On the Average Depth of Asymmetric LC-tries

Yuriy A. Reznik

RealNetworks, Inc.

2601 Elliott Avenue, Seattle, WA 98121

`yreznik@acm.org`

Abstract

Andersson and Nilsson have already shown that the average depth D_n of random LC-tries is only $\Theta\left(\log^* n\right)$ when the keys are produced by a symmetric memoryless process, and that $D_n = O\left(\log\log n\right)$ when the process is asymmetric. In this paper we refine the second estimate by showing that asymptotically (with $n \to \infty$): $D_n \sim \frac{1}{\eta} \log\log n$, where n is the number of keys inserted in a trie, $\eta = -\log\left(1 - h/h_{-\infty}\right)$, $h = -p\log p - q\log q$ is the entropy of a binary memoryless source with probabilities p, $q = 1 - p$ $(p \neq q)$, and $h_{-\infty} = -\log\min(p,q)$.

Key words: average case analysis of algorithms, trie, LC-trie.

1 Introduction

Level compressed trie (or *LC-trie*) introduced in 1993 by Andersson and Nilsson [1], represent a modification of the radix search tree (*trie*) structure [12], in which all complete subtrees are replaced by larger (multi-digit) nodes. Such a replacement is done in a recursive manner starting with a root node and results in a substantially flatter structure, requiring less number of steps for performing standard searching and sorting operations.

Main applications of LC-tries include algorithms for searching and sorting, image segmentation, image processing, geographic information systems, and robotics [1, 2]. Most of recent interest to LC-tries was due to their successful applications in the design of software Internet routers [14, 15]. Prior to LC-tries (and some specialized hashing schemes [21]) software IP-lookup algorithms have long been considered non-scalable to large networks, and routers used expensive CAM (Content Addressable Memory)-based architectures (see, e.g. [13]).

All early results on the average behavior of LC-tries belong to Andersson and Nilsson [1, 2]. Thus, they have shown that the average depth D_n of LC-tries over n keys produced by a symmetric memoryless process is only $\Theta\left(\log^* n\right)$. Under an asymmetric memoryless process (i.e. a process in which probabilities of possible outcomes are different), however, their average depth was shown to be $O\left(\log\log n\right)$ [2].

Devroye has recently published a more precise analysis of LC-tries in the symmetric case [7], showing, for instance, convergence in probability of their typical depth $D_n/\log^* n \to 1$ and height $H_n/\log_2 n \to 1$. At the same time, finding a more accurate characterization of LC-tries in the asymmetric model has been a long standing open problem. It was not clear, for example, how big is the factor in the $O\left(\log\log n\right)$ term of their average depth, or to which degree it can be affected by changing parameters of the input process.

In this paper we will attempt to answer these questions. Our main result[1], a first-term accurate asymptotic expression for D_n shows that LC-tries are, in fact, much more *sensitive to the asymmetry of the input process* than regular tries. Thus, if input data are generated by an extremely biased source it might not even make a practical sense to use level-compression, unless a special "symmetrization" technique [17] is employed to balance source's probabilities first.

---

[1] After this paper was submitted, the author has learned that the same formula has also been obtained by Devroye and Szpankowski [8]. Their proof, however, is different and relies on advanced probabilistic techniques.
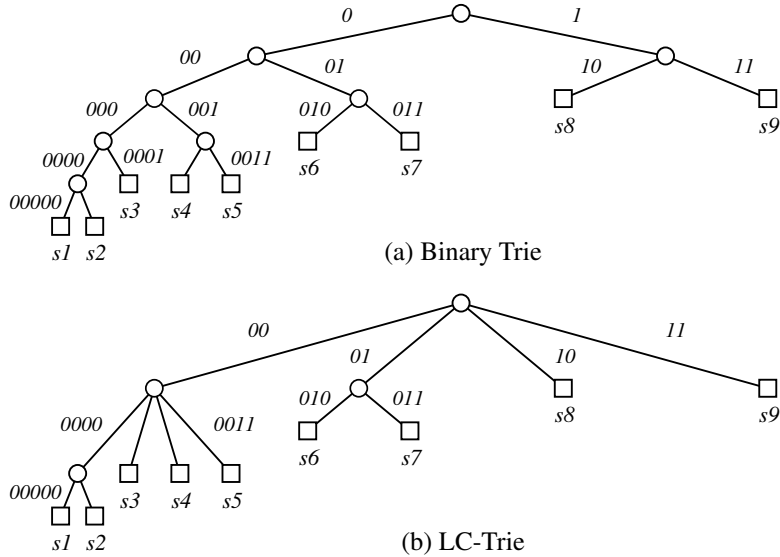
Figure 1: Examples of tries built from 9 binary strings: $s1 = 000000\ldots$, $s2 = 000010\ldots$, $s3 = 00011\ldots$, $s4 = 0010\ldots$, $s5 = 0011\ldots$, $s6 = 0100\ldots$, $s7 = 0110\ldots$, $s8 = 101\ldots$, $s9 = 110\ldots$.

