Dijkstra, E. W. (1967). Recursive Programming, in *Programming Systems and Languages*, Saul Rosen (ed.), McGraw-Hill, New York.

Griswold, R. E. (1976). The SL5 Programming Language and Its Use for Goal-Directed Programming, *Proc. of the Fifth Texas Conf. on Computing Systems*, pp. 1-5.

Griswold, R. E. *et al.* (1968). *The SNOBOL4 Programming Language*, second edition, Prentice-Hall, Englewood Cliffs, New Jersey

Griswold, R. E. and Hanson, D. R. (1977). An Overview of SL5, *SIGPLAN Notices*, Vol. 12 No. 4, pp. 40-50.

Hanson, D. R. (1976a). The Syntax and Semantics of SL5, SL5 Project Document S5LD2b, Dept. of Computer Science, The University of Arizona, Tucson, Arizona.

Hanson, D. R. (1976b). Variable Associations in SNOBOL4, *Software—Practice and Experience*, Vol. 6 No. 2, pp. 245-254.

Hanson, D. R. (1976c). *Procedure-Based Linguistic Mechanisms in Programming Languages*, Ph.D. Dissertation, Dept. of Computer Science, The University of Arizona, Tucson, Arizona.

Hanson, D. R. (1976d). A Procedure Mechanism for Backtrack Programming, *Proc. of the ACM Annual Conference*, pp. 401-405.

Hanson, D. R. and Griswold, R. E. (1978). The SL5 Procedure Mechanism, *CACM*, Vol. 21, to appear.

Hoare, C. A. R. (1972). Notes on Data Structuring, *Structured Programming*, Academic Press, London, pp. 127-130.

Kernighan, B. W. and Plaugher, P. J. (1976). *Software Tools*, Addison-Wesley, Reading, Mass.

Knuth, D. E. (1969). *The Art of Computer Programming*, Vol. 2, Seminumerical Algorithms, Addison-Wesley, Reading, Mass., p. 9.

McIlroy, M. D. (1968). Coroutines, Technical Report, Bell Laboratories, Murray Hill, New Jersey.

Ritchie, D. and Thompson, K. (1974). The UNIX Time-Sharing System, *CACM*, Vol. 17 No. 7, pp. 365-375.

van Wijngaarden, A. *et al.* (1976). Revised Report on the Algorithmic Language ALGOL 68, *Acta Informatica*, Vol. 5 No. 1, pp. 1-236.

Wulf, W. A. *et al.* (1971). Bliss: A Language for Systems Programming, *CACM*, Vol. 14 No. 12, pp. 780-790.

Book reviews

Compiler Construction (An Advanced Course), Second edition, edited by F. L. Bauer and J. Eickel, 1976; 638 pages. (*Springer-Verlag*, DM 24)

This book is a reprint of the course notes used in an *Advanced Course in Compiler Construction*, organised in Germany in March 1974 and repeated in March 1975. The course was presented by a number of persons prominent in the academic community, and covered a wide range of subjects. It is extremely difficult to review a book of this nature concisely, since it consists of many lectures of widely differing quality.

McKeeman's introduction to the course is lucid and interesting even though the lecture deals with material which must surely be present in any undergraduate course in language implementation.

A number of other lectures in this book fall into the same class as this. These are: DeRemer's lecture on formalisms and notation, Waite's lectures on the relationship of languages to machines and on assembly and linkage, McKeeman's lectures on symbol table access methods and on programming language design, Horning's lectures entitled 'What the compiler should tell the user' and 'Structuring compiler development', and Griffiths' lecture on run time storage management.

Some lectures give fairly complete and interesting coverage of their subject matter. These are: Waite's on optimisation techniques, Poole's on portable and adaptable compilers—which also contains useful case study material; and Hill's on run time organisation for ALGOL68.

Griffiths' coverage of *LL*(1) grammars and analysers gives enough information about *LL*(1) techniques to enable the reader to construct a parser-generator. Unfortunately, his description of the automatic elimination of left-recursion is confusing, as is his brief introduction to *LL*(*k*) grammars.

Horning's description of *LR* grammars and analysers will also enable the reader to construct a parser-generator based on one of the various refinements of *LR*(1) techniques.

DeRemer's lecture on lexical analysis covers elementary material. His lecture 'Transformational grammars' deals with a useful method of approaching the specification of translators, and advocates a view of language processing—as tree manipulation—which is extremely fruitful.

Koster's lecture on two level grammars is obscure. The material could easily have been presented less formally in about half the space. This reviewer would have preferred such a treatment to be augmented by an indication of the relevance to the compiler writer of two level grammars.

Waite's lecture on code generation is inadequate. Its first section is written in such an abstract style as to be almost incomprehensible to those who are not familiar with the model of code generation which he presents. Furthermore, his terminology changes from paragraph to paragraph—which adds confusion.

Koster's series of lectures on using the CDL compiler compiler demonstrates the unwieldliness of CDL, rather than the utility of compiler compilers. A newcomer to compiler compilers could, after reading these lectures, be forgiven for concluding that such systems are of little more use in the building of compilers than 'ordinary' high level languages. This is far from being the case. It is unfortunate that the latest widely known review paper on automatic compiler generation—that of Feldman and Gries—dates back to 1968. Unfortunately, Griffiths' introductory lecture on compiler compilers doesn't remedy this deficiency in the literature.

In summary, this book does not amount to 'An Advanced Course in Compiler Construction', despite the excellence of a few of the lectures in it.

B. A. Sufrin (Colchester)

Reference

Feldman and Gries, (1968). Translator Writing Systems, *CACM*, Vol 11 No 2

Online Review, Vol. 1 No. 1, published by *Learned Information*, Oxford, $45 a year

In the context of this journal the term online describes the facility whereby data bases may be accessed remotely via terminals for a tariff.

In the UK these systems are rare; however in the United States many independent information services are in operation. A major use of these systems is by researchers interested in articles published by others on related topics. For this reason it is likely that usage will continue to be mainly by libraries, research institutes and major organisations.

Much emphasis is given to the number of items available from various data bases, frequency of update and the ease with which the required items may be accessed. One paper attempts to identify which features of a certain interrogation language are the most useful. There is some justification for hoping that as these systems spread only a handful of interrogation languages remain, allowing users easily to use a number of different information services.

A journal dedicated to online systems must at present be addressed to a limited audience. However several times contributors emphasised the need for just such a journal. It will be interesting to see how it and information services develop.

E. Gildersleeves (Norwich)