Counting in Phylogenetics

Artemisa Labi
August 24, 2007

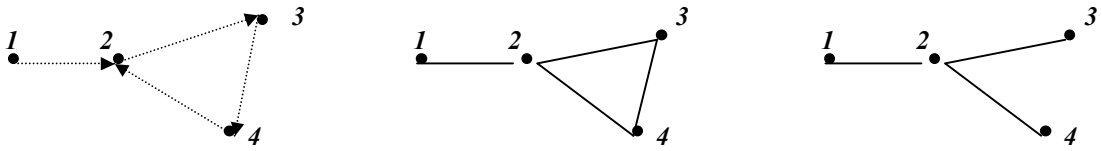
Abstract

In this project algorithms are developed to count the number of evolutionary trees. Trees are assumed to be unrooted and with labelled tips (leaves). Internal nodes are assumed to be unlabelled except the case when counting coalescent trees and fully labelled trees. Multifurcating unrooted trees are counted by adapting a simple recursion developed by J. Felsenstein for rooted trees. It can be used for phylogenies with valency 3 or higher. The algorithm is based on the recurrence relation of the number of trees with *n* leaves and *m* internal nodes. By implementing the recursion the results are tabulated and plotted on the log scale. The results presented are limited due to their size e.g. the number of different phylogenies with 20 nodes easily exceeds 4×10^{23} and consequently for large n (>30) programs are slow.

1. WHAT IS PHYLOGENETICS:

Phylogenetics is the science of the relationship of objects (sequences, individuals) that can be viewed as having an evolutionary history. The most famous example is undoubtedly the tree (phylogeny) relating the species on Earth. A species is a group of individuals in a bi-sexual population whose genetic material is subject to recombination. In tracing or describing the history of species there are used 3 graphs: the phylogeny, the pedigree and the ancestral recombination graph (ARG). The phylogenies are trees with labelled tips and unlabelled internal nodes while pedigrees are trees whose internal nodes are sex-labelled. The ARG show the how genes are passed on from the parents to their descendants.

A graph consists of nodes and edges. A graph can have *k* nodes $\{n_1,n_2,n_3,..,..n_k\}$ and *i* edges which connect pairs of nodes. Edges can be directed or undirected. If the node is directed, one can envision an arrow from the first node to the second node. If the edge is undirected the first and second node can be permuted without changing the graph.



The leftmost graph is directed and has nodes {1,2,3,4} and {(1,2),(2,3),(3,4),(4,2)}. The middle graph is the same as the first one except that it is undirected. The right most is the same as the middle, except the edge (3,4) has been removed. The rightmost graph is then a tree since it is connected (you can find a path from any node to any node) and there are no cycles (you cannot find a series of edges, that bring you back to the same node). The valency of a node is the number of edges touching it. A leaf is only touched by one node so it has a valency of 1.

Graphs used to represent phylogenies have both a continuous and a discrete component. The continuous component describes branch lengths. Branch lengths are specified on a continuous scale by parameterising dates using the real line. The discrete component describes which edges are present in the graph and is often called "the topology". Counting this, has been done by Cayley in non-biological contexts, while Felsenstein (1978) counted a series of elementary cases. Counting pedigrees has been done in restricted cases by Thomas and Cannings (2003) while counting the last combinatorial structure – the ARG – seems to be virgin territory.

The number of topologies of different genealogical structures is beneficial both as a measure of the hardness of problems involving that structure, but can also be a useful stepping stone towards a better understanding of the problem. Also, estimating the phylogeny of a group involves selecting one evolutionary tree among a large number of possibilities. In particular if a computer program is used to evaluate all the possible phylogenies, knowledge of their number is a useful check on whether all the possibilities are in fact being considered. In all the examples given below, the program used to count the phylogenies of different types was implemented using Python (See Appendix)

2. DIFFERENT TYPES OF PHYLOGENIES:

Phylogenies can be *rooted* if there is a specific node that is the most ancient in the tree. In this case the root determines the time flow in the tree so that it is possible for the internal nodes to be ranked. In coalescent trees the branch lengths are proportional to calendar time (Figure 2). If the root is not known or it is not of interest to study it then the tree is *unrooted*.

