# Note on the number of rooted complete $N$-ary trees

Stephan Dominique Andres
Zentrum für angewandte Informatik Köln
Weyertal 80, 50931 Köln, Germany
andres@zpr.uni-koeln.de

May 22, 2006

**Abstract**

We determine a recursive formula for the number of rooted complete $N$-ary trees with $n$ leaves, which generalizes the formula for the sequence of Wedderburn-Etherington numbers. The diagonal sequence of our new sequences equals to the sequence of numbers of rooted trees with $N+1$ vertices.

*Key words:* parenthesis structure enumeration, complete $N$-ary rooted tree, $N$-ary operation, adder topology

## 1 Introduction

A problem occurring in hardware design is the following: Given an n-operand addition that has to be realized by a set of binary adders, how many possibilities are there to arrange the adders? [1] To be precise we do not care about commutative operations, which can be executed on the adders without changing the arrangement. In a mathematical language, we seek for the number of interpretations of $x^n$ (or the number of ways to insert parentheses) when multiplication is commutative but not associative, or, from another point of view, we are looking for the number of isomorphism classes of n-leaf complete binary rooted trees (where every vertex has either 0 or 2 children). These numbers are known as Wedderburn-Etherington numbers, their sequence $(T_n)_n$ has the key A001190 in the On-Line Encyclopedia of Integer Sequences [2] and its generating function $B(x)$ satisfies the functional equation

$$B(x) = x + \frac{1}{2}\left(B(x)^2 + B(x^2)\right),$$

cf. [3]. So, a recursive method to calculate the sequence is given by

$$\begin{aligned}
T_1 &= 1, \\
T_{2n+2} &= \sum_{i=1}^{n} T_i \cdot T_{2n+2-i} + \frac{T_{n+1}(T_{n+1}+1)}{2} \qquad \text{for } n \geq 0, \\
T_{2n+1} &= \sum_{i=1}^{n} T_i \cdot T_{2n+1-i} \qquad \text{for } n \geq 1.
\end{aligned}$$

The aim of this paper is to generalize this formula from complete binary to complete $N$-ary trees (or from binary adders to $N$-ary adders as building blocks). Thus we want to determine the number $T_n^{(N)}$ of isomorphism classes of rooted trees with $n$ leaves with the property that each vertex has either $N$ or 0 children.

## 2 Recursive formula

To abbreviate notation, we call such a rooted complete $N$-ary tree with $n$ leaves an *$n$-tree* whenever $N$ is fixed. Let

$$p_i^{(D)} = \#\{(h_1, \ldots, h_D) | \forall l : h_l \geq 1, \sum_{l=1}^{D} h_l = i\} = \begin{pmatrix} i-1 \\ D-1 \end{pmatrix} \qquad (1)$$

be the number of unordered partitions of i into $D$ pieces. Then we may formulate

**Theorem 1** *For $N \geq 2$ and $n \geq 2$, $T_n^{(N)}$ can be calculated recursively via*

$$\begin{aligned}
T_1^{(N)} &= 1, \\
T_n^{(N)} &= \sum_{b=1}^{N} \sum_{\substack{(i_1,\ldots,i_b) \\ \sum_{j=1}^{b} i_j = N \\ i_j \geq 1\ \forall j}} \sum_{\substack{(k_1,\ldots,k_b) \\ 1 \leq k_1 < k_2 < \cdots < k_b \leq n \\ \sum_{j=1}^{b} i_j k_j = n}} \prod_{j=1}^{b} \sum_{D=1}^{i_j} \begin{pmatrix} i_j - 1 \\ D-1 \end{pmatrix} \begin{pmatrix} T_{k_j}^{(N)} \\ D \end{pmatrix}.
\end{aligned}$$

**Proof.** Clearly, $T_1^{(N)} = 1$. So consider an $n$-tree, $n > 1$. Then its root has $N$ children that are roots of $r_1$-, $r_2$-,$\ldots, r_N$-trees, respectively, with $r_s \geq 1$ for all $s$, and $\sum_{s=1}^{N} r_s = n$. $r_s$ is called *input size* of the $s$-th child's tree. Since we do not care about permutations we may assume without loss of generality that $r_1 \leq r_2 \leq \ldots \leq r_N$. We call the set of those $r_j$-trees with the same input size a *block*, and denote the number of blocks by $b$. The *block size* is the number of elements of a block. So we have blocks of (positive) block sizes $i_1, \ldots, i_b$ with respective input sizes $k_1, \ldots, k_b$ where w.l.o.g. $1 \leq k_1 < \cdots < k_b \leq n$. We state that

