The average size of ordered binary subgraphs †

Pieter H. Hartel
Computing Science Department, University of Amsterdam
Nieuwe Achtergracht 166, 1018 WV Amsterdam

Abstract
To analyse the demands made on the garbage collector in a graph reduction system, the change in size of an average graph is studied when an arbitrary edge is removed. In ordered binary trees the average number of deleted nodes as a result of cutting a single edge is equal to the average size of a subtree. Under the assumption that all trees with \( n \) nodes are equally likely to occur, the expected size of a subtree is found to be approximately \( \sqrt{\pi \frac{n}{2}} \). The enumeration procedure can be applied to graphs by considering spanning trees in which the nodes that were shared in the graph are marked in the spanning tree. A correction to the calculation of the average is applied by ignoring subgraphs that have a marked root. Under the same assumption as above the average size of a subgraph is approximately \( \sqrt{\pi \frac{n}{2}} - 2(m + 1) \), where \( m \) represents the number of shared nodes and \( m \ll n \).

Key words: binary graphs Catalan statistics combinator graph reduction subgraphs

1. Introduction
The \( \lambda \)-calculus\(^1\) can be viewed as a universal programming language. Its simplicity makes \( \lambda \)-expressions (programmes) amenable to direct mechanical evaluation.\(^2\) Functional programming

† This work is supported by the Dutch ministry of Science and Education, dienst Wetenschapsbeleid.
languages in essence are "sugared" versions of the $\lambda$-calculus. An ordered binary tree provides a natural representation for a $\lambda$-expression. A function application (a node) consists of the juxtaposition of the function (the left descendant of the node) and its argument (the right descendant of the node). A subexpression appears as a subtree, hence arbitrarily complex expressions can be represented. Graph reduction is generally preferred to tree reduction because it allows subexpressions to be shared in stead of copied. This saves both space and time, since a copy of an expression can be made by creating a new pointer to the representation of the expression. This improved efficiency is not without cost.

Let us assume, that initially a $\lambda$-expression is presented for mechanical evaluation, which contains at least one reducible expression (redex). Furthermore let there be a mechanism that decides which redex is to be evaluated next. The evaluation process then consists of a number of discrete reduction steps (e.g. $\beta$-reduction, $\alpha$-conversion), which take the expression to its final form, or at least a number of steps ahead. During this process the expression may take many different forms, which although semantically equivalent, require a varying amount of space for their graphical representation. Each reduction step typically causes some nodes to be deleted and some to be added to the graph. New nodes are added explicitly, for example as a result of $\beta$-reduction, but old nodes become unreachable without explicit notice. The reason is, that unless special precautions are taken, we can not know when the last reference to a possibly shared node is destroyed. This uncertainty has an important consequence for practical implementations of the $\lambda$-calculus and $\lambda$-based languages, because it necessitates a garbage collector, i.e. a device that recuperates storage occupied by parts of the graph that have become unreachable from the root. The cost of garbage collection is considerable. It may take up to an order of magnitude more time to recuperate the storage occupied by a node than it takes to allocate a node from a pool of free nodes. Many methods have been proposed to control the cost of garbage collection.

To arrive at a better understanding of the cost of garbage collection we will count the number of nodes that may be expected to turn into garbage during a reduction step. If we are prepared to
make some assumptions about the structure of the graphs being manipulated, interesting properties can be derived, even if the precise configuration is not known. For instance, under the assumption that all ordered trees with $n$ nodes are equally likely to occur, it can be shown that the average height of such trees is approximately $\sqrt{n\pi}$. The pre-order spanning trees of the graphs that occur during the evaluation of complicated expressions were found to behave in roughly the same way: their average height is proportional to $\sqrt{n\pi}$. Both ways of arriving at an average apparently lead to comparable results.

A method is presented that allows us to extend the results about ordered binary trees (Catalan statistics) to the graphs that occur during graph reduction. As a first approximation we regard graphs as trees, i.e. ignore the effects of sharing completely. This allows us to apply the basic Catalan statistics directly to our graphs (next section). In section 3 the effect of sharing is modelled by marking the nodes in a spanning tree that correspond to shared nodes in the program graph. New counting procedures are developed to take sharing into account. The results give a lower bound and an upper bound on the average number of nodes in a subgraph.

2. Enumeration of ordered binary trees

In the rest of the paper we assume, that the reader is familiar with the analysis as presented by Knuth pp. 388 - 389. To summarise the most important results, the construction method for ordered binary trees and the formulae for $b_n$ and $B(z)$ are reproduced here. An ordered binary tree with $n$ nodes can be built by taking a root and attaching an ordered binary tree with $i$ nodes to the left and an ordered binary tree with $n-i-1$ nodes to the right. To construct the set of ordered binary trees with $n$ nodes, $i$ must be varied over the range $[0..n-1]$ and the construction must be applied recursively to the subtrees. As an example, consider the set of ordered binary trees with 3 nodes shown in figure (1).
The recurrence relation and the boundary conditions for $b_n$, the number of ordered binary trees with $n$ nodes are:

$$b_n = \begin{cases} 
  b_0 b_{n-1} + b_1 b_{n-2} + \cdots + b_{n-2} b_1 + b_{n-1} b_0, & n > 0 \\
  1, & n = 0 
\end{cases}$$

Figure 1: The set of ordered binary trees with 3 nodes

The generating function $B(z)$ for the sequence $\{b_n\}$ is:

$$B(z) = \sum_{n=0}^{\infty} b_n z^n = \frac{1 - \sqrt{1 - 4z}}{2z}$$

The closed form for the number of ordered binary trees with $n$ nodes:

$$b_n = \frac{1}{n+1} \left(\begin{array}{c} 2n \\ n \end{array}\right)$$

With Stirling’s approximation to $n!$ for large $n$ we have:

$$b_n = \frac{4^n}{n \sqrt{n} \pi} + O \left(\frac{4^n}{n^2 \sqrt{n}}\right)$$

This concludes the exposé of the basic results in Catalan statistics.