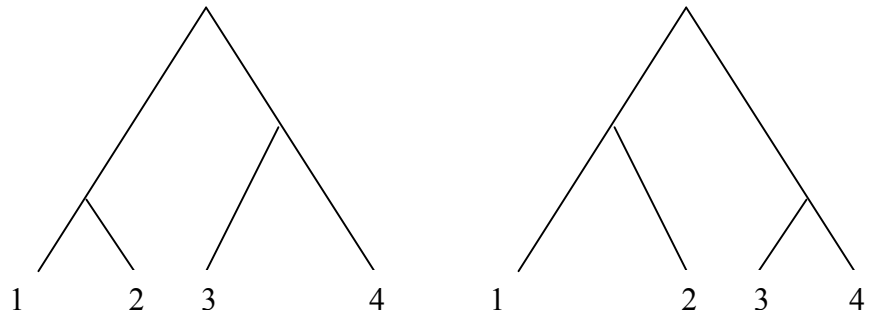


Figure 2. These coalescent trees are different even though they have the same topology. In the first tree 1 and 2 merge after 3 and 4.

Phylogenies often have labelled leaves and unlabelled non-leaf nodes (internal nodes). The degree of a node is the number of edges leaving a node. Multifurcations can happen as well where one node has a degree higher then 2 (Figure 2). In this case the number of different trees depends on the number of labelled leaves 'n' and the number of unlabelled internal nodes 'm'.

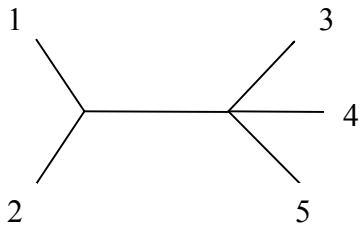


Figure 3. The leftmost internal node to the left has a valency of 3 while the right most internal node has a valency of 4

3.COUNTING BIFURCATING TREES:

In bifurcating trees each of the internal nodes have a valency 3. Counting binary trees can be done very basically using only paper and pen for a start, it can be made systematic and more advanced by using dynamic programming scanning along the sequences or to trace backwards in time. It is unclear how far the dynamic programming approach can count as it is conceivable that simplifications in the problem can be found.

If we want to count unrooted tree topologies, where internal nodes have valency 3, it is clear that there is only one with 3 labelled leaves (Figure 3). Let T(k) be the number of tree topologies with k leaves and internal valency 3. Thus T(3)=1. If a fourth leaf is added to this tree then there are only three possible edges where the new leaf can be placed.

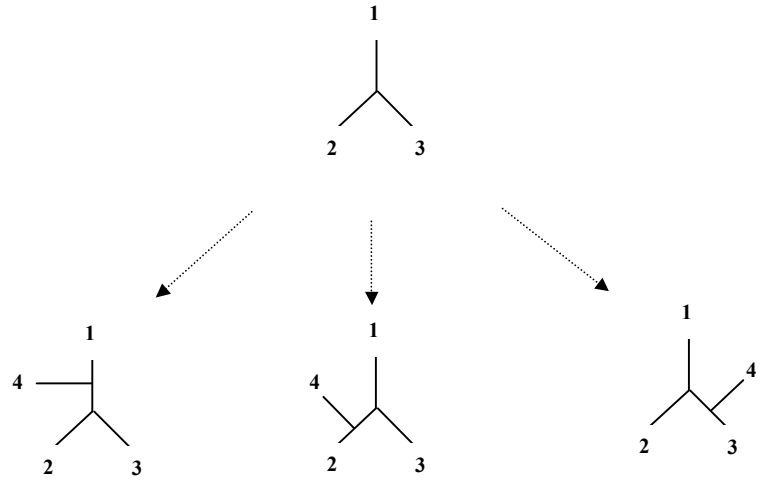


Figure 3. Illustration of simple recursive counting argument. There is only one unrooted tree topology with three labelled leaves and unlabelled inner node. The corresponding number with 4 leaves and only having duplications at inner nodes must be 3 as there are 3 edges at which the 4th leaf can be added. This argument allows tree counting for larger number of leaves.