$$\sum_{j=1}^{b} i_j k_j = n, \qquad \sum_{j=1}^{b} i_j = N.$$

We will count at first the number of possibilities for a block with block size $i$ and input size $k$, a number which we denote by $B(k,i)$. Then we derive the number $T(b,(i_1,\ldots,i_b),(k_1,\ldots,k_b))$ of trees for fixed $(r_1,\ldots,r_N)$, i.e., for fixed $b$, block sizes $i_1,\ldots,i_b$, and input sizes $k_1,\ldots,k_b$. Note that for distinct values of the 3-tupel $(b,(i_1,\ldots,i_b),(k_1,\ldots,k_b))$ trees cannot be isomorphic, since isomorphic trees must have the same number of blocks, corresponding block sizes, and corresponding input sizes, as we assume $k_1 < \cdots < k_b$. Therefore we obtain the total number of $n$-trees simply by adding $T\left(b,(i_1,\ldots,i_b),(k_1,\ldots,k_b)\right)$ over all possibilities for $(b,(i_1,\ldots,i_b),(k_1,\ldots,k_b))$, i.e.,

$$T_n^{(N)} = \sum_{b=1}^{N} \sum_{\substack{(i_1,\ldots,i_b) \\ \sum_{j=1}^{b} i_j = N \\ i_j \geq 1\ \forall j}} \sum_{\substack{(k_1,\ldots,k_b) \\ 1 \leq k_1 < k_2 < \cdots < k_b \leq n \\ \sum_{j=1}^{b} i_j k_j = n}} T(b,(i_1,\ldots,i_b),(k_1,\ldots,k_b)). \tag{2}$$

Now we consider a block with $i$ elements, each one having input size $k$. Let $D$ be the number of distinct $k$-trees occurring in the block. We have

$$\left( \begin{array}{c} T_k^{(N)} \\ D \end{array} \right)$$

possibilities to choose $D$ such structures, and $p_i^{(D)}$ possibilities to partition the $k$-trees of the block in $D$ subblocks each one having equal $k$-trees as elements. These choices are independent in the sense of non-isomorphism, so in total we have

$$p_i^{(D)} \left( \begin{array}{c} T_k^{(N)} \\ D \end{array} \right)$$

possibilities, and

$$B(k,i) = \sum_{D=1}^{i} p_i^{(D)} \left( \begin{array}{c} T_k^{(N)} \\ D \end{array} \right). \tag{3}$$

Now, what happens in different blocks is independent of each other block again, thus we conclude

$$T(b,(i_1,\dots,i_b),(k_1,\dots,k_b)) = \prod_{j=1}^{b} B(k_j,i_j). \tag{4}$$

Combining (1), (2), (3), and (4) yields the theorem. Note that in the right-hand-side of the recursive formula the expression $T_n^{(N)}$ does not occur, since $k_b < n$ whenever $N \geq 2$. $\square$

# 3 Final remarks

A very interesting sequence is the diagonal sequence

$$(T_{N^2}^{(N)})_{N=2,3,4,\dots} = 2, 4, 9, 20, 48, 115, \dots$$

**Theorem 2** $T_{N^2}^{(N)}$ *equals to the number of rooted trees with* $N+1$ *vertices.*

**Proof.** We construct an isomorphism between rooted complete $N$-ary trees with $N^2$ leaves and rooted trees with $N+1$ vertices, $N \geq 2$, in the following way. Let $T$ be a rooted complete $N$-ary tree with $N^2$ leaves. Delete all leaves to obtain a rooted tree with $N+1$ vertices. (Note that there are always $N+1$ inner vertices in $T$.) On the other hand, let $R$ be a rooted tree with $N+1$ vertices. So every vertex has at most $N$ children. Construct a complete $N$-ary tree from $R$ by adding children in such a way that every vertex from $R$ has exactly $N$ children, and every new vertex has no children. Obviously, these mappings are one-to-one. $\square$

**Theorem 3** $\lim\limits_{N\to\infty} T_{(n+1)(N-1)+1}^{(N)} = T_{n^2}^{(n)}$

**Proof.** Let $T$ be a rooted complete $N$-ary tree with $(n+1)(N-1)+1$ leaves, and $N > n$. This means that $T$ has exactly $n+1$ inner vertices, hence every inner vertex has at most $n$ children which are inner vertices. So for every inner vertex we may delete $N-n$ of its children which are leaves to obtain a rooted complete $n$-ary tree with $(n+1)((N-1)-(N-n))+1 = n^2$ leaves. This construction can be reversed by adding children in an appropriate way. $\square$

## References


[1] A. Kinane, “Adder topologies for Multi-Operand Addition”, Centre for Digital Video Processing, Dublin City University, *internet communication*

[2] N.J.A. Sloane et al., On-Line Encyclopedia of Integer Sequences, `http://www.research.att.com/~njas/sequences`

[3] R.P. Stanley, “Enumerative Combinatorics”, Vol. 2, Cambridge Univ. Press, 1999, p.245, Exercise 6.52