Analysis of a Class of Tries with Adaptive Multi-Digit Branching

Yuriy A. Reznik

RealNetworks, Inc.
2601 Elliott Avenue, Seattle, WA 98121
`yreznik@acm.org`

Abstract. We study a class of adaptive multi-digit tries, in which the numbers of digits r_n processed by nodes with n incoming strings are such that, in memoryless model (with $n \to \infty$):

$$r_n \to \frac{\log n}{\eta} \quad (pr.)$$

where η is an algorithm-specific constant. Examples of known data structures from this class include LC-tries (Andersson and Nilsson, 1993), "relaxed" LC-tries (Nilsson and Tikkanen, 1998), tries with logarithmic selection of degrees of nodes, etc. We show, that the average depth D_n of such tries in asymmetric memoryless model has the following asymptotic behavior (with $n \to \infty$):

$$D_n = \frac{\log \log n}{-\log\left(1 - h/\eta\right)} \left(1 + o\left(1\right)\right)$$

where n is the number of strings inserted in the trie, and h is the entropy of the source. We use this formula to compare performance of known adaptive trie structures, and to predict properties of other possible implementations of tries in this class.

1   Introduction

Radix search trees or *tries*, introduced in late 1950's by R. E. de la Briandeis [3] and E. Fredkin [14] have long become one of the most basic and much appreciated concepts in computer science. It is well known, that they enable access to n variable-length strings in $O(\log n)$ time (on average), while using only $O(n)$ of space [15]. These fundamental properties remain in force for a large class of statistical models [8, 5, 28], and are inherited by most of their modifications, such as *digital search trees* [6, 13], *Patricia tries* [16], *bucket tries* [26, 15, 12], and others [4]. Most of the original research on these structures has been done in 1960s–70s.

Nevertheless, in recent years, tries have seen a resurgence of interest in connection with several new modifications, allowing search operations to be executed much faster, typically in $O(\log \log n)$ time on average [1, 2, 18, 20, 10, 22, 25, 11]. The basic technique that enables such a speed up is called *adaptive multi-digit branching* [1, 22].

Fig. 1. Examples of tries built from 9 binary strings: $s1 = 00000\ldots$, $s2 = 00001\ldots$, $s3 = 00011\ldots$, $s4 = 0010\ldots$, $s5 = 0011\ldots$, $s6 = 01000\ldots$, $s7 = 01001\ldots$, $s8 = 1010\ldots$, $s9 = 1100\ldots$.

The best known example of such a structure is a *level compressed trie* (or *LC-trie*) of Andersson and Nilsson [1], which simply replaces all complete subtrees in a trie with larger multi-digit nodes[1] (see Fig. 1.a,b). Other known implementations include "sparse" LC-tries of Nilsson and Tikkanen [19, 20], which allow the resulting multi-digit nodes to have a certain percentage of empty positions (see Fig. 1.c), tries with logarithmic branching [22, 25], etc.

To the best of our knowledge, only LC-tries have been thoroughly analyzed in the past. First results, suggesting that the average depth D_n^{LC} of LC-tries in asymmetric memoryless model is $O(\log \log n)$ and only $\Theta(\log^* n)$ if model is symmetric, were reported by Andersson and Nilsson [1, 2]. A more refined estimate in the symmetric case: $D_n^{LC} \sim \log^* n$ has been obtained by Devroye [10]. In the asymmetric case, Devroye and Szpankowski [11], and Reznik [24] have inde-

---

[1] It is assumed that multi-digit nodes can be represented by lookup tables, so the time required to parse such a structure is still proportional to the number of nodes in each particular path.

pendently arrived at the following expression: $D_n^{LC} = \frac{\log \log n}{-\log(1-h/h_{-\infty})} (1 + o(1))$ where h and $h_{-\infty}$ are Shannon and Rényi entropies correspondingly [28]. Analysis of the expected height (longest path length) of LC-tries in memoryless model can be found in [10, 11].

