A More Precise Solution to Two Problems on Tries
(Preliminary Version)

Gonzalo Navarro Patricio V. Poblete*

Abstract

We use the Binomial Transform to address the problem of determining the average size and leaf depth of a trie. These problems lead to the need to find the poles of the function $\frac{1}{1-p^{s+1}-q^{s+1}}$, which we solve exactly up to low-order terms. This improves previous asymptotic approximations, which although solve the problem to a higher precision, do not allow to compute exact values for the main terms.

1 Introduction

1.1 Tries or Digital Search Trees

A *trie* or *digital search tree* over an alphabet A is a tree where each non-leaf node has $|A|$ sons, one for each element of A [2]. The trie is intended to store a set of strings over A, and retrieve any of them in time proportional to its length, in the worst case.

Insertion into a trie proceeds as follows: we scan the characters of the string in order, and follow the trie at the same time, starting at the root of the trie and at the beginning of the string. At each step, we follow the branch of the trie dictated by the current character of the string. If no string is a prefix of another(this can be achieved by adding to all strings a terminator character which is not part of the alphabet), then the process continues until we reach a leaf of the trie. The leaf may or may not hold another string. If the leaf does hold another string (which shares a prefix with the one we are inserting, at least up to the examined characters), then the leaf has to be converted into an internal node, and the insertion process continues creating internal nodes, until the two string differ. In that moment, the last node corresponding to a coincident character will have two leaves, one for each string. Then the insertion process terminates.

Searching into a trie proceeds much as insertion. We scan the string and follow the trie accordingly. If the string ends in an internal node, we see whether the string is stored in that node. If we reach a leaf, we do the same. Again, if we use a special termination character, then the string ends always in a leaf.

Tries have many applications, for example in lexical analysis, in lexicographical indexes for text databases, in the Lempel-Ziv algorithm, and in general in any application which needs to retrieve a string from a set in a time independent on the size of the set.

*This work has been supported in part by grant FONDECYT(Chile) 1940271

1.2 The problems

Since the average retrieval cost on a trie corresponds to its average leaf depth, this parameter is an important measure to analyze in order to get an idea of the efficiency of this data structure. Another interesting problem in tries is the average size of a trie built up from n strings. This size is not immediate, since for example if two strings with a long common prefix are inserted, the size of the trie may become proportional to their length).

We model these problems by assuming that the strings (or *keys*) to insert are infinite in length, so all of them are stored at the leaves of the trie. In order to simplify the problem, we take a binary alphabet, where the characters are 0 and 1. Finally, we take a probabilistic model in which the bits are randomly and independently generated: at each string position, there is a 0 with probability p, and a 1 with probability q, where $p + q = 1$. Note that in this case, the total number of internal nodes is the number of leaves minus one, and since there is at most one string per leaf, the important parameter is the number of empty leaves. The answer to the second problem is, then,

$$Trie\ Size(n) \ = \ 2 \times (n + Empty\ Leaves(n)) - 1$$

1.3 Previous approaches

The case $p = q$ is completely solved in [2], by using the Mellin Transform. The case $p \neq q$ is much harder. Recently, in [1], the presence of an oscilatory component was detected for this case, which was previously believed to exist only for $p = q$.

The known solution for the case $p \neq q$ is exact up to $O(n^{-1})$ for the problem of the average leaf depth [1], but although the presence of an $O(1)$ oscillatory component is detected, no method is provided to compute its values. Our attempt is to use the Binomial Transform to solve the case $p \neq q$ exactly for the main terms of the cost (that is, up to $o(1)$). The same approach may be used to solve the problem of the average size of a trie, exactly for the main term (that is, $o(n)$).

