5. Conclusion

In this paper we first introduced a new programming concept to solve exclusion problems and second, based on this concept, we provided a method to detect deadlocks. The concept of excluding regions may be viewed as a counterpart to the concept of conditional critical regions. The one concept should be used for exclusion problems, whereas the other is intended for solving synchronisation problems. Having an appropriate programming concept for exclusion problems greatly supports clearly structured solutions of these problems, as can be seen from the given examples. The concept of excluding regions may be extended to a concept similar to the monitor concept (Hoare, 1974), thereby delivering a counterpart to the monitor concept. The validation method may also be transferred to such an extended concept.

The major advantage of the validation method used is that, based on the syntactical notation of the excluding regions, it can be carried out automatically at compile time. A crucial point is the rate at which the number of states increases with the number of regions and the number of processes involved. But investigation of practical examples suggests that the number of states and hence the amount of storage and computation time can be kept within reasonable bounds. From the investigated examples, we learn that the use of appropriate synchronisation primitives greatly reduces the number of states necessary.

References

Brinch-Hansen, P. (1973). *Operating System Principles*, Prentice-Hall, Englewood Cliffs, N.J.

Belpaire, G., and Wilmotte, P. (1973). A Semantic Approach to the Theory of Parallel Processes, in A. Guenther *et al.*, (eds.), *International Computing Symposium* 1973, Davos.

Courtois, P. J., Heymans, F., and Parnas, D. L. (1971). Concurrent Control with 'Readers' and 'Writers', *CACM*, Vol. 14, No. 10, pp. 667-668.

Gilbert, P., and Chandler, W. J. (1972). Interference Between Communicating Parallel Processes, *CACM*, Vol. 15, No. 6, pp. 427-437.

Habermann, A. N. (1972). Synchronisation of Communicating Processes, *CACM*, Vol. 15, No. 3, pp. 171-176.

Hoare, C. A. R., and McKeag, R. M. (1973). *Nucleus of a Structured Multiprogramming System*, Technical Report of the Department of Computer Science, The Queen's University of Belfast.

Hoare, C. A. R. (1974). Monitors: an Operating System Structuring Concept, *CACM*, Vol. 17, No. 10, pp. 549-557.

Lister, A. M. (1974). Validation of Systems of Parallel Processes, *The Computer Journal*, Vol. 17, No. 2, pp. 148-151.

Book reviews

Methods of Optimization, by G. R. Walsh; 1975; 200 pages. (*John Wiley*, £3·75)

This text is based on a course of lectures given to third year mathematics undergraduates in the University of York. It is offered as a companion volume to the author's *An Introduction to Linear Programming*, published by Holt, Rinehard and Winston in 1971. The six chapters are entitled: Introduction, Nonlinear programming, Search methods for unconstrained optimization, Gradient methods for unconstrained optimization, Constrained optimization and Dynamic programming. There is a good set of exercises for the student at the end of each chapter (for which answers are provided) and 80 references (dating from 1972 and earlier). The topics covered are explained clearly and comprehensively, though some important approaches based on linear programming: integer programming, separable programming and the various versions of the reduced gradient method are not mentioned. As the author says, wherever possible, an historical approach has been used, for it is interesting to see why, as well as how, certain methods were developed. This leads at times to a less illuminating exposition than is possible in the light of later experience. For example the discussions of Lagrange multipliers for equality constraints in Chapter 1 and for inequality constraints in Chapter 2 are not integrated as closely as they might be.

E. M. L. Beale (Milton Keynes)

Apology

The review of *Revised report on the algorithmic language ALGOL 68*, edited by A. van Wijngaarden *et al.*, which appeared in the February issue of *The Computer Journal* on page 36, was prepared by Anne Rogers of Bath. We apologise that her name was omitted from the bottom of the review.

Computer Programming for Spatial Problems, by E. Bruce MacDougall, 1976; 160 pages. *Edward Arnold*, £5·50

This book is the text for a course in programming to produce line printer maps. It reveals its origin in lecture notes by starting with Babbage and the binary system and finishing with a thumbnail sketch of languages other than FORTRAN and notes on how to use an IBM card punch. In between there are two chapters forming an introduction to FORTRAN and two more on techniques for mapping qualitative and quantitative data in isarithmic, choropleth and proximal forms. There follows a section on the use of plotters, and a further chapter on FORTRAN programming techniques, together with a glossary and an index; most chapters include exercises for the reader.

With a few minor exceptions the subject matter is lucidly described and illustrated and the book is well produced. There is some confusion over the effect a virtual operating system has on FORTRAN efficiency and it is a pity that the existence of 10 lines per inch (square grid) printers is not mentioned. The claim on the back cover of the book that data sampling and statistical analysis methods are presented is a somewhat unusual use of the English language, just as the punched card on page 23 appears to have been produced by an unusual card punch.

If students of geography, planning and the like are to learn FORTRAN then this book, with its emphasis on relevant applications, would be an excellent text. If these same students wish simply to generate and use line printer maps, their time would be better spent learning to use one of the many existing packages.

D. T. Muxworthy (Edinburgh)