Recording the structure of trees in their leaves.

The following results are by no means new. I found the result for the binary tree in the late fifties; the result for the general tree — or something very similar to it — was discovered in the late sixties by C.S.Scholten and, if I remember it correctly, by D.T.Ross as well. The reasons for writing this short note are twofold. Firstly, with a renewed interest in trees I realized that I had never seen the results properly recorded — they may be somewhere in Knuth's encyclopedia! —; secondly, thanks to a recent remark by C.S.Scholten I now understand them much better than before.

Offspring of the results are the following theorems.

Theorem 1. Consider for positive integer n the values $r(n)$ given by

$$r(1) = 1$$

$r(n) =$ the sum of all products $r(i) * r(j)$, such that $i + j = n$ with $0 < i, j < n$.

Then $r(n) < 4^n$. (Note that $r(n)$ equals the number of rooted trees with n leaves such that each internal node has exactly two outgoing arrows.)

Theorem 2. Consider for positive integer n the values $r(n)$ given by

$$r(1) = 1$$

$r(n) =$ the sum of all products $r(i) * \cdots * r(j)$, such that $i + \cdots + j = n$ with $0 < i, \ldots, j < n$.

Then $r(n) < 6^n$. (Note that $r(n)$ equals the number of rooted trees with n leaves such that each internal node has at least two outgoing arrows.)

*
*
*

For a tree that is not a leaf the so-called "traversal order" of its leaves is recursively defined as the traversal orders of the leaves of its subtrees dealt with in the order of those subtrees, e.g.
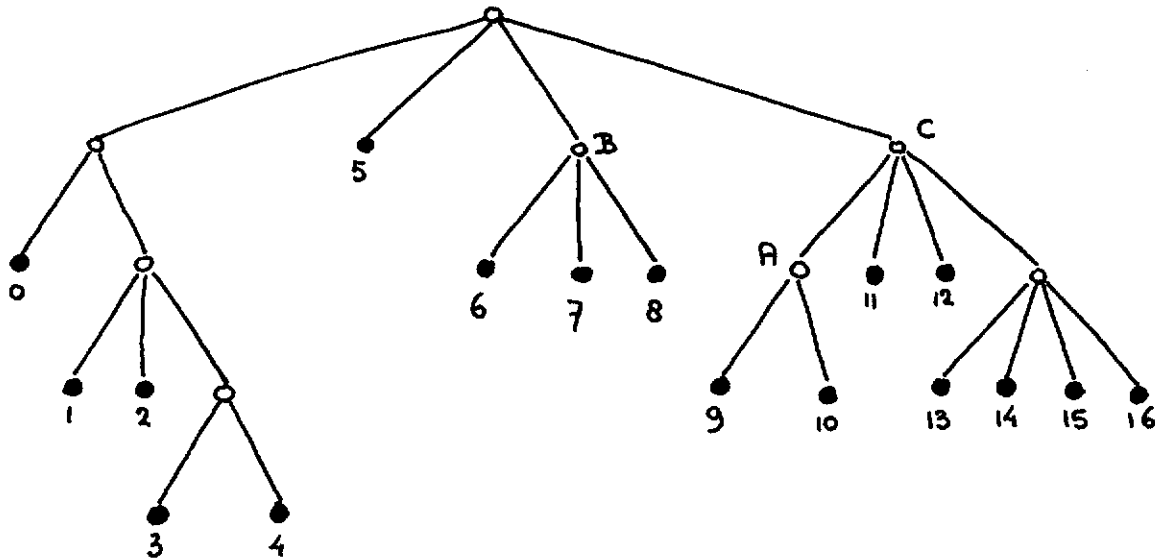
We shall characterize the structure of the trees by attaching to each leaf a label from a finite alphabet and listing those labels in traversal order. The stated theorems follow from the fact that — as we shall show — for the binary and the general rooted tree labels from a four-character alphabet and a six-character alphabet respectively suffice.

The labels serve two purposes. Firstly, they have to

define which nodes — leaves to start with — are "sons" of the same "father". Secondly, because only labels of leaves have been given, the label of a father will be defined by the labels of its sons. (Regretfully enough, we cannot avoid also attaching an irrelevant label to the root of the tree.)

Each internal node has one so-called "first son" and one so-called "last son"; it may have "middle sons". (In our example: 0,1,3, and A are first sons; 2,5,7, and B are middle sons; 4,8,10, and C are last sons.) In order that the labels determine which nodes — leaves to start with — are sons of the same father, we shall attach distinct labels to first sons, to middle sons, and to last sons. Because the middle son may be missing, the labels of the first and last sons of a father must determine the father's label.

With x possibilities for a first son's label and y possibilities for a last son's label, and an arbitrary label for the father, we conclude the requirement

$$x * y \geq \text{size of the alphabet} . \qquad (1)$$

For the binary tree — no middle sons — the distinction between between first and last sons requires

$$\text{size of the alphabet} \geq x+y . \qquad (2)$$

Relations (1) and (2) can be solved for size = 4. For the general tree, in which we attach to each middle son the same label, we have, instead of (2)

$$\text{size of the alphabet} \geq x+1+y . \qquad (3)$$

Relations (1) and (3) can be solved for size = 6.

The following codings show that these alphabet sizes indeed suffice.

For the binary tree we represent each label by the pair (b,c) with $0 \leq b,c \leq 1$. A father with label (b,c) has:

a first son with label $(0,b)$

a last son with label $(1,c)$,

i.e. distinct labels that satisfy the stated inequalities.

For the general tree we represent each label by the pair (b,c) with $0 \leq b \leq 1$ and $0 \leq c \leq 2$. A father with label (b,c) has:

a first son with label $(0,b)$

zero or more middle sons with label $(0,2)$

a last son with label $(1,c)$,

i.e. distinct labels that satisfy the stated inequalities.

Plataanstraat 5
5671 AL NUENEN
The Netherlands

21 April 1980
prof. dr. Edsger W. Dijkstra
Burroughs Research Fellow