It is easy to see that an unrooted tree topology with internal nodes valency 3 and k leaves, has 2k-3 edges. This is because as two edges join together in a node the number of edges is reduced by one. So there are k-1 internal nodes leading to tips and k-2 segments leading to internal nodes. Given that each segment has a node at its upper end the total number of edges is 2k-3. Thus in adding a kth leaf to an unrooted

tree topology with internal nodes valency 3 and k-1 leaves, there are 2k-5 possible edges, to which the new leaf can be added. i.e.

$$T(k)=(2k-5)*T(k-1)$$

This recursion allows tabulation of the number of possible topologies and in this simple case a closed formula also exists:

$$T(k)=\frac{(2k-5)!}{(k-3)!*2^{k-3}}$$

This expression is also written as:

$$T(k)=(2k-5)!!$$

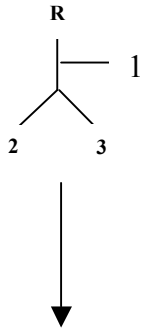
Where the double factorial is used to denote the product of every other integer.
The recursion above is implemented using Python and the results are shown in Table 1.Given that the numbers in Table 1 are very large they are plotted on a log scale (base 10) in Graph 1.

Table 1. **THE NUMBER OF UNROOTED TREES WITH n LABELLED LEAF NODES AND UNLABELLED INTERIOR NODES.**

N	Binary trees	Multifurcating trees
1	-	-
2	-	-
3	1	1
4	3	4
5	15	26
6	105	236
7	945	2752
8	10395	39208
9	135135	660032
10	2027025	12818912
11	34459425	282197824
12	654729075	6939897856
13	13749310575	188666182784
14	316234143225	5617349020544
15	7905853580625	181790703209728
16	213458046676875	6353726042486272
17	6190283353629375	238513970965257728
18	191898783962510625	9571020586419012608
19	6332659870762850625	408837905660444010496
20	221643095476699771875	1852230541036498690662 4

4. COUNTING UNLABELLED MULTIFURCATING TREES:

For rooted trees of n labelled leaves and m unlabelled internal nodes there are m+n edges because each edge leads to either a tip or a node.



In this case the tree has 3 leaves and 2 internal nodes. Let E_r (n, m) be the number of edges of a rooted tree with n unlabeled tips and m internal nodes. $E_r(3, 2)=5$.

When the root is removed the new tree has two less edges and one less node but the number of leaves does not change. Let E_u(n, m) be the number of edges of an unrooted tree with n tips and m internal nodes. $E_u(3, 1)=3$

An unrooted tree with m-1 nodes has two nodes less then the rooted tree with the same number of leaves and m nodes.

$$E_u(n, m-1)= E_r (n, m)-2 \quad (1)$$

It is clear that

$$E_r(n, m-1)=n+m-1 \qquad (2)$$

If we substitute (2) in (1) one we find that $E_u(n, m-1)= E_r(n, m-1)-1$. Therefore any unrooted tree has one edge less than any rooted tree with the same number of tips and internal nodes.

Let T(n, m) be the number of unrooted trees with n labelled tip species and m unlabelled internal nodes. There are two ways in which species n can be added to a tree with n-1 leaves and obtain a tree with m nodes:

- The nth node could be added in one of the nodes in a tree with n-1 leaves and m nodes. For each of the T(n-1, m) trees there are m possibilities where the leaf can be added.

- If a tree with n-1 leaves and m-1 nodes is considered then the m^{th} node can be placed in one of the edges where the nth leaf can arise from. Each of the T(n-1, m-1) trees has m+n-3 edges where the node can be placed

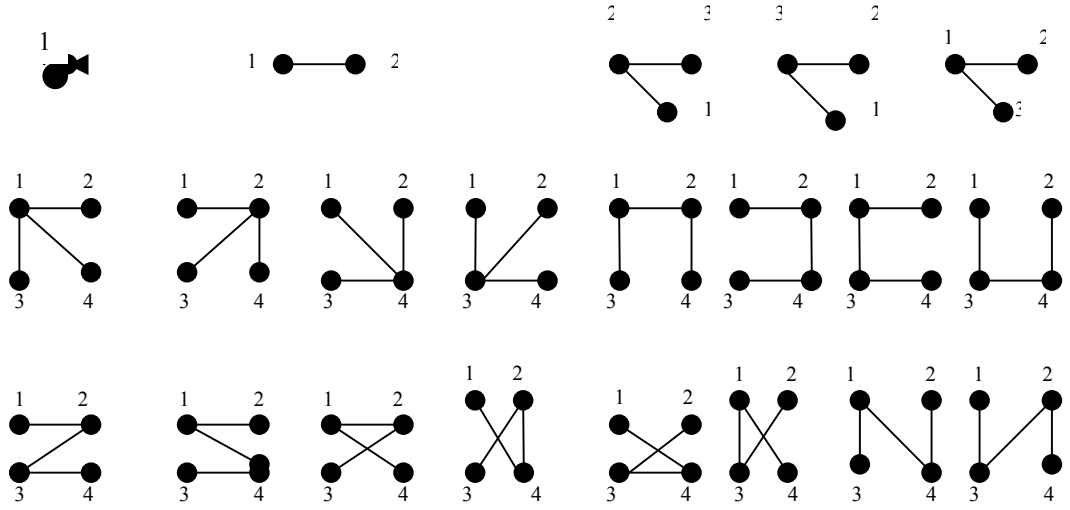
For the case when there is only one leaf and no internal nodes there is only one possible tree and all the other T(1, i)=0 for i >1. Therefore for n>1 the following recursion developed by Felsenstein can be applied:

