GENERATING K-ARY TREES LEXICOGRAPHICALLY

by

Shmuel Zaks

Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, Illinois 61801

November, 1977

† This work was supported in part by the National Science Foundation under grant number NSF MCS-73-03408.
TABLE OF CONTENTS

1. Introduction
2. Definitions and Notations
3. Existing Algorithms and Results; The Generating Algorithm
4. The Ranking Function and the Unranking Procedure
5. Summary
I. INTRODUCTION

We show a 1-1 correspondence between all regular k-ary trees with n internal nodes, all 0,1-sequences \( x = \{x_i\}_{i=1}^{kn} \) with n 1's and \((k-1)n\) 0's in which \( k-1 \) times the number of 1's in any prefix \( \{x_i\}_{i=1}^{\ell} \) \((1 \leq \ell \leq kn)\) is at least as the number of 0's in that prefix, and all integer sequences \( z = \{z_i\}_{i=1}^{n} \) in which \( 0 < z_1 < z_2 < \ldots < z_n \) and \( z_i \leq ki - (k-1) \) for \( i = 1, 2, \ldots, n \).

Working with the reverses of these sequences, we deal with the following three steps:

1. Generating: the sequences are generated one-by-one in lexicographic order.
2. Ranking: we show how to compute the function that, given a sequence, determines its position in that lexicographic order.
3. Unranking: a procedure is discussed that, given a position in that lexicographic order, constructs the sequence which occupies this position.

We also discuss relations to existing related algorithms.
II. DEFINITIONS AND NOTATIONS

2.1 We state here definitions and notations that we shall follow throughout this paper. We follow [KN] and [LI] for our basic terms.

A tree means an ordered tree. The terms root, son, internal node and leaf are used as defined in the literature. A \textit{k-ary} tree is a tree all of which internal nodes have at most \( k \) sons each. A \textit{regular} \( k \)-ary tree is a tree all of which internal nodes have exactly \( k \) sons each. \( |T| \) denotes the number of vertices in a tree \( T \), and \( T_i \) is the subtree rooted at the \( i \)-th son of the root of \( T \). \( T(k,n) \) denotes the set of all the regular \( k \)-ary trees with \( n \) internal nodes. The number of elements in \( T(k,n) \) is known to be 
\[
\frac{1}{(k-1)n+1} \binom{kn}{n}
\]
(see [KN: p. 584]).

2.2 In order to generate trees "lexicographically", we first establish a 1-1 correspondence between them and certain integer sequences, and then we generate these sequences lexicographically. Given a tree \( T \in T(k,n) \) we label each internal node with 1 and each leaf with 0. We then read these labels in preorder (root-left-right)*, and thus obtain a sequence of \( n \) 1's and \( (k-1)n + 1 \) 0's. The last visited node is a leaf, and we omit the corresponding 0 for simplifying matters. Denote the resulting sequence by 
\[
f(T) = x = \{x_i\}_{i=1}^{kn},
\]
and let \( F(T) = X \) be the reverse of \( f(T) \).

Instead of working with the sequences \( x \) and \( X \) of length \( kn \), one can prefer using the sequences \( g(T) = z = \{z_i\}_{i=1}^{n} \) in which \( z_i \) = position of the \( i \)-th 1 in \( f(T) = x \), and its reverse \( G(T) = Z \). Note that both \( z \) and \( Z \) are of length \( n \). See figure 1 for an example. The 0,1-sequence \( x = f(T) \) is also discussed in [BM], [GA] and [KL].

* The sons are read from left to right.
Sequences Corresponding to a k-ary Tree

Figure 1

We note that the reconstruction of a tree $T$ given $X = F(T)$ or $x = f(T)$, and the transformations $x \leftrightarrow z$ and $X \leftrightarrow Z$, are all quite easy, and the details are left to the reader.

