A Recurrent Enumeration of Free Hypermultisets

Vincenzo Manca

Department of Computer Science
University of Verona, Italy
vincenzo.manca@univr.it

Abstract. A recurrent formula enumerating unlabeled membrane structures is presented, which is deduced by means of elementary combinatorial arguments by considering them as hypermultisets built from the empty set.

1 Introduction

Tree enumeration formulae are an old subject [1], firstly investigated in connection to chemical structures. In Knuth's book [3] many classical results are reported for different kinds of trees. In particular, labeled unordered rooted trees are enumerated by Caley's formula n^{n-1} (n is the number of nodes). Unlabeled ordered rooted trees and labeled ordered rooted trees are enumerated by using Catalan numbers. Unlabeled unordered rooted trees, correspond to membrane structures of membrane computing [7]. For them no exact analytical formula is available, but a recurrent formula, obtained by means of generating functions, was given in [5], where also a complex asymptotic formula is presented.

In this paper we improve and extend an approach developed in [4], by providing, by elementary combinatorial reasoning, a new recurrent enumeration of membrane structures, viewed as hypermultisets built from the empty set. We call them *free hypermultisets*, where the attribute "free" means existence-free, in the sense of *free logics*, that is, structures without individuals.

2 Free Hypermultisets

A multiset can be obtained from a set by associating a *multiplicity*, to any of their elements, which provides a (non-null natural) number of *occurrences*. The sum of two multisets M_1 and M_2 is the multiset $M_1 + M_2$ where the elements of M_1 and those of M_2 occur with multiplicities which are the sums of the multiplicities they have in the original multisets (a null multiplicity is given to any element which does not occur in a multiset). Of course, the singleton set of an element a, which we denote by $[a]$, is a special case of a multiset. Now, starting from the empty set, which we denote by $[\ ]$, we inductively define the set $\mathbb{M}$ of the *finite free hypermultisets* by setting:

$$\begin{array}{ll} [\ ] \in \mathbb{M} & \textit{Base step} \\ X,\ X_1,\ X_2 \in \mathbb{M} \implies [X],\ X_1 + X_2 \in \mathbb{M} & \textit{Inductive step} \end{array}$$

It is easy to realize that free hypermultisets represent (unlabeled) membrane structures and therefore unlabeled unordered rooted trees.

J. Kelemen and A. Kelemenová (Eds.): Paun Festschrift, LNCS 6610, pp. 16–23, 2011.

The enumeration method we present is based on the following partitions of $\mathbb{M}$. A multiset has the empty-membership property if the empty set occurs in it (with a non null multiplicity). Therefore $\mathbb{M}$ is the disjoint union of the following sets $\mathbb{M} = \mathbb{M}_0 \cup \mathbb{M}_1 \cup \mathbb{M}_2$, where $\mathbb{M}_0$ is the set of elements of $\mathbb{M}$ having the empty-membership property, $\mathbb{M}_1$ is the set of elements of $\mathbb{M}$ which are singletons and do not have the empty-membership property, and $\mathbb{M}_2$ is the set of elements of $\mathbb{M}$ which do not belong to $\mathbb{M}_0$ or to $\mathbb{M}_1$.

We call the elements of sets $\mathbb{M}_0, \mathbb{M}_1, \mathbb{M}_2$ *neo-agglomerates*, *proto-agglomerates*, and *conglomerates* respectively, for this reason we shortly call the free hypermultisets of $\mathbb{M}$ also *agglomerates* (see Fig. 1).

We will denote by $\mathbb{N}$ the set of natural numbers and by i, j, k, m, n (possibly indexed) variables in $\mathbb{N}$, and by $\lfloor x \rfloor$ and $\lceil x \rceil$ the *floor* and the *ceiling* of a real number x, respectively. Given a set A of n elements, the number of different k-multisets built on elements of A (k is the sum of all the multiplicities of the multiset) is given by the following formula [3]:

$$\binom{n+k-1}{k} \tag{1}$$

and by using it, the following recurrent formula, given in Knuth's book ([3], sect. 2.3.4.4), provides the number $T(n)$ of unlabeled unordered rooted trees of n nodes ($T(1) = 1$):

$$T(n) = \sum_{k_1 \cdot n_1 + k_2 \cdot n_2 + \ldots + k_j \cdot n_j = n-1} \;\; \prod_{i=1,\ldots,\, j} \binom{T(n_i) + k_i - 1}{k_i} \tag{2}$$