At the same time, not much is known about other implementations of adaptive tries. Most relevant results in this direction include analysis of basic statistics of multi-digit nodes [24, 25], and experimental data collected for some specific implementations [19, 20]. Why useful by themselves, however, these results are not sufficient for complete characterization of the average behavior of such structures.

In this paper, we offer a general analysis of a *class of tries with adaptive multi-digit branching* in memoryless model, which, in various special cases, leads to expressions of average depths for all above mentioned implementations. We use these expressions to compare performance of these structures, and to predict properties of other possible implementations of tries in this class.

2 Definitions and Main Results

Consider a set $S = \{s_1, \ldots, s_n\}$ of n distinct strings to be used for trie construction. For simplicity, we will assume that these strings contain symbols from a binary alphabet $\Sigma = \{0, 1\}$.

Definition 1. *A* multi-digit trie *with parameter r ($r \geqslant 1$): $T^{(r)}(S)$ over S is a data structure defined recursively as follows. If $n = 0$, the trie is an* empty node. *If $n = 1$, the trie is an* external node *containing a pointer to a string in S. If $n > 1$, the trie is an r-digit internal node *containing pointers to 2^r child tries: $T^{(r)}(S_{0^r}), \ldots, T^{(r)}(S_{1^r})$, constructed over sets of* suffixes *of strings from S beginning with the corresponding r-digit words $S_v = \{u_j \mid v\, u_j = s_i \in S\}$, $v \in \Sigma^r$.*

In the simplest case, when $r = 1$, this structure turns in to a regular *binary trie* (see Fig. 1.a). When r is fixed, this structure is a 2^r-ary trie, which uses r-bits units of input data for branching. When r is variable, we say that this trie belongs to a class of *adaptive multi-digit tries*. Below we define several important implementations of such data structures.

Definition 2. *An* LC-trie $T^{LC}(S)$ *over S is an adaptive multi-digit trie, in which parameters r are selected to reach the first levels at which there is at least one external or empty node: $r = \min\left\{s : \sum_{v \in \Sigma^s} 1\{|S_v| \leqslant 1\} \geqslant 1\right\}$.*

Definition 3. *An ε-sparse LC-trie $T^{\varepsilon - LC}(S)$ over S is an adaptive multi-digit trie, in which parameters r are selected to reach the deepest levels at which the ratio of the number of empty nodes to the total number of nodes is not greater than ε: $r = \max\left\{s : \frac{1}{2^s} \sum_{v \in \Sigma^s} 1\{S_v = \emptyset\} \leqslant \varepsilon\right\}$.*

Definition 4. *A logarithmic trie $T^{\lg}(S)$ over S is an adaptive multi-digit trie, in which parameters r for nodes with n incoming strings are calculated using $r = \lceil \log_2 n \rceil$.*

Examples of the above defined types of tries are provided in Fig.1. Observe, that all input strings $s_1, \ldots, s_n$ inserted in a trie can be uniquely identified by the paths from the root node to the corresponding external nodes. The sum of lengths of these paths $C(T) = \sum_{i=1}^{n} |s_i|$ is called an *external path length* of a trie T, and the value $D(T) = C(T)/n$ – an *average depth* of this trie.

In order to study the average behavior of tries we will assume that input strings S are generated by a binary *memoryless* (or *Bernoulli*) source [7]. In this model, symbols of the input alphabet $\Sigma = \{0, 1\}$ occur independently of one another with probabilities p and $q = 1 - p$ correspondingly. If $p = q = 1/2$, such source is called *symmetric*, otherwise it is *asymmetric* (or *biased*).

Using this model, we can now define the quantity of our main interest:

$$D_n := E\{D(T)\} = \frac{E\{C(T)\}}{n}, \tag{1}$$

where the expectations are taken over all possible tries over n strings when parameters of the memoryless source (p and q) are fixed. Average depths of LC-, ε-sparse, and logarithmic tries over n strings will be denoted by D_n^{LC}, $D_n^{\varepsilon-LC}$, and $D_n^{\lg}$ correspondingly.

