On the Smoothed Heights of Trie and Patricia Index Trees

Weitian Tong, Randy Goebel, and Guohui Lin*

Department of Computing Science, University of Alberta
Edmonton, Alberta T6G 2E8, Canada
{weitian,rgoebel,guohui}@ualberta.ca

Abstract. Two of the most popular data structures for storing strings are the Trie and the Patricia index trees. Let H_n denote the height of the Trie (the Patricia, respectively) on a set of n strings. It is well known that under the uniform distribution model on the strings, for Trie $H_n/\log n \to 2$ and for Patricia $H_n/\log n \to 1$, when n approaches infinity. Nevertheless, in the worst case, the height of the Trie on n strings is unbounded, and the height of the Patricia on n strings is in $\Theta(n)$. To better understand the practical performance of both the Trie and Patricia index trees, we investigate these two classical data structures in a smoothed analysis model. Given a set $\mathcal{S} = \{s_1, s_2, \ldots, s_n\}$ of n binary strings, we perturb the set by adding an $i.i.d$ Bernoulli random noise to each bit of every string. We show that the resulting smoothed heights of Trie and Patricia trees are both $\Theta(\log n)$.

1 Introduction

A *Trie*, also known as a *digital tree*, is an ordered tree data structure for storing strings over an alphabet Σ. It was initially developed and analyzed by Fredkin [6] in 1960, and is one of the first collected in "The art of computer programming" by Knuth [7] in 1973. Such a data structure is used for storing a dynamic set to be exploited as an associative array, where keys are strings. There has been much recent exploitation of such index trees for processing genomic data.

In the simplest form, let the alphabet be $\Sigma = \{0, 1\}$ and consider a set $\mathcal{S} = \{s_1, s_2, \ldots, s_n\}$ of n binary strings over Σ, where each s_i is a countable string of 0's and 1's. The Trie for storing these n binary strings is an ordered binary tree $T_{\mathcal{S}}$: first, each s_i defines a path (infinite if its length $|s_i|$ is infinite) in the tree, starting from the root, such that a 0 forces a move to the left and a 1 indicates a move to the right; if one node is the highest in the tree that is passed through by only one string $s_i \in \mathcal{S}$, then the path defined by s_i is truncated at this node, which becomes a leaf in the tree and is associated (i.e., labelled) with s_i. The *height* of the Trie $T_{\mathcal{S}}$ built over $\mathcal{S}$ is defined as the number of edges on the longest root-to-leaf path. Fig. 1 shows the Trie constructed for a set of six strings. The strings can be long or even infinite, but only the first 5 bits are shown, which are those used in the example construction.

* Correspondence author.

Z. Cai et al. (Eds.): COCOON 2014, LNCS 8591, pp. 94–103, 2014.

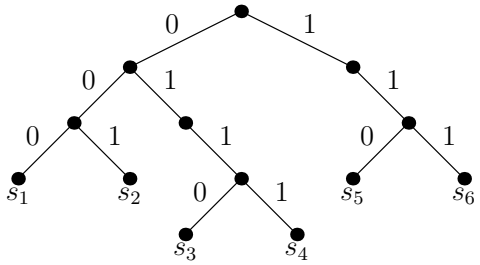

Fig. 1. The Trie constructed for $\{s_1 = 00001\ldots, s_2 = 00111\ldots, s_3 = 01100\ldots, s_4 = 01111\ldots, s_5 = 11010\ldots, s_6 = 11111\ldots\}$

Let H_n denote the height of the Trie on a set of n binary strings. It is not hard to see that in the worst case H_n is unbounded, because any two of the strings can have an arbitrary long common prefix. In the uniform distribution model, bits of s_i are *independent and identically distributed* (*i.i.d.*) Bernoulli random variables each of which takes 1 with probability $p = 0.5$. The asymptotic behavior of Trie height H_n under the uniform distribution model had been well studied in the 1980s [13,8,5,4,3,11,12,15,16], and it is known that *asymptotically almost surely* (*a.a.s.*)

