On Time-Space Efficiency of Digital Trees with Adaptive Multi-Digit Branching*

Yuriy A. Reznik
RealNetworks, Inc.
2601 Elliott Avenue, Seattle, WA
yreznik@ieee.org

March 27, 2003

Abstract

We consider a class of digital trees (*tries*) with adaptive selection of degrees of their nodes. This class includes *LC-tries* of Andersson and Nilsson (1993) which recursively replace all complete subtrees in the original tries with larger (multi-digit) nodes, as well as dynamic tries of Nilsson and Tikkanen (1998) which recursively replace all subtrees of bounded *sparseness* (a ratio of the number of missing nodes at the last level to the total number of nodes at this level).

In this paper we study the average behavior of such tries with respect to a hybrid *time/space efficiency criterion*. We demonstrate that there exists an interesting connection between the efficiency and sparseness of nodes in adaptive tries. In particular, we show that in a symmetric memoryless model, the optimal in a sense of time/space efficiency nodes, are $1/e$-times ($\approx 36.8\%$) sparse. On the other hand, if the source is asymmetric, the sparseness of the time/space efficient nodes is somewhat larger, asymptotically (with large number of strings) approaching 50%.

These results can be used to support the trie construction algorithm of Nilsson and Tikkanen, and suggest the optimal choice of constants in this procedure.

1 Introduction

Digital trees (also known as *radix search trees*, or *tries*) represent a convenient way of organizing alphanumeric sequences (strings) of variable lengths that facilitates their fast retrieving, searching, and sorting [14, 19, 26]. If we designate a set of n distinct strings as $S = \{s_1, \ldots, s_n\}$, and assume that each string is a sequence of symbols from a finite alphabet $\Sigma = \{\alpha_1, \ldots, \alpha_m\}$, then a trie $T(S)$ over S can be constructed 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 v pointers to the child tries: $T(S_1), \ldots, T(S_m)$, where each set S_i ($1 \leqslant i \leqslant m$) contains suffixes of all strings from S that begin with a corresponding first symbol. For example, if a string $s = uw$ (u is a first symbol, and w is a string containing the remaining symbols of s), and $u = \alpha_i$, then the string w will go into S_i. Thus, after all child tries $T(S_1), \ldots, T(S_m)$ are recursively processed, we arrive at a tree-like data structure, where the original strings $S = \{s_1, \ldots, s_n\}$ can be uniquely identified by the paths from the root node to non-empty external nodes (see Fig. 1.a).

It is well known, that the *average time of a successful search* in a trie is asymptotically $\frac{1}{h} \log n + O(1)$, where h is the entropy of a stochastic process used to produce n input strings (cf. [19, 7, 12, 16, 24, 18, 28, 17]). The *average number of internal nodes* in a trie is asymptotically $\frac{\log e}{h} n (1 + o(1))$[1].

*Author's translation from Russian original text, published in "Kibernetika i Sistemnyi Analiz" (Cybernetics and Systems Analysis), Vol 39, No 1, 2003, pp 177-188. Original paper was submitted on June 18, 2002.

[1]The bases of logarithms in these formulas correspond to a unit of information (e.g. *bits* or *nats*) used to measure the entropy of the source h

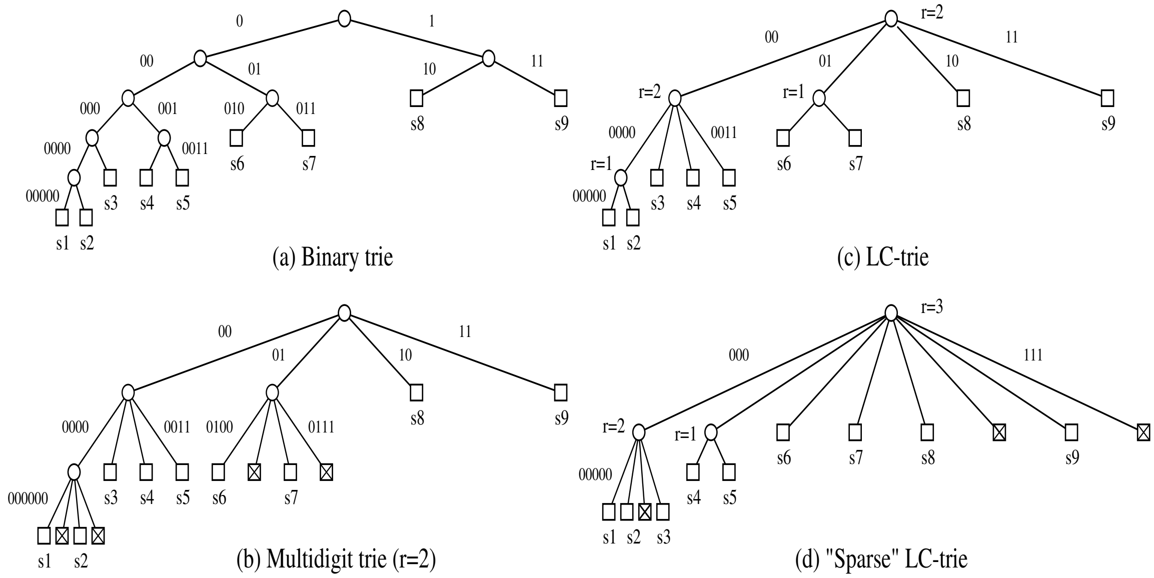

Figure 1: Examples of tries built from 9 binary strings: $s1 = 000000\ldots$, $s2 = 000010\ldots$, $s3 = 00011\ldots$, $s4 = 0010\ldots$, $s5 = 0011\ldots$, $s6 = 0100\ldots$, $s7 = 0110\ldots$, $s8 = 101\ldots$, $s9 = 110\ldots$.