2.1. The size of ordered binary subtrees

The collection of nodes that become unreachable from the root when an edge is cut is called a subtree. In addition the entire tree is considered as a subtree. As a consequence there is one subtree associated with each node in a tree. Let $s_{n,k}$ be the number of subtrees with $k$ nodes of the set of ordered binary trees with $n$ nodes. Suppose we want to find the number of subtrees with 2
nodes in the set of trees with 3 nodes. Counting subtrees in figure (1) yields the answer $s_{3,2} = 4$.

The configurations (a), (b), (d) and (e) each contribute one to the total. In general it is not practical to draw all configurations in order to count subtrees, hence we must use the construction method to arrive at the desired answer. The configurations (a) and (b) were constructed by attaching all possible configurations with two nodes to the right of the root combined with all possible configurations with zero nodes to the left. Together they contribute $b_0 s_{2,2} + s_{0,2} b_2$ to the number of subtrees with two nodes. Extending this procedure to the remaining configurations yields:

$$s_{3,2} = b_0 s_{2,2} + b_1 s_{1,2} + b_2 s_{0,2} + s_{0,2} b_2 + s_{1,2} b_1 + s_{2,2} b_0$$

$$= 1 \cdot s_{2,2} + 1 \cdot s_{1,2} + 2 \cdot s_{0,2} + 2 \cdot s_{0,2} b_2 + s_{1,2} \cdot b_1 + s_{2,2} \cdot b_0 = 4$$

The general rule is derived from the enumeration formula for ordered binary trees: where $b_i$ is multiplied by $b_{n-i-1}$ in (1), we must now multiply $s_{i,k}$ by $b_{n-i-1}$. This corresponds to the number of subtrees with $k$ nodes in the set of trees with $i$ nodes, multiplied by the number of times this configuration occurs to the left of the root. Similarly enumeration of subtrees with $k$ nodes in the right subtree yields the term $b_i s_{n-i-1,k}$. Since a tree is regarded as a subtree we have $s_{n,n} = b_n$. The recurrence relation and the boundary conditions for $s_{n,k} = \text{the number of subtrees with } k \text{ nodes among the set of trees with } n \text{ nodes are:}$

$$s_{n,k} = \begin{cases} b_0 s_{n-1,k} + b_1 s_{n-2,k} + \ldots + b_{n-2} s_{1,k} + b_{n-1} s_{0,k} + s_{0,k} b_{n-1} + s_{1,k} b_{n-2} + \ldots + s_{n-2,k} b_1 + s_{n-1,k} b_0, & 0 < k < n \\ b_n, & n \geq 0 \land k = n \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

To derive the closed form of (3), two auxiliary results are needed. Both can be proved by induction on $n$ from (3). The first yields the number of subtrees with one node:

$$s_{n,1} = n \cdot b_{n-1} \quad (4)$$

Proof: if $n=1$ we have $s_{1,1} = b_1 = 1 \cdot b_0$, since $b_0 = b_1 = 1$. Application of the induction
hypothesis $\forall i \in 1..n-1 : s_{i,1} = i \ b_{i-1}$ to (3) yields

$$s_{n,1} = b_0 s_{n-1,1} + \cdots + b_{n-2} s_{1,1} + b_{n-1} s_{0,1} +$$

$$s_{0,1} b_{n-1} + s_{1,1} b_{n-2} + \cdots + s_{n-1,1} b_0$$

$$= b_0 (n-1) b_{n-2} + \cdots + b_{n-2} b_0 + b_{n-1} 0 +$$

$$0 b_{n-1} + b_0 b_{n-2} + \cdots + (n-1) b_{n-2} b_0$$

$$= n (b_0 b_{n-2} + b_1 b_{n-3} + \cdots + b_{n-2} b_0)$$

$$= n \ b_{n-1}$$

The second auxiliary result (5) can be proved in a similar fashion.

$$s_{n,k} = s_{n-k+1,1} s_{k,k} , \quad 0 < k \leq n \quad (5)$$

Combining (3), (4) and (5) we find:

$$s_{n,k} = (n-k+1) b_{n-k} b_k , \quad 0 < k \leq n \quad (6)$$

The generating function for the sequence $\langle s_{n,k} \rangle$ with a fixed value of $k$ is:

$$S_k(z) = \sum_{n=0}^{\infty} s_{n,k} z^n = \sum_{n=0}^{\infty} (n-k+1) b_{n-k} b_k z^n = b_k z^k \frac{d}{dz} (z B(z)) = \frac{b_k z^k}{\sqrt{1-4z}} , \quad k > 0$$

Since every subtree is uniquely determined by its top node, the total number of subtrees must be equal to the total number of nodes in the set of $b_n$ trees. This can be proved directly from (3).

$$\sum_{k=1}^{n} s_{n,k} = n \ b_n$$

The values of $s_{n,k}$ grow quickly as $n$ increases. Table (1) shows the values for small $n$ and $k$.