In order to consolidate analysis of these (and possibly many other) implementations of adaptive multi-digit tries, we will assume, that in memoryless model, the numbers of digits r_n assigned to their internal nodes with n incoming strings have the following convergence (with $n \to \infty$):

$$r_n \to \frac{\log n}{\eta} \quad (pr.) \tag{2}$$

where η is an algorithm-specific constant.

For example, it is well known, that convergence (2) takes place for LC-tries [21] (see also [9, 2]). In this case, the constant η becomes:

$$\eta^{LC} = h_{-\infty}, \tag{3}$$

where $h_{-\infty} = -\log \min(p, q)$ is a special case of a Rényi's entropy [28]. In a case of ε-sparse LC-tries, an extension of analysis [25] suggests that: $r_n \to \frac{\log n}{\eta^{\varepsilon-LC}} + f(\varepsilon)\sqrt{\log n} \quad (pr.)$, where $f(.)$ is a monotonic function, such that $f(1/2) = 0$, and

$$\eta^{\varepsilon-LC} = h_g, \tag{4}$$

where $h_g = -\log(\sqrt{p\,q})$ is another constant depending on the probabilities of the source. It is clear, that our model (2) is sufficient to describe ε-sparse LC-tries with $\varepsilon = 1/2$. Finally, the behavior of logarithmic tries can obviously be modelled by (2) with

$$\eta^{\lg} = \log 2. \tag{5}$$

Our main result for a class of adaptive multi-digit tries is formulated in the following theorem.

Theorem 1. *Consider a class of adaptive multi-digit tries, in which the numbers of digits r_n processed by nodes with n incoming strings, in binary memoryless model (with $n \to \infty$):*

$$r_n \to \frac{\log n}{\eta} \quad (pr.)$$

where η is a constant, such that: $h < \eta \leqslant h_{-\infty}$, where $h = -p \log p - q \log q$ is the Shannon's entropy of the source, $h_{-\infty} = -\log \min(p, q)$, and it is assumed that $p \neq q$.

    Then, the average depth D_n of such tries over n strings is asymptotically (with $n \to \infty$):

$$D_n = \frac{\log \log n}{-\log(1 - h/\eta)} \left(1 + o(1)\right) . \tag{6}$$

3   Discussion

Using the result of Theorem 1 and the values of constants η for each respective algorithm (3-5), we can now compare them. The results of such a comparison are presented in Fig.2. For completeness, we also show the behavior of the average depths of regular (binary) tries $D_n^{bin} = \frac{1}{h} \log n + O(1)$ (cf. [15, 8, 27]).
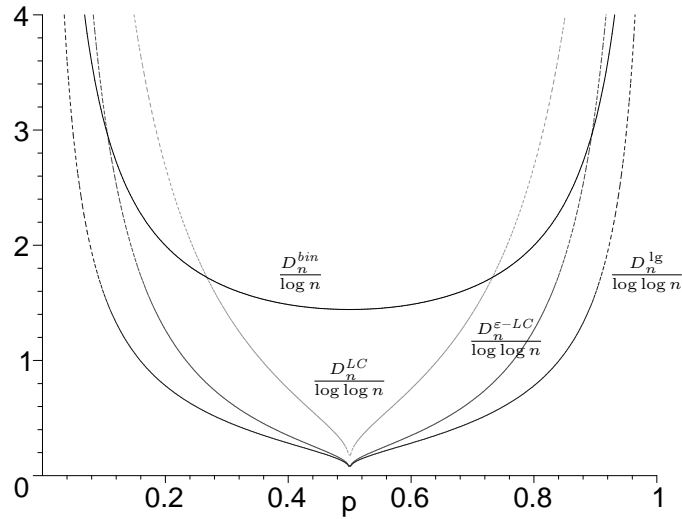

Fig. 2. Comparison of tries in binary memoryless model with $\Pr(0) = p$. D_n^{bin}, D_n^{LC}, $D_n^{\varepsilon-LC}$, and $D_n^{\lg}$ denote average depths of binary, LC-, "sparse" LC-, and logarithmic tries correspondingly.

