Enumeration of Tree Properties by Naïve Methods

Lars Ericson
New York University — INRIA Rocquencourt

Colm Ó Dúnlaing*
Dublin University

Abstract

This paper studies the problem of how much space is saved, on average, when a TRIE is pruned back to a minimal discriminating prefix. Exact figures (on average, half the space is saved) are given for binary trees. All trees with n nodes and m leaves are assumed equally likely. The calculations are based on an unusual form of recurrence relation.

1 Introduction

Consider a set S of strings over a fixed alphabet, none of which is a prefix of another in the set. The set can be represented in a TRIE in the usual way, so a given input string x can be matched with a string in S by traversing the trie downwards from the root (see, for instance, [1]). If the string is known to be in S then the search is significant only at those nodes of the TRIE which have more than one child. In particular, nodes of the tree which have exactly one leaf descendant are redundant. How much space is saved by discarding these redundant nodes?

We frame this as a question about tries, and for the case of binary tries compute the average saving is as follows:

(1.1) Theorem If all binary trees with n nodes of which m are leaves are equally probable, then the average ratio of[1] $|T'|/|T|$, where T ranges over all such binary trees and T' is obtained by discarding redundant nodes from T, is

$$\frac{m-1}{2m-1} + \frac{m}{n}.$$

Most of this paper is aimed at a proof of the above theorem. The result is in strong contrast with known estimates for compressing tries formed from random *strings*, since the distribution of such tries is non-uniform (different strings are unlikely to have long prefixes in common [1]).

*First author's address: CIMS, 251 Mercer Street, New York, NY 10012, U.S.A. Second author's address: Mathematics, Trinity College, Dublin 2, Irish Republic; electronic address odunlain@maths.tcd.ie. This work was supported by the European Community under Esprit BRA 3075 (ALCOM); presented at the Alcom Workshop on Probabilistic Methods, Patras, Greece, March 1990.

[1] $|T|$ denotes the number of nodes in T

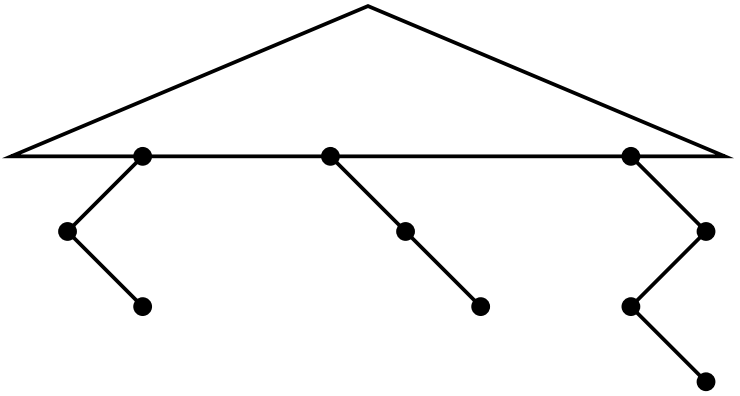

Figure 1: 3 new branches, total 7 nodes

2 Enumerating 'trim' binary trees

For brevity, a tree with n nodes of which m are leaves will be said to have size (n, m). A tree is *trim* if every leaf has a sibling. Equivalently, every internal node has two or more leaf descendants. The *trim prefix* T' of a tree T is obtained by deleting all nodes in T which have exactly one leaf descendant. Let $B_{n,m}$ denote the number of binary trees of size (n, m), and let $P_{k,m}$ denote the number of trim binary trees of size (k, m).

We first calculate, given such a fixed trim tree T' the number of extensions T of size (n, m) which it possesses. There are m leaves in T' and each leaf can be extended by a branch, as long as the total length of added branches is $n - k$. One can distribute branch lengths as a tuple $(i_1, \ldots i_m)$, consisting of nonnegative integers adding up to $n - k$. For each such m-tuple there are a total of 2^{n-k} branch configurations (see Figure 1), and there are

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

such arrangements of branches, by the well-known 'stars and bars' counting trick.[2] Therefore

(2.1) $$B_{n,m} = \sum_{k \le n} \binom{n-k+m-1}{m-1} 2^{n-k} P_{k,m}$$

This can be inverted directly using an inverse pair of relations [2]: yielding

(2.2) $$P_{k,m} = \sum_{n=2m-1}^{n=k} (-2)^{k-n} \binom{m}{k-n} B_{n,m}$$

As we shall see later (equation 3.2) $B_{n,m} = K_m \binom{n-1}{2m-2} 2^{n+1-2m}$ where $K_m = \frac{1}{m}\binom{2m-1}{m-1}$ is the number of full binary trees with m leaves. This can be substituted into equation (2.2). It is, of course, not easy to convert this to closed form. However, it can be done directly for small values of m[3] and we can quickly reach the following conjecture:

(2.3) $$P_{k,m} = K_m 2^{k+1-2m} \binom{k-1-m}{m-2}$$

[2] Imagine any such tuple written as a list of unary numbers with $m - 1$ comma separators. This is a list of ones and commas mixed, of length $n - k + m - 1$. Therefore the number of tuples is the number of choices of $m - 1$ (commas) out of $n - k + m - 1$ locations.