2 Definitions and Main Results

Consider a set $S = \{s_1, \ldots, s_n\}$ of n distinct strings, where each string s_i $(i = 1 \ldots n)$ is a sequence of digits from a binary alphabet $\Sigma = \{0, 1\}$. [2].

Definition 1. A *binary trie* $T(S)$ over the set S is a data structure defined recursively as follows. If $n = 0$, the trie is *empty*. If $n = 1$ (i.e. S has only one string), the trie is an *external node* containing a pointer to this single string in S. If $n > 1$, the trie is an *internal node* containing pointers to two child tries: $T(S_0)$ and $T(S_1)$, constructed over two sets of *suffixes* of strings in S beginning with digits 0 and 1 correspondingly: $S_\alpha = \{u_j \mid \alpha\, u_j = s_i \in S\}$, $\alpha \in \Sigma$.

Definition 2. An *LC-trie* $T_{LC}(S)$ over S is a data structure defined recursively as follows. If $n = 0$, the trie is empty. 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* $(r \geqslant 1)$ containing pointers to 2^r child LC-tries: $T_{LC}(S_0), \ldots, T_{LC}(S_{2^r-1})$, constructed over suffixes of strings from S beginning with the corresponding r-digit words $S_v = \{u_j \mid v\, u_j = s_i \in S\}$, $v \in \Sigma^r$. The number of digits r is selected to be *the smallest number* such that at least one child trie becomes empty or turns into an external node: $\sum_{v \in \Sigma^r} 1\{|S_v| \leqslant 1\} \geqslant 1$.

Examples of a binary trie and its level-compression version are shown in Fig.1.

In order to study the average behavior of tries we assume that our input strings S are generated by a binary *memoryless* (or *Bernoulli*) source [4]. In this model, symbols of the 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*.

Our main attention will be focused on two closely-related quantities: the *average external path length* (i.e. the sum of lengths of paths from root to all external nodes in a trie) C_n, and the *average depth* D_n of LC-tries over n strings:

$$D_n = C_n/n. \tag{1}$$

Our main finding is formulated in the following theorem.

---

[2]We use binary alphabet for the simplicity of presentation only. All our results should remain correct (with the appropriate re-formulations of constants and bases of logarithms) for any finite alphabet
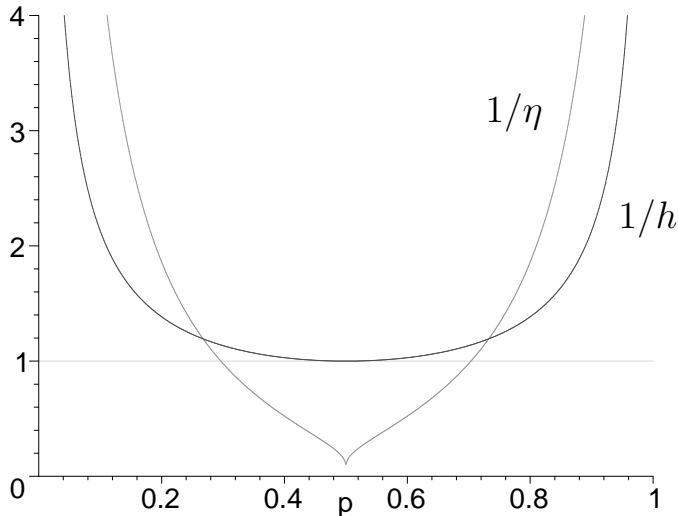
Figure 2: Plots of the $1/h$ and $1/\eta$ factors in the leading terms of depths of tries and LC-tries.

Theorem 1. *The average depth of LC-tries in the asymmetric memoryless model satisfies (with $n \to \infty$):*

$$\frac{D_n}{\log \log n} \to \frac{1}{\eta} \quad (pr.), \tag{2}$$

where n is the number of keys inserted in a trie, $\eta = -\log(1 - h/h_{-\infty})$, $h = -p \log p - q \log q$ is the (Shannon's) entropy of the binary source with probabilities p, $q = 1 - p$, $h_{-\infty} = -\log \min(p, q)$, and $\log := \log_b$, where b is a unit of information (e.g. bits or nats) being used.