We first notice that when the source is nearly symmetric $p \to 1/2$:

$$\frac{D_n^{LC}}{\log \log n} \to 0\,, \quad \frac{D_n^{\varepsilon-LC}}{\log \log n} \to 0\,, \quad \frac{D_n^{\lg}}{\log \log n} \to 0\,,$$

which suggests that in a symmetric case, the average depths of these structures should have a much smaller order. We already know that this is true for LC-tries (whose depth in symmetric case is only $O(\log^* n)$ [1, 10]), but now we can predict such an effect for any adaptive trie with $\eta \to h$ ($p \to 1/2$).

We next observe that behavior of LC-tries and their "sparse" variants is not very much different. Thus, by plugging their respective constants, it can be shown that (with $n \to \infty$):

$$2 \;\leqslant\; \frac{D_n^{LC}}{D_n^{\varepsilon-LC}} \;\leqslant\; \max_{p \in [0,1]} \frac{\log(1 - h_g/h)}{\log(1 - h_{-\infty}/h)} \approx 2.165367...\,,$$

which suggests that sparse LC-tries should be approximately twice faster than LC-tries, and that, contrary to the intuition, this ratio cannot be influenced much by increasing ε.

We also observe that both sparse and dense LC-tries are much more sensitive to the asymmetry of the source than regular tries. Thus, it can be seen that with $p \to 0$:

$$\frac{D_n^{LC}/\log \log n}{D_n^{bin}/\log n} \to \infty\,, \quad \frac{D_n^{\varepsilon-LC}/\log \log n}{D_n^{bin}/\log n} \to \infty\,.$$

At the same time, the sensitivity of logarithmic tries on the asymmetry of the source remains similar to one of regular tries. Thus, it can be shown that with $p \to 0$:

$$\frac{D_n^{\lg}/\log \log n}{D_n^{bin}/\log n} \to \log 2\,.$$

As obvious, logarithmic tries are the fastest among implementations that we have considered so far. Using the fact that the number of branches in their nodes is approximately equal to the number of passing strings, we can conjecture that the amount of space required by logarithmic tries is $O(n \log \log n)$, which is slightly larger than $O(n)$ space used by regular tries and LC-tries [15, 27, 2].

We conclude by pointing out that by using even larger nodes, e.g. with $h < \eta < \log 2$ in our model (2), it is possible to design tries that are faster than logarithmic tries. However, the amount of space required by such tries becomes $\Omega\left(n^{\frac{\log 2}{\eta}}\right)$, which in a practical world, might be too much of a price to be paid for a relatively small (in this case, limited to a constant-factor) improvement in speed.

4 Analysis

We start with deriving recurrent expressions for external path lengths of tries with r_n-digit nodes.

Lemma 1. *Parameters C_n (average external path length of adaptive multi-digit tries over n strings) in a binary memoryless model satisfy:*

$$C_n = n + \sum_{k=2}^{n} \binom{n}{k} \sum_{s=0}^{r_n} \binom{r_n}{s} \left(p^s q^{r_n-s}\right)^k \left(1 - p^s q^{r_n-s}\right)^{n-k} C_k; \qquad (7)$$

$$C_0 = C_1 = 0.$$

Proof. Consider an r_n-digit node processing n strings. Assuming that each of its 2^{r_n} branches have probabilities $p_1, \ldots, p_{2^{r_n}}$, and using the standard technique for enumeration of C_n in tries [15, 6.3-3], we can write:

$$C_n = n + \sum_{k_1 + \ldots + k_{2^{r_n}} = n} \binom{n}{k_1, \ldots, k_{2^{r_n}}} p_1^{k_1} \ldots p_{2^{r_n}}^{k_{2^{r_n}}} \left(C_1 + \ldots + C_{2^{r_n}}\right),$$

$$= n + \sum_{k=0}^{n} \binom{n}{k} \left(p_1^k (1 - p_1)^{n-k} + \ldots + p_{2^{r_n}}^k (1 - p_{2^{r_n}})^{n-k}\right) C_k. \qquad (8)$$