These estimates are known to be correct for a number of standard stochastic processes, such as *memoryless*, *Markovian*, ψ-mixed models [24, 17, 29].

In an effort to reduce the search time several modifications of the original trie structure have been proposed (cf. [27, 21, 5, 19]. For example, *multi-digit tries* (we use this term after [2]) accelerate search by processing some fixed number $r \geqslant 2$ of symbols in each node (see Fig.1.b). Assuming that branching is implemented using lookup tables, such tries should be approximately r-times faster than regular tries. However, such an improvement comes at a cost of about m^r/r-times more memory, since r-digit nodes must have m^r pointers, most of which are wasted if r is large [19].

This motivated the development of *adaptive multi-digit tries* (cf. [2, 22, 25]), in which the parameter r (the number of digits to be processed) can be changed from one node to another. The best known example of such a structure is a *level-compressed trie* (or *LC- trie*) of Andersson and Nilsson [2], which recursively replaces all complete subtrees of the original (m-ary) trie with multi-digit nodes (see Fig.1.c). It has been shown (cf. [23, 8, 3]), that in a memoryless model an LC-trie creates nodes with $r \to (\log n - \log\log n)/h_{-\infty}$, where n is the number of strings processed by a node, and $h_{-\infty} = -\log(\min\{p_i\})$, and p_i ($1 \leqslant i \leqslant m$) are the probabilities of symbols produced by the source. When the memoryless source is symmetric ($p_i = 1/m$), the expected search time in an LC-trie is only $\sim \log^* n$ [2, 9], however, it becomes $O(\log\log n)$-large in the asymmetric case [3].

Even faster versions of adaptive tries can be constructed by allowing some additional *sparse* levels (i.e. levels containing empty sub-tries) to be included in multi-digit nodes (see Fig.1.d). Such a strategy is also desirable for *dynamic construction* of such tries (since empty positions can be used to avoid frequent resizing of their nodes [22]). However, an overly aggressive usage of sparse levels will also lead to an increased memory usage, which brings a problem of finding an *optimal (in time/space sense) strategy* for construction of such tries.

An interesting experimental study on this subject has been recently published by Nilsson and Tikkanen [22]. They proposed to construct dynamic multi-digit tries by imposing upper and lower bounds on the *sparseness* (a ratio of the number of pointers to empty sub-tries to the total number of pointers in a node) of their nodes. For example, if during an insertion of a new string, the sparseness of a current node falls below a given lover bound, their algorithm increments parameter r for this node (i.e. adds the top-most level from its child sub-tries). On the other hand, if at some point, the sparseness of a current node became higher than a given upper bound, the algorithm decrements parameter r for this node (i.e. removes the last level and merges its (single-digit) nodes with the

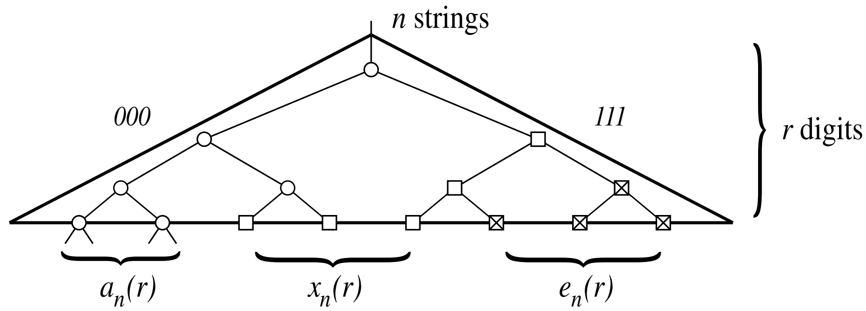

Figure 2: Parameters of an r-digit node processing n strings.

remaining sub-tries). Nilsson and Tikkanen have concluded that constraining sparseness of nodes in $25\% \ldots 50\%$ range provides a good compromise for various practical applications.

The paper of Nilsson and Tikkanen [22], however, left open the possibility (and emphasized the need) for an *analytic explanation of the relationship between the sparseness of nodes and the average time/space performance of multi-digit tries*. Finding such an explanation is the main goal of this paper.

In this paper, we study several natural parameters of nodes in multi-digit tries, and argue, that the effectiveness of an internal multi-digit node from a successful search-time perspective is reflected by the number of pointers to external nodes immediately following this node (i.e. by the number of strings separated (uniquely identified) by this node). On the other hand, the usage of space is reflected by the total number of pointers in the node (e.g. m^r for a node combining r levels in m-ary alphabet). Hence, an appropriate hybrid *time/space efficiency* metric is provided by a ratio of the number of pointers to external nodes to the total number of pointers the a node.

We analyze the average behavior of both sparseness and time/space efficiency of multi-digit nodes in a memoryless model, and derive their asymptotic expressions when the number of strings n passing through such nodes is large.

Using these expressions we show that the optimal in a sense of time/space efficiency nodes are actually $1/e$-times ($\approx 36.8\%$) sparse when the source is symmetric. When the source is asymmetric, the sparseness of the time/space efficient nodes is somewhat larger, asymptotically (with large number of strings) approaching 50%.