The numbers on the diagonal are the values for $b_n$ (Catalan numbers) since $s_{n,n} = b_n$. 
Table 1: $s_{n,k}$ for small values of $n$ and $k$.

2.2. Average size of subtrees

Under the assumption that all ordered binary trees with $n$ nodes occur with the same probability, we find for the size of the average subtree:

$$s_n = \frac{\sum_{k=0}^{n} k s_{n,k}}{n b_n} = \frac{\sum_{k=0}^{n} k (n-k+1) b_{n-k} b_k}{n b_n}$$

(7)

The generating function for the sequence $\langle \sum_{k=0}^{n} k s_{n,k} \rangle$ is:

$$\sum_{n=0}^{\infty} \sum_{k=0}^{n} (n-k+1) b_{n-k} k b_k z^n = z B'(z) \frac{d}{dz} (z B(z))$$

$$= \frac{1}{1-4z} + \frac{1}{2z} - \frac{1}{2z \sqrt{1-4z}}$$

$$= \sum_{n=0}^{\infty} 4^n z^n + \frac{1}{2z} - \frac{1}{2z} \sum_{n=0}^{\infty} \left( \frac{-1}{n} \right) (-4)^n z^n$$
\[= \sum_{n=0}^{\infty} \left( 4^n - \frac{1}{2} \left( \frac{-\sqrt{2}}{n+1} \right) (-4)^{n+1} \right) z^n\]

\[= \sum_{n=0}^{\infty} \left( 4^n - (2n+1) b_n \right) z^n\]

With this result (7) becomes:

\[s_n = \frac{4^n - (2n+1) b_n}{n b_n}, \quad n > 0\]

Let us apply this result to graph reduction. Using Stirling's approximation we find for large \(n\) that \(s_n \approx \sqrt{\pi n}\). Hence a reduction step may be expected to delete a subgraph containing \(\sqrt{\pi n}\) nodes from the programme graph. Unless a reduction step adds a similar number of new nodes to the graph, it is reduced in size until the average number of nodes added per reduction step and that deleted are in equilibrium. From our experiments we know that such an equilibrium may be reached when far fewer than \(\sqrt{\pi n}\) nodes are added per reduction step. The effect of sharing that is completely ignored in the current approximation, should at least be taken into account.

3. Trees with marked nodes

Our objective is to count the number of nodes that may be deleted from a programme graph by cutting a single edge. To be able to apply the results obtained so far, we must eliminate the effect of the extra edges that turn a tree into a graph. In general there are many subsets of edges that may be considered redundant. From our experience with the determination of the average height of spanning trees we found that the results are virtually independent of the particular set of edges that is removed. We have no reason to believe that in the current case the results should be sensitive to the choice of spanning tree, since the averaging processes in determining the expected height of trees and the expected size of subtrees are so similar. The first step is therefore to consider an arbitrary spanning tree of a graph in stead of the graph itself. Such a tree has the same set of nodes as the graph and one fewer edge than the graph has nodes. In the previous
section we saw that without further refinement the results with binary subtrees are still unsatisfactory. We therefore introduce a “marking” mechanism by which we can remember which nodes were originally shared. The enumeration procedures must take this marking into account. There are two issues that deserve further attention. In the first place we have to cope with nodes that have a different number of incident edges. In the second place the origin of the edges incident upon the same node may be expected to play a role. Since we wish to study the effect of removing a single edge from a graph, we need not be too concerned about the actual number of incident edges to a node as long as it is greater than one. Usually a shared node remains reachable from the root if an incident edge is removed. A connected component in a graph may be attached to the rest of the graph via a single edge. If that edge is cut, the entire component becomes unreachable from the root. However, since our experiments indicate that cycles are rare in practical graph reduction, we ignore this effect and treat a shared node as one that can not be removed by cutting a single edge.

![Diagram of two binary graphs with one marked node](image)

Figure 2: Two binary graphs with one marked node

The origin of the edges can not be completely ignored. Figure (2-a) shows, that a marked node may originate from two different kinds of graphical structure. If the left descendant of the root (marked with the letter "x") is removed, the right subgraph of node "x" also becomes disconnected, since all edges incident upon the shared node are successively removed. The right subgraph of node "x" remains connected if it is removed from the graph shown in figure (2-b). The marked nodes in trees corresponding to the graph as in figure (2-a) are called removable, the type of marking as in figure (2-b) is called permanent. We will study both kinds of marking.
Based on the graphical configurations as illustrated in figure (2) we develop the rules by which subtrees of marked trees are enumerated. Ultimately we are interested in the expected size of subtrees if all trees are equally likely to occur. The averaging process therefore considers every node once as the root of a subtree that must be counted:

- If the root of a subtree is marked the subtree is not counted, regardless of the type of marking. This is in accordance with the origin of a marked node as a shared node in a programme graph.

- If the root of a subtree is not marked, the subtree is counted, including the removable marked nodes and their subtrees but excluding the permanent marked nodes and their subtrees.