Recall now that our strings are generated by a binary memoryless source with probabilities p, and $q = 1 - p$. This means that:

$$p_i = p^{s_i} q^{r_n - s_i}, \qquad (9)$$

where s_i is the number of occurrences of symbol 0 in a string leading to a branch i $(1 \leqslant i \leqslant 2^{r_n})$. Combining (8) and (9), we arrive at the expression (7) claimed by the lemma. □

In order to find a solution of (7) we will use the following, very simple technique. We already know, that for a class of our tries $D_n = O(\log \log n)$, hence we can say that $C_n = \xi n \log \log n$ and plug it in (7). Ultimately, this will give us upper and lower bounds for the parameter ξ such that the recurrence (7) holds. If these bounds are tight, then we have successfully deduced the constant factor in the $O(\log \log n)$ term.

We will need the following intermediate results. For simplicity, here and below we use natural logarithms.

Lemma 2. *Consider a sum:*

$$f(n, \theta, \lambda) = \sum_{k=2}^{n} \binom{n}{k} \theta^k (1 - \theta)^{n-k} k \ln(\lambda + \ln k), \qquad (10)$$

where $\theta \in (0, 1)$, and $\lambda > 1$ are some constants. Then, there exists $0 < \zeta < \infty$, such that for any $n \geqslant 2$:

$$n\theta \ln(\lambda + \ln(n\theta)) - \zeta \;\leqslant\; f(n, \theta, \lambda) \;\leqslant\; n\theta \ln\left(\lambda + \ln(1 - \theta + n\theta)\right). \qquad (11)$$

Proof. We start with a representation:

$$f(n, \theta, \lambda) = \sum_{k=1}^{n} \binom{n}{k} \theta^k (1 - \theta)^{n-k} k \ln(\lambda + \ln k) - n\theta(1 - \theta)^{n-1} \ln \lambda$$

where the last term can be easily bounded by:

$$n\theta(1-\theta)^{n-1}\ln\lambda \;\leqslant\; \frac{\theta e^{-1}}{(\theta-1)\ln(1-\theta)}\ln\lambda \;=:\; \zeta\,.$$

Next, by Jensen's inequality for $x\ln(\lambda+\ln x)$:

$$\sum_{k=1}^{n}\binom{n}{k}\theta^k\,(1-\theta)^{n-k}\,k\ln(\lambda+\ln k)$$

$$\geqslant \left(\sum_{k=1}^{n}\binom{n}{k}\theta^k\,(1-\theta)^{n-k}\,k\right)\ln\left(\lambda+\ln\left(\sum_{k=1}^{n}\binom{n}{k}\theta^k\,(1-\theta)^{n-k}\,k\right)\right)$$

$$= n\theta\ln(\lambda+\ln(n\theta))\,.$$

where convexity for $k\geqslant 1$ is assured by picking $\lambda>1$.

To obtain an upper bound we use Jensen's inequality for $-\ln(\lambda+\ln(1+x))$:

$$\sum_{k=1}^{n}\binom{n}{k}\theta^k\,(1-\theta)^{n-k}\,k\ln(\lambda+\ln k)$$

$$= n\theta\sum_{k=0}^{n-1}\binom{n-1}{k}\theta^k\,(1-\theta)^{n-1-k}\ln(\lambda+\ln(1+k))$$

$$\leqslant n\theta\ln\left(\lambda+\ln\left(1+\sum_{k=0}^{n-1}\binom{n-1}{k}\theta^k\,(1-\theta)^{n-1-k}\,k\right)\right)$$

$$= n\theta\ln\left(\lambda+\ln\left(1-\theta+n\theta\right)\right)\,.$$

$\square$

Lemma 3. *Consider a sum:*

$$g\left(n,\theta,\alpha,\beta\right)=\sum_{k=0}^{n}\binom{n}{k}\theta^k\,(1-\theta)^{n-k}\ln(\alpha+\beta k)\,, \tag{12}$$

where $\theta\in(0,1)$, $\alpha,\beta>0$, and $\alpha>\beta$. Then, for any $n\geqslant 1$:

$$\ln(\alpha-\beta(1-\theta)+\beta\theta n) \;\leqslant\; g\left(n,\theta,\alpha,\beta\right) \;\leqslant\; \ln(\alpha+\beta\theta n)\,. \tag{13}$$