Even more importantly, we discover, that in a symmetric model, it is possible to construct tries with a constant ($O(1)$) average time/space efficiency of nodes. This suggests, that symmetric $1/e$-sparse tries should be $O(1)$-fast and $O(n)$-large at the same time. In the asymmetric case, however, the average time/space efficiency of nodes is asymptotically decreasing as $O\left(\frac{1}{\sqrt{\log n}}\right)$, making them somewhat less appealing compared to the N-trees and recursive hashing schemes (cf.[10, 11, 30, 20]).

This paper is organized as follows. In the next section, we give formal definitions and present our main results. All proofs are delayed until Section 3, which is also used to provide a brief description of the required tools of the asymptotic analysis.

2 Definitions and Main Results

Consider an r-digit node processing n binary strings depicted in Fig.2. We have the following parameters:

$a_n(r)$ the number of pointers to internal nodes attached to this node;

$x_n(r)$ the number of pointers to external nodes (strings);

$e_n(r)$ the number of empty pointers (i.e. pointers to empty child tries).

Observe that the total number of pointers in such a node is

$$a_n(r) + x_n(r) + e_n(r) = 2^r. \tag{1}$$

We now can introduce two derivative parameters, reflecting the effectiveness of such a node in both time- and space- domains.

Definition 1. A *sparseness* of an r-digit node processing n strings $\varepsilon_n(r)$ is a ratio of the number of pointers to empty nodes to the total number of pointers in this node:

$$\varepsilon_n(r) = \frac{e_n(r)}{2^r}. \tag{2}$$

Definition 2. A time/space *efficiency* of an r-digit node processing n strings $\eta_n(r)$ is a ratio of the number pointers to the external nodes (i.e. the number of strings uniquely identified by this node) to the total number of pointers in this node:

$$\eta_n(r) = \frac{x_n(r)}{2^r}. \tag{3}$$

Remark 1. The actual meaning of the second parameter $\eta_n(r)$ deserves some additional explanation. Observe, that $\eta_n(r) = 1$ implies that $x_n(r) = n = 2^r$, which means that n strings form a complete $r = \log_2 n$-level trie, which is fully covered by a single multi-digit node. A value $\eta_n(r) = 1/2$ corresponds to a number of possible situations, such as, for example, when a node with $r = \log_2 n$ levels contains only $x_n(r) = n/2$ external nodes, or when a node that successfully parses all n strings ($x_n(r) = n$) has twice many pointers: $2^r = 2n$, and so on. Lower values of $\eta_n(r)$ reflect even worse ability of nodes to parse strings at a rate comparable with the growth of their size.

Remark 2. It can be shown (using the *ratio inequality* [15]) that the efficiencies of individual nodes have the following simple connection with the parameters of the entire trie:

$$\frac{n}{S_n} = \frac{\sum_j x_{n_j}(r_j)}{\sum_j 2^{r_j}} \leqslant \max_j \left\{ \frac{x_{n_j}(r_j)}{2^{r_j}} \right\} = \max_j \left\{ \eta_{n_j}(r_j) \right\}, \tag{4}$$

where j is used to scan all internal nodes, and S_n is the total number of pointers in a trie constructed from n strings. The equality in 4 is attained when all nodes in a trie have the same efficiencies.

Remark 3. It shall be stressed that a formula (3) for the parameter $\eta_n(r)$ cannot be immediately used for guiding the dynamic construction of tries. For example, we know that all nodes without immediately attached external nodes have $\eta_n(r) = 0$, so that a node with at least one external node, regardless of its size will be considered as a better choice. In practice, this will lead to extreme variation of sizes of nodes created by maximizing their $\eta_n(r)$. This observation, however, does not affect the usefulness of this metric in studying the *average behavior* of tries. In fact, what we will show, is that the expected behavior of $\eta_n(r)$-optimal nodes is indeed, quite reasonable, and that it can be successfully simulated by imposing the appropriate constrains on the sparseness of nodes.

In order to study the average behavior of tries we will assume that our input strings S are generated by a binary *memoryless* (or *Bernoulli*) source [6]. In this model, symbols of the alphabet $\Sigma = \{0, 1\}$ occur independently of one another, so that if x_j is the j-th symbol produced by this source, then for any j: $\Pr\{x_j = 0\} = p$, and $\Pr\{x_j = 1\} = q = 1 - p$. If $p = q = 0.5$, such source is called *symmetric*, otherwise it is *asymmetric* (or *biased*).

Now, we can define the quantities of our main interest:

$$\bar{\eta}_n(r) \;:=\; E\{\eta_n(r)\} = \frac{E\{x_n(r)\}}{2^r}, \tag{5}$$

$$\bar{\varepsilon}_n(r) \;:=\; E\{\varepsilon_n(r)\} = \frac{E\{e_n(r)\}}{2^r}, \tag{6}$$

where the expectations are taken over all possible tries over n strings when parameters of the memoryless source (p and q) are fixed.

We first present our result regarding the expected efficiency of nodes.