Consider as an example the trees of figure (3). If the marking in figure (3-a) is considered permanent, there are 3 subtrees with one node, 1 subtree with two nodes and 1 subtree with four nodes. If the marking is considered removable, there are 3 subtrees with one node, 1 with four and 1 with six nodes. Similarly with permanent marking figure (3-b) shows 3 subtrees of one and 1 subtree of two nodes. With removable marking we count 3 subtrees of one, 1 of two and 1 of six nodes.

![Diagram](image)

Figure 3: Two marked binary trees

A more careful consideration of the graphical structure in figure (2) reveals that the permanent or removable status of a marked node is a property that varies with the choice of the first node to be deleted. For example if rather than the node "x" we remove the root in figure (2-b), even the permanent node and its subtree should disappear. However we can afford to ignore this
complication, since the expected size of subtrees with only permanent or only removable marked nodes form the extremes of a range that encompasses realistic values of the expected number of nodes to be deleted in programme graphs (provided the other assumptions prove to be realistic).

It will be shown, that in a tree with only removable marked nodes, the enumeration for unmarked trees can be modified in a straightforward manner by discounting all subtrees with a marked root. Permanent marking causes considerable complication since each subtree has to be "searched" for permanent marked nodes, which "truncate" the subtree to a smaller size. First the results are generalised to reason about trees with one removable or permanent marked node. This allows us to develop the necessary intuition to solve the problem if an arbitrary number of nodes is marked.

3.1. The size of subtrees in trees with one removable marked node

In the collection of the \( n b_n \) trees with one removable marked node, each subtree appears once with a marked root and \( n-1 \) times with an unmarked root. Since the total number of subtrees with \( k \) nodes in the \( n b_n \) trees with \( n \) nodes is \( n s_{n,k} \), the number of subtrees without marked nodes is:

\[
r_{n,k} = (n-1) s_{n,k}, \quad n > 0
\]

The total number of subtrees must be equal to the total number of nodes in all \( n b_n \) trees, minus the number of marked nodes:

\[
\sum_{k=1}^{n} r_{n,k} = n (n-1) b_n
\]
3.2. Average size of subtrees in trees with one removable marked node

Making the same assumption as before about the probability distribution, we find that the average size of subtrees in trees with one removable marked node is:

\[
\bar{r}_n = \frac{\sum_{k=0}^{n} k r_{n,k}}{n^2 b_n} = \frac{(n-1) \sum_{k=0}^{n} k s_{n,k}}{n^2 b_n} = \left[1 - \frac{1}{n}\right] s_n, \quad n > 0
\]  \hspace{1cm} (8)

3.3. The size of subtrees in trees with one permanent marked node

The construction method for trees is extended to take the location of the marked node into account. Any one of the \( n \) nodes in a tree may be marked. Therefore, the total number of trees that must be considered is \( n b_n \). The recurrence relation of (1) is rewritten to reflect this situation:

\[
n b_n = \left[ b_0 (n-1) b_{n-1} + 0 b_0 b_n \right] + \left[ b_1 (n-2) b_{n-2} + 1 b_1 b_{n-2} \right] + \cdots + \left[ b_{n-1} 0 b_0 + (n-1) b_{n-1} b_0 \right] + b_n
\]  \hspace{1cm} (9)

In the pairs of terms, the first subterm corresponds to the configuration with the marked node residing in the right subtree. The second subterm has the same relation to the left subtree. Both subterms therefore represent configurations with an unmarked root. The last term of (9) enumerates the \( b_n \) trees of \( n \) nodes with a marked root.

The recurrence relation and boundary conditions for \( p_{n,k} \) the number of subtrees with \( k \) nodes in the set of trees with \( n \) nodes of which one is permanent marked can now be formulated by enumerating the relevant number of subtrees in each term of (9):
The first two subterms of (10) correspond to configurations with an unmarked root. The marked node either resides in the left subtree or in the right subtree. The second subterm accounts for the fact that if one subtree (say with size $i$) contains the marked node, the other (with size $n-i-1$) is unmarked. This gives a contribution of $i$ $b_i s_{n-i-1,k}$ subtrees, taken over all possible values of $i$. The factor $i$ originates from the fact that the marked node may reside at any one of the $i$ places in the subtree.

If the root of the tree is marked, the situation resembles that of the unmarked trees. This accounts for the subterm $s_{n,k}$, but with $k < n$, since there are no subtrees with $n$ nodes in any marked tree with $n$ nodes. The last subterm of (10) takes into account, that each subtree with a marked root of size $k$ "prunes" a branch of the main tree, such that an unmarked tree with $n-k$ nodes remains.

