11. B. Liskov, Distributed programming in Argus. *Comm. ACM* **31** (3), pp. 300–312 (1988).
12. C. Low, A shared, persistent object store. *Proceedings of the Second European Conference on Object-Oriented Programming, Oslo*, pp. 390–408 (1988).
13. E. Miranda, BrouHaHa – a portable Smalltalk Interpreter. *Proceedings OOPSLA '87.*
14. J. R. Olson, G. M. Olson, L. A. Mack and P. Wellner, Concurrent editing: the group's interface. *Proceedings of a Conference on Human–Computer Interaction, INTERACT '90, Cambridge, England*, pp. 835–840 (1990).
15. S. K. Shrivastava, G. Dixon, G. D. Parrington, F. Hedayati, S. Wheater and M. Little, The design and implementation of Arjuna. *Proceedings of the Third Conference on Object-Oriented Programming, Nottingham* (1989).
16. D. Steel, Distributed object-oriented programming: mechanism and experience. *Proceedings, Tools USA 91, Santa Barbara.*
17. J. G. Steiner, C. Neuman and J. I. Scheller, *Kerberos: An Authentication Service for Open Network Systems.* Project Athena Report, MIT (1988).
18. D. Thomas and K. Johnson, Group Object-Oriented Programming: Orwell – A Configuration Management System for Team Programming, Carleton University, Canada. *Proceedings, OOPSLA '88.*

Book Review

VALERIE ILLINGWORTH (Editor), *Dictionary of Computing* (third edition), Oxford University Press. £6.99. ISBN 0-19-286131-X.

Not even a reviewer could be expected to read a dictionary straight through, so a review has to be derived from browsing. In my browsing I concentrated on two areas: (i) terms that are commonly misused; (ii) terms that I feel unsure of and would like to understand better. While looking up such items my eye, inevitably, kept finding other entries of interest, so I would look at them on the way (and sometimes forget what I was originally searching for). The result of my browsing is remarkably favourable and I congratulate the general editor, two consultant editors, 45 contributors and the publishers on a splendid product at a reasonable price.

So far I have found only two errors that a proof-reader should have spotted – a remarkable degree of accuracy. These are (i) a missing decimal point in the example of BNF production rules; (ii) 11 words missing from the definition of 'F distribution'.

Many definitions are timeless, but some entries are necessarily out of date when such a publication appears. I doubt, for instance, whether it is still reasonable to say that ICL is 'a wholly British company'.

A few other points are worth mentioning in hoping for an even better next edition. (Incidentally, why do the publishers put the ridiculous words 'New edition' on the front? Of course it is new when it is new, and of course it will soon cease to be so.) The definitions of some of the searching techniques would not have told me to use a binary search when wanting to know where a function crosses zero, but a golden section search (or a *Fibonacci* search) when looking for a maximum or minimum. The impression is given that these are merely alternatives. The definition of BNF correctly uses angle-brackets, not to be confused with less-than and greater-than signs, in its example, but unfortunately less-than and greater-than are used where BNF is employed in some other definitions. In spite of being a British publication, there are some Americanisms, the oddest being where 'colored' is used in the definition of 'coloured book', and I strongly dislike 'named for' instead of 'named after'. The word 'contemporary' is misused in the entry for 'IBM system 360', where 'contemporary IBM mainframes' is presumably meant to mean 'present-day' rather than contemporary with the 360.

I was surprised at finding no entry for 'dangling else', and even 'else' on its own is treated as arising only in connection with decision tables. To find a description of the problem you have to look under 'ambiguous grammar'.

I was disappointed in the definition of *K*. Perhaps I am out of date, but I was certainly taught 30 years ago that *K* had been introduced to computing as a quick simple way of indicating 1024. Except mnemonically it had nothing to do with *k* meaning kilo- which, of course, remained as 1000. Unfortunately *K* has been stolen from us, and misused to mean 1000, by administrators and accountants, but I regret finding a dictionary of computing that merely says

k (or **K**) *Symbols for* kilo-

and then defines kilo- as indicating 1024 when the binary system is used. Their advice to avoid the capital *K* is not followed elsewhere by themselves.

I have enjoyed comparing this dictionary with the glossary at the back of Lord Bowden's classic *Faster than Thought* (1953). That glossary contained only 55 terms, compared with over 4000 in this dictionary, but 40% of them have survived as basically the same term with the same meaning. One term, microprogramming, appears in both lists but with disjoint meanings. I am sorry that 'Hartree constant' has gone ('The time which is expected to elapse before a particular electronic computing machine is finished and working'); perhaps it does not now apply so strongly to hardware, but it is surely still a relevant concept in terms of when computers will all understand natural language and accept voice input. This has been 'about 5 years' as long as I can remember. One rather sad note comes from Lord Bowden's borrowing from Dr Johnson's definition of 'lexicographer' in defining 'programmer' as 'a harmless drudge' – the present dictionary has to have an entry for 'hacker' that was absent then.

My complaints are few indeed. I strongly recommend this excellent dictionary. There are, after all, not many dictionaries where one could find such intriguing entries as:

mother *Another name for* parent, rarely used.
worm *See* virus.

I. D. HILL