1.4 The Binomial Transform

The Binomial Transform [3] is a reversible transformation from sequences into sequences. It is defined by

$$\widehat{a_s} \ = \ \mathcal{B}_s a_n \ = \ \sum_{n=0}^{s} (-1)^n \binom{s}{n} a_n$$

and its inverse is the same, so

$$a_n = \mathcal{B}_n \mathcal{B}_s a_n$$

We summarize here the properties of the transform we need. For details, we refer the reader to [3]. Observe that any rule stands for two, the other is obtained by applying $\mathcal{B}_n$ to both sides.

P1. $\mathcal{B}$ is a linear operator.

P2. $\mathcal{B}_s[n = k] = (-1)^k \binom{s}{k}$. $[cond]$ is a function which takes the value 1 when $cond$ holds, and 0 otherwise.

P3. $\mathcal{B}_s H_n = -\frac{[s>0]}{s}$. H_n is defined as $\sum_{k=1}^{n} \frac{1}{k}$, and it holds $H_n = \log n + \gamma + O(\frac{1}{n})$.

P4. $\mathcal{B}_s \sum_{k=0}^{n} \binom{n}{k} p^k q^{n-k} a_k = p^s \widehat{a_s}$.

P5. $\mathcal{B}_s \frac{1}{n+x} = \frac{n!}{x^{\overline{n+1}}}$. $a^{\overline{n}}$ stands for $a(a+1)(a+2)\ldots(a+n-1)$.

P6. $\mathcal{B}_s[n>k]y_n = [s>k]\left(\widehat{y_s} - \sum_{i=0}^{k}(-1)^i\binom{s}{i}y_i\right)$.

P7. $\mathcal{B}_s n a_n = s(\widehat{a_s} - \widehat{a_{s-1}})$.

P8. $\mathcal{B}_s a_{n-1} = -\sum_{k=0}^{s-1}\widehat{a_k}$.

2 Average leaf depth of a trie

We begin by solving the first problem, that is, the average leaf depth of a trie. We do this by considering the following model: at first, the trie holds all the n strings in a single leaf. If there are two or more strings in the same leaf, it is converted into an internal node with two leaves: in one of them we put the strings whose next character is a 0; in the other those which have a 1. The process continues until no leaf has more than one string. Note that there may be empty leaves.

We will take a string at random and mark it. Then, we will study the average depth it gets in the trie when the process terminates. To ease manipulations, we assume we have $n+1$ keys, one of them is the marked one.

We call $P_n(z)$ the generating function for the probability to have the marked key at a given depth, after inserting $n+1$ keys (including the marked one), so the average leaf depth is $a_n = P_n'(1)$. Thus, we have

$$P_0(z) = 1$$

since if only the marked key is inserted, it is at depth 0. For $n > 0$,

$$P_n(z) = \sum_{i=0}^{n}\binom{n}{i}p^i q^{n-i}(zpP_i(z) + zqP_{n-i}(z))$$

the first three factors express the probability to generate a partition with i keys with 0 and $n-i$ with 1 as the next character, the last factor expresses the probability for the marked key to lie at each partition, the increment in cost (z), and what happens next. Observe that the recurrence ends when there are no more keys in the leaf, appart from the marked one.

This way, we reach the complete recurrence for this problem:

$$P_n(z) = \sum_{i=0}^{n}\binom{n}{i}p^i q^{n-i}(zpP_i(z) + zqP_{n-i}(z)) + (1-z)[n=0] \tag{1}$$

Now we apply the binomial transform. By rewriting (1) as

$$P_n(z) = \sum_{i=0}^{n}\binom{n}{i}p^i q^{n-i}zpP_i(z) \;+\; \sum_{i=0}^{n}\binom{n}{i}q^i p^{n-i}zqP_i(z) \;+\; (1-z)[n=0]$$

and by properties P1 and P4 we can transform the sequence $P_n(z)$ for any z to get

$$\widehat{P_s}(z) = p^s zp\widehat{P_s}(z) + q^s zq\widehat{P_s}(z) + (1-z)$$

that is

$$(1 - z(p^{s+1} + q^{s+1}))\widehat{P_s}(z) = 1 - z \tag{2}$$

Since the transform is linear, it commutes with the derivation operator; and since at any moment we can take any particular value for z, our answer is obtained by untransforming $\widehat{a_s} = \widehat{P_s}'(1)$. So, we derive (2):