Proof. We use the same technique as in the previous Lemma. By Jensen's inequality for $-\ln(\alpha+\beta x)$:

$$g\left(n,\theta,\alpha,\beta\right)\leqslant\ln\left(\alpha+\beta\sum_{k=0}^{n}\binom{n}{k}\theta^k\,(1-\theta)^{n-k}\,k\right)=\ln\left(\alpha+\beta\theta n\right)\,.$$

The lower bound follows from Jensen's inequality for $x \ln(\alpha - \beta + \beta x)$:

$$g(n, \theta, \alpha, \beta) = \frac{1}{\theta(n+1)} \sum_{k=1}^{n+1} \binom{n+1}{k} \theta^k (1-\theta)^{n+1-k} \, k \ln(\alpha - \beta + \beta k)$$

$$\geqslant \frac{1}{\theta(n+1)} \left(\sum_{k=1}^{n+1} \binom{n+1}{k} \theta^k (1-\theta)^{n+1-k} \, k \right) \times$$

$$\times \ln \left(\alpha - \beta + \beta \sum_{k=1}^{n+1} \binom{n+1}{k} \theta^k (1-\theta)^{n+1-k} \, k \right)$$

$$= \ln(\alpha - \beta(1-\theta) + \beta\theta n) \, .$$

It is clear, that convexity and continuity in both cases is assured when $\alpha > \beta > 0$.

$\square$

We are now prepared to solve our recurrence (7). For simplicity we assume that $p > 1/2$. Let $C_n = \xi n \ln(\lambda + \ln n)$, where $\lambda > 1$ is a constant. Then, according to Lemma 2:

$$C_n = n + \sum_{s=0}^{r_n} \binom{r_n}{s} \sum_{k=2}^{n} \binom{n}{k} \left(p^s q^{r_n - s} \right)^k \left(1 - p^s q^{r_n - s} \right)^{n-k} \xi k \ln(\lambda + \ln k)$$

$$\leqslant n + n\xi \sum_{s=0}^{r_n} \binom{r_n}{s} p^s q^{r_n - s} \ln \left(\lambda + \ln \left(n \, p^s q^{r_n - s} + 1 - p^s q^{r_n - s} \right) \right)$$

$$= n + n\xi \sum_{s=0}^{r_n} \binom{r_n}{s} p^s q^{r_n - s} \ln \left(\lambda + \ln \left(n \, p^s q^{r_n - s} \right) + \ln \left(1 + \frac{1}{n \, p^s q^{r_n - s}} - \frac{1}{n} \right) \right) \, .$$

Next, by our assumed property (2), we can pick $\varepsilon > 0$, such that the probability that

$$\left| r_n - \frac{\ln n}{\eta} \right| \leqslant \varepsilon \, . \tag{14}$$

holds true is 1 with $n \to \infty$. If we further assume that $\eta \leqslant -\ln q$ [2], then

$$n \, p^s q^{r_n - s} \geqslant n \, q^{r_n} \geqslant n \, q^{\frac{\ln n}{\eta} + \varepsilon} = n^{1 - \frac{\ln q}{\eta}} q^\varepsilon \geqslant q^\varepsilon \, .$$

and consequently:

$$\ln \left(1 + \frac{1}{n \, p^s q^{r_n - s}} - \frac{1}{n} \right) \leqslant \ln \left(1 + q^{-\varepsilon} - \frac{1}{n} \right) < \ln \left(1 + q^{-\varepsilon} \right) = O(\varepsilon) \, .$$

---

[2] A case when $\eta = -\ln q = h_{-\infty}$ corresponds to LC-tries. Smaller η correspond to tries with larger nodes.

