ELSEVIER

Information Processing Letters 48 (1993) 73–75

Information Processing Letters

Enumerating k-way trees

M.D. Atkinson *, Louise Walker

Department of Mathematical and Computational Sciences, North Haugh, St Andrews, Fife KY16 9SS, Scotland, United Kingdom

Communicated by R.S. Bird
Received 28 May 1993
Revised 9 August 1993

Key words: Combinatorial problems; Data structures; Trees

This paper makes a contribution to the enumeration of trees. We prove a new result about k-way trees, point out some special cases, use it to give a new proof of Cayley's enumeration formula for labelled trees, and observe that our techniques allow various types of k-way trees to be generated uniformly at random. We recall the definition: a *k-way tree* is either empty or it consists of a root node and a sequence of k k-way subtrees. These structures are often used in software systems to store data, with $k = 2$ (binary trees) being the commonest case. It is usual to represent a k-way tree by a diagram in which the root node is connected to its non-empty subtrees by edges which point in one of k fixed directions. Fig. 1 depicts a 3-way tree with 5 directions of the first type, 4 of the second type and 3 of the third type.

Theorem. *The number of k-way trees on $n = 1 + \sum_{i=1}^{k} a_i$ nodes with a_i edges in the ith direction, for $1 \leqslant i \leqslant k$, is*

$$\frac{1}{n}\prod_{i=1}^{k}\binom{n}{a_i}.$$

* Corresponding author.

The theorem depends on an encoding of k-way trees. The encoding generalises the binary tree encoding given in [6] but we justify our encoding in a somewhat simpler manner. Let Γ be any k-way tree on n nodes. For any node q in Γ we define a (0, 1) column vector of length k, called the *type* of q: the ith component of the type vector is 0 if the ith subtree of q is empty, and is 1 otherwise. The entire tree is encoded by a $k \times n$ matrix $M(\Gamma)$ consisting of the types of the nodes in the order of a pre-order traversal. For example, the tree of Fig. 1 is encoded as:

$$\begin{matrix} 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{matrix}$$

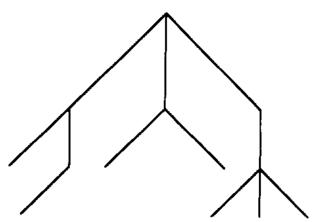

Fig. 1. A 3-way tree.

SSDI 0020-0190(93)E0147-C

To every k-way tree Γ there is an associated ordinary rooted ordered tree Γ^* obtained from Γ by forgetting the directions on the edges (but retaining the same order). Of course, the ordinary tree Γ^* arises from many different k-way trees. An ordinary tree can be encoded as the pre-order sequence of the down-degrees of its nodes; the encoding $M(\Gamma)$ may be regarded as a generalisation of this degree coding. The columns of $M(\Gamma)$ give the more subtle information required to describe the subtrees of a k-way tree node.

The following lemma records some elementary properties of M(Γ).

Lemma 1. (1) *$M(\Gamma)$ has exactly $n-1$ entries equal to* 1.

(2) *The number of* 1*'s in the ith row of $M(\Gamma)$ (the row sum) is equal to the number of edges of Γ in the ith direction.*

(3) *The number of* 1*'s in the jth column of $M(\Gamma)$ (the column sum) is equal to the down-degree of the jth node (in the pre-order traversal) of Γ.*

(4) *The vector of column sums is the degree coding of the ordinary tree Γ^*.*

Not every $k \times n$ (0, 1) matrix with $n-1$ entries equal to 1 is the matrix of some k-way tree. In fact, as the next lemma shows, only one such matrix in every n corresponds to a k-way tree.

Lemma 2. *Let A be any $k \times n$ (0, 1) matrix with $n-1$ entries equal to* 1. *Let A_i be the matrix obtained by rotating the columns of A cyclically to the right through i positions. Then $A_0, A_1, \ldots, A_{n-1}$ are all distinct and precisely one of them is the matrix of a k-way tree.*

Proof. The column sums of A define a vector $\mathbf{d} = (d_1, d_2, \ldots, d_n)$ of non-negative integers with sum $n-1$. By a theorem of Raney [8] the n rotations of this vector are all distinct and there is a unique rotation which is the pre-order degree sequence for an ordinary tree. Once this rotation is identified (and applied to A) the columns of (the new) A define the directions at each node of a unique k-way tree Γ such that $A = M(\Gamma)$. □

Note. The proof given in [8] identified which rotation produced the pre-order degree sequence. Put $d_i' = d_i - 1$ and let $\mathbf{d}' = d_1'd_2' \ldots d_n'$. Raney showed that the necessary rotation of $\mathbf{d}$ is the one that brings to the front that segment of $\mathbf{d}'$ which has largest sum (and, if there are several such, the longest is required). The problem of detecting the segment of a vector with largest sum has achieved some celebrity both as an example of good algorithm design (see [2] and [7]) and as an example of deriving algorithms from their formal specification (see [1] and [3]). The linear time solution in [2] can easily be adapted to accommodate the extra requirement of finding the longest when there are several candidates.

