharged, and similarly for the validation conditions arising from the system requirements. The top-level operations of the system are specified, then two refinement steps are given, the first of which includes all the relevant proofs. All in all, the case study perfectly fulfills its objective of drawing together the material of the preceding sections in a realistic setting.

The final section consists of a brief description of data types not already treated and a directory of theorems. The former discusses how to axiomatize such data types and where relevant discusses the resultant problems. In a sense, this chapter is not strictly necessary but merely ties up some loose ends. On the other hand, the directory of theorems is a hive of information, collecting together theorems and axioms, grouped according to data type. To anyone embarking upon a proof, such a directory is invaluable.

Throughout the text there are many exercises of varying degrees of difficulty. The solutions of the exercises are available by FTP, and mirror the thoroughness and clarity of thought manifested in the main text. However, the main text itself is rather less readable than the solutions—the small typeface and narrow margins combine to give the test a rather cramped appearance, though this is presumably to keep down the price of the book.

In the past, anyone wishing to perform proof on formal specification has had to ask the question: "where does one start?". With the publication of this book, this question has been answered totally. Not only does the book explain what proof is, when it should be used, why it should be used and how to use it, but it does so in a style that does not patronize the expert and simultaneously engenders confidence in the novice.

P. MUKHERJEE
University of Birmingham

LESLIE PACK KAELBLING
Learning in Embedded Systems. MIT Press. 1993. ISBN 0-262-11174-8. £26.95. 176 pp. hardbound.