$$-(p^{s+1} + q^{s+1})\widehat{P_s}(z) + (1 - z(p^{s+1} + q^{s+1})\widehat{P_s}'(z) = -1$$

and evaluate both sides at $z = 1$. Since $P_n(1) = 1$, $\widehat{P_s}(1) = [s = 0]$ (by property P2, for $k = 0$):

$$-(p^{s+1} + q^{s+1})[s = 0] + (1 - (p^{s+1} + q^{s+1})\widehat{P_s}'(1) = -1$$

Finally,

$$\widehat{a_s} = \frac{(p^{s+1} + q^{s+1})[s = 0] - 1}{1 - p^{s+1} - q^{s+1}} = -\frac{[s > 0]}{1 - p^{s+1} - q^{s+1}} \tag{3}$$

The problem is to untransform this expression, which we address in the next section.

3 Finding the poles

We solve the problem of inverting expression (3) by expanding it into partial fractions, so the first step is to find poles of the expression on the right side, that is, the complex solutions of

$$p^{s+1} + q^{s+1} = 1 \tag{4}$$

By writing $s = a + bi$, we get

$$p^{a+1}e^{b\log(p)i} + q^{a+1}e^{b\log(q)i} = 1$$

by replacing $e^{xi} = \cos x + i \sin x$, and calling $\alpha = b \log p$ and $\beta = b \log q$, the complex equation (4) turns into the real system

$$\begin{aligned} p^{a+1}\sin\alpha + q^{a+1}\sin\beta &= 0 & (5)\\ p^{a+1}\cos\alpha + q^{a+1}\cos\beta &= 1 & (6) \end{aligned}$$

from the modulus and (6) we infer $a \leq 0$. We then analyze the case $a = 0$, to get the simpler equations

$$\begin{aligned} p\sin\alpha + q\sin\beta &= 0 & (7)\\ p\cos\alpha + q\cos\beta &= 1 & (8) \end{aligned}$$

again, by looking at the modulus in (8), we get $\cos\alpha = \cos\beta = 1$, which implies $\alpha = 2\pi k$ and $\beta = 2\pi k'$, for any integers k and k' (observe it also satisfies (7)). That is,

$$b = \frac{2\pi k}{\log p} = \frac{2\pi k'}{\log q}$$

which implies

$$\frac{\log p}{\log q} = \frac{k}{k'}$$

that is, if between p and q holds an algebraic equation of the form

$$p^{k_1} = q^{k_2}$$

for some integers and relative primes k_1 and k_2, then all purely imaginary solutions to the equation are

$$s = \frac{2\pi k_2 k}{\log p} i$$

for any integer k. If, instead, no equation of that kind holds, then the only purely imaginary solution is $s = 0$. This may be seen as if $k_2 = 0$.

We now turn our attention to the poles lying at $a < 0$. Recalling property P5, we show now that if $z = x + yi$ is a complex number where $x > 0$, then

$$\left| \mathcal{B}_n \frac{1}{s+z} \right| = \left| \frac{n!}{z^{\overline{n+1}}} \right| \leq \frac{n!}{x^{\overline{n+1}}} = \frac{\Gamma(n)\Gamma(x)}{\Gamma(n+x)} = \frac{\Gamma(x) n^n e^{x+n} \left(1 + O(\frac{1}{n})\right)}{e^n (x+n)^{x+n} \left(1 + O(\frac{1}{n+x}\right)}$$

which for large n is

$$\Gamma(x) e^x e^{-x} O(1) \frac{1}{(x+n)^x} \longrightarrow 0$$

Since any pole wich lies at $a < 0$ will be expressed as $1/(s - a - bi)$, where $-a$ is positive, the untransformed term tends to zero as n grows, so it is $o(1)$. Therefore, we have found all poles up to $o(1)$.