Unfortunately, formula (2) is not manageable for an effective computation, because it is based on integer partitions, which grow, according to Hardy-Ramanujan's exponential asymptotic formula [2]. The following Otter's recurrent formula (see [5], formulae (6) and (7)), with $S_n^{(i)} = 0$, for $i > n$, was obtained by advanced analytical methods based on generating functions:

$$S_n^{(i)} = S_{n-i}^{(i)} + T(n+i-1)$$

$$nT(n+1) = 1T(1)S_n^{(1)} + 2T(2)S_n^{(2)} + \ldots nT(n)S_n^{(n)}.$$

We denote by $M(n) = T(n+1)$ the number of agglomerates which, apart the skin membrane, have n membranes (pairs of matching brackets), and by $P(n), N(n), C(n)$, the number of proto-agglomerates, neo-agglomerates, and conglomerates, respectively, having n membranes (apart the skin). Although unlabeled unordered rooted trees and agglomerates are equivalent notions, we will continue to speak in terms of agglomerates, because the intuition behind our analysis is directly related to the notion of a hypermultiset, where a membrane corresponds to the multiset construction. It easy to realize that an agglomerate with n membranes, when it is put inside a further membrane, provides a proto-agglomerate with $n+1$ membranes, while united with the neo-agglomerate [[ ]] provides a neo-agglomerate with $n+1$ membranes. The following lemmas easily follow from the tripartite classification of agglomerates.

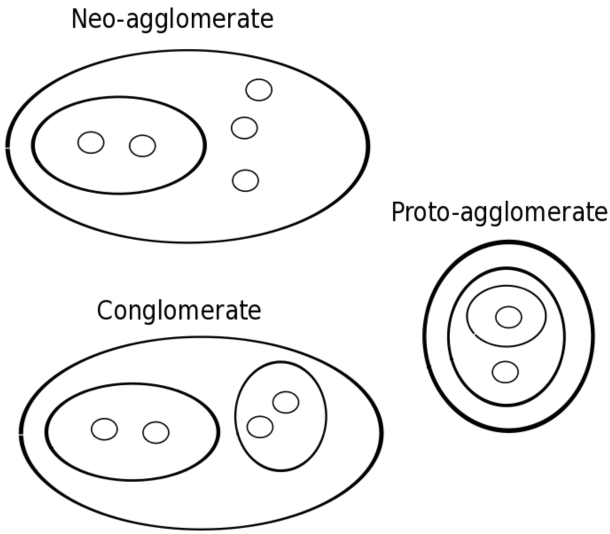

Fig. 1. Different types of agglomerates

Lemma 1. *For* $n > 0$ *the following equations hold:*

$$M(n) = N(n+1) = P(n+1).$$

Lemma 2. *For* $n > 0$

$$M(n+1) = 2M(n) + C(n+1). \tag{3}$$

Proof. The term $2M(n)$ refers to the number of proto-agglomerates and neo-agglomerates with $n+1$ membranes. From the tripartition of agglomerates, the remaining agglomerates of $n+1$ membranes are conglomerates.

Lemma 3. $C(1) = C(2) = C(3) = 0$. *For* $n > 2$

$$C(n+1) < M(n).$$

Proof. Removing the external membrane in a component of a conglomerate with $n+1$ membranes, provides an agglomerate with n membranes, but not all agglomerates with with n membranes can be obtained in this way. Therefore conglomerates with $n+1$ membranes are less than $M(n)$.

Putting together Lemmas 2 and 3 we get the following lemmas.

Lemma 4. *For* $n > 2$

$$2M(n) < M(n+1) < 3M(n).$$

$$2^n < M(n+1) < 3^n.$$

In the number $M(n+1)$ the part $2M(n)$ refers to proto-agglomerates plus neo-agglomerates. Therefore, if $M(n)$ is known, the real problem for the computation of $M(n+1)$ is the evaluation of $M(n+1) - 2M(n) = C(n+1)$, that is, the number of conglomerates with $n+1$ membranes.