$$T(n, m)= \begin{cases} mT(n-1,m) \\ +(m+n-3)T(n-1,m-1) \end{cases} \qquad (m>1)$$

This recursion is implemented and results are displayed in Table 1.

5. COUNTING LABELLED UNROOTED TREES

Let T_n be the number of labelled multifurcating trees with n species then simple enumeration gives T_1=1, T_2=1, T_3= 3, T_4=16 with the trees shown below:



Cayley has shown that there are n^{n-2} different labelled trees with n species. To count this using dynamic programming a recursion for partly labelled trees will be used. Let U(n, m) equal the number of trees with n labelled species that can be internal or tip nodes, and m unlabelled internal nodes. To construct a tree with n labelled species and m unlabelled interior nodes then the n th species can be added to a tree in these ways:

- By introducing it as a new labelled internal node into one of the n+m-2 segments of any of the U(n-1, m) trees.
- By placing a new unlabelled internal node in one of the n+m-3 segments in any of the U(n-1, m-1) trees and have species n occur from it.
- By labelling one of the m+1 unlabelled nodes in any of the U(n-1, m+1) trees.
- By adding species n as a new leaf in one of the n+m-1 nodes in any of the U(n-1, m) trees.

For U(1, 0)=1 and for m>1 the number of trees U(1, m)=0. For n>1 the recursion is:

$$U(n, m)= (n+m-3)*U(n-1, m-1) \quad (m>0)$$
$$+ (m+1)*U(n-1, m+1) \quad (n-1>m+1)$$
$$+ (2n+2m-3)*U(n-1, m) \qquad (3)$$

By computing U(n, 0) the following results are obtained that agree with the results obtained using Cayley's Formula.

There is another recursion for counting fully labelled unrooted trees which is shown below:

$$V(n)=n*V(n-1)*\left(\frac{n}{n-1}\right)^{n-3} \qquad (4)$$

Table2. **THE NUMBER OF ROOTED MULTIFURCATING TREES WITH FULLY LABELLED NODES U(n, 0) AND BINARY COALESCENT TREES C(n) WITH *n* LEAVES.**

n	U(n, 0)	C(k)
1	1	1
2	1	1
3	3	3
4	16	18
5	125	180
6	1296	2700
7	16807	56700
8	262144	1587600
9	4782969	57153600
10	100000000	2571912000
11	2357947691	141455160000
12	61917364224	9336040560000
13	1792160394037	728211163680000
14	56693912375296	66267215894880000
15	1946195068359375	6958057668962400000
16	72057594037927936	834966920275488000000
17	2862423051509815793	113555501157466368000000
18	121439531096594251776	17373991677092354304000000
19	5480386857784802185939	2970952576782792585984000000
20	262144000000000000000000	564480989588730591336960000000

6. COUNTING THE NUMBER OF ROOTED TREES WITH RANKED INTERNAL NODES (COALESCENT TREES)

If a binary coalescent tree with k-1 leaves is considered than it contains k-1 periods. Let C(k) be the number of coalescent trees that have k leaves. It is clear that if we put a k^{th} leaf in any of the C(k-1) trees then there are k-1 leaves where the new leaf can be

placed. A new period will occur when the k^{th} leaf is positioned, so there are k possibilities to rank the internal nodes.

$$C(k)= \frac{k*(k-1)*C(k-1)}{2} = \frac{k!*C(k-1)}{2*(k-2)!} = \binom{k}{2} * C(k-1)$$

The division by 2 occurs because of the symmetry present in evolutionary trees (see below). The results obtained are show in Table 2 and Graph 2.