The next step is to find the constant which multiplies the fractions. Thus,

$$\lim_{s \to \frac{2\pi k_2 k}{\log p} i} \frac{s - \frac{2\pi k_2 k}{\log p} i}{1 - p^{s+1} - q^{s+1}} = \lim_{s \to \frac{2\pi k_2 k}{\log p} i} \frac{1}{-\log(p) p^{s+1} - \log(q) q^{s+1}} = \frac{1}{-p \log p - q \log q}$$

which shows that all of them are single order poles. Thus we get

$$\widehat{a_s} \;=\; -\frac{[s>0]}{1 - p^{s+1} - q^{s+1}} \;=\; -f(s) \;-\; \frac{[s>0]}{p \log \frac{1}{p} + q \log \frac{1}{q}} \sum_{k \in Z} \frac{1}{s + \frac{2\pi k_2 k}{\log p} i} \;+\; \mathcal{B}_s o(1)$$

where $f(s)$ is the analytic difference function, which is left when one subtracts all the partial fractions from the original function. We call $f(n) = \mathcal{B}_n f(s)$.

The inverse of any of the summation terms is studied in [3], in reference to skip lists. The problem in that case is to invert

$$\sum_{k \in Z} \frac{1}{s + \sigma_k} \;=\; \sum_{k \in Z} \frac{1}{s + \frac{2\pi k}{\log p} i}$$

which is our case when $k_2 = 1$. Their solution involves the definition of a family of functions

$$\widehat{F_s^{[r]}} = \sum_{k \in Z - \{0\}} \frac{\sigma_k^{\overline{r}}}{s + \sigma_k}$$

that are studied in that paper. The untransformed versions are as follows:

$$F_n^{[r]} \;\approx\; 2 \sum_{k\geq 1} Re\left(\Gamma(r\sigma_k)e^{-\sigma_k \log n}\right)$$

which are found to be oscilatory, with period $\log_p n$ and very little amplitude, in the order of 10^{-6}, but they do not tend to zero as n grows.

To adapt this result to our problem we just replace σ_k by $k_2\sigma_k$. Since in our case $r = 0$, the final value is

$$2 \sum_{k\geq 1} Re\left(e^{-\sigma_k k_2 \log n}\right) = 2 \sum_{k\geq 1} \cos\left(\sigma_k k_2 \log n\right) \approx F_{n^{k_2}}^{[0]}$$

so different values for k_2 change the period, which gets longer as k_2 is bigger. In fact, k_2 is the quotient between both periods. Recall that if $\log p/\log q$ is not a rational, this oscilatory term does not exist.

The factor $[s > 0]$ can be eliminated from all terms of the sum, except from the one which corresponds to $k = 0$, since for $s = 0$, the sum becomes $\sum_{k\in Z-\{0\}} 1/\frac{2\pi k_2 k}{\log p} i$, which is already zero (the justification for the convergence of this particular summation can be found in [3], where the same result is extracted by expanding denominators of the form $s^2 + (2\pi k/\log p)^2$, which do converge). Observe also that the sign of the sum is only significant for the term corresponding to $k = 0$.

This (still untransformed) term is $[s > 0]/s$, whose inverse transform is $-H_n$ (recall P3). So, the untransformed result is

$$a_n \;=\; -f(n) + \frac{1}{p\log\frac{1}{p} + q\log\frac{1}{q}} \;\left(H_n + F_{n^{k_2}}^{[0]}\right) + o(1)$$

What is left is to find the analytic difference function $f(s)$.