We have $M(0) = 1$, $M(1) = 1$, $M(2) = 2$, $M(3) = 4$, as it is indicated in the following schema where skin membranes are denoted by braces.

0	{ }			
1	{ [ ] }			
2	{ [ ] , [ ] }	{ [[ ]] }		
3	{ [ ], [ ], [ ] }	{ [[[ ]]] }	{ [ ], [[ ]] }	{ [[ ], [ ]] }

From Lemma 2 we evaluate immediately $M(4) = 2M(3) + 1 = 9$. In fact, $C(4) = 1$, because there is only a conglomerate with 4 non-skin membranes: {[[ ]], [[ ]]}. Analogously, $M(5) = 2M(4) + 2 = 18 + 2 = 20$, because there are two conglomerates with 5 membranes: {[[ ]], [[ ], [ ]]}, and {[[ ]], [[[ ]]]}. The sequence from $M(1)$ up to $M(12)$ (sequence A000081 of The On-Line Encyclopedia of Integer Sequences [8]) provides the following table.

n	1	2	3	4	5	6	7	8	9	10	11	12
M(n)	1	2	4	9	20	48	115	286	719	1842	4766	12486

Now we consider an enumeration method which is based on the number $C_{min(k)}(n)$ of conglomerates having their smallest components of $k+1$ membranes ($k > 0$).

Lemma 5. *For* $n > 3$

$$C_{min(1)}(n+1) = M(n-1) - M(n-2)$$

Proof. The number $C_{min(1)}(n+1)$ counts the conglomerates of $n+1$ membranes (apart the skin) having [[]] as smallest components. These conglomerates are of two different types: i) those having more than two components, one of them equal to [[]]; ii) those having a component equal to [[]] plus only a further component. The number of conglomerates i) is $C(n-1)$; the number of conglomerates ii) is $M(n-2)$ (after removing [[]] from the $n+1$ original membranes , the skin of the further component is not counted). Therefore, they are $C(n-1) + M(n-2)$. However, $M(n-1) = 2M(n-2) + C(n-1)$, therefore $C(n-1) = M(n-1) - 2M(n-2)$, so that $C(n-1) + M(n-2) = M(n-1) - M(n-2)$.

Lemma 6. *For* $n > 3$

$$C_{min(k)}(n) = 0 \ \ if \ \ k > \ (n-2)/2$$

and

$$M(n+1) = 2M(n) + M(n-1) - M(n-2) + \mathop{}\limits_{k=2,\ (n-1)/2} C_{min(k)}(n+1) \quad (4)$$

Proof. Any conglomerate has at least two components, therefore no conglomerate with a component having more than $(n-2)/2$ membranes can exist (two membranes need for the skins of the components). Therefore, conglomerates having at least 2 membranes inside their components can be partitioned into disjoint classes, by providing a total number of elements given by the summation in the right member of the formula above.

The function M_k, defined by the following equations, will be used in the next lemma.

$$M_k(i) = \begin{cases} M(i-1) & \text{if} \quad i > k+1 \\ 0 & \text{if} \quad 0 < i \leq k+1 \\ 1 & \text{if} \quad i = 0 \end{cases} \tag{5}$$

The computation of terms $C_{min(k)}(n)$ concludes our enumeration method of conglomerates. Let

$$C_{min_{>j}}(m) = \overset{j<i\leq\lfloor(m-2)/2}{\underset{i}{}} \qquad C_{min(i)}(m) \tag{6}$$

Lemma 7. *For* $n > 3$ *and* $(n-2)/2 \geq k \geq 2$

$C_{min(k)}(n) =$

$$\overset{n/(k+1)}{\underset{i=1}{}} \quad \begin{matrix} M(k)+i-1 \\ i \end{matrix} \quad (M_k(n-i(k+1)) + C_{min_{>k}}(n-i(k+1))) \tag{7}$$

Proof. Conglomerates with their smallest components having $k+1$ membranes are of two disjoint types: i) those with one or more components of $k+1$ membranes plus a remaining component with more than $k+1$ membranes (if they exist); and ii) those with one or more components of $k+1$ membranes plus a conglomerate having minimal components with more than $k+1$ membranes (if they exist). For $m > 1$, we define m-multiplicity of a conglomerate as the number of its components with m membranes. Then, the two types i) and ii) of conglomerates are summed in formula (7), for each value i of their $(k+1)$-multiplicity, and their sum is multiplied by a binomial coefficient giving the number of different conglomerates with $(k+1)$-multiplicity i. The definition of the function M_k guarantees that, in the case of a conglomerate with $(k+1)$-multiplicity i which has $i+1$ components, the component, having a number of membranes different from $k+1$, has more than $k+1$ membranes.