7. COUNTING UNLABELLED TREES WITH INTERNAL NODE DEGREE BOUNDED BY d.

In this case it is assumed that the internal nodes have a degree of at most d. To count these phylogenies it is important to keep track of the number of nodes of different degrees. Let *m* be a list [m_2, m_3, m_4 ….., m_{d-1}, m_d] where m_i is the number of nodes present in a tree with degree *i*. The bound on the degree is one more then the length of the list m.

d=length(m)+1

Let $k=m_2+m_3+\ldots..m_{d-1}+m_d$ so $k= \sum_{i=2}^{d} m_i$

For a given degree and number of leaves there can be more than one possible list. For example if a tree with three leaves and degree at most three is considered then there are two possible lists :
m_1= [2, 0…..0] and m_2= [0, 1, 0…….0] with the respective trees shown below:

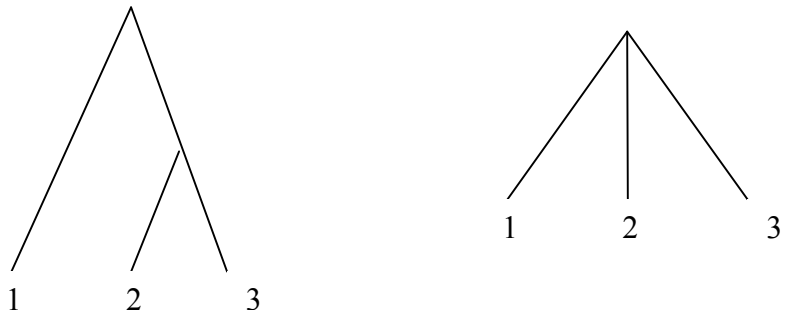


Figure 5. The left tree has two nodes of degree two and no other node of a higher degree. The right tree has one node of degree 3 and no other node of a different degree

There are again two ways in which a tree with n leaves and m nodes can be constructed from a tree with n-1 leaves and m internal nodes:

- There are k+n -2 segments present in a tree with one less internal node and one less leaf. The n^{th} leaf can be placed in any one of these segments present

in a tree that has one less node of degree two then the tree we want to construct.

- The new leaf can be added in one of the nodes of a tree with n-1 leaves. In this case the new tip node can be added only to those nodes that have a degree less than the set bound d.

Clearly for n=1 and d=0 T(1)=1 else for degree higher then 0 no such trees exist. The resulting recursion is:

$$T(n,m)=\begin{cases}(n+k-2)*T(n-1,m_2-1,m_3,m_4,.....m_d)+\\ \sum_{i=3}^{d}(m_{i-1}+1)*T(n-1,m_2...,m_{i-1}+1,m_i-1,...m_d)\end{cases} \quad \textbf{(5)}$$

Let $T_d(n)$ be the number of trees with n leaves and internal node degree bounded by *d*. By computing the number of trees with *n* leaves and all the possible m lists and adding them $T_d(n)$ can be calculated..

$$T_d(n)=\sum_{m1,m2...md} T(n,m)$$

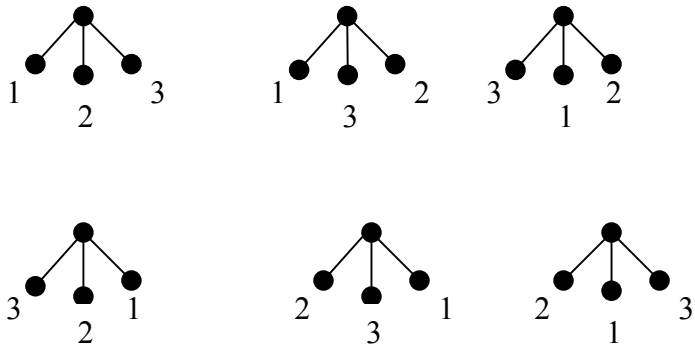
To verify the results from the first recursion another approach was considered for unlabelled trees with bounded degree on internal nodes. Instead of taking into account the ways in which a leaf can be added to a tree with n-1 leaves, it is possible to look at how a whole level (subtree) can be added to construct a tree with n leaves and with nodes that have degree of at most *d*.

