augment these by macros to map the appropriate inner syntax. This is, in fact, a typical example of the use of a macro processor to implement an extensible language.

Conclusion

The evidence provided by this work shows that the division of a programming language into outer and inner syntax can be done in a natural and successful way. This is reinforced by the views of Newey, Poole and Waite (1972), who, in the conclusions of their paper, indicate that their thoughts are moving in this direction.

One of the problems in designing programming languages is to reconcile generality with compactness. It is clear that, at least in one application, the problem can be solved viably and practically by designing an extensible language.

Corrections to Wilkes' paper

Professor Wilkes has asked me to take this opportunity of pointing out two corrections that should be made to his paper. Firstly, in the second paragraph the italicised words 'outer' and 'inner' should be reversed. Secondly, in the third paragraph the symbol p_i is omitted from the list of defined symbols.

References

BROWN, P. J. (1972). Levels of language for portable software, *Comm. ACM*. Vol. 15, pp. 1059-1062.

NEWEY, M. C., POOLE, P. C., and WAITE, W. M. (1972). Abstract machine modelling to produce portable software—a review and evaluation, *Software P.E.*, Vol. 2, pp. 107-136.

WILKES, M. V. (1968). The outer and inner syntax of a programming language, *The Computer Journal*, Vol. 11, No. 3, pp. 260-263.

Book reviews

Queueing theory in OR, by E. Page, 1972; 187 pages. (*Butterworths*, £3·60)

This book is the third to be published in the Butterworth Operational Research Series and maintains the high standard already set by its predecessors on Graphs and Networks, and on Scientific Inventory Control. The publication of a specialist text on Queues is timely in view of the sophistication of currently-available software for queueing situations, and of the current need for a coherent account of recent developments in the mathematical treatment of the more complex situations.

The author has nevertheless selected an elementary introduction to the subject which will be readily interpreted by the student as well as inspiring interest in the subject. Commencing with an analysis of industrial queues, and extraction of the elements of typical queueing situations, the author develops the various forms of statistical distribution encountered in practice. The theory of simple queues leads naturally to the multi-server queue, and queues with arrivals dependent on queue size. The final chapter introduces priority queues in which queue disciplines other than first-come, first-served are considered.

The book is fully equipped with Tables for Average Waiting Times, and probabilities of customers receiving immediate service, under the various queueing constraints considered in the text. A useful glossary conforms to current examination conventions and industrial practice. The list of References provides ready access to supplementary literature, much of which is already internationally recognised as authoritative source material. This is expected in a book which itself succeeds in attaining a similar standard of expert presentation.

The student is provided with problems dispersed throughout the text, to which a set of solutions at the end of the book applies. Examples of typical problems and methods of attack are frequently included within the theoretical sections, and renders the book a more convenient medium for selecting appropriate methods for queuing problems encountered in day-to-day business activities. The Index is detailed, but the Contents list could well be expanded in a future edition to reveal the location of the rich theoretical content of each Chapter.

The standard of printing and binding will ensure that the book will withstand the frequent usage to which it will inevitably be subjected by practitioners in all fields of activity in which queueing situations occur. The price is very modest for a text of such high calibre.

W. PAKER (Woking)

More Fortran Programs for Economists, by Lucy J. Slater; 1972. (*Cambridge University Press*, £2·20)

This volume follows the format of Miss Slater's earlier volume—*Fortran Programs for Economists*—and gives for seven selected topics the mathematical description, the flow chart and a FORTRAN program.

Five of the chapters collect together routine work in the field and require little further comment other than to say it is convenient to find them collected in one volume. The five topics covered are:

An inversion routine for a symmetrical matrix with only the lower triangle stored in a one dimensional array. The method is modified Gauss-Jordan. Unfortunately some of the test results are incorrect.

The calculation of an optimum convex surface is designed to find the set of hyperplanes in n-dimensions which envelop a set of points with all co-ordinates positive.

The RAS technique is used to transform a matrix by linear operations so that rows and columns will sum to two arbitrary vectors.

Probit Analysis is discussed in both the quantile and quantitative cases.

The final chapter of the book discusses the model for a single linear expenditure system.

Two earlier chapters require more detailed comment:

An improved method of regression analysis discusses the wellknown Efroymson type step-wise regression procedure and points out that this is in fact the simplex type procedure used in linear programming which although computationally fast has the two disadvantages of roundoff error and unnecessary cycling. To get round this latter problem Miss Slater introduces some more recent work which involves recalculation of the significance levels at each stage of the process.

The chapter on Large Sparse Matrices is the least satisfactory of the book. The method used is to store elements of sparse matrices as triplets, containing the row and column position, together with the value of the element. The row and column positions are compressed into 1 word which is then stored together with the data in a real array. The use of a real array rather than a separate integer array to store integer pointers is questionable and inefficient, especially since all unpacking of the two actual pointers from the real location is by successive real subtraction.

This chapter illustrates one reservation the reviewer has regarding flow charts. In this case the program which occupies three pages is far easier to follow than the flowcharts which occupy eight pages.

M. J. GARSIDE (Canterbury)