[3] Some of these calculations used Reduce and Maple computer algebra systems.

We need only verify that equation (2.3) satisfies equation (2.1): ignoring the factors K_m we need to verify that

$$\binom{n-1}{2m-2}2^{n+1-2m} = \sum_{k=2m-1}^{n} 2^{n-k}\binom{n-k+m-1}{m-1}2^{k+1-2m}\binom{k-m-1}{m-2}$$

or, equivalently,

(2.4) $$\binom{n-1}{2m-2} = \sum_{k=2m-1}^{n}\binom{k-m-1}{m-2}\binom{n-k+m-1}{m-1}.$$

This last identity is easily verified: think of $k-m$ as indexing the $m-1$st element in a choice of $2m-2$ items from $n-1$.

3 Calculating $B_{n,m}$

The standard way to calculate the quantities $B_{n,m}$ is to embed them in a bivariate generating function $B(x,y)$, and consider a generic tree in terms of its root and left and right subtrees, leading to a convolution formula applying to the $B_{n,m}$, which translates into the following equation in $B \equiv B(x,y)$:

$$xB^2 - B + 1 - x + xy = 0.$$

This does not lead to easy solutions[4]. The standard recursive descriptions consider the formation of trees from the root, a top-down recurrence. An alternative bottom-up recurrence can be obtained as follows.

Let us imagine building up a tree by repeatedly adding leaves. In how many ways can you add a leaf to a tree of size (n,m)? You can either make it a child of a leaf of the tree, or you can make it a child of a defective internal node (one with only one child currently). It is clear that for any fixed leaf you can add a new child in two possible ways, yielding a tree of size $(n+1,m)$.

Clearly, if an internal node is defective (having just one child), a new leaf can be added as its other child in just one way: the resulting tree has size $(n+1,m+1)$. Moreover, a tree of size (n,m) possesses exactly $n+1-2m$ defective nodes. Thus by adding a single leaf, $2m$ trees of size $(n+1,m)$ can be formed and $n+1-2m$ of type $(n+1,m+1)$.

Viewing this from another standpoint, a tree of size $(n+1,m+1)$ can be reduced to one of size $(n,m+1)$ or (n,m) by deleting any one of its $m+1$ leaves. How many of each type depends on the tree, but we conclude that the tree can be constructed in exactly $m+1$ different ways by adding just one leaf to a tree of n nodes. (The situation is summarised in Figure 2.) Hence

(3.1) $$(m+1)B_{n+1,m+1} = 2(m+1)B_{n,m+1} + (n+1-2m)B_{n,m}.$$

Experiments with computer algebra, for small values of m, suggest a solution

$$B_{n,m} = A_m(n-1)(n-2)\cdots(n-2m+2)2^n, n \geq 2m-1,$$

[4]At the Patras meeting where this material was presented, P. Flajolet (private communication) set up and solved this recurrence (modified slightly) using Lagrange's inversion formula. The authors had not known of solution techniques beyond using the Binomial series. In another private communication, J-M. Steyaert calculated both $B_{n,m}$ and $P_{n,m}$ using generating function methods.

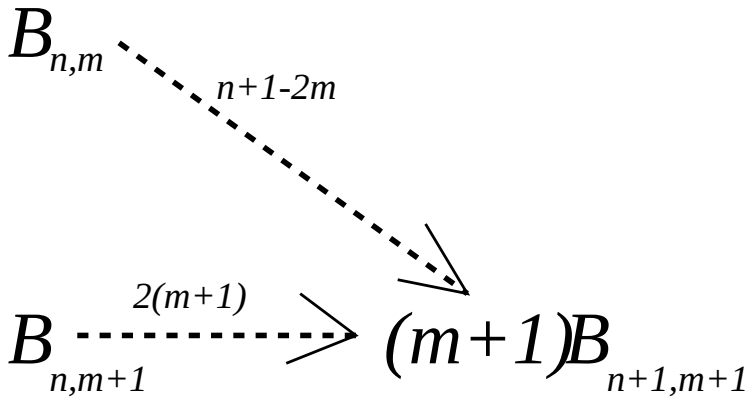

Figure 2.

where A_m is independent of n. The figure is zero if $n < 2m-1$. Let us therefore assume a solution of this form, and recast it to bring in binomial coefficients (this adjustment is made so that $B_{n,m} = K_m$ when $n = m-1$).