In this case it is assumed that the levels are always added to the root node. The recursion for T(n) should go through all the possible valencies of the top node that is being added. For each valency the function goes through the different ways of distributing the leaves on the subtrees.

If binary trees are considered then there are $T_2(m)$ different subtrees with m leaves and degree 2. This subtree will be added to the remaining part of the tree which has (n-m) leaves. There are $T_2(n-m)$ such trees. Also there are $\binom{n}{m}$ ways of labelling the leaven in the subtrees. Therefore the number of different trees with n leaves is:

$$T_2(n)=\frac{1}{2!}\sum_{m=1}^{n-1}T(m)*T(n-m)\binom{n}{m} \quad \textbf{(6)}$$

The expression is divided by the degree factorial because of the symmetry that is present in phylogenies. For example, consider the different rooted trees with 3 leaves and degree exactly 3 that can be formed. There are 3! ways of labelling the leaves with the trees shown below:



It is clear that there are only three different trees present since each of the trees in the bottom level of the diagram is the mirror image of the one directly above it. The number of topologies can be calculated by dividing the number of ways leaves can be labelled by the degree factorial (1 in this example). The number of different trees can be determined by multiplying with $\binom{n}{m}$ which in this case is $\binom{3}{1}$.

From (5) the number of different tree topologies t(n) that arise from adding the subtree to the rest of the tree is:

$$t(n) = \frac{1}{2!} \sum_{m=1}^{n-1} T(m) * T(n-m)$$

For the general case when the trees has nodes with degree at most *d*:

$$T_d(n) = \frac{1}{2!} \sum_{m=1}^{n-1} T(m) * T(n-m) \binom{n}{m} +$$

$$\frac{1}{3!} \sum_{i=2}^{n-1} \sum_{j=1}^{i=1} T(n-i) * T(i-j) * T(j) \binom{n}{i-j, j} +$$

.
.
.
.

$$\frac{1}{d!} \sum_{i2} \sum_{i2} \cdots \sum_{id-1} \sum_{id} T(i_2) * T(i_3) * \ldots\ldots T(i_d) \left(\frac{n!}{\prod_{j=1}^{d} i_j!} \right)$$

Where $\binom{n}{i-j,\,j} = \left(\frac{n!}{(n-i)!(i-j)!(j!)} \right)$ and T (1)=1

In the cases when d=2 and d=n the results obtained are the same as the ones shown in Table1 for binary and multifurcating trees respectively. The results from both programs are concordant and since d ≤ n, the results can be represented in a triangular table.(Table 3)