A sequence $x$ is called **feasible** if there is a tree $T \in T(k,n)$ s.t. $x = f(T)$. The same definition holds for $X, z$ and $Z$. $x(k,n), X(k,n), z(k,n)$ and $Z(k,n)$ denote the classes of the feasible $x, X, z$ and $Z$ sequences, respectively, corresponding to the trees in $T(k,n)$.

Let $a = \{a_i\}_{1}^{kn}$ be a sequence consisting of $n$ 1's and $(k-1)n$ 0's. We say that $a$ has the **$k$-dominating property** if in each prefix $\{a_i\}_{1}^{\ell}, 1 \leq \ell \leq kn$, the number of 1's is at least $\left\lceil \frac{\ell}{k} \right\rceil$; in other words, the number of 0's is

* $\lceil t \rceil$ denotes the smallest integer not smaller than $t$. $\lfloor t \rfloor$ denotes the largest integer not larger than $t$. 
at most \((k-1)\) times the number of 1's in any such prefix. This property is closely related to the generalized box-office problem* (see [DM], [ZA]). The case \(k = 2\) is discussed in various references (see, for example, [YY: Problem 83], or [GN: p. 37]).

2.3 In this paper we work with the reverse sequences, for which we obtain the following:

1. Generating: we generate \(Z(k,n)\) lexicographically (corresponds to a lexicographic generation of \(X(k,n)\); see Theorem 2).

2. Ranking: we show how to compute the function \(\text{index}(X)\) that assigns to a sequence \(X \in X(k,n)\) its position in the lexicographic ordering of \(X(k,n)\).

3. Unranking: given an integer \(t\), we construct the sequence \(X \in X(k,n)\) s.t. \(\text{index}(X) = t\).

In a previous work [ZA], we proved that generating \(x(k,n)\) lexicographically corresponds to generating the trees in \(T(k,n)\) according to the following ordering (see also Theorem 2):

**Definition 1:** Given \(T, T' \in T(k,n)\), we say that \(T < T'\) if

1. \(T\) is empty, or

2. \(T\) is not empty, and for some \(i, 1 \leq i \leq k\), we have
   a) \(T_j = T'_j\) for \(j = 1, 2, \ldots, i-1\), and
   b) \(T_i < T'_i\).

Consequently, generating \(X(k,n)\) (= the reverses of the feasible \(x\)

* The generalized box-office problem: \(kn\) people are waiting in a line at a box-office. \(n\) of them have \(k-1\) 1-dollar bills, and the other \((k-1)n\) have one \(k\)-dollar bill. A ticket costs \(k-1\) dollars, and each customer buys one ticket. When the box-office opens, there is no money in the till. In how many ways can they stand so that none of them will have to wait for change?
sequences) corresponds to generating $T(k,n)$ according to the "reverse" ordering, defined as follows:

**Definition 2:** Given $T, T' \in T(k,n)$, we say that $T < T'$ if

1. $T$ is not empty, and for some $i, 1 \leq i \leq k$, we have:
   a) $T_j = T'_j$ for $j = k, k-1, \ldots, i+1$, and
   b) $T_i < T'_i$, or
2. $T$ is empty.

As defined, the $x = f(T)$ sequence corresponds to a preorder traversing of the 0,1 labels of the vertices of $T$; therefore, the $X = F(T)$ corresponds to a postorder (right-left-root) traversing of these labels. This preorder/postorder nature of these sequences is very briefly reflected in these two definitions.
III. EXISTING ALGORITHMS AND RESULTS; THE GENERATING ALGORITHM

We mention here related algorithms, and discuss results that will be used in the sequel. Theorems 1, 2 and 3, and the generating algorithm following them are not proved here. They are all modifications of results we proved in [ZA].

3.1 In [RH] a binary tree is represented by the sequence of the levels of its leaves from left to right, and the generating, ranking and unranking steps are discussed. In [KN: 2.2.1, ex. 2,4,5; 2.3.1, ex. 6], [TR1, 2] and [KNO] a tree with n vertices is represented by an appropriate permutation of (1, 2, ... , n). The generating, ranking and unranking steps are discussed in [TR1, 2]. Both discussions use definition 1 (as proved in [ZA: Theorems 4.8 and 7.8]).

