

Bauer - Goos, Informatik, Zweiter Teil.

The second volume of the introductory survey of computing science has appeared within a remarkably short period of time after the publication of the first volume, that contained the first four chapters.

The second volume starts with the fifth chapter. This chapter (48 pages) is devoted to dynamic storage allocation and restricts itself -very wisely at this stage- to a single level store. The main part of this chapter deals with the accomodation of variables with nicely nested life times, it explains the stack, the concept of blockheight and the notion of a display; it does so by showing a specific implementation of these concepts in rather great detail. Your reviewer is not quite happy with this way of presentation: the amount of detail makes the chapter quite hard to read and to my taste it is insufficiently stressed that the implementation shown is just one out of many possibilities. Finally the problem of garbage collection -a poor term for which I think the German translation "Speicherbereinigung" a great improvement- is dealt with in only the last seven pages. In view of the comparatively meagre treatment of the allocation problems when life times are no longer nicely nested, one could have the feeling that in this respect the fifth chapter is somewhat unbalanced.

The sixth chapter (56 pages) tries to do the impossible: to give a concise and elegant treatment of a number of very extensive and sometimes very messy subjects. It starts with a dense but adequate survey of all sorts of peripheral equipment and storage media (8 pages). Channels, I/O commands and interrupts are introduced in the next 4 (!) pages; it is the reader's first confrontation with parallellism and a form of non-determinism in the sense that interrupts occur at unpredictable and irreproducible moments. After having told that this could lead to irreproducible results, the authors think it sufficient to remark "Derartige Schwierigkeiten lassen sich durch korrektes Programmieren vermeiden." Although the first half of this section mentions the possibility of a multi-processor configuration, its second half -introducing the interrupt- has clearly been written with a single processor in mind. Is the previous section misleading, the next section "Functional Characterization of the I/O Processes" (17 pages) is just obscure. It does not state the problems to be solved nor the goals aimed at and the result is

a long enumeration of definitions (in the style of "We can do this, we can introduce that...") while on the whole it fails to explain their purpose. The text is not always too carefully written: "die Angabe des ersten nicht mehr geschriebenen Zeichens" (pg.57) reminds me of the classical advertisement for self-checking tape equipment with the claim "No undetected error has ever been found.". The final part of this section introduces so-called "formats" at great length. I have always been quite happy without them and until someone has convinced me of the contrary, I shall continue to regard the introduction of formats in programming languages as a mistake, as an ad hoc concept that has done more to obscure than to clarify. The authors have included the concept unchallenged. The section "Functional Characterization of the I/O Processes" leans heavily on the baroque apparatus of ALGOL 68, a circumstance that could be responsible for its unsatisfying nature. The next section (6 pages) deals with Data Management; it makes the reader encounter the usual terminology but leaves him with a vague picture of a rather elusive subject. The final section (20 pages) of the sixth chapter deals with Operating Systems. It explains multiprogramming in a very satisfactory way. The very condense treatment of synchronisation problems ends with an example of such a twisted nature that I would never recommend it, but has the merit of mentioning the prevention of deadlock as a difficult problem. The section on main storage management is quite clear and has the merit to point out that in a widely used type of machine (guess which!) the accessibility of the base registers prevents refined management of main store. It mentions different systems and system goals (batch processing, time sharing etc.). The final two pages on compilers, loaders, assemblers and linkage editors are not enough: the underlying assumptions or prior decisions that call for this organisation are left undiscussed.

The seventh chapter "Automata and Formal Languages" (45 pages) is of quite a different nature: in contrast to the previous chapter it is full of script capital letters, so beloved by the Mathematician. Besides that the chapter contains 35 pictures and the authors deserve a compliment of their expert exploitation of graphical means for the purpose of explanation. The pictures range from transition diagrams for finite state machines to parsing trees. It gives a clear example of a syntactic ambiguity, classifies context-free grammars and discusses the parsing problem in a quite attractive way.

The eight chapter "Syntactical and Semantical Definition of Algorithmic

Languages" (30 pages) has outstanding features. For three different algorithmic languages it shows a facsimile of a page of its defining report. It has a short section on axiomatic definition of semantics in relation to the possibility of giving correctness proofs for programs. It contains the remark, that it is practically impossible to establish with the aid of testruns the absence of program bugs; in spite of the overwhelming importance of this observation, the remark, however, is placed in a footnote. After mentioning syntax directed compilers, a few algorithmic languages are compared.

An appendix of fifteen pages gives a survey of the history of automatic computing that is a pleasure to read. Personally I think it very valuable that such a historical survey is included in a book that will be placed in the hands of many students.

Reviewing the two volumes as a whole, we can state without reservation that it gives a fair survey of the ^{field}~~field~~ of computing science. The price paid for this encyclopaedic aim is twofold. Firstly, many areas of the ^{field}~~field~~ could not be covered, they could be touched only. Secondly, the book faithfully mirrors the characteristic lack of coherence between all sorts of computer connected mental activities, but for this incoherence not the book, but the still immature science of computing should be blamed.

Eindhoven, March 1972

prof.dr.Edsger W.Dijkstra