(3.2) $$B_{n,m} = K_m \binom{n-1}{2m-2} 2^{n+1-2m}$$

If we substitute this into equation (3.1) and factor out 2^{n-2m} we obtain the form

$$K_{m+1}\left\{\binom{n}{2m} - \binom{n-1}{2m}\right\} \;=\; \frac{n-2m+1}{m+1}\binom{n-1}{2m-2}2K_m$$

or

$$\binom{n-1}{2m-1}K_{m+1} \;=\; 2K_m\frac{n-2m+1}{m+1}\binom{n-1}{2m-2} \;=\; 2K_m\frac{2m-1}{m+1}\binom{n-1}{2m-1},$$

so

$$K_{m+1} = \frac{2(2m-1)}{m+1}K_m.$$

If we introduce a factor m in the numerator and denominator of the above factor of K_m, and note that $K_1 = 1$, it follows easily that

$$K_m = \frac{1}{m}\binom{2m-2}{m-1}$$

serves to make formula (3.2) satisfy the recurrence (3.1). It does, of course, furnish the known estimate (Catalan numbers) for the number of full binary trees with m leaves.

4 The average ratio.

We want to calculate the average value of $|P|/|T|$ where T is a random tree with size (n,m) and P is its trimmed version. The probability of P having size $k \le n$ is $P_{k,m}/B_{n,m}$, so the quantity we seek is

(4.1) $$\textstyle\sum_{k=2m-1}^{n} \frac{k}{n}\frac{P_{k,m}}{B_{n,m}}.$$

Let ρ denote this quantity. Substituting the known expressions for $P_{k,m}$ and $B_{n,m}$ (equations 2.3 and 3.2), we obtain

(4.2) $$n\binom{n-1}{2m-2}\rho \;=\; \sum_{k=2m-1}^{n} k\binom{k-m-1}{m-2}\binom{n-k+m-1}{m-1}$$

We can make the following substitution, justified by simple algebra:

$$k\binom{k-m-1}{m-2} \;=\; (m-1)\binom{k-m}{m-1} + m\binom{k-m-1}{m-2},$$

and break the right-hand side of equation (4.2) into two sums. The first is

$$\sum_k \binom{n-k+m-1}{m-1}\binom{k-m}{m-1} \;=\; \binom{n}{2m-1};$$

this identity can be justified by thinking of $k-m+1$ as indexing the mth item in a subsequence of $2m-1$ from n items. The second is

$$\sum_k \binom{n-k+m-1}{m-1}\binom{k-m-1}{m-2} \;=\; \binom{n-1}{2m-2}.$$

This identity we have seen already (2.4). Finally, we can factorise $\binom{n-1}{2m-2}$ from each part of equation (4.2), and obtain

$$\rho \;=\; \frac{m-1}{2m-1} + \frac{m}{n}$$

concluding the proof of Theorem 1.1.

5 Further remarks

It is tempting to generalise these results to q-ary trees for arbitrary q. For the case, for instance, of ternary trees, one can easily formulate the following recurrence analogous to equation (3.1): the number $T_{n,m}$ of ternary trees with n nodes of which m are leaves satisfies

(5.1) $$(m+1)T_{n+1,m+1} = 3(m+1)T_{n,m+1} + (2n+1-3m)T_{n,m}.$$

(and indeed the formula easily generalises to q-ary trees.) However, experimentation with small values of m reveals a divergence from the binomial coefficients:

$$T_{n,3} \;=\; \frac{(n-1)(n-2)(n-3)(3n-10)}{4}3^{n-5}.$$

While patterns may still be found here, it seems that an exact solution similar to that of Theorem 1.1 would be elusive, or at least difficult to calculate.

6 Acknowledgements

The authors would like to acknowledge the alternative derivations of $B_{n,m}$ and $P_{k,m}$ obtained by Philippe Flajolet and Jean-Marc Steyaert using Lagrange's inversion formula (mentioned in a footnote).

7 References

1. K. Mehlhorn, *Data Structures and Algorithms 1: Sorting and Searching,* Springer-Verlag 1984.

2. J. Riordan, *Combinatorial Identities,* Wiley 1968.