By incorporating this bound and using Lemma 3:

$$C_n \leqslant n + n\xi \sum_{s=0}^{r_n} \binom{r_n}{s} p^s q^{r_n - s} \ln\left(\lambda + \ln\left(n\, p^s q^{r_n - s}\right) + \ln\left(1 + q^{-\varepsilon} - 1/n\right)\right)$$

$$= n + n\xi \sum_{s=0}^{r_n} \binom{r_n}{s} p^s q^{r_n - s} \ln\left(\lambda + \ln n + r_n \ln q + s \ln(p/q) + \ln\left(1 + q^{-\varepsilon} - 1/n\right)\right)$$

$$\leqslant n + n\xi \ln\left(\lambda + \ln n - h\, r_n + \ln\left(1 + q^{-\varepsilon} - 1/n\right)\right),$$

where $h = -p \ln p - q \ln q$ is the entropy. Now, by applying (14), we have:

$$C_n \leqslant n + n\xi \ln\left(\lambda + \ln n \left(1 - \frac{h}{\eta}\right) + h\,\varepsilon + \ln\left(1 + q^{-\varepsilon} - 1/n\right)\right),$$

and by plugging $C_n = \xi n \ln(\lambda + \ln n)$ in the left side of the above inequality, we finally obtain:

$$\xi \leqslant \frac{1}{-\ln\left(1 - \frac{h}{\eta} + \frac{\lambda + h\,\varepsilon + \ln(1 + q^{-\varepsilon} - 1/n)}{\ln n}\right) + \ln\left(1 + \frac{\lambda}{\ln n}\right)}$$

$$= \frac{1}{-\ln\left(1 - \frac{h}{\eta}\right)} \left(1 + O\left(\frac{\varepsilon}{\ln n}\right)\right). \tag{15}$$

The procedure for finding a lower bound is very similar:

$$C_n = n + \sum_{s=0}^{r_n} \binom{r_n}{s} \sum_{k=2}^{n} \binom{n}{k} \left(p^s q^{r_n - s}\right)^k \left(1 - p^s q^{r_n - s}\right)^{n - k} \xi k \ln(\lambda + \ln k)$$

$$\geqslant n + n\xi \sum_{s=0}^{r_n} \binom{r_n}{s} p^s q^{r_n - s} \log\left(\lambda + \ln\left(n\, p^s q^{r_n - s}\right)\right) - \zeta$$

$$= n + n\xi \sum_{s=0}^{r_n} \binom{r_n}{s} p^s q^{r_n - s} \ln\left(\lambda + \ln n + r_n \ln q + s \ln(p/q)\right) - \zeta$$

$$\geqslant n + n\xi \ln\left(\lambda + \ln n - h\, r_n - q \ln(p/q)\right) - \zeta,$$

$$\geqslant n + n\xi \ln\left(\lambda + \ln n \left(1 - \frac{h}{\eta}\right) - h\varepsilon - q \ln(p/q)\right) - \zeta,$$

which (after plugging $C_n = \xi n \ln(\lambda + \ln n)$ in the right side) leads to the following inequality:

$$\xi \geqslant \frac{1 - \zeta/n}{-\ln\left(1 - \frac{h}{\eta} + \frac{\lambda - h\varepsilon - q \ln(p/q)}{\ln n}\right) + \ln\left(1 + \frac{\lambda}{\ln n}\right)}$$

$$= \frac{1}{-\ln\left(1 - \frac{h}{\eta}\right)} \left(1 + O\left(\frac{\varepsilon}{\ln n}\right)\right). \tag{16}$$

By combining our bounds (15) and (16) and taking into account the fact that for any $\varepsilon > 0$, the probability that they both hold true is approaching 1 with $n \to \infty$, we can conclude that:

$$\xi \to \frac{1}{-\ln\left(1 - \frac{h}{\eta}\right)} \left(1 + o(1)\right)$$

in probability.