The values $F_n^{[0]}$, although may not be considered a closed form expression, may be computed from [3] in the following way: since that equation is

$$\frac{[s>0]}{p^s - 1} = -\frac{[s>0]}{2} + \frac{[s>0]}{s\log p} + \frac{1}{\log p}\widehat{F_s}^{[0]}$$

it can be put this way

$$\widehat{F_s}^{[0]} = [s>0]\left(\frac{\log p}{p^s - 1} + \frac{\log p}{2} - \frac{1}{s}\right)$$

so any desired value for $F_n^{[0]}$ may be computed by applying the definition of $\mathcal{B}_n$ numerically, on the right side of the equation, which is composed of known functions. This way, the values of F_n can be tabulated, for example. In this sense, F_n becomes a known function.

This completes the solution of our first problem: the expected leaf depth of a trie after inserting n random strings over a binary alphabet with probability p for the 0 is

$$\frac{1}{p\log\frac{1}{p} + q\log\frac{1}{q}} \;\left(H_{n-1} + F_{(n-1)^{k_2}}^{[0]}\right) - f(n-1) + o(1)$$

if $\log p/\log(1-p) = k_1/k_2$ (relative primes); and

$$\frac{1}{p\log\frac{1}{p} + q\log\frac{1}{q}}\; H_{n-1} - f(n-1) + o(1)$$

otherwise.

4 Average size of a trie

We have mentioned another problem, namely the average number of empty leaves of a trie, which we have shown directly related with the total number of nodes. We will show that the same technique used to solve the previous problem may be applied to this one.

Calling $P_n(z)$ the generating function for the probability to have a given number of empty leaves after inserting n keys (then our answer is $x_n = P_n'(1)$, we have

$$\begin{aligned} P_0(z) &= z \\ P_1(z) &= 1 \\ P_n(z) &= \sum_{i=0}^{n} \binom{n}{i} p^i q^{n-i} P_i(z) P_{n-i}(z) \end{aligned}$$

where in the last case $n > 1$ is assumed. The intuition is as follows: if there are no keys, there is a single empty leaf, with probability 1; if there is one key, there are no empty leafs, with probability 1; if there are more than one key, the first three factors are as in the other problem, and the number of empty leaves must be added from both sides of the trie, weighted by their probability, and this is exactly what the final two factors do. A closed form for P_n is

$$P_n(z) = \sum_{i=0}^{n} \binom{n}{i} p^i q^{n-i} P_i(z) P_{n-i}(z)\ [n > 1]\ +\ [n = 1]\ +\ z[n = 0]$$

by deriving the above equation we get

$$P_n'(z) = \sum_{i=0}^{n} \binom{n}{i} p^i q^{n-i} (P_i(z) P_{n-i}'(z) + P_i'(z) P_{n-i}(z))\ [n > 1] + [n = 0]$$

and by evaluating at $z = 1$, that leads to

$$x_n = \sum_{i=0}^{n} \binom{n}{i} p^i q^{n-i} (x_i + x_{n-i})\ [n > 1] + [n = 0]$$

By applying the binomial transform to the above equation, and using properties P4 and P6 we get

$$\widehat{x_s} = [s > 1]\,((p^s + q^s)\,\widehat{x_s} - 2 + s) + 1$$

(since $x_0 = 1$ and $x_1 = 0$). Thus,

$$\widehat{x_s} \;=\; \frac{1 + (s-2)[s > 1]}{1 - [s > 1](p^s + q^s)} \;=\; [s \le 1] \;+\; \frac{[s > 1]}{1 - p^s - q^s}(s-1)$$

which is quite similar to the one we solved. Indeed, recalling that that sequence was called a_n,

$$\widehat{x_s} = [s \le 1] - (s-1)\widehat{a_{s-1}}$$

Let's call

$$u_n = \sum_{k=0}^{n-1} a_k$$

by using properties P2, P7 and P8, we get

$$x_n = 1 - n - u_n - n(u_{n-1} - u_n) = 1 - n - u_n + na_{n-1}$$

by expanding the expression for u_n, we obtain

$$x_n = 1 - n + na_{n-1} - \sum_{k=0}^{n-1} a_k$$

that is

$$\begin{array}{rc} x_n = & 1 - n - nf(n-1) + no(1) + \sum_{k=0}^{n-1} f(k) - no(1) + \\ & \frac{1}{p\log\frac{1}{p}+q\log\frac{1}{q}}\left(nH_{n-1} + nF^{[0]}_{(n-1)^{k_2}} - \sum_{k=0}^{n-1} H_k - \sum_{k=0}^{n-1} F^{[0]}_{k^{k_2}}\right) \end{array}$$

