> inner languages can be 'plugged'. The task of implementing a given inner language is made more self-contained, although perhaps at the cost of a small amount of redundancy, if any data types that have been defined in the outer language are ignored.

If similar data types exist in both outer and inner languages, the nomenclature must, of course, be chosen to keep them separate. One could, for example, use **counter** in the outer syntax, and **integer** in the inner syntax.

The remarks made above about the **for** statement apply also to the **switch,** which also needs integers for its implementation.

We have now obtained a very clean (although not necessarily unique) cut between the outer and inner syntax. It is not, however, possible for one of the variables in the **for** statement—the range for example—to be changed by the program. We would, therefore, need to provide a transfer function which would enable an integer belonging to the outer syntax to be set equal to an integer belonging to the inner syntax. If the same data structure is used in both the inner syntax and the outer syntax, this transfer function is, of course, the identity operator. In other cases, a subroutine would have to be specially written when the inner syntax was being implemented. Naturally, in order to write this subroutine, a knowledge of the data representation used for storing integers in the outer language would be needed.

Conclusion

The primary purpose of this note will have been served if it has been made clear that there are two influences controlling the form of a programming language; these influences operate from opposite ends. There are features of the language controlled entirely by the predilections of the programmer, ranging from genuine minimum requirements as to what he is able to say, to fads and fancies. There are other features controlled by the nature of the data types and representations that the language is designed to manipulate, and the storage allocation scheme that is used to accommodate them; in this class, the most fundamental concern data types and the operations to be performed on the data.

It is not to be expected that any existing language will separate cleanly into the two parts. It has, however, been shown that, in the case of ALGOL, the separation can be effected without great cost. No doubt it could be effected in ways other than the one given here, since there is a certain amount of arbitrariness about what is regarded as belonging to the outer language and what is regarded as belonging to the inner language. In whatever way the separation is done, we shall be left with an outer language of great manipulative power, to which can be fitted a variety of inner languages, each designed to manipulate a different set of data structures. It is suggested that along these lines might be found a solution to the problem of enabling a sufficiently skilled programmer to extend a language by adding new data types while preserving intact the external form of the language.

Book Review

The Information Centre, by Morton F. Meltzer, 1968; 160 pages. (London: *Bailey Bros. and Swinfen Ltd.*, 85s. 6d.)

This well-written book presents to top management the case for establishing a comprehensive information centre to deal not only with purely scientific or technical information relevant to the business but one which also acquires, files and distributes, as needed, all relevant commercial information as well.

In presenting his case the author finds himself in a difficulty. Because he demands the critical attention of busy senior executives he cannot afford to be lengthy. But he also needs to display all the specialist techniques that the manager of a technical information centre can command (thereby also justifying the salary he claims), but convincing demonstrations of specialist skills rapidly get wordy or technically intricate, and therefore unreadable by executives. So the author has to compromise. The result is a readable report or essay of 126 pages (rather expensive at 8d. per page) which, to the uninitiated, teasingly mentions rather than explains many facets of information work.

So, in spite of its brevity, the book is comprehensive. The author helpfully admits that information services cost money and boldly produces some realistic estimates of the initial and running costs of an information department. His weakest chapter is on 'Determining the return on investment' on an information centre. It is unconvincing because no one has yet been able to quantify the return on information *per se*. Information is only one of many elements in the complex of human activity engaged in productive enterprises and is peculiarly resistant to isolation and objective quantification.

The greatest danger that faces the author is that the book could succeed too well. The executive who reads it might discern that the role assigned by the author to the manager of the technical information centre he describes would make the manager a mighty power in the business. The executive might decide that this book could best be regarded as an outline systems analysis of an information service which would more safely be provided by upgrading, not the manager of the technical information centre, but the firm's computer.

Foreword, acknowledgements, notes, glossary, selected bibliography and index all help to provide a grand total of 160 pages.

B. C. Brookes (London)