Ranking and unranking for k = 2 are discussed in [KNO], using a different ordering which is also used - for an arbitrary k - in [TR1].

In [ZR] definition 1 is used to generate, rank and unrank k-ary trees - and larger classes of trees - using a pure combinatorial approach.

3.2 In [ZA] we proved that there is a 1-1 correspondence between T(k,n), all 0,1-sequences \( x = \{x_i\}_{i=1}^{k^n} \) with n 1's and \((k-1)n\) 0's having the k-dominating property, and all integer sequences \( z = \{z_i\}_{i=1}^{n} \) s.t. \( 0 < z_1 < z_2 < \ldots < z_n \) and \( z_i < k_i - (k-1) \) for \( i = 1, 2, \ldots, n \). Modification of this result to the reverse sequences gives a characterization of \( X(k,n) \) and \( Z(k,n) \), as follows:

**Theorem 1:** The following sets are in a 1-1 correspondence with one another:

1. \( T(k,n) \)
2. All 0,1-sequences \( X = \{x_i\}_{i=1}^{k^n} \), with n 1's and \((k-1)n\) 0's, the reverses of which have the k-dominating property.
3. All integer sequences \( Z = \{z_i\}_{i=1}^{n} \) s.t. \( z_1 > z_2 > \ldots > z_n > 0 \) and \( z_{n-i+1} \leq k_i - (k-1) \) for \( i = 1, 2, \ldots, n \).

Following the results proved in [ZA] for the \( x \) and \( z \) sequences, and
Theorem 2: Let $T, T' \in T(k,n)$, and let $x, Z, z, Z$ and $x', X', z', Z'$ be the corresponding sequences associated with them. Then

1. $T < T'$ according to definition 1 $\Leftrightarrow x < x' \Leftrightarrow z > z'$.
2. $T < T'$ according to definition 2 $\Leftrightarrow X < X' \Leftrightarrow Z < Z'$.

3.3 In [ZA], we proved that a 0,1-sequence $x = \{x_i\}_{i=1}^{kn}$ is feasible iff erasing any $10^{k-1}$ pattern*, as long as possible, results in the empty sequence. A graphical interpretation of this reduction is discussed there. Modifying this to the $X$ sequences yields the following:

Theorem 3: A 0,1-sequence $X = \{X_i\}_{i=1}^{kn}$ is feasible iff erasing any $0^{k-1}$ pattern, as long as possible, results in the empty sequence.

It is Theorem 3 that enables us to compute the ranking function for $X(k,n)$ (see Theorem 5 in the next section), while the mentioned result enabled us to compute it for $x(k,n)$ only for $k = 2$.

3.4 In [ZA], we generated the feasible $z$ sequences lexicographically. A modification of this algorithm will generate the $Z$ sequences, as characterized by Theorem 1:

Algorithm GENERATE (Generating the sequences $Z \in Z(k,n)$ lexicographically):

1. Begin with $Z = \{Z_1\} = \{n, n-1, \ldots, 1\}$
2. Find the largest $\ell$ s.t. $Z_\ell < k(n-\ell+1) - (k-1)$.
   
   If $\ell = 1$ then goto 3.
   
   If $\ell > 1$ then: if $Z_\ell < Z_{\ell-1} - 1$ then goto 3 else go on looking for such an $\ell$.

   If no such $\ell$ exists - goto 5.
3. The sequence $Z' = \{Z'_1\}$ next to $Z$ is built as follows:

* $a^m$ means a string consisting of $m$ consecutive $a$'s.
Let $Z'$ be called $Z$.


5. End.

Example: In figure 2 we list the sequences $x$, $X$, $z$ and $Z$ corresponding to the ternary trees with 3 internal nodes ($T(3,3)$). The table is arranged according to lexicographic ordering of the $x$ sequences (= antilexicographic ordering of the $z$ sequences) on the left, and lexicographic ordering of the $X$ sequences (= lexicographic ordering of the $Z$ sequences) on the right.