Theorem 1. *The expected efficiency $\bar{\eta}_n(r)$ of an r-digit node processing n binary strings from a memoryless source satisfies:*

$$\bar{\eta}_n(r) = n2^{-r} \sum_{s=0}^{r} \binom{r}{s} p^s q^{r-s} \left(1 - p^s q^{r-s}\right)^{n-1}. \tag{7}$$

If $p \neq q$ and

$$r = \frac{\log n}{h_\varepsilon} + x\sigma_\varepsilon \sqrt{\log n}, \tag{8}$$

where

$$h_\varepsilon = -\frac{1}{2}\log p - \frac{1}{2}\log q, \tag{9}$$

$$h_\varepsilon^{(2)} = \frac{1}{2}\log^2 p + \frac{1}{2}\log^2 q, \tag{10}$$

$$\sigma_\varepsilon^2 = \frac{h_\varepsilon^{(2)} - h_\varepsilon^2}{h_\varepsilon^3}, \tag{11}$$

and $x = O(1)$, then, asymptotically, with $n \to \infty$:

$$\bar{\eta}_n(r) = \frac{1}{\sqrt{2\pi}\sigma_\varepsilon h_\varepsilon \sqrt{\log n}} e^{-\frac{x^2}{2}} + O\left(\frac{1}{\log n}\right). \tag{12}$$

Observe, that the asymptotic expression (12) has a clear point of maximum when $x = 0$. Hence, based on the precision of our original approximation (8), we can conclude that an r-digit node has a maximum average efficiency when $r = r^*$, where

$$r^* = \frac{\log n}{h_\varepsilon} + o\left(\sqrt{\log n}\right). \tag{13}$$

This statement holds in both symmetric and asymmetric cases (in the symmetric case, we simply notice that $\bar{\eta}_n(r) \sim n\,2^{-r} e^{-n\,2^{-r}}$, where maximum is attained when $n\,2^{-r} = 1$).

We now turn our attention the expected sparseness of multi-digit nodes. We discover the following.

Theorem 2. *The expected sparseness $\bar{\varepsilon}_n(r)$ of an r-digit node processing n binary strings from a memoryless source satisfies:*

$$\bar{\varepsilon}_n(r) = 2^{-r} \sum_{s=0}^{r} \binom{r}{s} \left(1 - p^s q^{r-s}\right)^{n}. \tag{14}$$

If $p \neq q$ and

$$r = \frac{\log n}{h_\varepsilon} + x\sigma_\varepsilon \sqrt{\log n}, \tag{15}$$

where $x = O(1)$, and h_ε, σ_ε are as defined in (9-11), then, asymptotically, with $n \to \infty$:

$$\bar{\varepsilon}_n(r) = \Phi\left(x - x^2 \frac{\sigma_\varepsilon h_\varepsilon}{2\sqrt{\log n}}\right) - \gamma \frac{1}{\sqrt{2\pi}\sigma_\varepsilon h_\varepsilon \sqrt{\log n}} e^{-\frac{x^2}{2}} + O\left(\frac{1}{\log n}\right), \tag{16}$$

where

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-\frac{t^2}{2}} dt, \tag{17}$$

is the distribution function of the standard normal distribution [1], and $\gamma = 0.5772\ldots$ is the Euler constant.

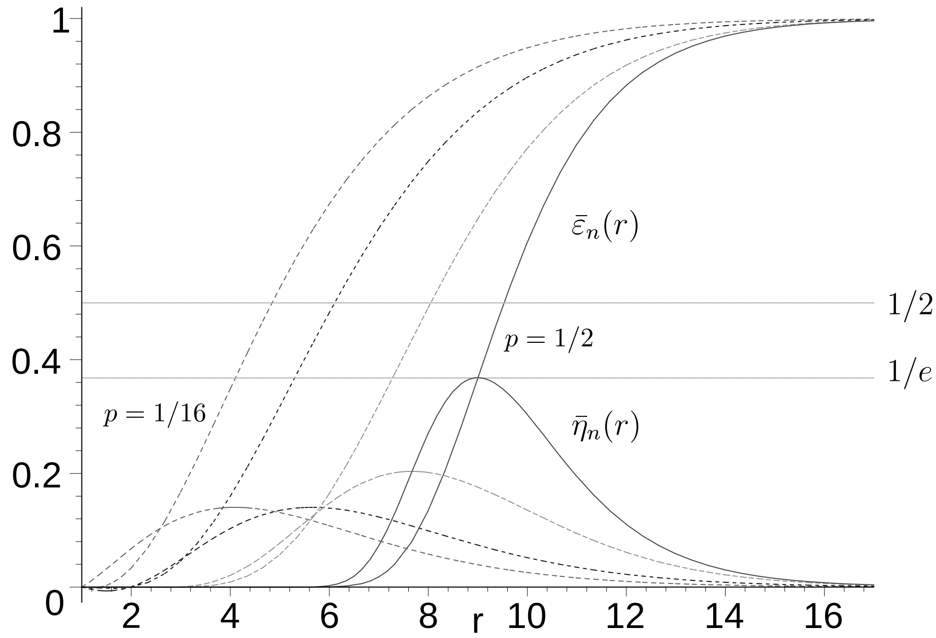

Figure 3: Plots of two parameters $\bar{\eta}_n(r)$ and $\bar{\varepsilon}_n(r)$ of nodes under memoryless sources with probabilities: $p = 1/2$ (solid line), $p = 1/4$, $p = 1/8$, and $p = 1/16$ (dotted lines, lowering the p moves centers of both $\bar{\eta}_n(r)$ and $\bar{\varepsilon}_n(r)$ to the left). Horizontal lines correspond to levels $1/2$ and $1/e$. Plots are rendered for $n = 512$.

