the wife cannot be entered without details of the husband being present. It is clear that the only way that an entry can be made is for the details of both husband and wife to be entered at the same time. Such circular dependencies can arise when separate constraints are being combined, and they have to be detected as such.

4.2. *Use of mathematical logic*

It is usually agreed that systems design must involve creative imagination, but there is hope that more and more of the design effort will be carried on by routine techniques, in the same way that engineering design involves routine calculations. It is, therefore, pertinent to ask whether mathematical logic as used above could become a routine design tool. At this early time some tentative observations are:

1. Mathematical logic is obviously a tool for specialised theoretical studies.
2. Logical notation is a concise shorthand for expressing certain kinds of design information. It is reasonably easy to learn, and can therefore be expected to find some general use. It will be reinforced in use if certain types of database programming languages come into use (see Codd, 1971).
3. The direct manipulation of logical formulae is so tedious as to be unusable as a practical design tool. Methods of manipulating logical formulae by computer have been demonstrated by several research workers (see Robinson, 1967); the performance achieved so far, falls far short of that needed for system design. The future of such developments remains speculative.
4. Progress could be made by studying specific efforts, where concrete insights could be combined with logic. An example might be in developing tests to recognise circular dependencies in constraints.

At the present time it has been found that systematic attempts to express constraint conditions in logical notation are a valuable aid to examining the nature of the constraints. In particular, it is found in practise that various alternative ways of expressing the constraint become apparent, and this is valuable in providing more insight.

5. Conclusions

An introduction to one general aspect of the preservation of accuracy of stored data has been given by isolating the notion of the consistency of stored data. Accuracy needs in commercial applications have already given rise to several methods of input data validation which have been subjected to basic theoretical research. On the other hand, auditing of stored data is usually the result of an ad hoc collaboration between accountants and systems designers, and does not appear to have been the subject of systematic research. Since computerised systems are much more flexible than manual ones, it seems that a slow evolution of auditing practises will not be adequate to cope with changing and expanding demands. It will be necessary to devise an overall design approach to auditing that will allow the designer to produce effective audit procedures for new, and untried, systems.

Acknowledgements

The author would like to acknowledge stimulating discussions on the fundamental theory of databases with P. Hopewell (IBM UK Laboratories Ltd.), Dr. M. G. Notley and Dr. T. W. Rogers (IBM Scientific Centre, Peterlee). The application of logic to the theory of computation has been discussed with C. D. Allen and C. B. Jones (IBM UK Laboratories Ltd.). The author is also grateful for encouragement and comments from his colleagues Professor P. J. H. King and J. Inglis.

References

Codd, E. F. (1970). 'A Relational Model of Data for Large Shared Data Banks', *Comm. ACM*, Vol. 13, No. 6, pp. 377-387

Fraser, A. G. (1969). 'Integrity of a Mass Storage Filing System', *The Computer Journal*, Vol. 12, No. 1, pp. 1-5.

Mendelson, E. (1964). *Introduction to Mathematical Logic*, Van Nostrand, New York.

Robinson, J. A. (1967). 'A Review of Automatic Theorem Proving', *Proc Symposia in Applied Mathematics*, Vol. 19, American Mathematical Society, Providence, R.I., pp. 1-18.

Wilkes, M. V. (1972). 'On Preserving the Integrity of Data Bases', *The Computer Journal*, Vol. 15, No. 3, pp. 191-194.

Book review

Pattern Recognition, Learning and Thought, by L. Uhr, 1973; 506 pages. (*Prentice-Hall Inc.*, Englewood Cliffs, N.J. £7·30)

Professor Uhr has written an unusually *personal* book. In many ways, his text reads as though it were an interesting, albeit onesided, discussion with the author. The style is informal, colloquial, and jargon-free.

Nevertheless, the subject matter is treated thoroughly and it is possible to support the author's implied claim that he has written a useful, self-contained textbook 'as an introduction to simulation models of cognitive processes and artificial intelligence'.

The first nine chapters of this rather long book deal exhaustively with the great variety of programs which have been designed to solve pattern recognition problems. In common with many authors, Professor Uhr has concentrated his attention on character recognition techniques, and, within this rather restricted area, has produced an admirable survey. He introduces a programming language (EASEy) which he describes as 'a general-purpose list-processing, pattern-matching language . . . closely modelled on SNOBOL', and uses his language to provide programmed illustrations for all the main features of pattern recognition methodology. In this way, some of the apparent differences between similar programs written in different language structures, are ironed out.

Descriptions of useful applied pattern recognition systems, where 'production' pattern processing is being achieved, are not given, nor are the problems concerned with grey-scale pictures and scene analysis discussed. Thus, these chapters are thorough in their treatment of pattern recognition only up to the point where the problems begin to look insuperable.

The second half of the book looks at problem solving, game playing, theorem proving and (computer) learning methods. Finally, these various concepts are combined with the pattern recognition ideas to suggest flexibly structured programs which learn to recognise patterns in fairly simple classes of input picture.

There is a good bibliography (25 pages of references) and a short and rather unhelpful glossary. Anyone wishing to get a good idea from scratch as to what pattern recognition is all about could do a lot worse than study this book. Those who are old hands in the field will find the summary discussions, following each chapter, a useful survey of what can be done in recognition of line patterns. Incidentally, it is possible to read this book adequately without bothering to learn EASEy. Programs are summarised in plain language as well as appearing in full detail.

M. J. B. Duff (London)