Proof of the Theorem. By Lemma 1 the k-trees described in the statement of the theorem have encodings which are $k \times n$ (0, 1) matrices with $n-1$ entries equal to 1 and with a_i entries equal to 1 in the ith row, for each $i = 1, \ldots, k$. In general there are $\prod_{i=1}^{k}\binom{n}{a_i}$ $k \times n$ matrices with a_i entries in the ith row equal to 1 since the ith row may be chosen in $\binom{n}{a_i}$ different ways. By Lemma 2 exactly $(1/n)\prod_{i=1}^{k}\binom{n}{a_i}$ of these are encodings of the k-way trees. □

Note. Exactly the same technique establishes that the number of k-way trees on n nodes is

$$\frac{1}{n}\binom{kn}{n-1}$$

since there are $\binom{kn}{n-1}$ $k \times n$ (0, 1) matrices with $n-1$ entries equal to 1 but only one in n of them is an encoding of a k-way tree. Another proof of this formula appears in [4, p. 584].

Corollary. *The number of binary trees on n nodes with i left branches and $n-i-1$ right branches is*

$$\frac{1}{n}\binom{n}{i}\binom{n}{i+1}.$$

Somewhat surprisingly, our theorem gives a simple proof of Cayley's theorem (see [4, pp. 389–391]) on free (= unordered, unrooted) labelled trees.

Corollary. *The number of free labelled trees on n nodes is* n^{n-2}.

Proof. The number of $(n-1)$-way trees with $a_1 = a_2 = \cdots = a_{n-1} = 1$ is, by the Theorem, n^{n-2}. These trees have exactly one branch in each of $n-1$ different directions. There is a simple one-to-one correspondence between such trees and free labelled trees. Given any such $(n-1)$-way tree we can label its nodes with $1, 2, \ldots, n$ as follows: the root is labelled n and any other node is labelled by the direction that leads to it. Conversely, from any free labelled tree we can reconstruct an $(n-1)$-way tree; we orient the tree by choosing the root node to be the node labelled with n, and then give directions to each edge according to the label on its lower end point. This one-to-one correspondence proves the corollary. □

Finally we observe that our encoding of trees allows arbitrary k-way trees to be generated uniformly at random in time $O(nk)$ with arithmetic involving only integers of $O(n)$ in value. We merely generate a random $k \times n$ (0, 1) matrix with $n-1$ entries equal to 1 and shift it using Bentley's algorithm until it is the matrix of a k-way tree. The random matrix can be constructed from a random combination generator (see [5, p. 137]). If a random k-way tree with specified numbers of directions of each type is required we just have to generate the initial random matrix row by row so that the ith row has a_i entries equal to one; again a combination generator can be used.

References

[1] J.L. Bates and R.L. Constable, Proofs as programs, *ACM Trans. Programming Language Systems* **7** (1985), 113-136.

[2] J.L. Bentley, Algorithm design techniques, *Comm. ACM* **27** (1984) 865-871.

[3] D. Gries, The maximum segment sum problem, in: E.W. Dijkstra, ed., *Formal Development of Programs and Proofs* (Addison-Wesley, Reading, MA, 1990) 33-36.

[4] D.E. Knuth, *The Art of Computer Programming, Vol. 1: Fundamental Algorithms* (Addison-Wesley, Reading, MA, 1973).

[5] D.E. Knuth, *The Art of Computer Programming, Vol. 2: Semi-numerical Algorithms* (Addison-Wesley, Reading, MA, 2nd ed., 1981).

[6] J.F. Korsh, Counting and randomly generating binary trees, *Inform. Process. Lett.* **45** (6) (1992) 291-294.

[7] U. Manber, Introduction to Algorithms (Addison-Wesley, Reading, MA, 1989).

[8] G.N. Raney, Functional composition patterns and power series reversion, *Trans. Amer. Math. Soc.* **94** (1960) 441-451.