From the above asymptotic expression (16) it should be clear that when a node reaches a point $r = r^*$ (13) of its the maximum average efficiency, its sparseness is asymptotically:

$$\bar{\varepsilon}_n(r^*) \sim \frac{1}{2} - \gamma \frac{1}{\sqrt{2\pi}\sigma_\varepsilon h_\varepsilon \sqrt{\log n}},$$

and approaching $1/2$ when n is sufficiently large.

When the source is symmetric, formula (14) becomes $\bar{\varepsilon}_n(r) = \left(1 - 2^{-r}\right)^n$, and it is clear that for large n: $\bar{\varepsilon}_n(r^*) \to 1/e$.

In Fig.3 we plot the results of numerical calculations using exact expressions (7, 16) for both functions $\bar{\eta}_n(r)$ and $\bar{\varepsilon}_n(r)$ under and several types of memoryless sources ($p = 1/2 ... 1/16$). Here, is can be seen that the maximum points of $\bar{\eta}_n(r)$ are very close (within γ) to the corresponding median points of $\bar{\varepsilon}_n(r)$, thus confirming findings of our asymptotic analysis.

3 Analysis

Consider an r-digit node processing n binary strings. By $a_n^k(r)$ we denote *the number of its child nodes containing exactly k strings* from the original set of n. Our previously introduced quantities, such as the number of pointers to external nodes $x_n(r)$ and the number of empty pointers $x_n(r)$ are, indeed, just special cases of $a_n^k(r)$:

$$\begin{aligned} e_n(r) &= a_n^0(r), & (18) \\ x_n(r) &= a_n^1(r). & (19) \end{aligned}$$

The next lemma provides an exact formula for the average value of $a_n^k(r)$ in a memoryless model.

Lemma 1. *The quantity $\bar{a}_n^k(r) := E\left\{a_n^k(r)\right\}$ in a memoryless model satisfies:*

$$\bar{a}_n^k(r) = \binom{n}{k} \sum_{s=0}^{r} \binom{r}{s} \left(p^s q^{r-s}\right)^k \left(1 - p^s q^{r-s}\right)^{n-k}. \quad (20)$$

Proof. Consider an r-digit node processing n strings. Assuming that each of its 2^r branches have probabilities $p_1, \ldots, p_{2^r}$, and using the standard technique for enumeration of nodes in tries [19, 6.3-3], we can write:

$$\begin{aligned}
\bar{a}_n^k &= \sum_{l_1+\ldots+l_{2^r}=n} \binom{n}{l_1 \ldots l_{2^r}} p_1 \ldots p_{2^r} \left(\delta_{kl_1} + \ldots + \delta_{kl_{2^r}}\right), \\
&= \sum_{l=0}^{n} \binom{n}{l} \left(p_1^l \left(1-p_1\right)^{n-l} + \ldots + p_{2^r}^l \left(1-p_{2^r}\right)^{n-l}\right) \delta_{kl}, \\
&= \binom{n}{k} \left(p_1^k \left(1-p_1\right)^{n-k} + \ldots + p_{2^r}^k \left(1-p_{2^r}\right)^{n-k}\right), \qquad (21)
\end{aligned}$$

where δ_{ij} is a Kronecker delta. Recall now, that we are actually working with an r-digit node, so given the probabilities of each digit (p and $q = 1-p$ for symbols 0 and 1 correspondingly) we can write:

$$p_i = p^{s_i} q^{r-s_i}, \qquad (22)$$

where s_i is the number of occurrences of symbol 0 in a string leading to a branch i ($1 \leqslant i \leqslant 2^r$). Combining (21) and (22), we arrive at the expression (20) claimed by the lemma. □

Using this result and our definitions of the average efficiency (5) and the average sparseness (6) of r-digit nodes we arrive at:

$$\begin{aligned}
\bar{\eta}_n(r) &= \frac{\bar{a}_n^1}{2^r} = n\, 2^{-r} \sum_{s=0}^{r} \binom{r}{s} p^s q^{r-s} \left(1 - p^s q^{r-s}\right)^{n-1}, \\
\bar{\varepsilon}_n(r) &= \frac{\bar{a}_n^0}{2^r} = 2^{-r} \sum_{s=0}^{r} \binom{r}{s} \left(1 - p^s q^{r-s}\right)^{n},
\end{aligned}$$

which proves the first pair of expressions (7) and (14) in our Theorems.

In order to study asymptotic behavior of these formulas for large n, we first convert them into alternating sums:

$$\begin{aligned}
\bar{\eta}_n(r) &= n\, 2^{-r} \sum_{k=0}^{n-1} \binom{n-1}{k} (-1)^k \left(p^{k+1} + q^{k+1}\right)^r \\
&= -2^{-r} \sum_{k=0}^{n} \binom{n}{k} k (-1)^k \left(p^k + q^k\right)^r, \qquad (23) \\
\bar{\varepsilon}_n(r) &= 2^{-r} \sum_{k=0}^{n} \binom{n}{k} (-1)^k \left(p^k + q^k\right)^r, \qquad (24)
\end{aligned}$$

and apply Rice's integral method (cf. Knuth [19, Ex.5.2.2-54], Flajolet and Sedgewick [12, 13]).

We quote the following formulation of this method from [29].