$$H_n/\log_2 n \to 2, \text{ when } n \to \infty.$$

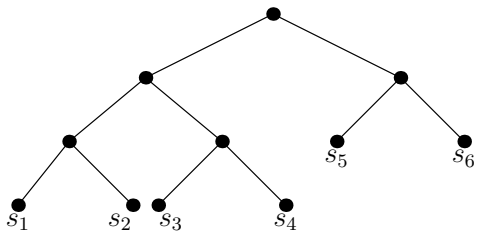

Fig. 2. The Patricia constructed for $\{s_1 = 00001\ldots, s_2 = 00111\ldots, s_3 = 01100\ldots, s_4 = 01111\ldots, s_5 = 11010\ldots, s_6 = 11111\ldots\}$

A Patricia index tree is a space-optimized variant of the Trie data structure, in which every node with only one child is merged with its child. Such a data structure was firstly discovered by Morrison [9] in 1968, and then well analyzed in "The art of computer programming" by Knuth [7] in 1973. Fig. 2 shows the Patricia tree constructed for the same set of six strings used in Fig. 1. Again let H_n denote the height of the Patricia tree on a set of n binary strings. In the worst case, $H_n \in \Theta(n)$. Under the same uniform distribution model assumed for an average case analysis on Trie height, Pittel showed that a.a.s. the height of Patricia is only 50% of the height of Trie [11], that is,

$$H_n / \log_2 n \to 1, \text{ when } n \to \infty.$$

The average case analysis is intended to provide insight on the practical performance as a string indexing structure. Recently, Nilsson and Tikkanen experimentally investigated the height of Patricia trees and other search structures [10]. In particular, they showed that the heights of the Patricia trees on sets of $50,000$ random uniformly distributed strings are 15.9 on average and 20 at most. For real datasets consisting of $19,461$ strings from geometric data on drill holes, $16,542$ ASCII character strings from a book, and $38,367$ strings from Internet routing tables, the heights of the Patricia trees are on average 20.8, 20.2, 18.6, respectively, and at most 30, 41, 24, respectively.

Theoretically speaking, these experimental results suggest that worst-case instances are perhaps only isolated peaks in the instance space. This hypothesis is partially supported by the average case analysis on the heights of Trie and Patricia structures, under the uniform distribution model, that suggests the heights are a.a.s. logarithmic. Nevertheless, these average case analysis results on the specific random instances generated under the uniform distribution model could be inconclusive, because the specific random instances have very special properties inherited from the model, and thus would distinguish themselves from real-world instances. To overcome the fact that real-world instances are not captured by a single probabilistic distribution, Spielman and Teng introduced the idea of *smoothed analysis* [14], which can be considered as a hybrid of the worst-case and the average-case analyses, and inherits the advantages of both. In brief, an given string instance is perturbed by adding a slight random noise to generate an instance neighborhood and the average performance on this neighbor is evaluated; the smoothed performance is then taken as the worst among all these local average performances. One can image that when the magnitude of random noise approaches 0, the smoothed analysis becomes the worst case analysis; when the magnitude of random noise approaches infinity, the smoothed analysis becomes the average case analysis under the probabilistic distribution assumed on the random noise. In practice, such a magnitude is set to be small; then a good smoothed analysis result under certain reasonable probabilistic distribution assumed on the random noise generally implies a good practical performance in real world applications. One key reason underlying this hypothesis is that real world instances are often subject to a slight amount of noise, especially when they are obtained from measurements of real world phenomena. The classic example is the Simplex method for solving linear programming. The Simplex method is one kind of practical algorithm for solving linear programming, all of which have worst case exponential running time. Spielman and Teng showed that Simplex algorithms have polynomial smoothed running time [14], which explained their practical performance.

In this paper, we conduct the smoothed analysis on the height of Trie and Patricia structures, to reveal certain essential properties of these two data structures. In the next section, we first introduce the string perturbation model, and we show an a.a.s. upper bound $O(\log n)$ and an a.a.s. lower bound $\Omega(\log n)$ on

the Trie height H_n. The consequence is that the smoothed height of the Trie on n strings is in $\Theta(\log n)$. In Section 3, we achieve similar results for the smoothed height of the Patricia tree on n strings.

