those propounding this essentially computational view of design must admit the existence of at least three characteristics of creative design, as practised by architects using pencil and paper, which cause problems for computer-based support tools. Firstly, shapes, or combinations of shapes, are ambiguous in the sense that they may be open to interpretation in a number of ways. This idea of "emergent shapes", or shapes which somehow emerge unexpectedly as a result of combining other components in a design in a particular way, is one which seems very important in the business of architectural design. Secondly, the rules of a shape algebra are often unstable or context-dependent. Thirdly, critical reasoning about architectural designs is essentially non-monotonic, in that judgements about whether some aspect of a design has been satisfactorily completed may later be reversed in the light of further considerations. This last factor makes it hard to decide when a design is finished.

Early tools for CAAD emphasized rigid structures for designs by specifying shapes in terms of points, edges and surfaces. This was at first felt to be an advantage, but later came to be recognized as unnecessarily restrictive. The trend in more recent years has been to attempt to provide semantically richer and more flexible descriptions of both the initial shape vocabularies and the designer's knowledge about how elements of these vocabularies can be transformed or combined. This trend is exemplified in a number of papers in the book. For example Stiny and Tan suggest more structured and flexible ways of describing shapes in terms of schemata and high-level relationships between their components, and Schmitt et al. suggest attempting to code a number of different types of expert designer knowledge into a system for CAAD. Rules, frames and scripts all make an appearance, as do Lisp and Prolog in roughly equal measures. Approaches based on the use of neural nets and hypermedia are also discussed in two papers.

These later tools have begun to tackle the problems of ambiguity or discontinuity in shape interpretation and the need for flexibility in rules about design to provide more useful support to the expert architectural designer for a wider range of activities than was previously possible. There is still, however, much progress to be made, and in particular, the problem of non-monotonicity has apparently not been cracked.

The book as a whole is well-produced and has a good index, though one or two of the papers seem to have suffered a little in translation (contributions from Europe, Israel and Japan appear alongside those from Canada, Australia and the US). Its subject matter is at first sight rather esoteric (and this is reflected in its price!), but on slightly closer inspection, yields an interesting insight into the applicability of KBS techniques in an ususual but important area of application.

Approximate reasoning models by Ramon Lopez de Mántaras, Ellis Horwood, Chichester, 1990, pp 109.
Search, inference and dependencies in artificial intelligence by Murray Shanahan and Richard Southwick, Ellis Horwood, Chichester, 1989, pp 140.

A distinguishing feature of intellectual man* is his ability to solve a diverse range of problems. However, in modern professional life the need to solve problems of such complexity, or requiring such wide domain knowledge, or with such frequency, is making computer-based assistance an increasingly necessary tool. An expert in a given field will have access to relevant domain knowledge and theories to enable him to identify solutions to a given problem, and make a judgement on the efficacy of acting on those solutions. He will, in general, be able to reason with uncertain and incomplete information, hypothesise and revise solutions as further information becomes available or if the solutions are found to violate constraints imposed on the solution space.

The difficulty is to find a model of human reasoning which is sufficiently rich to capture all these aspects. To date, in AI there has been a long tradition of using numerical techniques to cover uncertainty and vagueness in reasoning. More recently, there has been an increasing use of Reason Maintenance Systems (RMSs) to model a more symbolic style of problem solving and hypothetical

* Out of laziness, and in want of a better word, I use the male term to represent the generic of personkind.

reasoning. Two recent additions to the Ellis Horwood series in Artificial Intelligence address, respectively, each of these reasoning styles. *Approximate Reasoning Models* by Ramon Lopez de Mántaras covers probabilistic, evidential (dempster–shafer) and fuzzy/possibilistic approaches. *Search, Inference and Dependencies in Artificial Intelligence* by Murray Shanahan and Richard Southwick covers the application of RMSs to constraint satisfaction problems, theorem proving and hypothetical reasoning.

The question which, as yet, has no definitive answer is whether any of the numerical techniques do ultimately express anything which it is not possible to express in any of the others. A critical dispassionate evaluation of probability versus dempster–shafer versus possibility would be a valuable addition to the current literature on these subjects. However, this is not really a concern of *Approximate Reasoning Models,* which contains a short undiluted introduction to the formal theory of each model, but no discussion of their relative merits. Where the author's own interests lie should be apparent from the contents page. The book is divided into three chapters, with Chapters 1 and 2, covering probability and dempster–shafer respectively, being allocated a quarter of the book each, and the final chapter, on fuzzy logic and possibility theory, being allocated the remaining half.