References

1. A. Andersson and S. Nilsson, Improved Behaviour of Tries by Adaptive Branching, *Information Processing Letters* **46** (1993) 295–300.
2. A. Andersson and S. Nilsson, Faster Searching in Tries and Quadtries – An Analysis of Level Compression. *Proc. 2nd Annual European Symp. on Algorithms* (1994) 82–93.
3. R. E. de la Briandeis, File searching using variable length keys. *Proc. Western Joint Computer Conference.* **15** (AFIPS Press, 1959).
4. J. Clement, P. Flajolet, and B. Vallée, The analysis of hybrid trie structures. *Proc. Annual ACM-SIAM Symp. on Discrete Algorithms.* (San Francisco, CA, 1998) 531–539.
5. Clement, J., Flajolet, P., and Vallée, B. (2001) Dynamic sources in information theory: A general analysis of trie structures. *Algorithmica* **29** (1/2) 307–369.
6. E. G. Coffman Jr. and J. Eve, File Structures Using Hashing Functions, *Comm. ACM*, 13 (7) (1970) 427–436.
7. T. M. Cover and J. M. Thomas, *Elements of Information Theory.* (John Wiley & Sons, New York, 1991).
8. L. Devroye, A Note on the Average Depths in Tries, *SIAM J. Computing* **28** (1982) 367–371.
9. L. Devroye, A Note on the Probabilistic Analysis of PATRICIA Tries, *Rand. Structures & Algorithms* **3** (1992) 203–214.
10. L. Devroye, Analysis of Random LC Tries, *Rand. Structures & Algorithms* **19** (3-4) (2001) 359–375.
11. L. Devroye and W. Szpankowski, Probabilistic Behavior of Asymmetric LC-Tries, *Rand. Structures & Algorithms* – submitted.
12. R. Fagin, J. Nievergelt, N. Pipinger, and H. R. Strong, Extendible Hashing – A Fast Access Method for Dynamic Files, *ACM Trans. Database Syst.*, 4 (3) (1979) 315–344.
13. P. Flajolet and R. Sedgewick, Digital Search Trees Revisited, *SIAM J. Computing* **15** (1986) 748–767.
14. E. Fredkin, Trie Memory, *Comm. ACM* **3** (1960) 490–500.
15. D. E. Knuth, *The Art of Computer Programming. Sorting and Searching. Vol. 3.* (Addison-Wesley, Reading MA, 1973).
16. D. A. Morrison, PATRICIA – Practical Algorithm To Retrieve Information Coded in Alphanumeric, *J. ACM*, 15 (4) (1968) 514–534.
17. S. Nilsson and G. Karlsson, Fast IP look-up for Internet routers. *Proc. IFIP 4th International Conference on Broadband Communication* (1998) 11–22.
18. S. Nilsson and G. Karlsson, IP-address look-up using LC-tries, *IEEE J. Selected Areas in Communication* **17** (6) (1999) 1083–1092.

19. S. Nilsson and M. Tikkanen, Implementing a Dynamic Compressed Trie, *Proc.* 2nd *Workshop on Algorithm Engineering* (Saarbruecken, Germany, 1998) 25–36.

20. S. Nilsson and M. Tikkanen, An experimental study of compression methods for dynamic tries, *Algorithmica* **33** (1) (2002) 19–33.

21. B. Pittel, Asymptotic Growth of a Class of Random Trees, *Annals of Probability* **18** (1985) 414–427.

22. Yu. A. Reznik, Some Results on Tries with Adaptive Branching, *Theoretical Computer Science* **289** (2) (2002) 1009–1026.

23. Yu. A. Reznik, On Time/Space Efficiency of Tries with Adaptive Multi-Digit Branching, *Cybernetics and Systems Analysis* **39** (1) (2003) 32–46.

24. Yu. A. Reznik, On the Average Depth of Asymmetric LC-tries, *Information Processing Letters* – submitted.

25. Yu. A. Reznik, On the Average Density and Selectivity of Nodes in Multi-Digit Tries, *Proc.* 7th *Workshop on Algorithm Engineering and Experiments and* 2nd *Workshop on Analytic Algorithmics and Combinatorics (ALENEX/ANALCO 2005)* (SIAM, 2005).

26. E. H. Sussenguth, Jr. (1963) Use of Tree Structures for Processing Files, *Comm. ACM*, 6 (5) 272–279.

27. W. Szpankowski, Some results on V-ary asymmetric tries, *J. Algorithms* **9** (1988) 224–244.

28. W. Szpankowski, *Average Case Analysis of Algorithms on Sequences* (John Wiley & Sons, New York, 2001).