From (10) we find the open form of the generating function for the sequence $<p_{n,k}>$ with a fixed value of $k$:

$$P_k(z) = \sum_{n=0}^{\infty} \left( 2 \sum_{i=0}^{n-1} b_i p_{n-i-1,k} + 2 \sum_{i=0}^{n-1} i b_i s_{n-i-1,k} + s_{n,k} + s_{n,n-k} \right) z^n$$

$$- \left( s_{k,k} + s_{k,0} \right) z^k$$

(11)

The correction term in (11) is necessary to compensate for $p_{k,k}$, which if the recurrence relation in (10) is used for $k=n$ (hence outside its domain) yields $s_{k,k}+s_{k,0}$ instead of 0.
From (6) we find that:
\[ s_{n,k} + s_{n,n-k} = (n-k+1) b_{n-k} b_k + (k+1) b_k b_{n-k} = (n+2) b_k b_{n-k} \]

Substitution of this result in (11) yields:
\[
P_k(z) = \sum_{n=0}^{\infty} \left( 2 \sum_{i=0}^{n-1} b_i p_{n-i-1,k} + 2 \sum_{i=0}^{n-1} i b_i s_{n-i-1,k} + (n+2) b_k b_{n-k} \right) z^n - (k+2) b_k b_0 z^k
\]

\[
= 2 z B(z) P_k(z) + 2 z^2 B'(z) S_k(z) + \frac{b_k}{z} \frac{d}{dz} \left[ z^{k+2} (B(z) - 1) \right]
\]

This equation can be solved for \( P_k(z) \) yielding:
\[
P_k(z) = \frac{1}{1 - 2z B(z)} \left[ 2 z^2 B'(z) S_k(z) + \frac{b_k}{z} \frac{d}{dz} \left[ z^{k+2} (B(z) - 1) \right] \right]
\]

\[
= b_k z^k \left\{ \frac{k+1}{2z} \left[ \frac{1}{\sqrt{1-4z}} - 1 \right] - \frac{k+1}{\sqrt{1-4z}} + \frac{2z}{(1-4z)^{1/2}} \right\}
\]

\[
= b_k z^k \left\{ \frac{k+1}{2} \sum_{n=0}^{\infty} \left( -\frac{1}{2} \right)^n (n+1) z^n - (k+1) \sum_{n=0}^{\infty} \left( -\frac{1}{2} \right)^n (4z)^n 
\right. 
\left. + 2 \sum_{n=1}^{\infty} \left( -\frac{1}{2} \right)^n (-4z)^n \right\}
\]

\[
= \sum_{n=0}^{\infty} (n+2) (n-k) b_{n-k} b_k z^n
\]

Hence:
\[
p_{n,k} = (n+2) (n-k) b_{n-k} b_k , \quad 0 < k \leq n
\]

The total number of subtrees must be equal to the total number of nodes in all \( n \ b_n \) trees minus the number of marked nodes:
3.4. Average size of subtrees in trees with one permanent marked node

With a uniform probability distribution, the average size of subtrees in trees with one permanent marked node is:

\[
\tilde{\beta}_n = \frac{\sum_{k=0}^{n} k p_{n,k}}{n^2 b_n} = \frac{\sum_{k=0}^{n} k (n+2) (n-k) b_{n-k} b_k}{n^2 b_n} 
\]  

(13)

The generating function for the sequence \(\sum_{k=0}^{n} k p_{n,k}\) is:

\[
\sum_{n=0}^{\infty} \sum_{k=0}^{n} (n+2) (n-k) b_{n-k} b_k z^n = \frac{1}{z} \frac{d}{dz} \left( z^2 B'(z) \right)^2 
\]

\[
= \frac{2 (1-2z)}{(1-4z)^2} - \frac{2}{(1-4z)^{1/2}} 
\]

(14)

After simplification of the binomial coefficient, the result can be substituted in (13) such that:

\[
\tilde{\beta}_n = \frac{(n+2) 4^n - (2n+1) (2n+2) b_n}{n^2 b_n}, \quad n > 0
\]

3.5. The size of removable marked subtrees

In the set of \(\binom{n}{m} b_n\) trees with \(m\) removable marked nodes, each subtree appears \(\binom{n-1}{m-1}\) times with a marked root. The total number of subtrees with \(k\) nodes in the \(\binom{n}{m} b_n\) trees is \(\binom{n}{m} s_{n,k}\).
Therefore $r_{n,k}^m$ is the number of subtrees with $k$ nodes in the set of trees with $n$ nodes, of which $m$ are removable marked is:

$$r_{n,k}^m = \binom{n}{m} - \binom{n-1}{m-1} \quad s_{n,k} = \binom{n-1}{m} \quad n > 0$$

The total number of subtrees must be equal to the total number of nodes in the set of $\binom{n}{m} b_n$ trees, minus the number of marked nodes:

$$\sum_{k=1}^{n} r_{n,k}^m = \binom{n}{m} (n-m) b_n$$

### 3.6. Average size of removable marked subtrees

With a uniform probability distribution, the average size of subtrees in removable marked trees is:

$$\overline{r}_n = \frac{\sum_{k=0}^{n} k \cdot r_{n,k}^m}{\binom{n}{m} b_n} = \frac{\binom{n-1}{m} \sum_{k=0}^{n} k \cdot s_{n,k}}{\binom{n}{m} n b_n} = \left[1 - \frac{m}{n}\right] \overline{s}_n, \quad n > 0$$

### 3.7. The size of permanent marked subtrees

The generalisation of the result to trees with an arbitrary number of permanent marked nodes follows along the lines set out in the previous paragraphs. We commence by characterising the subtrees of all $\binom{n}{m} b_n$ marked trees, by the way the $m$ permanent marked nodes are distributed over the total $n$ nodes. Using Vandermonde’s convolution and (1) we find that:

$$\binom{n}{m} b_n = \binom{n-1}{m} b_n + \binom{n-1}{m-1} b_n$$
\[ t_{nm,k} = \sum_{i=0}^{m} \sum_{j=0}^{n-1} \left[ \binom{j}{i} b_j \binom{n-j-1}{m-i} b_{n-j-1} \right] b_{n-j-1} \] (16)

The first sum in (16) corresponds to configurations with an unmarked root, the second to those with a marked root. In both “inner” sums, the index \( j \) ranges over all possible combinations of subtrees. The “outer” sums distribute the marked nodes over the subtrees, starting with all marked nodes to the left and none to the right, one to the left and all but one to the right etc.

