the area and all opportunities for reminders to related or earlier work are used, even at the cost of repetition.

The insights and connections offered by this book are especially valuable since the system it refers to, MOBAL, is a real working system not just a theoretical possibility. In addition to detailing the development of a specific system, the book attempts to address a wide set of issues in the construction of knowledge acquisition system and will prove useful to all modelling such systems.

Reviewed by Rashmi Pandya, Department of Computer Science, University of Exeter, UK

Algebraic specification techniques and tools for software development—the ACT approach by I. ClaBen, H. Ehrig and D. Wolz, AMAST Series in Computing, Vol 1. World Scientific Publishing Co., USA, 1993, pp 237, £27.00, ISBN 981-02-1227-5.

The authors present a formal algebraic approach (namely ACT) that aims to support "the entire software development process from semi-formal requirements, via functional requirement specification, design specification and executable specification (or prototype) up to the efficient software system version or product." The method is rooted in universal algebra and category theory, and so readers without some grounding in theoretical computer science beware.

In algebraic specification approaches one defines data types with a number of operations on them. The operations are defined in terms of declarative equations relating constructors (primitive objects) and other operations. The advantage of the approach is that it allows a formal semantics to be assigned to specifications, proofs of key properties are possible and mechanisable by term rewriting, and certain classes of specification can be executed, again by term rewriting. Despite this little use has been made of such techniques in industry; this book is trying to change that situation.

In chapter 1 there is an introduction to the use of algebraic specification methods and the theory underlying them. It is a strange mix covering life-cycle issues, underlying theory and tool support. I found this chapter rather intimidating as an introduction because of the passing references to results of some fairly heavy theoretical computer science. Still, this should not deter the uninitiated as most of the rest of the book concentrates on the application of the method. It is also a little simplistic in its treatment of life-cycle issues, and talks about formal refinement in a very matter of fact way, however the real difficulties of doing this for large, complex systems is glossed over.

The ACT ONE language is presented in Chapter 2. The two main building blocks of the language are type definitions and parameter specifications. It is possible to build parameterised specifications and combine definitions with extension, union, renaming and actualisation mechanisms. We are gradually taken through the features of the language with increasingly hard examples of mini-specifications for natural numbers, lists, stacks, sets and maps, culminating with a symbol table for a block structured language, which is built from a stack of maps. We are also given an overview of the language syntax and semantics before a 39 page case study development of a syntax directed editor. The authors are very honest about limitations of the language, and point out the features where the language might change in the future. At this point I was seduced by the generality and power of the approach, in particular how specifications can be built by combining smaller units. But I was also left wishing I could experiment with the ACT ONE system. My previous experience of algebraic specifications was with OBJ. I remember the initial frustration with the interface and the strictness of the system, followed (much later) by the jubilation of a specification that could have proofs performed about it.

In Chapter 3 we learn how to develop modular systems using the ACT approach. A module consists of four parts: an export interface, import interface, parameter part and body part. The export interface is sufficient for other modules to use the module. The import interface contains a number of formal parameters that can be matched to the exports of other modules. The parameter part allows modules to specified generically and actualised on use. The body defines the operations

defined in the interface. Once again the authors back this up with a detailed case study, this time a material requirements planning system.

In Chapter 4 the ACt Environment is described. This helps provide a good feel of how one would use the actual system. Of particular note is a facility to generate visualisations of algebraic terms. I feel that this chapter is effective as it can be, however there is no real substitute for first hand experience of the system.

Appendix A gives a more detailed treatment of the formal theory underpinning ACT. Realistically, one needs a good grasp of category theory to understand this. Finally, Appendix B gives some brief user instructions for the ACT Environment.

I hearily recommend the book to all computer science students at Technical University of Berlin (the authors' institution)! It should also be very useful to others studying algebraic techniques in general, and industrialists wanting to gain an understanding of what is possible with algebraic specifications. The balance between theory and examples has been struck, with plenty of case studies to back up the concepts. To summarise, if the area interests you and you are prepared to grapple with a little discrete mathematics, then buy it.

Reviewed by Stephen P. Wilson, University of York, UK

The uncertain reasoner's companion: a mathematical perspective by J. B. Paris, Cambridge Tracts in Theoretical Computer Science 39, Cambridge University Press, UK, 1994, pp 212, £25.00, ISBN 0 521 46089 1.

Uncertain reasoning is a very controversial field, where very different approaches are competing. After the era of Mycin-like "certainty factors", very popular in the 1970s, a lot of mathematical work was carried out in the 1980s to base the field on solid foundations. Dempster-Shafer Evidence theory, Fuzzy logic and related possibility theory, and many logics have been developed for addressing the uncertain reasoning problem. The old probability theory, a little bit rejected in the 1970s, is now becoming the most used formalism, in the modern form of belief networks. Some people, while convinced of the suitability of probabilities for representing uncertainty, find the constraints imposed by the belief network formalism too strong, or unrealistic, and thus have proposed methods for allowing inference from knowledge bases represented as a non-organized set of beliefs. Among these approaches, the ones based on Maximum Entropy are the most popular. The author of *The Uncertain Reasoner's Companion* is one of the most famous advocates of this last approach, and his book, while based on mathematical grounds, is clearly biased toward the author's preference.

The book can be seen to be constructed in four parts: the first part (Chapter 1) describes the motivation, and a mathematical formulation of the problem of uncertain reasoning. In the second part, three approaches are presented and discussed, namely: probabilities, which are briefly introduced in Chapter 2 and discussed in Chapter 3; the Dempster-Shafer theory, described in Chapter 4; and the so-called "truth functional theories", which contain fuzzy logic and possibility theory as special cases, are presented in Chapter 5. Chapters 2 to 5 contain detailed justifications and criticisms on each proposed approach, and they are thus controversial, even if the main content of these chapters, as is the case for the remaining of the book, is mainly a set of theorems and their demonstration. The third part is mainly devoted to probabilistic inference. Inference processes are defined in Chapter 6, and the usual inference processes are described for each approach. Chapter 7 is the most important of the book, since it contains 10 properties (principles) of common sense reasoning, according to the author's opinion. The extent to which these properties are met by the inference processes is discussed, and the main results, not demonstrated in the book but referenced, is that the maximum entropy approach is the only one consistent with all the principles.

Chapter 8 describes three methods for belief revision: Bayesian updating is the most usual one, and it is shown to be a special case of Jeffrey's updating, which in turn is a special case of Minimum