2 The Smoothed Height of Trie

We consider an arbitrary set $\mathcal{S} = \{s_1, s_2, \ldots, s_n\}$ of n strings over alphabet $\{0, 1\}$, where each string may be infinitely long. Let $s_i(\ell)$ denote the ℓ-th bit in string s_i, for $i = 1, 2, \ldots, n$ and $\ell = 1, 2, 3, \ldots$. Every string s_i is perturbed by adding a noise string ν_i, giving rise to the perturbed string $\tilde{s}_i = s_i + \nu_i$, where $\tilde{s}_i(\ell) = s_i(\ell)$ if and only if $\nu_i(\ell) = 0$. The noise string ν_i is independently generated by a memoryless source, which assigns 1 to every bit of string ν_i independently and with a small probability $\epsilon \in [0, 0.5]$. More formally, $Pr\{\nu_i(\ell) = 1\} = \epsilon$ for each $\ell = 1, 2, 3, \ldots$. Essentially the perturbation flips each bit of every string independently and with a probability ϵ. Let $\tilde{\mathcal{S}} = \{\tilde{s}_1, \tilde{s}_2, \ldots, \tilde{s}_n\}$ denote the set of perturbed strings.

Let p_{ij}^{ℓ} be the probability of the event $\{\tilde{s}_i(\ell) = \tilde{s}_j(\ell)\}$. We have

$$p_{ij}^{\ell} = \begin{cases} 2\epsilon(1-\epsilon) \triangleq p, & \text{if } s_i(\ell) \neq s_j(\ell), \\ \epsilon^2 + (1-\epsilon)^2 = 1 - p \triangleq q, & \text{if } s_i(\ell) = s_j(\ell). \end{cases} \tag{1}$$

We can clearly note that $q \geq p$, since $\epsilon \leq 0.5$. Let C_{ij} denote the length of the longest common prefix between $\tilde{s}_i$ and $\tilde{s}_j$. Since $C_{ij} = k$ if and only if $\tilde{s}_i(\ell) = \tilde{s}_j(\ell)$ for $\ell = 1, 2, \ldots, k$ but not for $\ell = k+1$, the probability of $\{C_{ij} = k\}$ for any $k \geq 0$ is

$$Pr\{C_{ij} = k\} = \left(\prod_{\ell=1}^{k} p_{ij}^{\ell}\right)(1 - p_{ij}^{k+1}).$$

From the fact that $\{C_{ij} = k\}$ and $\{C_{ij} = m\}$ are disjoint events when $k \neq m$, we have for any $k \geq 1$

$$Pr\{C_{ij} < k\} = \sum_{m=0}^{k-1} \left(\prod_{\ell=1}^{m} p_{ij}^{\ell} - \prod_{\ell=1}^{m+1} p_{ij}^{\ell}\right) = 1 - \prod_{\ell=1}^{k} p_{ij}^{\ell}.$$

Consequently, the probability that the longest common prefix between $\tilde{s}_i$ and $\tilde{s}_j$ is at least k long is

$$Pr\{C_{ij} \geq k\} = 1 - Pr\{C_{ij} < k\} = \prod_{\ell=1}^{k} p_{ij}^{\ell}. \tag{2}$$

2.1 An a.a.s. Upper Bound

We use a slight abuse of notation H_n to also denote the height of the Trie constructed for $\tilde{\mathcal{S}}$. We can express H_n in terms of C_{ij} as

$$H_n = \max_{1 \leq i < j \leq n} C_{ij} + 1.$$

By Boole inequality [2], we have

$$Pr\{H_n > k\} = Pr\{\max_{1\le i<j\le n} C_{ij} \ge k\} \le \binom{n}{2}\prod_{\ell=1}^{k} p_{ij}^{\ell} \le \binom{n}{2} q^k,$$

where the last equality holds when all the n strings $\{s_1, s_2, \ldots, s_n\}$ have the same prefix of length k. By setting $k = 2(1+\delta)\log_{1/q} n$ for a constant $\delta > 0$, we have

$$Pr\{H_n > k\} \le \binom{n}{2} q^{2(1+\delta)\log_{1/q} n} \le n^{-2\delta} \to 0,$$

as $n \to \infty$. Therefore, $H_n \le 2\log_{1/q} n$ with high probability, when n approaches infinity.