Lemma 2 (S.O.Rice). *Let $f(z)$ be of polynomial growth at infinity, and analytical left to the vertical line $\left(\frac{1}{2} - m - i\infty\,,\ \frac{1}{2} - m + i\infty\right)$. Then:*

$$\begin{aligned}
&\sum_{k=m}^{n} \binom{n}{k} (-1)^k f(k) \\
&= \frac{1}{2\pi i} \int_{\frac{1}{2}-m-i\infty}^{\frac{1}{2}-m+i\infty} f(-z) B(n+1, z)\, dz \qquad (25) \\
&= \frac{1}{2\pi i} \int_{\frac{1}{2}-m-i\infty}^{\frac{1}{2}-m+i\infty} f(-z) n^{-z} \Gamma(z) \left(1 + O\left(\frac{1}{n}\right)\right) dz,
\end{aligned}$$

where $B(x,y) = \Gamma(x)\Gamma(y)/\Gamma(x+y)$ is the beta-function.

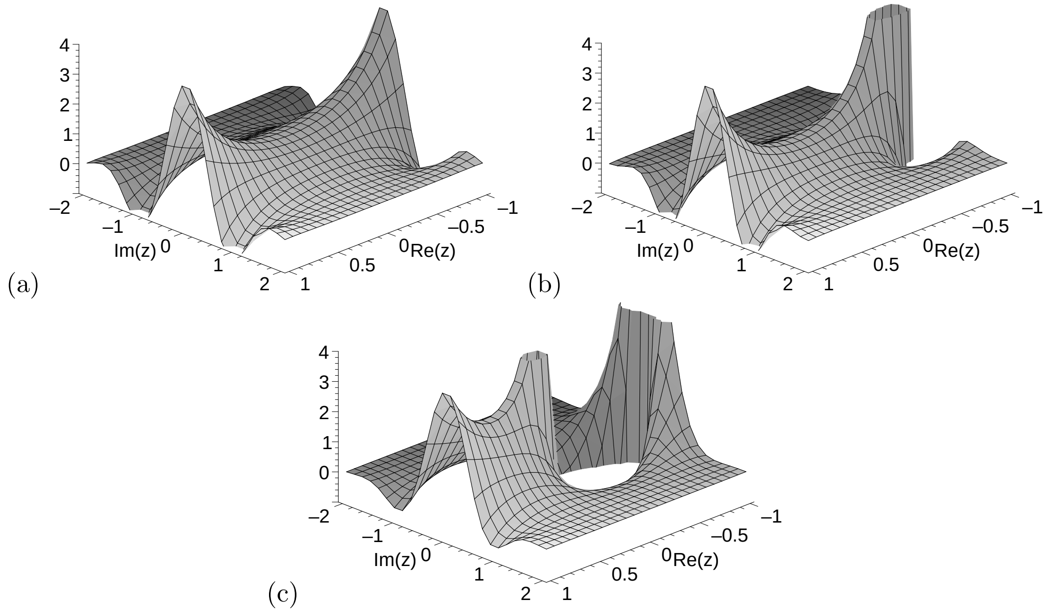

Figure 4: Plots of the following expressions: (a) $f_n(z) = n^{-z} 2^{-r} \left(p^{-z} + q^{-z}\right)^r$ – has a saddle point near $z = 0$; (b) $\Gamma(1+z)\, f_n(z)$ – has a saddle point and a distant pole (at $z = -1$); (c) $\Gamma(z)\, f_n(z)$ – has a saddle point coinciding with a pole ($z = 0$) of $\Gamma(z)$. Plots are rendered for $n = 1024$, $r = 8$, and $p = 0.23$.

Thus, our sums (23) and(24) are asymptotically equivalent to the following two integrals:

$$
\begin{aligned}
I_\eta(n) &= \frac{1}{2\pi i} \int_{\frac{1}{2}-i\infty}^{\frac{1}{2}+i\infty} \Gamma(1+z)\, f_n(z) dz, && (26) \\
I_\varepsilon(n) &= \frac{1}{2\pi i} \int_{\frac{1}{2}-i\infty}^{\frac{1}{2}+i\infty} \Gamma(z)\, f_n(z) dz, && (27)
\end{aligned}
$$

where

$$
f_n(z) = n^{-z} 2^{-r} \left(p^{-z} + q^{-z}\right)^r. \tag{28}
$$

We present plots of the function $f_n(z)$ and full expressions under both integrals in Fig.4. Observe, that under condition (8), $f_n(z)$ has a saddle point, which in the case of our first integral (26) is relatively distant from a nearest pole of the Γ-function, while in the second case (27) the saddle point coincides with it.

While it is clear that the first integral can be taken directly using the *saddle point method* [4], the situation with a coinciding pole is a bit more complicated, but still can be handled using the *Van der Waerden technique* [31]. Due to the space constrains, below we will only sketch the main steps in taking these integrals, but it is not difficult to make them rigorous.