This result can be compared to a well-known asymptotic expression of average depth of regular tries: $\frac{1}{h} \log n + O(1)$ (see, e.g. [12, 5, 19]). While the order of the first term here is much larger ($\log n$ vs. $\log \log n$ of LC-tries), its factor $1/h$ appears to be much more robust with respect to the asymmetry of the source.

We show the behavior of these two quantities $1/\eta$ and $1/h$ in Fig.2. Notice than when the source is close to become symmetric $1/\eta \to 0$, which explains cancellation of all terms in D_n up to $\log^* n$. At the same time, the factor $1/h$ in a symmetric case becomes $1/\log 2 > 0$ leaving the order of the leading term in the expression of depth of regular tries unchanged.

Looking at Fig.2, we can also notice that there exist points $p = \zeta$ and $p = 1 - \zeta$ where curves of $1/h$ and $1/\eta$ intersect. Numerical calculations show that this constant $\zeta \approx 0.2675709462$. It is clear, that when p is outside the $[\zeta, 1 - \zeta]$ range, the quantity $1/\eta$ becomes much larger than $1/h$, and this distance is rapidly increasing with the asymmetry of the source. In particular, it is easy to show that $\lim_{p \to 0} h/\eta = \infty$.

Described behavior of $1/\eta$ confirms the fact that LC-tries are much more sensitive to the asymmetry of the source than regular tries. If source is heavily biased, the number of complete sub-trees in a trie decreases, and at some point level-compression just wouldn't be able to speed-up such a structure. On the other hand, sharp increase of $1/\eta$ with $p \to 0$ suggests that probability-equalization techniques, such as "symmetrization" mapping [17] should provide very effective means for improving performance of LC-tries.

3 Analysis

We start with quoting an important result (cf. Pittel [16], Devroye [6], and Knessl and Szpankowski [11]) regarding the number of complete levels F_n in a random trie.

Proposition 1. *Let F_n be the number of complete levels in a trie oven n strings produced by a binary memoryless source. Then, when $p \neq q$ and $n \to \infty$:*

$$\frac{F_n - \frac{\log n}{h_{-\infty}}}{\log \log \log n} \to -\frac{1}{h_{-\infty}} \quad (pr.), \tag{3}$$

where $h_{-\infty} = -\log \min(p, q)$.

In order to construct an LC-trie one needs to find the first incomplete level r:

$$r = r_n := F_n + 1, \tag{4}$$

and then create a 2^{r_n}-digit root node. This process is recursively repeated as long as $n \geqslant 2$. We immediately discover the following.

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

$$
\begin{aligned}
C_n &= n + \sum_{k=2}^{n} \binom{n}{k} \sum_{s=0}^{r_n} \binom{r_n}{s} \left(p^s q^{r_n - s} \right)^k \left(1 - p^s q^{r_n - s} \right)^{n-k} C_k; \\
C_0 &= C_1 = 0.
\end{aligned} \tag{5}
$$

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 [12, 6.3-3], we can write:

$$
\begin{aligned}
C_n &= n + \sum_{k_1 + \ldots + k_{2^{r_n}} = n} \binom{n}{k_1, \ldots, k_{2^{r_n}}} p_1^{k_1} \ldots p_{2^{r_n}}^{k_{2^{r_n}}} \left(C_1 + \ldots + C_{2^{r_n}} \right), \\
&= n + \sum_{k=0}^{n} \binom{n}{k} \left(p_1^k (1 - p_1)^{n-k} + \ldots + p_{2^{r_n}}^k (1 - p_{2^{r_n}})^{n-k} \right) C_k.
\end{aligned} \tag{6}
$$

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

$$p_i = p^{s_i} q^{r_n - s_i}, \tag{7}$$

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 (6) and (7), we arrive at the expression (5) claimed by the lemma. $\qquad\square$

It shall be stressed that due to existence of variables r_n in (5) the analysis of this recurrence becomes much more difficult than in a case of regular tries (see e.g. Knuth [12, 6.3-3], Flajolet and Sedgewick [10], or Szpankowski [19]). It is not clear, for example, if there exists a closed form expression for this recurrence (all standard techniques from [12] seem to fail in this case). Without such a conversion, one still can try to use multivariate generation functions followed by their complex-domain singularity analysis (see, e.g. [9], [20]), but these are quite laborious and delicate (in their own sense) techniques.