Looking at figure (3) again and interpreting the marked nodes as permanent, it appears that there are two kinds of subtrees eligible to be counted. If the root of the main tree is unmarked, it is counted as a subtree. This subtree is called a top-tree. It will prove useful, to extend the definition of top-trees to the case where the root of main tree is marked. In that case a top-tree is considered to have 0 nodes. The remaining subtrees (i.e. non top-trees) are isolated from the root by a marked node. In figure (3-a) there is one subtree of either kind and in (3-b) there are two isolated subtrees (if subtrees of top-trees are ignored). The recurrence relation and the boundary conditions for \( t_{nm,k} \) = the number of top-trees with \( k \) nodes in the set of trees with \( n \) nodes, of which \( m \) nodes are permanent marked are:

\[
\begin{cases}
  t_{nm,k}^{m} = \\
  \left\{ \begin{array}{l}
    \sum_{i=0}^{m} \sum_{j=0}^{k-1} \sum_{h=0}^{n-j-1} t_{j,h}^{m-i} t_{n-j-1,k-h-1}^i, & n > 0 \wedge m > 0 \wedge k > 0 \wedge k+m \leq n \\
    \binom{n-1}{m-1} b_n, & n > 0 \wedge m = 0 \wedge k = 0 \\
    b_n, & n > 0 \wedge m = 0 \wedge k = n \\
    0, & \text{otherwise}
  \end{array} \right. 
\end{cases} 
\] (17)

The recurrence in the above expression draws upon the “unmarked root” sum in (16), with the constraint that if the left subtree has \( h \) nodes, the right subtree must host the remaining \( k-h-1 \) nodes. The second clause in (17) gives the number of times that the root is marked. This is the number of times a top-tree with 0 nodes must be counted in a marked tree.

We conjecture that from (17) it can be proved by induction:
\[ t_{n,k}^m = \frac{(k+1)}{n-k} \binom{n+m}{m-1} \binom{2n-2k}{n-k-m} b_k, \quad n > 0 \land m > 0 \land k \geq 0 \land k + m \leq n \] (18)

With this result we can formulate the recurrence relation and the boundary conditions for \( p_{n,k}^m \) the number of subtrees with \( k \) nodes in the set of trees with \( n \) nodes, of which \( m \) nodes are permanent marked:

\[
\begin{align*}
   p_{n,k}^m &= \begin{cases} 
   q_{n,k}^m + q_{n,k}^{m-1} + t_{n,k}^m, & n > 0 \land m \geq 0 \land k > 0 \land k + m \leq n \\
   1, & n = 0 \land m = 0 \land k = 0 \\
   0, & \text{otherwise}
   \end{cases} \\
   q_{n,k}^m &= 2 \sum_{j=0}^{n-k-1} \sum_{i=0}^{m} \binom{j}{i} b_j p_{n-j-1,k}^{m-i}
\end{align*}
\] (19) (20)

Where \( q_{n,k}^m \) is defined as:

\[
q_{n,k}^m = 2 \sum_{j=0}^{n-k-1} \sum_{i=0}^{m} \binom{j}{i} b_j p_{n-j-1,k}^{m-i}
\] (20)

The first two subterms in (19/20) exhibit a strong resemblance with (16). The first subterm corresponds to configurations with an unmarked root, the second subterm to those with a marked root. The top-trees contribute the term \( t_{n,k}^m \). Since \( p_{n,k}^0 = s_{n,k} \), we must have that \( p_{0,0}^0 = 1 \). We will proof by generalised induction over \( n \) and \( m \) (with \( k \) fixed) that:

\[
\begin{align*}
   p_{n,k}^m &= \binom{n+m+1}{m} \binom{2n-2k}{n-k-m} b_k, \quad n > 0 \land m \geq 0 \land k > 0 \land k + m \leq n
\end{align*}
\] (21)

In section 2.1 we have proved that (21) holds for \( m = 0 \lor n = 0 \). By the generalised induction principle we may assume that (21) holds for \( 0 < m' < m \) and \( 0 < n' < n \). Substitution of (21) in (20) yields:

\[
q_{n,k}^m = 2 b_k \sum_{j=0}^{n-k-1} \sum_{i=0}^{m} \binom{j}{i} b_j \binom{n-j+m-i}{m-i} \binom{2n-2j-2k-2}{n-j-k-1-m+i}
\]