In both cases we start with the following representation

$$
f_n(z) = e^{-g_n(z)}, \tag{29}
$$

where

$$
g_n(z) = z \log n + r \log 2 - r \log\left(p^{-z} + q^{-z}\right). \tag{30}
$$

Now, in order to apply the saddle point method, we must write

$$
\begin{aligned}
g_n(z) &= \left(\log n - r h_\varepsilon\right) z - r \frac{h_\varepsilon^{(2)} - h_\varepsilon^2}{2} z^2 + O\left(r\, z^4\right) \\
&= \alpha - \beta \left(z - s_0\right)^2 + O\left(\log n\, z^4\right), && (31)
\end{aligned}
$$

where

$$\beta = r\frac{h_\varepsilon^{(2)} - h_\varepsilon^2}{2} = \frac{1}{2}\sigma_\varepsilon^2 h_\varepsilon^2 \left(\log n + x\sigma_\varepsilon h_\varepsilon \sqrt{\log n}\right), \tag{32}$$

$$\begin{aligned} \alpha &= \frac{(\log n - rh_\varepsilon)^2}{2r\left(h_\varepsilon^{(2)} - h_\varepsilon^2\right)} = \frac{x^2}{2}\frac{1}{1 + x\sigma_\varepsilon h_\varepsilon/\sqrt{\log n}} \\ &= \frac{x^2}{2} + O\left(\frac{1}{\sqrt{\log n}}\right), \end{aligned} \tag{33}$$

$$s_0 = \frac{\log n - rh_\varepsilon}{r\left(h_\varepsilon^{(2)} - h_\varepsilon^2\right)} = \frac{-x}{\sigma_\varepsilon h_\varepsilon \sqrt{\log n}} + O\left(\frac{x^2}{\log n}\right), \tag{34}$$

and h_ε, $h_\varepsilon^{(2)}$, σ_ε, and x are as defined in (9-11,8).

Let now $z = s_0 + it$ $(-\infty < t < \infty)$ (i.e. we shift the path of the integration to cross the saddle point). To evaluate the contribution of $\Gamma(1+z)$ around $z = s_0 \to 0$, we use:

$$\begin{aligned} \Gamma(1+z) &= 1 - \gamma z + \left(\frac{\pi^2}{12} + \frac{\gamma^2}{2}\right) z^2 + O\left(z^3\right) \\ &= 1 - \left(\frac{\pi^2}{12} + \frac{\gamma^2}{2}\right) t^2 + O\left(s_0\right) + O\left(s_0 t^2\right) + O\left(t^4\right) \\ &\quad + i\left\{-\gamma t + O\left(s_0 t\right) + O\left(t^3\right)\right\}, \end{aligned}$$

and putting everything together we obtain:

$$\begin{aligned} I_\eta(n) &= \frac{1}{2\pi}\, e^{-\alpha} \int_{-\infty}^{\infty} \Gamma\left(1 + s_0 + it\right)\, e^{-\beta t^2} \times \\ &\qquad \times \left(1 + O\left(\log n\ (s_0 + it)^4\right)\right) dt \\ &= \frac{1}{2\sqrt{\pi\beta}}\, e^{-\alpha} + O\left(\frac{1}{\log n}\right). \end{aligned} \tag{35}$$

where the cancellation of the imaginary terms and convergence of the real ones are due to the following properties of Gauss integral (see, e.g. de Bruijn [4, Chapter 4]):

$$\begin{aligned} &\frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\infty} t^k e^{-\beta t^2} dt \\ &= \left[\begin{array}{ll} 0, & \text{if } k = 1, 3, 5, \ldots, \\ \frac{1}{\sqrt{2\beta}}\frac{k!}{(k/2)!\, 2^k \beta^{k/2}}, & \text{if } k = 0, 2, 4, \ldots. \end{array}\right. \end{aligned} \tag{36}$$

Now by expanding (33) and (32) in (35) we produce:

$$\begin{aligned} I_\eta(n) &= \frac{e^{-\frac{x^2}{2}\frac{1}{1+x\sigma_\varepsilon h_\varepsilon/\sqrt{\log n}}}}{\sqrt{2\pi}\sigma_\varepsilon h_\varepsilon \sqrt{\log n + x\sigma_\varepsilon h_\varepsilon \sqrt{\log n}}} + O\left(\frac{1}{\log n}\right) \\ &= \frac{1}{\sqrt{2\pi}\sigma_\varepsilon h_\varepsilon \sqrt{\log n}}\, e^{-\frac{x^2}{2}} + O\left(\frac{1}{\log n}\right), \end{aligned}$$

which is the expression (12) claimed in Theorem 1.

In our second integral (27) we also can use a decomposition of $g_n(z)$ (31), revealing its saddle point at $z = s_0$. However, since for large n: $s_0 \to 0$, we now also have to take into account a pole of $\Gamma(z)$ at $z=0$. Using its Laurent series:

$$\begin{aligned}\Gamma(z) &= \frac{1}{z} - \gamma + \left(\frac{\pi^2}{12} + \frac{\gamma^2}{2}\right) z + O\left(z^2\right) \\ &= \frac{s_0}{s_0^2+t^2} - \gamma + O\left(s_0\right) + O\left(t^2\right) \\ &\quad + i\left\{-\frac{t}{s_0^2+t^2} + O\left(t\right)\right\},\end{aligned}$$

we can show that

$$\begin{aligned}I_\varepsilon(n) &= \frac{1}{2\pi} e^{-\alpha} \int_{-\infty}^{\infty} \Gamma\left(s_0+it\right) e^{-\beta t^2} \times \\ &\qquad \times \left(1 + O\left(\log n \, (s_0+it)^4\right)\right) dt \\ &= \frac{1}{2\pi} e^{-\alpha} \int_{-\infty}^{\infty} e^{-\beta t^2} \frac{s_0}{s_0^2+t^2} dt \qquad (37) \\ &\quad - \gamma \frac{1}{\sqrt{2\pi}\sigma_\varepsilon h_\varepsilon \sqrt{\log n}} e^{-\frac{x^2}{2}} + O\left(\frac{1}{\log n}\right), \qquad (38)\end{aligned}$$

where the remaining integral (37) is due to a principal part of the Laurent series of $\Gamma(z)$. The other terms lead to a combination of the standard Gauss integrals, converging to (38).