Here, we will use much simpler approach. Since we already know that $D_n = O(\log \log n)$, we can try to substitute $C_n = \xi n \log \log n$ in (5) and then find upper and lower bounds for the parameter ξ such that recurrence (5) holds. If these bounds are tight, then we have successfully deduced the constant factor in the $O(\log \log n)$ term.

In order to realize this idea, we will need the following intermediate results. For simplicity, here and below we use natural logarithms.

Lemma 2. *Let $\theta \in (0, 1)$, $n \geqslant 2$, and $\lambda > 1$. Then there exists $0 < \zeta < \infty$, such that:*

$$n\theta \ln(\lambda + \ln(n\theta)) - \zeta \leqslant \sum_{k=2}^{n} \binom{n}{k} \theta^k (1 - \theta)^{n-k} k \ln(\lambda + \ln k) \leqslant n\theta \ln\left(\lambda + \ln\left(1 - \theta + n\theta\right)\right). \tag{8}$$

4

Proof. We start with a representation:

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

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

$$\ln(\alpha - \beta(1-\theta) + \beta\theta n) \;\; \leqslant \;\; \sum_{k=0}^{n} \binom{n}{k} \theta^k (1-\theta)^{n-k} \ln(\alpha + \beta k) \;\; \leqslant \;\; \ln(\alpha + \beta\theta n) \,. \tag{9}$$

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

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

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

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

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

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

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

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 (5). 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:

$$
\begin{aligned}
C_n &= n + \sum_{s=0}^{r_n} \binom{r_n}{s} \sum_{k=2}^{n} \binom{n}{k} \left(p^s q^{r_n-s}\right)^k \left(1 - p^s q^{r_n-s}\right)^{n-k} \xi k \ln(\lambda + \ln k) \\
&\leqslant n + n\xi \sum_{s=0}^{r_n} \binom{r_n}{s} p^s q^{r_n-s} \ln\left(\lambda + \ln\left(n\, p^s q^{r_n-s} + 1 - p^s q^{r_n-s}\right)\right) \\
&= n + n\xi \sum_{s=0}^{r_n} \binom{r_n}{s} p^s q^{r_n-s} \ln\left(\lambda + \ln\left(n\, p^s q^{r_n-s}\right) + \ln\left(1 + \frac{1}{n\, p^s q^{r_n-s}} - \frac{1}{n}\right)\right).
\end{aligned}
$$

Next, by Proposition 1, we know, that for any $\varepsilon > 0$, the probability that

$$
\left| r_n - \frac{\ln n}{-\ln q} + \frac{\ln\ln\ln n}{-\ln q} - 1 \right| \leqslant \varepsilon \ln\ln\ln n, \tag{10}
$$

holds true is approaching 1 with $n \to \infty$. Then:

$$
n\, p^s q^{r_n-s} \geqslant n q^{r_n} \geqslant n q^{-\frac{\ln n}{\ln q} - \left(\frac{1}{-\ln q} - \varepsilon\right)\ln\ln\ln n + 1} = q\left(\ln\ln n\right)^{1+\varepsilon\ln q}. \tag{11}
$$

and consequently:

$$
\ln\left(1 + \frac{1}{n\, p^s q^{r_n-s}} - \frac{1}{n}\right) \leqslant \ln\left(1 + \frac{1}{q\left(\ln\ln n\right)^{1+\varepsilon\ln q}} - \frac{1}{n}\right) =: \delta(n,\varepsilon), \tag{12}
$$

which is a relatively small $\left(\delta(n,\varepsilon) = O\left((\ln\ln n)^{-1-\varepsilon\ln q}\right)\right)$ when $1 + \varepsilon\ln q > 0$) quantity.

By incorporating this bound and using Lemma 3:

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

where $h = -p\ln p - q\ln q$ is the entropy. Now, by applying (10), we have:

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

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

$$
\begin{aligned}
\xi &\leqslant \frac{1}{-\ln\left(1 - \frac{h}{-\ln q} + \frac{\lambda + h\left(\frac{1}{-\ln q} + \varepsilon\right)\ln\ln\ln n - h + \delta(n,\varepsilon)}{\ln n}\right) + \ln\left(1 + \frac{\lambda}{\ln n}\right)} \\
&= \frac{1}{-\ln\left(1 - \frac{h}{-\ln q}\right)}\left(1 + O\left(\frac{\varepsilon \ln\ln\ln n}{\ln n}\right)\right). \tag{13}
\end{aligned}
$$