<table>
<thead>
<tr>
<th>$z$</th>
<th>$x$</th>
<th>index</th>
<th>$X$</th>
<th>$Z$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1,4,7</td>
<td>1 0 0 1 0 0 1 0 0</td>
<td>1</td>
<td>0 0 0 0 0 0 1 1 1</td>
<td>3,2,1</td>
</tr>
<tr>
<td>1,4,6</td>
<td>1 0 0 1 0 1 0 0 0</td>
<td>2</td>
<td>0 0 0 0 0 1 0 1 1</td>
<td>4,2,1</td>
</tr>
<tr>
<td>1,4,5</td>
<td>1 0 0 1 1 0 0 0 0</td>
<td>3</td>
<td>0 0 0 0 0 1 1 0 1</td>
<td>4,3,1</td>
</tr>
<tr>
<td>1,3,7</td>
<td>1 0 1 0 0 0 1 0 0</td>
<td>4</td>
<td>0 0 0 0 1 0 0 1 1</td>
<td>5,2,1</td>
</tr>
<tr>
<td>1,3,6</td>
<td>1 0 1 0 0 1 0 0 0</td>
<td>5</td>
<td>0 0 0 0 1 0 1 0 1</td>
<td>5,3,1</td>
</tr>
<tr>
<td>1,3,5</td>
<td>1 0 1 0 1 0 0 0 0</td>
<td>6</td>
<td>0 0 0 0 1 1 0 0 1</td>
<td>5,4,1</td>
</tr>
<tr>
<td>1,3,4</td>
<td>1 0 1 1 0 0 0 0 0</td>
<td>7</td>
<td>0 0 0 1 0 0 0 1 1</td>
<td>6,2,1</td>
</tr>
<tr>
<td>1,2,7</td>
<td>1 1 0 0 0 0 1 0 0</td>
<td>8</td>
<td>0 0 0 1 0 0 1 0 1</td>
<td>6,3,1</td>
</tr>
<tr>
<td>1,2,6</td>
<td>1 1 0 0 0 1 0 0 0</td>
<td>9</td>
<td>0 0 0 1 0 1 0 0 1</td>
<td>6,4,1</td>
</tr>
<tr>
<td>1,2,5</td>
<td>1 1 0 0 1 0 0 0 0</td>
<td>10</td>
<td>0 0 1 0 0 0 0 1 1</td>
<td>7,2,1</td>
</tr>
<tr>
<td>1,2,4</td>
<td>1 1 0 1 0 0 0 0 0</td>
<td>11</td>
<td>0 0 1 0 0 0 1 0 1</td>
<td>7,3,1</td>
</tr>
<tr>
<td>1,2,3</td>
<td>1 1 1 0 0 0 0 0 0</td>
<td>12</td>
<td>0 0 1 0 0 1 0 0 1</td>
<td>7,4,1</td>
</tr>
</tbody>
</table>

The Sequences Corresponding to $T(3,3)$

Figure 2
IV. THE RANKING FUNCTION AND THE UNRANKING PROCEDURE

4.1 In this section, we find the position $\text{Index}(T)$ of a given tree $T \in T(k,n)$. For this purpose, we take $X = F(T)$, and find the position $\text{index}(X)$ of $X$ in the lexicographic ordering of $X(k,n)$. The modification to $Z(k,n)$ is left to the reader. First we observe the following:

Theorem 4: There is a 1-1 correspondence between all the sequences in $X(k,n)$ which begin with $0^{\ell+k-1}1$, and those in $X(k,n-1)$ which begin with $0^{\ell}$.

