

numbers of vertices. It can be seen that Method GS was fastest for all the configurations considered. It can also be seen from Fig. 7 that the time required by Method GS seems to be proportional to the number of points considered, thus giving it an advantage over the other methods. The two best methods, GS and E, are further compared in Fig. 7. It should be pointed out that all the times refer to our own implementations of the various algorithms, except for method E where the program

CONVEX described by Eddy (1977) was used.

References

- BARNETT, V. (1976). The ordering of multivariate data (with discussion), *J. Roy. Statist. Soc. (A)*, Vol. 139, pp. 318-354.
- BENTLEY, J. L., and SHAMOS, M. I. (1977). Divide and conquer for linear expected time, Carnegie-Mellon University Computer Science Technical Report.
- EDDY, W. F. (1977). A new convex hull algorithm for planar sets, *ACM Transactions on Mathematical Software*, Vol. 3, pp. 398-403 and 411-412.
- GRAHAM, R. L. (1972). An efficient algorithm for determining the convex hull of a finite planar set, *Information Processing Letters*, Vol. 1, pp. 132-133.
- JARVIS, R. A. (1973). On the identification of the convex hull of a finite planar set, *Information Processing Letters*, Vol. 2, pp. 18-21.
- PREPARATA, F. P., and HONG, S. J. (1977). Convex hulls of finite sets of points in two and three dimensions, *CACM*, Vol. 20, pp. 87-93.

Book reviews

Natural Language Communication with Computers, edited by L. Bolc, 1978; 292 pages. (Springer-Verlag, \$13.50)

Mainly, the book contains descriptions of systems which are capable of analysing, and transforming into internal code, classes of natural language sentences. The systems vary according to their purpose (application) and the extent of analysis performed (morphological/syntactical/semantical). Chapter 1 describes an ambitious system providing for the building and subsequent querying of a knowledge structure, including the possibility of temporal changes. Natural language sentences are parsed by a so-called feature grammar (a two-level extension of a Chomsky grammar), and coded in predicate logic with extra operators which reflect the 'state-of-the-world'. Chapter 2 is an account of a less general but perhaps more streamlined system designed for the use with, for example, a pharmacological data base. Queries are translated (again, by a feature grammar) into logical expressions (set language). The system described in Chapter 3 is designed for the monitoring of water pollution. Input statements or queries are analysed by an upgraded ATN (Augmented Transition Network) and translated into formulae. The book also includes a very detailed tutorial on ATNs (Chapter 5). Chapter 6 outlines a project whose aim is to analyse syntactically a subset of Polish, using the programming language PROLOG which, together with its formal semantics (or rather, a second approximation of that), is described in Chapter 4. Two sophisticated PROLOG programs are also given in Chapter 4, a complete compiler for a subset of ALGOL and an 'intelligent' interactive system.

In more than one respect, the book is less homogeneous than its title suggests; the subject inevitably attracts people with varying interests. On the whole I expect Chapters 1, 3, 5 and 6 to appeal to the linguist more than to the computer scientist, and vice versa for 2 and 4. In places, Chapter 5 reads like 'a linguist's first course in formal methods', whereas Chapter 4 is almost an anti-tutorial and 1 is also difficult to read. Although, in my opinion, the book does justice to its declared aim of 'providing information on the present state of the research in the area', there is only one contribution, namely Chapter 2, by Kraegeloh and Lockemann, which managed to convey to me quite clearly what has been achieved and, importantly, what has been left out. I regret that nowhere is the problem of what can or cannot be achieved discussed in detail. While the tenor of the contributions points to the recognition that 'syntax' and 'semantics' of natural languages tend to be inextricably interwoven, I also find traces of the opinion that the two can be considered separately. This touches the question of whether or not natural languages, or non-trivial subsets thereof, are suitable as machine languages. I think that a decision has to be made whether one is content with leaving

natural languages to communication amongst humans (in which case the title of the book is a misnomer), or one strives for an improved COBOL, to put it in extreme terms (in which case the title is an over-estimation). The very notion 'natural' clashes with the term 'computer' which denotes, after all, an artificial device. It is a discussion of such issues which I most dearly miss, not being an expert in 'artificial intelligence'; it is also unfortunate that the responsibility of the editor did not extend to cover these problems and, thereby, to put the set of contributions into a more specific framework. For these reasons I can recommend the book in the first place only to those specialising in the area and wishing to find out what is happening 'elsewhere'.

E. BEST (Newcastle)

Portability of Numerical Software, Workshop, Oak Brook, Illinois, 1976. Edited by W. Cowell, Lecture Notes in Computer Science Vol. 8, 1977. 539 pages. (Springer-Verlag, \$18.30)

This volume should be compulsory reading for all members of ANSI X3J3, the Committee responsible for FORTRAN 77. Nonetheless, despite the considerable criticism of FORTRAN within this book, the overall feeling emanating from the papers is one of gratitude for its existence. Indeed the low point in the book comes with a predictable series of bleats in favour of ALGOL 68, but as L. M. Delves states in his paper 'ALGOL 68 compiler writers would give the biggest boost to the language if they could let the user call FORTRAN routines from his ALGOL 68 program'.

The book is a large volume and its very nature inevitably means that there is a fair amount of repetition, but the editors have found an extremely natural order for the papers and added some useful cohesive comment. The book gets off to a fairly slow start with a discussion of the restrictions created by the physical limitations of computers, in particular the word length. It starts to come to life with the papers from those responsible for successful general purpose libraries, in particular those concerning PFORT and PORT from members of Bell Telephone Laboratories. It contains an excellent paper from B. T. Smith on 'FORTRAN poisoning' and a valuable paper by J. M. Boyle surveying contemporary computerised source management systems. The NAG project emerges with considerable credit, and the final paper in the book contains, under a section headed 'Overflow control', a classic example of the kind of problem obstructing the path to program portability.

There is just one sour note to sound. The papers themselves are beautifully laid out. What a pity the examples of coding within them do not separate code and comment as well as the typist has separated examples and text.

D. L. FISHER (Leicester)