Putting together the two previous lemmas we get the final proposition.

Proposition 1. *For any* $n \in \mathbb{N}$*, the number* $M(n+1)$ *is given by formulae (4), (5), (6), and (7) above.*

The following tables provide MATLAB functions computing the enumeration formula asserted in the previous proposition, with full accordance with Sloane's sequence A000081.

```
% CM = computeCM(n,k)
%
% This function calculates C_{min>k}(n)
%
% n +1 = number of membranes
% k +1= size of smallest components


function CM = computeCM(n,k);
CM = 0;
if ((k+1) <= (n-2)/2 )
   for i = k+1 : (n-2)/2
      CM = CM + computeC(n,i);
   end
end
```

```
% MP = computeMP(i,k)
%
% This function calculates M'_k(i)
%
% k +1= size of smallest components
% i = number of minimal components

function MP = computeMP(i,k);
if (i == 0)
   MP = 1;
elseif (i > (k+1))
   MP = computeM(i-1);
elseif (0 < i <= (k+1))
   MP = 0;
end
```

```
% C = computeC(n,k)
%
% This function calculates C_{min(k)}(n) the number of conglomerates of n membranes
% having their smallest components with k+1 membranes.
%
% n +1 = number of membranes
% k +1 = size of smallest components

function C = computeC(n,k);
if (k > (n-2)/2 +1);
   C = 0;
else
   C = 0;
   for i = 1 : n/(k+1)
      C = C + ((factorial(computeM(k)+i-1) /
            (factorial(i) · factorial(computeM(k)-1))) ·
            ((computeMP(n-i·(k+1),k)) +
            computeCM(n-i·(k+1),k)));
   end
end
```

```
% M = computeM(n)
%
% This function calculates the number M(n)
%
% n+1 = number of membranes

function M = computeM(n);
if (n > 0)
   n = n − 1;
end
if (n == 0)
   M = 1;
elseif (n == 1)
   M = 2;
elseif (n == 2)
   M = 4;
elseif (n == 3)
   M = 9;
elseif (n > 3)
   k = 2;
   if (k <= (n − 2))
      if (k > ⌊(n − 1)/2⌋)
         M = 2 · computeM(n) + computeM(n − 1) − computeM(n − 2);
      else
         sum = 0;
         for i = 2 : ⌊(n − 1)/2⌋
            sum = sum + computeC(n + 1, i);
         end
         M = 2 · computeM(n) + computeM(n − 1) −
                computeM(n − 2) + sum;
      end
   end
end
```

Acknowledgments

The author is grateful to Gheorghe Păun for drawing his attention to the combinatorial analysis of membrane structures [6], and to Sara Compri for implementing and testing the enumeration formulae, by computer programs (MATLAB and JAVA) on the values of sequence A000081 of [8].

References

1. Cayley, A.: On the analytical forms called trees, with application to the theory of chemical combinations. Mathematical Papers 9, 427–460 (1875)
2. Hardy, G.H., Ramanujan, S.: Asymptotic Formulae in Combinatory Analysis. Proc. London Math. Soc. 17, 75–115 (1918)
3. Knuth, D.: The Art of Computer Programming. Fundamental Algorithms, vol. 1. Addison Wesley, Reading (1968)

4. Manca, V.: Enumerating Membrane Structures. In: Corne, D., et al. (eds.) WMC 2008. LNCS, vol. 5391, pp. 292–298. Springer, Heidelberg (2009)
5. Otter, R.: The Number of Trees. The Annals of Mathematics, 2nd Ser. 49(3), 583–599 (1948)
6. Păun, G.: Personal Communication (October 1998)
7. Păun, G.: Membrane Computing: an introduction. Springer, Heidelberg (2002)
8. Sloane, N.J.A.: The On-Line Encyclopedia of Integer Sequences. Notices of The American Mathematical Society 59(8), 912–915 (2003)