Proof: Suppose there are $u$ sequences in $X(k,n)$ beginning with $0^{\ell+k-1}1$. Call them $A_1 = 0^{\ell+k-1}1Y_1, A_2 = 0^{\ell+k-1}1Y_2, \ldots, A_u = 0^{\ell+k-1}1Y_u$. By omitting the first occurrence of $0^{k-1}$ from each of the $A_i$'s we get $B_1 = 0^{\ell}Y_1, B_2 = 0^{\ell}Y_2, \ldots, B_u = 0^{\ell}Y_u$, each of which is a sequence in $X(k,n-1)$ by Theorem 3. It is clear that $B_i \neq B_j$ for $i \neq j$, and that all the sequences in $X(k,n-1)$ beginning with $0^{\ell}$ are matched in this way (by Theorem 3); the theorem has thus been proved. □

Note: It is clear that the correspondence mentioned above preserves lexicographic order, i.e. $A_i < A_j$ iff $B_i < B_j$. This fact will be used in the next theorem - which is used recursively to compute the ranking function - as follows:

Theorem 5: Let $X = 0^{\ell}1Y$ belong to $X(k,n)^*$. Then

$$\text{index}(X)^* = \begin{cases} 
1 & \text{if } X = 0^{a}1^{b} \text{ for some } a \text{ and } b \\
(a((k-1)n - (\ell+1),n,k) + \text{index}(0^{\ell-k+1}1Y) & \text{otherwise}
\end{cases}$$

where $a((k-1)n - (\ell+1),n,k)$ is the number of sequences in $X(k,n)$ which begin with $0^{\ell+1}$.\[\]

Proof: Immediate, by the nature of lexicographic order and theorem 4. □

* By Theorem 4, we have $\ell \geq k - 1$.

** We use here the same name for the functions $\text{index}(X)$ corresponding to a given $n$ and $k$. $\text{index}(X)$ gives the position of $X$ in $X(k,n)$, while $\text{index}(0^{\ell-k+1}1Y)$ gives the position of the feasible sequence $0^{\ell-k+1}1Y$ in $X(k,n-1)$.\[\]
4.2 The following discussion concerns the evaluation of the numbers \( a(i,j,k) \) as defined in Theorem 5. We make use of the following geometric interpretation (see [YY: Problem 83]): a sequence \( a = \{a_i\}_{i=1}^{kn} \), with \( n \) 1's and \((k-1)n\) 0's, corresponds to a path from \( B((k-1)n,n) \) to \( A(0,0) \) in the rectangular lattice defined by \( B \) and \( A \) (see figure 3), where 1 means "one step down" and 0 means "one step to the left." This sequence is feasible iff this path never goes below the line \( AB \) (see Theorem 1), which is \( i = (k-1)j \).

![Figure 3](image)

\( a(i,j,k) \) for \( k = 3 \) and \( j \leq 5 \)

The following labeling of the lattice points gives the number of ways to go from a point \((i,j)\) to \(A\), without going below the line \( AB\):

\[
a(i,j,k) = \begin{cases} 
1 & i = 0 \\
0 & 1 > (k-1)j \\
\ a(i,j-1,k) + a(i-1,j,k) & \text{otherwise}
\end{cases}
\]

see example in figure 3.
The solution to this recurrence relation is as follows:

**Theorem 6:** The solution to the recurrence relation

\[
a(i,j,k) = \begin{cases} 
1 & \text{if } i = 0 \\
0 & \text{if } i > (k-1)j \\
a(i,j-1,k) + a(i-1,j,k) & \text{otherwise}
\end{cases}
\]

where \( i, j \geq 0 \), is given by

\[
a(i,j,k) = \binom{i+j-1}{i} - \sum_{t=1}^{[\frac{i-1}{k-1}]} \binom{i+j-1-kt}{j-t} \frac{1}{(k-1)t+1} \binom{kt}{t}.
\]

where \( \binom{s}{t} = 0 \) if \( s < t \).

**Proof:** We prove by induction on \( i \) and \( j \). For \( i = 0 \) and any \( j \), we have \( a(0,j,k) = 1 \).

We assume the formula holds for \( i-1 \), and prove it for \( i \), as follows:

Let \( i = (k-1)x + y \), \( 1 \leq y \leq k - 1 \). For \( j \leq x \), we get \( i = (k-1)x + y \geq (k-1)j + y \geq (k-1)j \) and in this case the formula must give us the boundary condition \( 0 \), and it really does: as \( i = (k-1)x + y \), and \( 1 \leq y \leq k - 1 \), thus \( \frac{i-1}{k-1} = x \).