Application of (22), see exercise 31, page 70 in Knuth's book, which states that:

\[
\begin{pmatrix} a \\ c \end{pmatrix} \begin{pmatrix} b \\ d \end{pmatrix} = \sum_{x} \begin{pmatrix} c-a+b \\ x \end{pmatrix} \begin{pmatrix} d+a-b \\ d-x \end{pmatrix} \begin{pmatrix} a+x \\ c+d \end{pmatrix}, \quad \text{integer } c \geq 0, \text{ integer } d \geq 0
\] (22)

and changing the order of summation yields:
\[ q_{n,k}^m = 2 b_k \sum_{x=0}^{k+1} \sum_{j=0}^{n-k-1} \sum_{i=0}^{k+l} b_j \left( \begin{array}{c} k+1 \\ x \end{array} \right) \left( \begin{array}{c} n-j-2k-2 \\ m-x-i \end{array} \right) \left( \begin{array}{c} 2n-2j-2k-2+x \\ n-j-k-1 \end{array} \right) \]

Application of Vandermonde’s convolution to perform the summation over \( i \) and substitution of (2) yields:

\[ q_{n,k}^m = 2 b_k \sum_{x=0}^{k+1} \frac{1}{2j+1} \left( \begin{array}{c} 2n-2j-2k-2+x \\ n-j-k-1 \end{array} \right) \left( \begin{array}{c} n-2k-2 \\ m-x \end{array} \right) \]

Application of Rothe’s non-symmetric addition theorem\(^9\) to perform the summation on \( j \) yields:

\[ q_{n,k}^m = 2 b_k \sum_{x=0}^{k+1} \left( \begin{array}{c} k+1 \\ x \end{array} \right) \left( \begin{array}{c} 2n-2k-1+x \\ n-k-1 \end{array} \right) \left( \begin{array}{c} n-2k-2 \\ m-x \end{array} \right) \]

Using (22) again but in opposite direction to perform the summation on \( x \) we obtain:

\[ q_{n,k}^m = 2 b_k \left( \begin{array}{c} 2n-2k-1 \\ n-k-m-1 \end{array} \right) \left( \begin{array}{c} n+m+1 \\ m \end{array} \right) \]

With (18) and (19) it is readily verified that (21) holds for \( n \) and \( m \), which concludes the proof of (21).

The total number of subtrees in all permanent marked trees must be equal to the number of unmarked nodes:

\[ \sum_{k=1}^{n} p_{n,k}^m = \left( \begin{array}{c} n \\ m \end{array} \right) (n-m) b_n \]

3.8. Average size of permanent marked subtrees

For a uniform probability distribution of trees we have:

\[ \overline{p_n} = \frac{\sum_{k=0}^{n} k p_{n,k}^m}{\sum_{k=0}^{n} \frac{n}{m} \left( \begin{array}{c} n+m+1 \\ m \end{array} \right) \left( \begin{array}{c} 2n-2k \\ n-k-m \end{array} \right) b_k} = \frac{\sum_{k=0}^{n} k \left( \begin{array}{c} n \\ m \end{array} \right) b_n}{\sum_{k=0}^{n} \frac{n}{m} \left( \begin{array}{c} n \\ m \end{array} \right) b_n} \]

Unfortunately, there is no simple general solution for the sum in the numerator of (23). To see this, let \( \Delta_n^m \) be defined as:
\[
\Delta^m_n = \frac{\sum_{k=0}^{n} k p_{n,k}^{m-1}}{\binom{n+m}{m-1}} - \frac{\sum_{k=0}^{n} k p_{n,k}^m}{\binom{n+m+1}{m}}, \quad 0 < m < n
\]

With (21) and \( m \) fixed it can be proved by induction on \( n \) that:

\[
\Delta^m_n = \binom{2n+1}{n-m}
\]

If the sum of \( \Delta^m_n \) for all possible values of \( m \) is calculated, we find that all but the first and last terms cancel out:

\[
\sum_{i=1}^{m} \Delta^i_n = \sum_{k=0}^{n} \frac{k p_{0,n,k}^{0}}{\binom{n+1}{0}} - \sum_{k=0}^{n} \frac{k p_{1,n,k}^{1}}{\binom{n+2}{1}} + \sum_{k=0}^{n} \frac{k p_{1,n,k}^{0}}{\binom{n+2}{1}} - \ldots + \sum_{k=0}^{n} \frac{k p_{m-1,n,k}^{m-1}}{\binom{n+m}{m}} - \sum_{k=0}^{n} \frac{k p_{m,n,k}^{m}}{\binom{n+m+1}{m}}
\]

Using this result and the fact that \( p_{0,n,k}^{0} = s_{n,k} \), we calculate \( \sum_{k=0}^{n} k p_{n,k}^m \) we obtain:

\[
\sum_{k=0}^{n} k p_{n,k}^m = \binom{n+m+1}{m} \left( \sum_{k=0}^{n} k s_{n,k} - \sum_{i=1}^{m} \Delta^i_n \right)
\]

\[
= \binom{n+m+1}{m} \left( 4^n - \sum_{i=0}^{m} \binom{2n+1}{n-i} \right) \quad (24)
\]

\[
= \binom{n+m+1}{m} \left( 4^n - \sum_{i=0}^{n} \binom{2n+1}{i} + \sum_{i=0}^{n-m-1} \binom{2n+1}{i} \right)
\]