The claim on the inside cover is that the "book provides an up-to-date analysis and description of the main numerical approaches to modelling approximate reasoning . . . and presents them in a rigorous yet easy to read manner". Two out of three for being up-to-date and for rigour. Two pages in the chapter on probability are devoted to Pearl's algorithm and Bayesian networks. The vast bulk of this chapter, however, is devoted to the much criticized uncertainty handling in PROSPECTOR and MYCIN. A great deal of research activity has been expended to address the criticisms of the MYCIN and PROSPECTOR approaches, and yet we only have the briefest discussion of Pearl's work. Indeed there is no mention at all of the Lauritzen and Spiegelhalter algorithm and its provision of a highly efficient and rigorous probabilistic update mechanism as embodied in, for example, the expert system MUNIN and its associated shell HUGIN.

This more recent work requires the elicitation of the expert's knowledge in the form of an inference network where the nodes represent knowledge items (e.g., diseases, findings, physiological states), and the links, relations between them. In the probabilistic case, prior and conditional probabilities are then assigned to the relevant nodes and links. Some graph-theoretical techniques may then be exploited to enable the efficient update of probabilities as evidence nodes are conditioned on. The pity that no more than a passing mention is given to networks in this book is that the graph theory is applicable no matter what calculus is used to propagate information through the network. The technique is as applicable to dempster–shafer and possibility as it is to probability.

As regards differences in expressiveness, one distinction is that dempster–shafer allows an expression of ignorance. A belief mass may be assigned to a set of propositions if one cannot differentiate further between the elements of the set; I may be 80% sure that the murderer was one of Peter, Paul or Mary, but be quite unable to say any more than that. In the dempster–shafer approach, a belief mass of 0.8 would be assigned to the set {Peter, Paul, Mary}. The dempster rule of combination then allows belief to be transferred to appropriate subsets of {Peter, Paul, Mary} from that assigned to non-disjoint sets by other sources of evidence.

Fuzzy logic and possibility theory allow for notions of vagueness as well as uncertainty. The use of vague propositions ("X is *old*") and fuzzy quantifiers ("X is *very* red") is modelled. The dempster–shafer and fuzzy/possibility models are dealt with here in much more rigour than probability. But it is not the easy reading promised on the inside cover. For example, in the discussion on dempster–shafer the condition for the "Belief" and "Plausibility" measures to be the same is given as "$A \cap B = \emptyset$ when $A \subseteq B$" for all the focal elements in X. The requirement is that the focal elements be disjoint sets, which is easy to understand and quite rigorous if stated in English. Equally characteristic of the style of presentation is that in the chapter on fuzzy logic, the symbol F is used for two pages before it is defined as the domain of a fuzzy set, the symbol Π_x is used before mentioning what it represents (a possibility distribution).

The whole book has an air of having been prepared in haste. The resulting carelessness in presentation can make some of the arguments very hard to follow. There are a number of excellent review articles available onreasoning with uncertainty, and they are certainly worth reading before attempting this book (some references are given at the end of this review).

In contrast, *Search, Inference and Dependencies in Artificial Intelligence* is a much more workmanlike book. This is perhaps a reflection of the fact that much of the book is based on Murray Shanahan's doctoral thesis and so has had to be subjected to greater critical evaluation than the previous book. The disadvantage of making a book from a PhD thesis is the resulting rather terse presentation, although depending on your degree of motivation, this may not be a problem.

The general role of a reason maintenance system is to assist a problem-solver in reaching a solution to a problem and/or revising a solution to a problem by recording a structure of data dependencies. The benefits gained by coupling an RMS to a problem solver are:

(1) Incrementality: an unguided search may entail a large amount of redundant recomputation. By recording dependencies as the search progresses, the problem solver may be prevented from unnecessarily rediscovering logical dependencies in another part of the search space.
(2) Selectivity: inconsistent sets of sentences are recorded to avoid the later exploration of any part of the search space which includes those sentences.
(3) Consistency management: if an inconsistency is discovered, the data dependencies may be exploited to identify those data that contributed to the inconsistency.
(4) Non-monotonicity: in commonsense and hypothetical reasoning, conclusions that were previously held to be valid may be withdrawn upon receipt of further information: I may revise the belief that I can get to work for 9.00 am on finding that there has been a heavy fall of snow. The recording of data dependencies enables the identification of those conclusions which have to be withdrawn or revised upon receipt of new information.

