$$\text{else } \gamma \leftarrow \begin{cases} \Sigma_{d-2}(\beta_{d-1}+1)^1 & \text{if } m_d = 1 \\ \Sigma_{d-2}(\beta_{d-1}+1)^1 1^1 & \text{if } m_d = 2 \\ \Sigma_{d-2}(\beta_{d-1}+1)^1 2^{(m_d-1)/2} & \text{if } m_d > 2 \text{ is odd} \\ \Sigma_{d-2}(\beta_{d-1}+1)^1 2^{(m_d-2)/2} 1^1 & \text{if } m_d > 2 \text{ is even} \end{cases}$$

exit.

2. If $\beta_d > 1$ then

$$\gamma \leftarrow \begin{cases} \Sigma_{d-1} 1^{\beta_d} & \text{if } m_d = 1 \\ \Sigma_{d-1} \beta_d^{m_d-1} 1^{\beta_d} & \text{otherwise} \end{cases}$$

exit.

3. If $d > 1$ then

$$\gamma \leftarrow \Sigma_{d-2} \beta_{d-1}^{m_{d-1}-1} 1^{m_d+\beta_{d-1}}$$

else $\gamma \leftarrow$ **nil.**

We see that Algorithm 9 is loop-free. However, if we use the k-tuple representation instead of the multiplicity representation, the resulting algorithm clearly cannot be loop-free. This last statement also holds for any algorithms which generate the successor of α in lexicographic or anti-lexicographic order (i.e. corresponding to Algorithm 5 and NEXPAR). It is, therefore, interesting to observe that if we used a 'mixed' representation with parts greater than 2 represented in k-tuple form and parts of size 2 and 1 represented by their (possibly zero) multiplicities (e.g. the partition $6^2 4^1 3^3 2^3 1^6$ is represented as the triple $<(6, 6, 4, 3, 3, 3), 3, 6>$), then the algorithm derived from Algorithm 9 will be loop-free. However, it is again not possible for the algorithms corresponding to Algorithm 5 and NEXPAR to be loop-free, even for this representation.

By inspection it is possible to deduce the structure of the M-ordering of $\mathscr{P}_n$. It is, in fact, a lexicographic ordering in which, instead of employing the usual collating sequence 1, 2, 3, . . . , $n-1$, n for comparisons between parts, the modified collating sequence 2, 3, 4, . . . , n, 1 is used. This result can be proved formally in a manner analogous to the proof of the corresponding result for the lexicographic order given in Section 4.

7. Conclusions

A representation of the set of partitions of an integer n as a binary tree T_n has been introduced. By considering two of the principal ways in which this tree may be traversed, we have obtained both recursive and non-recursive loop-free versions of algorithms for generating the set of partitions in two different orders, one of which is the natural lexicographic order. This same approach may also be used to derive algorithms for generating the set of partitions in other orders.

The coding of Algorithms 5 and 9 as computer programs entails a number of decisions concerning the ordering of the various tests, and also the extent to which code may be used in common for several of the various cases. Accordingly, any detailed analysis or comparison of these with other algorithms will depend on the particular decisions taken. The analytical results needed to evaluate any detailed coding of the algorithms may be found in Fenner and Loizou (1981). Moreover, these results may also be used as a basis for making the decisions on the ordering of the tests in an optimal fashion.

References

ANDREWS, G. E. (1976). *The theory of partitions*, Addison-Wesley, Reading, Mass.

BERGE, C. (1971). *Principles of combinatorics*, Academic Press, New York.

BRYLAWSKI, T. (1973). The lattice of integer partitions, *Discrete Math.*, Vol. 6, pp. 201-219.

EHRLICH, G. (1973). Loopless algorithms for generating permutations, combinations, and other combinatorial configurations, *JACM*, Vol. 20, pp. 500-513.

FENNER, T. I. and LOIZOU, G. (1981). An analysis of two related loop-free algorithms for generating integer partitions (submitted for publication).

KNUTH, D. E. (1973). *The art of computer programming*, Vol. 1, Addison-Wesley, Reading, Mass.

LEHMER, D. H. (1964). The machine tools of combinatorics, in *Applied combinatorial mathematics*, Beckenbach, E. F. (ed.), Wiley, New York.

MCKAY, J. K. S. (1965). Partition generator, Algorithm 263, *CACM*, Vol. 8, p. 493.

MCKAY, J. K. S. (1970). Partitions in natural order, Algorithm 371, *CACM*, Vol. 13, p. 52.

NARAYANA, T. V., MATHSEN, R. M. and SARANJI, J. (1971). An algorithm for generating partitions and its applications, *J. Combinatorial Theory*, Vol. 11, pp. 54-61.

NIJENHUIS, A. and WILF, H. S. (1975). *Combinatorial algorithms*, Academic Press, New York.

RIHA, W. and JAMES, K. R. (1976). Efficient algorithms for doubly and multiply restricted partitions, Algorithm 29, *Computing*, Vol. 16, pp. 163-168.

SCOTT, D. (1970). The lattice of flow diagrams, Oxford Univ. Comput. Laboratory, Rep. PRG-3.

WELLS, M. B. (1971). *Elements of combinatorial computing*, Pergamon Press, Oxford.

WHITE, J. S. (1970). Restricted partition generator, Algorithm 374, *CACM*, Vol. 13, p. 120.

Book review

Current Issues in Computer Simulation edited by N. R. Adam and A. Dogramaci, 1979; 292 pages. (*Academic Press*, $24.00)

This collection of refereed papers provides a comprehensive coverage of current work in most aspects of computer simulation. It is divided into three parts, each with an introductory chapter by the editors, roughly reflecting concerns about how computer simulations can be programmed, where they have been applied and how they can be used.

Part I concerns computer simulation languages, and includes chapters on particular languages as well as on more general issues. I particularly enjoyed a chapter by McCormack and Sargent on the choice of future event set algorithms. Applications of simulation dealt with in Part II include uses for corporate simulation, consumer choice, hospital occupancy, computer networks and the organisation of the US juvenile court system; unfortunately no applications of continuous simulation are discussed. A recurring theme here is the use of simulation models to predict the effect of different policies, but there is little mention of the perennial problem of making sure that the system being simulated actually exhibits the behaviour exemplified in the model. Part III contains more detailed statistical and OR material. It consists of papers on the uses of computer simulations in conjunction with analytic models in statistical experiments. Although it appears interesting I am not qualified to comment on its technical content.

The standard of presentation is high with many references, and this book can be warmly recommended to those interested in most aspects of computer simulation. P. J. L. WALLIS (Bath)