For sums of type \( \sum_{k=0}^{n} \binom{m}{k} \), where \( n < m \) no simple solution is known (see Knuth's book, page 64). The best that can be done is to compute \( \sum_{k=0}^{n} k p_{n,k}^m \) for specific values of \( m \). Although this may be done conveniently for small values of \( m \) using (24), the method used in the previous sections is employed once more, to show why the problem has no simple solution. For instance let \( m = 2 \). The generating function for the sequence \( \sum_{k=0}^{n} k p_{n,k}^{2} \) with a fixed value of \( k \) can be
found from $B(z)$ by observing that:

$$\binom{2n-2k}{n-k-2} z^{n-k} = \frac{(n-k)(n-k-1)}{(n-k+2)} b_{n-k} z^{n-k} = z^2 \frac{d^2}{dz^2} \left( \frac{1}{z^2} \int_0^z \zeta b_{n-k} \zeta^{n-k} d\zeta \right)$$

Hence:

$$\sum_{n=0}^{\infty} \left( \sum_{k=0}^{n} \binom{2n-2k}{n-k-2} b_k k \right) z^n = z B'(z) z^2 \frac{d^2}{dz^2} \left( \frac{1}{z^2} \int_0^z \zeta B(\zeta) d\zeta \right)$$

$$= \frac{1}{1-4z} + \frac{z-1}{2z^2 \sqrt{1-4z}} - \frac{(2z+1) \sqrt{1-4z}}{4z^3} - \frac{(1-4z)^{1/2}}{4z^3} + \frac{1}{2z} - \frac{1}{z^2} + \frac{1}{2z}$$

$$= \sum_{n=0}^{\infty} \left( (n+2)(n+3) 4^n - (2n+1)(3n^2+7n+6) b_n \right) z^n$$

To calculate $p_n^m$, the generating function for the sequence $\left\{ \binom{2n-2k}{n-k-m} \right\}$ has to be found, with $k$ and $m$ fixed. This requires $m-1$ integrations, followed by $m$ differentiation operations. Comparing (14) and (25) we may assert, that the structure of the function $B(z)$ does not permit a simple generalisation to such a procedure.

### 3.9. Approximation and numerical data

The values of $p_n^m$ for large $n$ and comparatively small $r$ are calculated by deriving an approximation for a combination of (23) and (24):

$$p_n^m = \binom{n+m+1}{m} \binom{4^n}{n b_n} - \sum_{i=0}^{m} \binom{2n+1}{n-i} \binom{m}{i} b_n$$

The quotients of binomial coefficients in this equation can be written as quotients of polynomials in $n$ with integer coefficients:
For any two polynomials \( u(n) \) and \( v(n) \) over the field of rational numbers, there exists a unique pair of polynomials \( q(n) \) and \( r(n) \) such that:\(^{10}\)

\[
u(n) = v(n) \times q(n) + r(n), \text{ } \deg(r) < \deg(v)
\]

Since \( \forall i \leq m+1 \) the leading coefficients of \( u_i(n) \) and \( v_i(n) \) are 1 and the degrees of both polynomials are equal we have \( q_i(n) = 1 \).

\[
\overline{p_n^m} = \left[ 1 + \frac{r_{m+1}(n)}{v_{m+1}(n)} \right] \left[ \frac{4^n}{n b_n} - \frac{(2n+1)}{n} \sum_{i=0}^{m} \left[ 1 + \frac{r_i(n)}{v_i(n)} \right] \right] = \sqrt{\pi n} - 2(m+1) + O \left( \frac{1}{\sqrt{n}} \right), \text{ } n \gg m
\]

Table (2) shows some values of \( \overline{p_n^m} \) for small \( n \) and \( m \), which where calculated using (26). The last line in the table gives approximated values using the formula \( \sqrt{8192 \pi} - 2(m+1) \). This shows that for small \( m \) the approximation is good, but for \( m \) near \( \sqrt{n} \) it has no significance.
Table 2: $p_n^m$ for some values of $n$ and $m$.

4. Conclusions

The practical importance of graph reduction has provided the incentive to develop a model for
the effects that reduction has on the structure of its graphs. The question that we have worked on
in this paper is: how many nodes may be expected to become unreachable when an arbitrary
edge is cut? In doing so a method has been developed that makes some of the standard results of
Catalan statistics applicable to graphs. The method works by treating a shared node in a graph as
a specially marked node in a tree. The standard enumeration method for ordered binary trees is
extended to exclude subtrees with marked nodes. Disconnecting a single edge in a graph may
cause shared nodes to become unreachable, because all paths to such shared nodes pass via that
single edge. Assuming that all shared nodes are from this particular (removable) type, an upper
bound on the expected size of a subtree is calculated ($p_n^m$). An estimate from below is obtained
by assuming that all shared nodes remain connected (permanent) if an arbitrary edge is cut. The average size of a subtree under this assumption is also derived ($p^m_n$).

It turns out that in both cases the expected number of nodes that become unreachable when an arbitrary edge is cut is larger than we have observed in practice. One reason may be that not all edges have the same probability of being cut. Reduction probably takes place more near the leaves than near the root of the graph. However, without further investigation we can only speculate on possible probability distributions. Although the formulae that we have derived are specific with respect to the uniform distribution that we have assumed, the method allows for other distributions to be used.

Acknowledgements

I am grateful to Henk Barendregt and Arthur Veen for comments on an earlier draft of the paper. Peter van Emde Boas and Michiel de Smid helped me sort out the literature.

References