So we get

\[
\binom{i+1-1-kt}{j-t} + \frac{1}{(k-1)t+1} \binom{kt}{t} = \binom{i+j-1}{i} - \binom{i+j-1-kt}{j-t} \frac{1}{(k-1)t+1} \binom{kt}{t} = \binom{i+j}{j} - \binom{i+j-1}{j}
\]

* *In [KN: p.58] we have

\[
\sum_{k \geq 0} \binom{r-tk}{k} \binom{s-t(n-k)}{n-k} \frac{r}{n-tk} = \binom{r+s-tn}{n}
\]

for integer \( n \) and \( r \neq tk \), \( 0 \leq k \leq n \). Setting \( k \leftarrow t \), \( r \leftarrow 1 \), \( t \leftarrow -k \), \( s \leftarrow i - (k-1)j - 1 \) and \( n \leftarrow j \) we get our summation \( \binom{i+j}{j} \) minus the term corresponding to \( t = 0 \).
So the formula gives
\[ a(i,j,k) = \binom{i+j-1}{i} + \binom{i+j-1}{j} - \binom{i+j}{j} = 0, \quad \text{as desired.} \]

Assuming it holds for \( j - 1 \), we continue as follows: if \( j > x \), we have
\[
i = (k-1)x + y < (k-1)i + y, \quad \text{or} \quad i \leq (k-1)j,
\]
and we show that \( a(i,j,k) \) satisfies
\[
a(i,j,k) = a(i,j-1,k) + a(i-1,j,k). \]
By the induction hypothesis we have
\[
a(i,j-1,k) = \binom{i+j-2}{i} - \sum_{t=1}^{i-1} \binom{i+j-2-kt}{j-1-t} \frac{1}{(k-1)t+1} \binom{kt}{t}
\]
and
\[
a(i-1,j,k) = \binom{i+j-2}{i-1} - \sum_{t=1}^{i-2} \binom{i+j-2-kt}{j-t} \frac{1}{(k-1)t+1} \binom{kt}{t}.
\]
Therefore,
\[
a(i,j-1,k) + a(i-1,j,k) = \binom{i+j-1}{i} - \sum_{t=1}^{i-1} (\ldots) + \sum_{t=1}^{i-2} (\ldots).
\]

It remains to show that
\[
\sum_{t=1}^{i-2} \binom{i+j-2-kt}{j-t} \frac{1}{(k-1)t+1} \binom{kt}{t} + \sum_{t=1}^{i-1} \binom{i+j-2-kt}{j-1-t} \frac{1}{(k-1)t+1} \binom{kt}{t} = \sum_{t=1}^{i-1} \binom{i+j-1-kt}{j-t} \frac{1}{(k-1)t+1} \binom{kt}{t}.
\]

If \( y > 1 \) then \( \frac{i-1}{k-1} = \frac{i-2}{k-1} = x \), and all the summations are \( \sum_{t=1}^{x} (\ldots) \), hence
\( (*) \) is correct. If \( y = 1 \), we have
\[
\frac{i-1}{x-1} = x, \quad \frac{i-2}{k-1} = x - 1,
\]
but then a term corresponding to \( t = x \) in the second summation on the left side of \( (*) \) is \( \binom{j-x-1}{j-x} \); but \( j > x \), so this is 0. The proof is thus completed. \( \square \)
4.3 As for the unranking procedure, we find it in a "reverse" interpretation of Theorem 5. We are given an integer $t$, and look for a tree $T \in T(k,n)$ s.t. $\text{Index}(T) = t$. For this we find a sequence $X \in X(k,n)$ s.t. $\text{index}(X) = t$. The modification for the $Z$ sequences is left to the reader.

