years old, but its contents are well worth reading now by even a novice TeX user. Like its peers in the "conference proceedings" world it lacks consistency of presentation style and the content (although well planned) cannot avoid being bitty. Broad subject areas are, however, grouped in sequential chapters and each chapter is remarkably concise.

A complete novice would perhaps like a longer preamble to bind the whole work together and place each chapter in context, but having said this individual chapters mostly have a good introduction which give the appropriate "this is for you" or "please try later" messages about content.

Chapters 2 and 3 give quite outstanding background accounts of the history, development and uses of TeX: these should be compulsory reading for any TeX user! Malcolm Clarke and Cathy Boot in their masterful overview, make no excuses for TeX's shortcomings, transforming these into positive benefits (see the last paragraph, p. 34). This (deserved) alchemy is characteristic of most of the papers in the book reflecting both the scope of the authors' vision and their peculiar endearment with TeX itself.

At chapter four the reader first becomes aware how just unfortunate are the omissions of some authors. "TeX Device Drivers Today", a topic which I would have been very interested in at the time, is merely six lines of abstract. This happens for six other chapters, some of which are equally frustrating.

Supplementing the basic TeX program with interfaces and other facilities occupies most of the rest of the book. In particular there are a couple of "front-end" attachments to TeX discussed in Chapters 14 and 15 which give a very interesting perspective for users who are concerned about the (acknowledged) steep learning curve required by novices.

Chapter 16 gives a fascinating and useful account of TeX in a university environment and the problems encountered. A similar account of a working installation is given by Gerlinde Petersen from GESYCOM in Chapter 9. Both stories are useful illustrations of TeX at work in large installations and how its applications are developing.

Foreign language TeX is tackled with varying degrees of success and labour in Chapters 18–20. Source code is given in the numerous examples in these chapters which suggest that TeX is not too easy to use for some foreign characters. These chapters are not for easy to master for the newly initiated, but provide useful reference material.

One of my biggest concerns with TeX has been with its graphics handling facilities. Graphics, lines, boxes and diagrams are covered very neatly in Chapters 21–25. They are rather heavy going but include source code and end-result examples.

The book concludes with some complex chapters on TeX and chemistry and a thorough index.

My main concern is that being 4 years old much of the information in this book has been superceded by the TeX community which is, after all, extremely active and dynamic. Only Chapters 2 and (especially) 3 provide full future-proofing in this sense as they deal with TeX as a whole rather than specific topics.

At the incredible price £44.95 I am not sure if even the most ardent TeX user is going to be able to afford this book anyway!

Dick Grune and Ceriel J. H. Jacobs
Parsing Techniques: A Practical Guide. Simon & Schuster International Group. 13–651431–6. £36.00

The subject of parsing techniques provides one of the success stories of computer science. Not only does it have wide application in computing (compiling, database interfaces, AI, etc.) but also is used in other diverse disciplines such as linguistics, chemistry, music, mathematics, geology and informatics. Its strength is that it has a well understood, formal mathematical basis. Early texts establishing this theory, by authors such as Hoperoft, Ullman and Aho, have become classics. More recently, a host of lesser publications have vied for the same market most offering some particular emphasis or special insights.

Most computing degrees cover parsing techniques but the problem generally confronting teachers of this material is the diversity of their audience. Some students will enjoy the mathematical insights which automata and formal language theory can offer; others will find the formalism difficult and daunting and want a totally pragmatic approach. In addition, the audience may not just comprise computer scientists. Whilst the computer science student can be expected to have a reasonably sophisticated programming ability and some in-depth knowledge of programming language syntax and semantics, data structures, etc., this is not a reasonable expectation for a music or linguistic student.

The authors of "*Parsing Techniques: A Practical Guide*" claim their book is written for such diverse audiences, the phrase they use being "for anyone who has parsing to do". Nevertheless, the reader will require reasonable programming ability and some mathematical skills. Providing they have this and give the book time and enthusiasm, they can gain much. The book is not a daunting tome but neither is it just a gentle introductory text. The important algorithms are presented with care and in detail. A very impressive, annotated bibliography running to just under 50 pages at the end of the book provides material of interest even to the specialist.

As well as the usual, classic, Chomsky hierarchy of languages and automata and associated parsing techniques, the book covers, in detail, the important parsing techniques for deterministic languages, i.e. LL, LR, SLR and LALR. There is a necessary emphasis on the use of toolkits and a useful chapter on some error handling methods. The general parsing techniques of Unger, Earley and Tomita are described and compared. There is also some brief mention of attribute grammars and two-level grammars, together with applications.