by introducing $I_n = \sum_{k=0}^{n-1} F_k^{[0]}$, the above expression is

$$x_n = 1 - n - nf(n-1) + \sum_{k=0}^{n-1} f(k) + o(n) + \frac{1}{p\log\frac{1}{p}+q\log\frac{1}{q}}\left(n - 1 + nF^{[0]}_{(n-1)^{k_2}} - I_{n^{k_2}}\right)$$

Finally,

$$x_n = \left(\frac{1}{p\log\frac{1}{p}+q\log\frac{1}{q}}\left(1 + F^{[0]}_{(n-1)^{k_2}} - \frac{I_{n^{k_2}}}{n}\right) - 1\right) n + \sum_{k=0}^{n-1} f(k) - nf(n-1) + o(n)$$

Since I_n may be computed from $F_n^{[0]}$, we take it as a known function. In this sense, the formula is exact up to lower order (i.e. sublinear) terms. In the case that $\log p/\log q$ is irrational, the formula reduces to

$$x_n = \left(\frac{1}{p\log\frac{1}{p}+q\log\frac{1}{q}} - 1\right) n + \sum_{k=0}^{n-1} f(k) - nf(n-1) + o(n)$$

So, the answer to our problem is that the expected size of a trie built up from n insertions of strings from a binary alphabet with probability p for the 0, is

$$2(n + x_n) - 1 = \frac{2}{p\log\frac{1}{p}+q\log\frac{1}{q}}\left(1 + F^{[0]}_{(n-1)^{k_2}} - \frac{I_{n^{k_2}}}{n}\right) n + \sum_{k=0}^{n-1} f(k) - nf(n-1) + o(n)$$

for example, if $p = q$, it is $\approx 2.88n$. This is the value for p which minimizes the size of the trie.

5 Conclusions

We have focused on the problem of finding the average leaf depth and size of a trie after n key insertions. The first measure is directly related to the time efficiency of this data structure, while the second one is related to its space utilization.

While previous approaches have solved this problem exactly for $p = q$, the case $p \neq q$ is much harder, and has been solved up to $O(n^{-1})$ for leaf depth and up to $O((\log n)^{-1})$ for size. However, these solutions do not allow to compute exact values for the oscillatory components present in the main terms.

We used an approach based on the Binomial Transform, to solve the problems up to $o(1)$ in the first case and up to $o(n)$ in the second, that is, less deep than previous solutions. However, our solutions do allow to compute exact values for the main terms of the expressions.

Further work on this subject needs to be carried out:

- We should find the expression for the analytic difference function, $f(s)$.
- Variance should be studied in both cases (to obtain exact results for the main terms).
- The results should be generalized for larger alphabets.
- The poles should be studied more in depth to find more exact terms of the solution. While we have a number of results about where the poles can be located, we need more information where $a \longrightarrow -0$.

References

[1] P. Jacquet and W. Szpankowski. A functional equation arising in the analysis of algorithms. In *Proceedings of* STOC'94, pages 780–789, May 1994. Montreal, Canada.

[2] D.E. Knuth. *The Art of Computer Programming*, volume 3. Sorting and Searching. Addison-Wesley, 1973.

[3] P.V. Poblete, J.I. Munro, and T. Papadakis. The binomial transform and its application to the analysis of skip lists. In P. Spirakis, editor, *Proceedings of* ESA'95, pages 554–569, September 1995. Corfu, Greece. LNCS 979.