Algorithm \textsc{UNRANK} (Finding the sequence $X \in X(k,n)$ s.t. $\text{index}(X) = t$ for a given $t$):

1. $A \leftarrow t$, $j \leftarrow n$
2. Find* $\ell_j$ s.t. $a(\ell_j, j, k) < A \leq a(\ell_j + 1, j, k)$
3. $A \leftarrow A - a(\ell_j, j, k)$
   \hspace{1cm} $j \leftarrow j - 1$
   \hspace{1cm} If $A > 1$ then goto 2.
   \hspace{1cm} Comment: we have now $t = 1 + \sum_{j=0}^{n} a(\ell_j, j, k)$
   \hspace{1cm} $(k-1)j_0 \leq j_0$
4. $X \leftarrow 0$
   \hspace{1cm} $m \leftarrow j_0 + 1$
   \hspace{1cm} $s \leftarrow (k-1)m - \ell_m$
5. Change $X$ as follows:
   \hspace{1cm} $X_i$ unchanged for $i \leq s - k$
   \hspace{1cm} $X_{s+1}$ $X_{s+2}$ \ldots $X_{km} \leftarrow X_{s-k+1}$ $X_{s-k+2}$ \ldots $X_{k(m-1)}$
   \hspace{1cm} $X_{s-k+1}$ $X_{s-1+2}$ \ldots $X_s \leftarrow 0^{k-1} 1$
6. $m \leftarrow m + 1$
   \hspace{1cm} If $m \leq n$ then set $s \leftarrow (k-1)m - \ell_m$ and goto 5.
7. End.

* This step is performed by using a table look-up for the pre-computed values of the $a(i,j,k)$'s, or by computing them on-line. In both cases, we use the formula derived in Theorem 6. Note that $\ell_j$ is unique because $a(i,j,k)$ is increasing in $i$. 
Example: For the sequence $X = 000100101$ (the 8th sequence in figure 2), we have - by our ranking function - the following:

$$\text{index}(X) = \text{index}(000100101) =$$
$$= a(2,3,3) + \text{index}(000101) =$$
$$= a(2,3,3) + a(0,2,3) + \text{index}(001) =$$
$$= a(2,3,3) + a(0,2,3) + 1 = 6 + 1 + 1 = 8.$$

As for the unranking procedure: we look for a tree $T \in T(3,3)$ s.t. $\text{Index}(T) = 8$. For this purpose we will find $X$ s.t. $\text{index}(X) = 8$. Applying our unranking algorithm, we have after its 4th step:

$$8 = a(2,3,3) + a(0,2,3) + 1.$$

By step 5 we first set $X$ to 001. Step 6 changes $X$ to 000101 - by inserting 001 after the first 0 - and then to 000100101.
V. SUMMARY

5.1 After discussing the feasibility criteria for the integer sequences associated with the k-ary trees with n internal nodes, we introduced the algorithm GENERATE (in 3.4) which generates the sequences \( Z \in Z(k,n) \) lexicographically. The recursive ranking function \( \text{index}(X) \) (computed by Theorem 5) finds the position of \( X \) in the lexicographic ordering of \( X(k,n) \), and is based on the reduction criterion (Theorem 4) and on the solution to the recurrence relation for \( a(i,j,k) \) (Theorem 6). The algorithm UNRANK (in 4.3) finds the sequence \( X \in X(k,n) \) s.t. \( \text{index}(X) = t \), and - as is always the case - is done by using the ranking procedure backwards.

5.2 There is a simple 1-1 correspondence between all regular k-ary trees with n internal vertices and all k-ary trees with n vertices (see [KN: p. 559] or [TR1: p. 3]), so our algorithms can be applied to these classes as well.

ACKNOWLEDGEMENT

I would like to thank Professor C. L. Liu for making valuable suggestions.
REFERENCES


We show a one-one correspondence between all the regular k-ary trees with n internal nodes and certain integer sequences. We then generate these sequences lexicographically, and discuss the ranking function and the unranking procedure. Relations to existing algorithms are discussed.