Dick Grune is a well known and respected Dutch computer scientist. Both he and his co-author, Cereil Jacobs, work at Vrije University in Amsterdam. They write clear, unambigious prose which is a pleasure to read. This text, like many others on the same subject, is fairly densely printed and is packed with illustrations and tables. The examples have been carefully chosen to illustrate the techniques being presented. There is a useful index and, as stated above, an excellent bibliography. There are very few misprints.

I am really most impressed with this book. I think it is an exceptionally good modern text on the subject and most certainly will be recommending it to my students.

V. J. RAYWARD-SMITH
Norwich

C. R. SNOW
Concurrent Programming. Cambridge University Press. 0-521-32796- -2, 0-521-33993-6. £30 hardback, £13.95 p/b

This is an introductory textbook on concurrent programming, aimed at undergraduates. The study of concurrent programming has often been hidden away inside courses on operating systems, which is unfortunate since students are more likely to encounter a need for concurrent programming techniques within applications programs; this is increasingly true as parallel computers become more commonplace. This new book is therefore a welcome addition to the relatively small number of books that deal with concurrent programming in its own right. Other recent titles that come to mind are Ben-Ari's *Principles of Concurrent and Distributed Programming* (1990) and *Concurrent Programming: Principles and Practice* by Andrews (1991).

The order of material in the first five chapters is fairly standard. The first two chapters introduce the basic concepts of concurrency and process invocation. Chapter 3 discusses interprocess communication by shared data, first using busy waiting and then semaphores; this chapter includes the ubiquitous mutual exclusion algorithms by Dekker and Peterson, and the bounded buffer example. Chapter 4 introduces higher-level shared-data facilities such as monitors, illustrated by a few classic examples: the bounded buffer, readers and writers, and disk head scheduling. The relatively short Chapter 5 deals with communication by message passing. A minor complaint here is that the description of some facilities is perhaps too closely identified with their use in a particular language, e.g. it is hard to distinguish the general properties of a channel from those of an Occam channel.

Chapter 6 is, in my view, the least satisfactory part of the book, which does not justify the 60 pages that it occupies (just over a quarter of the book's length). It describes, in some detail, the concurrent programming features of several languages, namely Concurrent Pascal, Concurrent Euclid, Mesa, Path Pascal, Ada, Pascal-M and Occam. The author claims that this is "a representative selection of languages available at the present time", but many of them are neither modern nor widely available. In particular, the description seems somewhat repetitive, partly because of the similarity between several of these languages and partly because all of them have already been introduced in earlier chapters.

The final chapter describes the implementation of a concurrency kernel in Pascal. This is reminiscent of the "Implementation Kit" in the appendix of Ben-Ari's 1982 book *Principles of Concurrent Programming*, except that it seems to be intended for pedagogical purposes rather than for actual implementation by the reader.

In summary, this is a readable and well written book, though rather dry in places. The only error that I noticed was a misspelling of Eratosthenes, a common fault! It is pitched at a more elementary level than the books mentioned above, especially the one by Andrews, which is far more detailed. In comparison with these other books, there is more detail of specific languages and less emphasis on concurrent algorithms or applications: relatively few examples are presented. This, I feel, is the book's major weakness: the student is likely to end up with a good knowledge of some of the various concurrent programming paradigms and languages but less idea of how to apply them to actual problems.

S. GREGORY
Bristol

D. DIAPER and N. HAMMOND (Eds)
People and Computers VI (Proceedings of the HCI '91 Conference). Cambridge University Press. 0-521-41694-9. £40.00

"People and Computers VI" is a collection of the 30 papers presented in the British Computer Society HCI '91 Conference. Consequently, the selected and refereed texts are of a consistently high standard and are drawn from across the field of HCI, ranging from abstract theoretical discussions to detailed descriptions of particular systems. Paper contributors are multinational, though predominantly British. The Editor's introduction gives an overview of the volume, as well as noting substantial changes in emphasis since the HCI '87 conference as this field develops and matures. The papers are divided into sub topics, such as HCI Frameworks, Graphical Interaction and Task Analysis, with two or three papers in each section.

The first of the two invited conference papers, by Brad Myers of Carnegie Mellon University, describes demon-