(1) a terminal symbol in quotes means that any identifier is acceptable at that place;
(2) the symbol $ at the end of a word (e.g. circuit$) means that either singular or plural form of the word is allowed;
(3) ∅ as an alternative for a syntactic class means that class need not be present.

⟨*command*⟩ ::= ⟨*retrieval command*⟩. | ⟨*other command*⟩.
⟨*retrieval command*⟩ ::= ⟨*request*⟩ ⟨*remember*⟩ ⟨*subject*⟩ ⟨*choice*⟩
⟨*subject*⟩ ::= *all circuits* | *above circuits* | '*dubname*' *circuits* | *circuits* ⟨*namelist*⟩
⟨*namelist*⟩ ::= ⟨*circname*⟩ | ⟨*circname*⟩, ⟨*namelist*⟩
⟨*circname*⟩ ::= '*manufacturer*' '*series no*' '*order no*'
⟨*choice*⟩ ::= *with* ⟨*choice expression*⟩ | ∅
⟨*choice expression*⟩ ::= ⟨*ce*1⟩ | ⟨*ce*1⟩ *or* ⟨*choice expression*⟩
⟨*ce*1⟩ ::= ⟨*choice term*⟩ | ⟨*choice term*⟩ *and* ⟨*ce*1⟩
⟨*choice term*⟩ ::= (⟨*choice expression*⟩) | ⟨*word property*⟩ ⟨*EO*⟩ ⟨*value*⟩ | ⟨*number property*⟩ ⟨*RO*⟩ ⟨*value*⟩ | ⟨*qualifier*⟩ ⟨*number property*⟩
⟨*word property*⟩ ::= *name* | *can type* | *logical function* | . . . etc.
⟨*number property*⟩ ::= *cost* | *stock level* | *propagation delay* | . . . etc.
⟨*EO*⟩ ::= *equal* | *not equal*
⟨*RO*⟩ ::= = | ≠ | < | ⩽ | > | ⩾
⟨*qualifier*⟩ ::= *smallest* | *smallest value of* | *greatest* | *greatest value of*
⟨*value*⟩ ::= ⟨*embedded command*⟩ | ⟨*value syntax*⟩ | *unknown*
⟨*value syntax*⟩ depends on the property being considered
e.g. ⟨*value syntax for cost*⟩ ::= *integer* / *integer* / *integer* | *integer* / *integer*
⟨*request*⟩ ::= *print* ⟨*pplist*⟩ *of* | *select* | ∅
⟨*pplist*⟩ ::= ⟨*pp*⟩ | ⟨*pp*⟩ *also* ⟨*pplist*⟩
⟨*pp*⟩ ::= *total number* | ⟨*word property*⟩ | ⟨*number property*⟩ | ⟨*qualifier*⟩ ⟨*number property*⟩ | *that*
⟨*remember*⟩ ::= *also dub* '*dubname*' | *dub* '*dubname*' | ∅
⟨*embedded command*⟩ ::= ⟨*request*1⟩ ⟨*remember*⟩ ⟨*subject*⟩ ⟨*choice*1⟩
⟨*request*1⟩ ::= ⟨*pp*⟩ *of* | *print* ⟨*pp*⟩ *of*
⟨*choice*1⟩ ::= *with* ⟨*choice term*⟩ | ∅
⟨*other command*⟩ ::= ⟨*forget*⟩ | ⟨*synonyms*⟩ | ⟨*directive*⟩
⟨*forget*⟩ ::= *forget* ⟨*dubnamelist*⟩
⟨*dubnamelist*⟩ ::= '*dubname*' | '*dubname*', ⟨*dubnamelist*⟩
⟨*synonyms*⟩ ::= *define synonyms* ⟨*synonym list*⟩
⟨*synonym list*⟩ ::= '*new*' *for* '*old*' | '*new*' *for* '*old*', ⟨*synonym list*⟩
⟨*directive*⟩ various commands to select input and output channels, and anything else required by a particular implementation.

## References


FELDMAN, J., and GRIES, D. (1968). Translator Writing Systems, *Comm. ACM*, Vol. 11, No. 2, p. 77.
FOSTER, J. M. (1967). Interrogation Languages, *Machine Intelligence* 1, p. 267. Edinburgh and London: Oliver and Boyd.
FOSTER, J. M. (1968). A Syntax Improving Program, *Computer Journal*, Vol. 11, No. 1, p. 31.
FOX, A. J., and EDWARDS, P. W. (1968a). The Selection of Integrated Circuits using RREAC—Language Manual. Ministry of Technology unpublished work.
FOX, A. J., and EDWARDS, P. W. (1968b). The Implementation of a Data Retrieval Scheme on RREAC using Syntax Techniques. Ministry of Technology unpublished work.
LANDIN, P. J. (1964). The Mechanical Evaluation of Expressions, *Computer Journal*, Vol. 6, No. 4, p. 308.


# Book Review

*Advanced Linear-Programming Computing Techniques*, by William Orchard-Hays, 1968; 355 pages. (McGraw-Hill, £5 17s. 0d.)

This is an ideal reference book for those involved in using, writing, or amending programs for linear programming. The application theory, and the detailed interpretation of the results are not examined. One objective of the book is that it will establish a notation and nomenclature to promote easier communication. It is not considered that the notation satisfies this aim.

The introductory chapters define the problem and discuss the simplex methods of solution. Although the book is claimed not to provide a mathematical background, it should be pointed out that it is largely concerned with mathematics and suggested algorithms for the writing of computer programs, and it is therefore advisable that readers are well versed in algebraic techniques. An excellent illustrative example of a 'widget' manufacturer is included, this continues through the book, which shows how these programs can be applied. For those assisting management in the formulation of company models, these sections will be of considerable assistance.

Another example, for the manual solving of simultaneous equations is rather unfortunate in that it makes an easy problem difficult. This is, however, a minor criticism.

The procedures for the ranging of coefficients in the modification, extension, and combination of company models are of particular interest. In certain sections mnemonics come thick and fast, PARROW, PARCOL, PARRHS, etc., as detailed computer programs are investigated. The development of decomposition techniques, including the Dantzig-Wolfe Algorithm, are well documented. The Appendices on Elementary Transformation and the Mathematical Programming System are excellent and there is also a useful index.

In conclusion, the author comments that the techniques developed over the last ten years have automated Linear Programming computational runs; but with the increased power of program systems and algorithms, it is likely that it will again be necessary for the analyst to direct the procedures.

This book can be recommended both for Scientific Computer Programmers and for Operational Research personnel who are involved in the application of linear programming.

G. P. D. MORRIS (Coventry)