The procedure for finding a lower bound is very similar:

$$
\begin{aligned}
C_n &= n + \sum_{s=0}^{r_n} \binom{r_n}{s} \sum_{k=2}^{n} \binom{n}{k} \left(p^s q^{r_n-s}\right)^k \left(1 - p^s q^{r_n-s}\right)^{n-k} \xi k \ln(\lambda + \ln k) \\
&\geqslant n + n\xi \sum_{s=0}^{r_n} \binom{r_n}{s} p^s q^{r_n-s} \log\left(\lambda + \ln\left(n\, p^s q^{r_n-s}\right)\right) - \zeta \\
&= n + n\xi \sum_{s=0}^{r_n} \binom{r_n}{s} p^s q^{r_n-s} \ln\left(\lambda + \ln n + r_n \ln q + s \ln(p/q)\right) - \zeta \\
&\geqslant n + n\xi \ln\left(\lambda + \ln n - h\, r_n - q \ln(p/q)\right) - \zeta \\
&\geqslant n + n\xi \ln\left(\lambda + \ln n \left(1 - \frac{h}{-\ln q}\right) + h\left(\frac{1}{-\ln q} - \varepsilon\right) \ln\ln\ln n - h - q\ln(p/q)\right) - \zeta.
\end{aligned}
$$

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

$$
\begin{aligned}
\xi &\geqslant \frac{1 - \zeta/n}{-\ln\left(1 - \frac{h}{-\ln q} + \frac{\lambda + h\left(\frac{1}{-\ln q} - \varepsilon\right) \ln\ln\ln n - h - q\ln(p/q)}{\ln n}\right) + \ln\left(1 + \frac{\lambda}{\ln n}\right)} \\
&= \frac{1}{-\ln\left(1 - \frac{h}{-\ln q}\right)} \left(1 + O\left(\frac{\varepsilon \ln\ln\ln n}{\ln n}\right)\right).
\end{aligned}
\tag{14}
$$

By combining our bounds (13) and (14) 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:

$$
\xi \to \frac{1}{-\ln\left(1 - \frac{h}{-\ln q}\right)} \quad (pr.).
$$

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] 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.

[4] T. M. Cover and J. M. Thomas, *Elements of Information Theory.* (John Wiley & Sons, New York, 1991).

[5] L. Devroye, A Note on the Average Depths in Tries, *SIAM J. Computing* **28** (1982) 367–371.

[6] L. Devroye, A Note on the Probabilistic Analysis of PATRICIA Tries, *Rand. Structures & Algorithms* **3** (1992) 203–214.

[7] L. Devroye, Analysis of Random LC Tries, *Rand. Structures & Algorithms* **19** (3-4) (2001) 359–375.

[8] L. Devroye, and W. Szpankowski, Probabilistic Behavior of Asymmetric LC-Tries, *Rand. Structures & Algorithms* – submitted.

[9] P. Flajolet and A. Odlyzko, Singularity Analysis of Generating Functions, *SIAM J. Discrete Math.*, **3** (2) (1990) 216–240.

[10] P. Flajolet and R. Sedgewick, Digital Search Trees Revisited, *SIAM J. Computing* **15** (1986) 748–767.

[11] C. Knessl, and W. Szpankowski, On the Number of Full Levels in Tries, *Rand. Structures & Algorithms* **25** (2004) 247–276.

[12] D. E. Knuth, *The Art of Computer Programming. Sorting and Searching. Vol. 3.* (Addison-Wesley, Reading MA, 1973).

[13] A. McAuley and I. Francis, Fast routing table lookup using CAMS, *Proc. INFOCOM* (1993) 1382–1391.

[14] S. Nilsson and G. Karlsson, Fast IP look-up for Internet routers. *Proc. IFIP 4th International Conference on Broadband Communication* (1998) 11–22.

[15] S. Nilsson and G. Karlsson, IP-address look-up using LC-tries, *IEEE J. Selected Areas in Communication* **17** (6) (1999) 1083–1092.

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

[17] Yu. A. Reznik and W. Szpankowski, Improved Behaviour of Tries by the Symmetrization of the Source, *Proc. IEEE Data Compression Conference (DCC'02)* (Snowbird, Utah, 2002) 253–262.

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

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

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

[21] M. Waldvogel, G. Varghese, J. Turner, and B. Plattner, Scalable high speed IP routing lookups. *Proc. ACM SIGCOMM'97* **27** (4) (1997) 25–36.