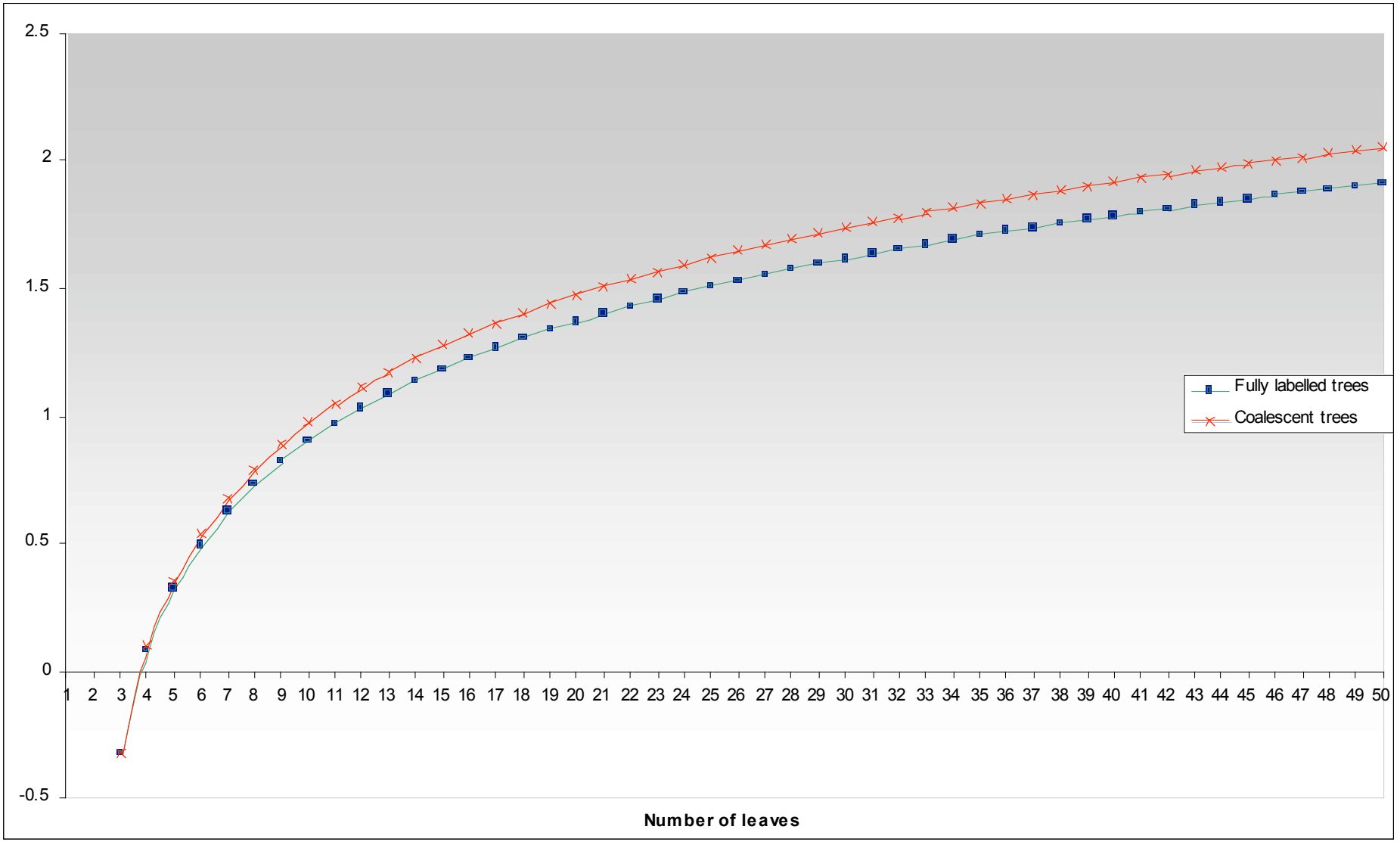
Table 3. THE NUMBER OF ROOTED TREES WITH *n* LEAVES AND INTERNAL NODE DEGREE BOUNDED BY *d*.

	d							
n	3	4	5	6	7	8	9	10
3	4							
4	25	26						
5	220	235	236					
6	2485	2730	2751	2752				
7	34300	38745	39179	39207	39208			
8	559405	649705	659281	659995	660031	660032		
9	10525900	12569095	12799066	12817756	12818866	12818911	12818912	
10	224449225	275556050	281583071	282102128	282136118	282137768	282137823	282137824

