Checking a table which has conformed to Pollack's convention is child's play. Completeness and ambiguity are easily checked because of the bifurcation principle upon which the table has been developed.

By the way, Mr. King's limited entry table examples are all mixed entry since the action entries are shown in extended entry format!

Yours faithfully,

D. A. CARRICK

15 Rockfield Avenue
Ware
Hertfordshire
25 November 1969

Reference

KING, P. J. H. (1969). The interpretation of limited entry decision table format and relationships among conditions, *The Computor Journal*, Vol. 12, No. 4, pp. 320–326.

To the Editor
The Computer Journal

Chain processing in PL/1

Sir,

I would like to suggest some features that could be implemented in PL/1 to make list processing easier and less liable to error for programmers not well versed in the field. Although most of the generality of PL/1 list processing is retained by the suggestions below, pointer manipulation provides many facilities not covered here, and the following is not considered as a replacement for pointers.

A unified data type (like LISP lists) and a small number of operating concepts are essential for simplicity. The data type suggested by the PL/1 list processor is the *chain*, a series of *items* (structures, as at present, containing data fields) connected up and down by pointers. In a chain processor, the pointers will not be manipulated directly by the programmer, but only indirectly, through the operations he performs on the chain. Such operations are creating or deleting items in or from the middle or ends of a chain. The data fields of only one item in a chain can be accessed at a time, and this can be thought of as the *current* item. Thus the chain processor requires an operation of moving up and down the chain from the current item, making the item above or below current. Among the data fields is allowed the name of a *subchain* (or names of subchains). If the subchain is the same as the first (i.e. has the same name), a bifurcating tree results. In the case of plural subchains of the same name, we get a polyfurcating tree. If we add to the chain processor a facility for linking a chain to itself or to other chains, rings and other interconnected structures may be built up.

These concepts give very simply a wide range of list processing facilities. Only four basic operations are required, whose names suggest their functions: CREATE_ITEM, DELETE_ITEM, MOVALONG and LINK_CHAIN. In addition, the chain processor would require a declarator DECLARE_CHAIN and two logical functions, END_CHAIN and NULL_CHAIN (to detect the top or bottom item, or when a chain has no items).

Operands for these facilities would include: the name of the chain; ABOVE, BELOW and SUB for CREATE_ITEM (to place a new chain is to be a subchain); UP, DOWN, TO TOP, TO BOTTOM, TO (key) for MOVALONG (where the key is a logical condition on the data fields, to be satisfied by the item sought); and assignments to the data fields of new items.

I believe that the above suggestions are of general interest, and that in practice they would add to the appeal of PL/1 and to the ease of teaching it. Since they have the effect of restricting PL/1 to a more streamlined form (from the user's point of view), they can be implemented experimentally in PL/1, using the macro facilities of the language. Such an experiment is in progress. While this work cannot indicate the efficiency of a chain processor, implemented for example as a part of PL/1, it will give an idea of its usefulness.

Yours faithfully,

T. H. MERRET

Upper Flat
Craigmore
Shore Road
Skelmorlie
Ayrshire
28 November 1969

To the Editor
The Computer Journal

Further comments on a line-thinning scheme

Sir,

In a recent communication Mr. E. S. Deutsch has suggested some modifications to the set of rules given by Rutovitz, the realisation of which yields a line-thinning algorithm (this *Journal*, Vol. 12, p. 412). The complete set of rules now assumes a somewhat formidable appearance, and I would like to present a simpler alternative which I have been using for several months.

The line-thinning scheme which is suggested here examines only edge-points and so requires an edge-following routine which yields a table or set of tables giving points on the edge or edges of the pattern. These points are examined to see if they can be eliminated (set to binary zero in the pattern), a new set of tables defining edges is found, and the process is continued until no further elimination is possible. The edge-following routine gives both external and internal boundaries, the latter being found by edge-following after inverting the bits inside and on the boundary of the closed region defined by an external boundary.

The crossing-number χ is as defined by Rutovitz but in calculating χ it is to be noted that if $\gamma(1)$ and $\gamma(3)$ are both 1 the decision on whether a point can be eliminated or not does not depend on $\gamma(2)$. In such a case $\gamma(2)$ is considered to be 1. Similar statements apply in the other three quadrants.

With this modification in the calculation of χ, unless $\chi = 2$ the point cannot be eliminated. If $\chi = 2$, a further investigation is required before deciding to eliminate, since an end satisfies this condition and there must be a means of ensuring that further iteration does not simply shorten a skeleton line.

A line end satisfies the additional condition

$$\sum_{k=1}^{8} \gamma(k) = 1$$

but it is desirable to eliminate some points for which the conditions

$$\chi = 2 \qquad \sum_{k=1}^{8} \gamma(k) = 1$$

are satisfied. The size of the original picture area (512 × 512 points) makes it impossible to keep the original picture in store and to build a separate modified picture. The elimination process is used to modify the original picture, and unless some care is taken 'spikes' are generated due to minor irregularities in the edge of the initial pattern.

The scheme has been applied to samples of all the digits, and to certain other patterns to which line-thinning is appropriate. Uniformly good results have been obtained and are available as graph-plotter drawings. The plotter drawings show that the final irreducible line is centrally

situated with respect to the points of the initial pattern and is not biased to left or right or top or foot.