To evaluate the remaining integral (37) we use a transformation $u=t^2$, which after some algebra yields:

$$\begin{aligned}&\frac{1}{2\pi}\int_{-\infty}^{\infty} e^{-\beta t^2} \frac{1}{s_0^2+t^2} dt \\ &\quad = \frac{1}{2\pi}\int_0^{\infty} e^{-\beta u} \frac{1}{(s_0^2+u)\sqrt{u}} du \qquad (39) \\ &\quad = \frac{1}{2\,s_0} e^{\beta s_0^2} \mathrm{Erfc}\left(s_0\sqrt{\beta}\right), \qquad (40)\end{aligned}$$

where

$$\mathrm{Erfc}(x) = \frac{2}{\sqrt{\pi}}\int_x^{\infty} e^{-t^2} dt \ ,$$

is a complementary error function [1].

Now, by noticing that

$$-\alpha + \beta s_0^2 = 0,$$

$$s_0\sqrt{\beta} = -\frac{x}{\sqrt{2}}\left(1 - x\frac{\sigma_\varepsilon h_\varepsilon}{2\sqrt{\log n}} + O\left(\frac{1}{\log n}\right)\right),$$

and

$$\frac{1}{2}\mathrm{Erfc}\left(-\frac{y}{\sqrt{2}}\right) = 1 - \Phi(-y) = \Phi(y),$$

and putting everything together, we finally arrive at

$$I_\varepsilon(n) = \Phi\left(x - x^2\frac{\sigma_\varepsilon h_\varepsilon}{2\sqrt{\log n}}\right) - \gamma\frac{1}{\sqrt{2\pi}\sigma_\varepsilon h_\varepsilon\sqrt{\log n}} e^{-\frac{x^2}{2}} + O\left(\frac{1}{\log n}\right),$$

which is the asymptotic expression (16) claimed by Theorem 2.

References

[1] Abramowitz, M. and Stegun, I. (1972) *Handbook of Mathematical Functions.* Dover.

[2] Andersson, A. and Nilsson, S. (1993) Improved Behaviour of Tries by Adaptive Branching. *Information Processing Letters* **46** 295–300.

[3] Andersson, A. and Nilsson, S. Faster Searching in Tries and Quadtries – An Analysis of Level Compression. *Proc. 2nd Annual European Symp. on Algorithms* 82–93.

[4] de Bruijn, N. G. (1981) *Asymptotic Methods in Analysis.* Dover.

[5] Coffman, E. G. Jr. and Eve, J. (1970) File Structures Using Hashing Functions, *Comm. ACM*, 13 (7) 427–436.

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

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

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

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

[10] Dobosiewitz, W. (1978) Sorting by Distributive Partitioning. *Information Processing Letters* **7** (1) 1–6.

[11] Ehrlich, G. (1981) Searching and Sorting Real Numbers. *J. Algorithms* **2** 1–14.

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

[13] Flajolet, P. and Sedgewick, R. (1995) Mellin Transforms and Asymptotics: Finite Differences and Rice's Integrals. *Theoretical Computer Science* **144** (101–124.

[14] Fredkin, E. (1960) Trie Memory. *Comm. ACM* **3** 490–500.

[15] Hofri, M. (1995) *Analysis of Algorithms: Computational Methods & Mathematical Tools.* Oxford University Press.

[16] Jacquet, P. and Régnier, M. (1986) Trie Partitioning Process: Limiting Distributions. *Lecture Notes in Computer Science* **214** 196–210. Springer-Verlag.

[17] Jacquet, P. and Szpankowski, W. (1991) Analysis of Digital Trees with Markovian Dependency, *IEEE Trans. Information Theory* **37** 1470–1475.

[18] Kirschenhofer, P. and Prodinger, H. (1986) Some further results on digital search trees. *Lecture Notes in Computer Science* **229** 177–185. Springer-Verlag.

[19] Knuth, D. E. (1973) *The Art of Computer Programming. Sorting and Searching. Vol. 3.* Addison-Wesley. 5259-271.

[20] Mahmoud, H., Flajolet, P., Jacquet, P. and Régnier, M. (2000) Analytic Variations on Bucket Selection and Sorting, *Acta Informatica* 36 735–760.

[21] Morrison, D. A. (1968) PATRICIA – Practical Algorithm To Retrieve Information Coded in Alphanumeric, *J. ACM*, 15 (4) 514–534.

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

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

[24] Pittel, B. (1986) Paths in a Random Digital Tree: Limiting Distributions, *Advances in Applied Probability* **18** 139–155.

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

Also in: (2000) *Lecture Notes in Computer Science* **1858** 148–158. Springer-Verlag.

[26] Sedgewick, R. and Flajolet, P. (1996) *An Introduction to the Analysis of Algorithms.* Addison-Wesley.

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

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

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

[30] Tamminen, M. (1983) Analysis of N-Trees, *Information Processing Letters* **16** (3) 131–137.

[31] Van der Waerden, B. (1951) On the Method of Saddle Points, *Applied Scientific Research* **B2** 33–45.