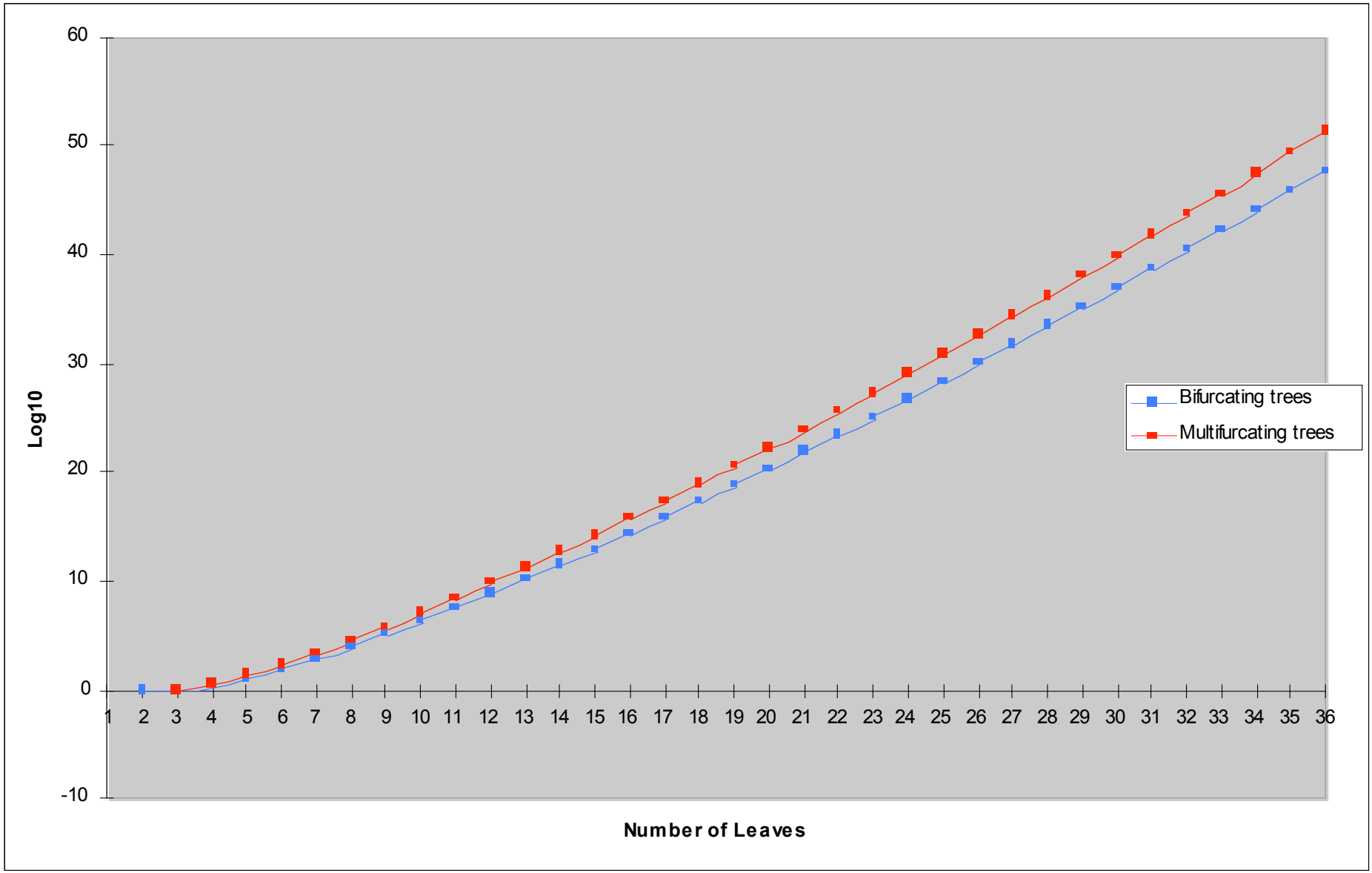
Conclusion:

The results in these tables are useful for taxonomists as a check whether every possible tree is being considered especially when this task is given to a computer. Given that in most problems in taxonomy the unrooted trees are not fully labelled, recursion (3) for unrooted trees with partly labelled internal nodes is more useful then (4) which counts only fully labelled trees.

In the case when rooted trees with bounded degree on their internal nodes are counted the second recursion is more memory efficient then (5) because one value has to be calculated each time. Also because some calculations appear several times it was useful to store the results in a Hash table so that the program could refer to it continuously. From Table 3 and from the recursion it is possible to find the number of trees with internal node degree exactly d.



GRAPH 2. THE LOG10(LOG10) OF THE NUMBER OF TREES WITH *n* LEAVES WHEN THEY ARE FULLY LABELLED TREES AND WHEN THEY ARE COALESCENT TREES.



GRAPH 1. SHOWS THE LOG10 OF THE RESULTS OBTAINED FOR MULTIFURCATING AND BIFURCATING TREES WITH *n* LEAVES.

Appendix:

Python

Python is an interpreted, interactive, object-oriented programming language. It incorporates modules, dynamic typing, very high level dynamic data types, and classes. Python combines remarkable power with very clear syntax and is extensible in C or C++. It is used in scientific and numeric computing such as in Bioinformatics.

Python is a good programming language for beginners because it has a very simple and consistent syntax and a large standard library and, most importantly, using Python in a beginning programming course permits to concentrate on important programming skills such as problem decomposition and data type design. With Python, basic concepts such as loops and procedures can be quickly introduced.

References

- Felsenstein, J (1978) The Number of Evolutionary Trees *Systematic Zoology*, Vol. 27, No. 1. 27-33.
- Griffiths, R.C. (1987). Counting genealogical trees. J.Math.Biol. 25, 422-432.
- Hein, J.J., Schierup, M.H. and Wiuf, C.H. (2005) Gene Genealogies, Variation and Evolution. Oxford University Press, 296 pages.