Fig. 1(*b*) in Mr. Deutsch's note raises questions of bias. It is to be noted that in Fig. 1(*b*) the top of the digit lies along the tenth row, in Fig. 1(*a*), the top of the digit lies along the ninth row, and that in the input diagram in Fig. 1, the tenth row corresponds to an internal boundary of the digit. It would be of interest to see the result produced by Mr. Deutsch's scheme when it is applied to a character the original lines of which were, say, twelve points thick.

In conclusion it should be noted that if the scheme described above is used with the pattern of Fig. 1(*b*) as input data, a further two points would be removed, namely the points forming the top right hand and bottom left hand corners.

Yours faithfully,

J. G. FRASER

Department of Computer Science
University of Strathclyde
Glasgow C1
8 December 1969

Reference

RUTOVITZ, D. (1966). Pattern Recognition, *Journal of the Royal Statistical Society, Series A*, Vol. 129, pp. 504–530

Book review

The Economics of Computers, by William F. Sharpe, 1969; 571 pages. (*Columbia University Press, New York*, £4.50

According to the preface this book. . . 'attempts to provide and apply a set of concepts from economic theory that may prove valuable to those who are now or may become decision-makers in the selection, financing, and/or use of computers'. 'The book is intended to appeal to three groups of readers': . . . 'those who now manage or set policy for computer installations', 'students in programs leading to degrees in computer science (or information science)', . . . and 'economists interested in the computer industry'.

There are two parts. Part I is entitled 'Theory' and I found the 178 pages of this part tedious and frustrating. It is a mash of economic ideas served up with a dash of computer sauce that failed to stimulate my palate. The discussion is conceptual and is primarily oriented towards the reasons computer manufacturers might have for some of their policies and decisions; but, of course, they might have other reasons for them too, and the symbols are merely symbols, representing nothing (or anything?). This part must be directed towards the students, but there are better texts in theoretical applied economics. Decision makers and economists are advised to skip, and students to find another course.

Part II is entitled 'Applications', and appears to include notes about every topic the author thought relevant to the subject (but he does not deal explicitly with many subjects of considerable importance to computer users); he managed to fill some 358 pages. As, mixed with the soggy dough, there are some very interesting morsels, a further description of the contents of this part might help some potential readers recognize chapters of interest to them.

The applications are sandwiched between two chapters about the computer industry. The first gives a potted history followed by a brief description of the sources of data about the past and current state of the industry and computer installations. The final chapter is a recital of all the bits about the industry which the author has not already worked in elsewhere: leasing companies, used computer market, the market for computer services and software (including a page or two on patent and copyright problems), personnel and communications costs, and finally as much as two pages on 'computers abroad'. These chapters may prove to be useful points of departure either for a general reader or, through the references, for someone seeking to pursue a particular point.

There are two chapters on the sale and rental of computers. The first is devoted to an examination of the terms and conditions offered by computer manufacturers to the U.S. government in 1966–7. It might be useful as a digest to the curious, or to specialists in government tendering procedures; but because of the distance in time, let alone space, there is little of immediate practical value. The second, devoted to 'constraints and issues', is again something of a hotch-potch; among the subjects dealt with are anti-trust action against IBM, and the relationships between rental, maintenance, extra-use charges, purchase prices and purchase options. This chapter does cover some useful background information. For example, the author makes it clear, both here and elsewhere, that actual and potential anti-trust suits are fundamental determinants of IBM policy; this is of immediate practical interest to computer users, even in the U.K., and may also be of general interest to those interested in monopoly legislation.

The chapter on pricing computer services is very poor, primarily because it treats the problems of internal pricing as a problem in theoretical economics rather than practical management. Furthermore, the version of economic theory expounded, here as elsewhere, is naïve and does scant justice either to the complexity of the problems or the sophistication of discussions on the supply of, for example, such services as electricity or transport infrastructure. The section on external pricing is not much better, while that on the pricing of university computer services may be relevant to the U.S.A. but not to the U.K.

The most interesting chapter is that entitled 'Cost and effectiveness of computer systems', and this is supported by another on the cost-effectiveness of memory. But the interest of these chapters is due more to the vista of interesting research opened up, than to the immediate practical value of the results presented. The studies described have produced some impressive and interesting explanatory relationships, but the basis for these is historical data; there is no theoretical justification for extrapolating these results, and some practical reason for believing that the underlying relationships are changing. The relative success of the work described in an area of such importance must draw others into the field, especially as, typically, no conclusive answers have been produced, and many other closely related questions have been brought to the surface.

This book is presented as a 'Rand Corporation Research Study' but much of it has the character of a specialised encyclopaedia. However, Professor Sharpe has convinced me, at least, that the economics of computers is a subject that should be developed and taught, which may be the real purpose of his pioneering book. But let us hope that those who decide to follow him in that respect avoid his obsession with machinery and appreciate that the economics of such a complex service as that provided by computers must recognize more costs than those of the machinery, and more interaction with users and computer applications than the price (omniscient?) users might be prepared to pay.

This book can hardly be enthusiastically recommended to any of its three target groups of readers. But many will dip into it and feel, as I do, that they have derived some benefit.

M. J. H. WEBB (London)