A classic use of RMSs is to assist in the solution of constraint satisfaction problems. As well as the more academic map-colouring, n-queens problems, and the like, these include genuine real-world applications such as VLSI circuit design, laying out circuit boards, configuring computer systems and solving planning problems. If a solution is found by the problem solver which violates one or more of the constraints, the RMS must assist in providing an efficient method of backtracking to enable the continued exploration of the search space for a solution which satisfies all the constraints. Various backtracking techniques are discussed here. Since the constraint satisfaction problem is NP-complete, it is not to be expected that it be possible to isolate one algorithm and say that this will be universally more efficient than any of the others. One has to identify the algorithm appropriate to a given problem. The algorithms are given in full in pseudo-code with a written description. This is followed by an empirical comparison of four of the algorithms (chronological backtracking, forward checking, selective backtracking and selective backtracking with nogoods) in one problem domain.

The empirical study showed some distinct advantages of the selective backtracking techniques in improving the efficiency with which constraint satisfaction problems may be solved. The next goal of the book is to explore the use of more intelligent backtracking schemes in theorem provers. Most Prolog interpreters use naive chronological backtracking to search for solutions to a goal. Although experienced Prolog programmers will write code which requires little or no backtracking to arrive at a solution, and guide the interpreter to finding efficiently all solutions if required, there is a case for augmenting the Prolog interpreter with a more intelligent backtracking technique. This is discussed in the later part of the book in which RMSs are discussed in the context of theorem proving and hypothetical reasoning. If I understand aright, this is the main research contribution of the book to this field. Unfortunately, I found it a little disappointing. There is but a cursory introduction to four logical representations of hypothetical reasoning; default logic, circumscription, negation-as-failure and abduction. This is followed by just two paragraphs on the possible use of an RMS in updating beliefs in the light of new information. Even if this is not a particular

research interest of the authors', it would have been useful to have seen some discussion of the state of the art of the application of RMS's to this field.

The main suggestion is that an RMS be used to implement an incremental theorem prover in which a record is kept of the dependencies in the search space it has explored to answer a query. If a small change is made to the database (adding or deleting one clause) the dependency information may be exploited by the RMS to enable the *same* query to be efficiently resolved after the change. This idea is explored with a simple blocks world example (a surface with moveable blocks stacked on it). However, it is not at all clear that, in general, the overhead of storing dependency structures as new queries are solved, and updating them after small changes in the database, will be justified by the frequency with which those queries are repeated. If the modified clauses are buried deep in the proof tree, instead of being at or near the leaf nodes, the computation of the revised dependency structure may be quite complex. There is no suggestion in the book of a real world application in which such an approach would be a positive advantage.

In summary, those parts of this book which discuss the algorithms for more intelligent backtracking schemes contain much useful information. I remain agnostic, however, as to the value of their application to automated theorem proving.

So, two books covering important theoretical and computational work on the foundations of AI. But we still await some good sound scholarly work to consolidate our current understanding of the various numerical and symbolic models of automated reasoning.

References

Buxton, R, 1989. "Modelling uncertainty in expert systems" *Int. J. Man-Machine Studies* **31** 415–476.

Clark, D A, 1990. "Numerical and symbolic approaches to uncertainty management in AI" *Artificial Intelligence Review* **4** 109–146.

Saffiotti, A, 1988. "An AI view of the treatment of uncertainty" *Knowledge Engineering Review* **2** 75–98.

Expert systems for business: concepts and applications by D V Pigford and Greg Baur, Chapman & Hall, London, 1990, pp 391, £15.95.

This paperback edition is aimed at students on artificial intelligence/expert systems undergraduate courses, and business studies students. According to the authors' preface, "Practising professionals are also a viable user population". They may indeed be *viable,* but I would expect them, like me, to be intensely irritated by aspects of this book.

The book is organized into two sections—Part I, "The Concepts", consists of nine chapters covering: the intelligent computer; expert systems technology; knowledge representation; reasoning; the knowledge base; the inference engine; the user interface; software engineering and expert systems development/integration and the future of expert systems.

Part II, "The Applications", covers eight tutorial modules using the expert systems shell program VP Expert—a cut down version of which is included in the purchase price (though not received or therefore reviewed).

In essence, this is a book about the application of "intelligent" technology to practical problems. I found it astonishing and irritating to discover an enormous number of typographical and other errors in the text, including seven in the first 14 pages! The application of a little intelligent proof reading seems to have been sacrificed in the rush to get to press!

The early chapters are quite well laid out, though, with the student reader in mind they could be improved with more use of bold text and initial capitals for defined terms. Each chapter concludes with a list of key points which are generally useful, though the eight page chapter on the "Future of expert systems" generates 35 key points! Again aimed at the student, some exercises and more detailed assignments are given. Although these are helpful in the main, errors are found here too. The diligent student could find it frustrating seeking a specific reference to work on in Chapter 3 which is not given a year of publication!