2.2 An a.a.s. Lower Bound

To estimate a lower bound, we will use the following Chunge-Erdös formulation of the second moment method on a set of events:

Lemma 1. (Chunge-Erdös) [1] *For any set of events $E_1, E_2, \ldots, E_n$,*

$$Pr\{\cup_{i=1}^{n} E_i\} \ge \frac{\left(\sum_{i=1}^{n} Pr\{E_i\}\right)^2}{\sum_{i=1}^{n} Pr\{E_i\} + \sum_{i\ne j} Pr\{E_i \cap E_j\}}.$$

Let A_{ij} denote the event $\{C_{ij} \ge k\}$, for every pair $\{i, j\}$ such that $1 \le i < j \le n$; also define the following two sums:

$$S_1 \triangleq \sum_{1\le i<j\le n} Pr\{A_{ij}\}, \text{ and } S_2 \triangleq \sum_{\{i,j\}\ne\{s,t\}} Pr\{A_{ij} \cap A_{st}\}.$$

Then by Chunge-Erdös formulation (Lemma 1), we have

$$Pr\{H_n > k\} = Pr\{\cup_{1\le i<j\le n} A_{ij}\} \ge \frac{S_1^2}{S_1 + S_2}. \tag{3}$$

Let's first estimate S_1. From Eq. (2), one clearly sees that

$$S_1 = \sum_{1\le i<j\le n} Pr\{A_{ij}\} = \sum_{1\le i<j\le n}\prod_{\ell=1}^{k} p_{ij}^{\ell}. \tag{4}$$

Recall the definition of p_{ij}^{ℓ} and its value in Eq. (1). The following Lemma 2 is then straight-forward:

Lemma 2. *For any $\ell \ge 1$ and any three perturbed strings $\tilde{s}_i, \tilde{s}_j, \tilde{s}_t$, if $p_{ij}^{\ell} = p_{it}^{\ell}$, then $p_{jt}^{\ell} = q$.*

Lemma 3. *For any three perturbed strings $\tilde{s}_i, \tilde{s}_j, \tilde{s}_t$,*

$$S_0 \triangleq \prod_{\ell=1}^{k} p_{ij}^{\ell} + \prod_{\ell=1}^{k} p_{it}^{\ell} + \prod_{\ell=1}^{k} p_{jt}^{\ell} \geq 3p^{\frac{2}{3}k} q^{\frac{1}{3}k}.$$

Proof. For the string pair (s_i, s_j), let Z_{ij} denote the number of $(0,1)$-pairs and $(1,0)$-pairs in $\{(s_i(\ell), s_j(\ell)), 1 \leq \ell \leq k\}$, that is, the number of bits where s_i and s_j have different values among the first k bits. Clearly from Eq. (1),

$$\prod_{\ell=1}^{k} p_{ij}^{\ell} = p^{Z_{ij}} q^{k-Z_{ij}}.$$

For the string triple (s_i, s_j, s_t), let x_{ij} denote the number of $(0,0,1)$-triples and $(1,1,0)$-triples in $\{(s_i(\ell), s_j(\ell), s_t(\ell)), 1 \leq \ell \leq k\}$; likewise, x_{it} and x_{jt} are similarly defined. Also let y denote the number of $(0,0,0)$-triples and $(1,1,1)$-triples in $\{(s_i(\ell), s_j(\ell), s_t(\ell)), 1 \leq \ell \leq k\}$. The following relationships are direct consequences of the definitions:

$$\begin{aligned} Z_{ij} &= x_{it} + x_{jt}, \\ Z_{it} &= x_{ij} + x_{jt}, \\ Z_{jt} &= x_{ij} + x_{it}, \\ k &= x_{ij} + x_{it} + x_{jt} + y. \end{aligned}$$

It follows that

$$\begin{aligned} S_0 &\triangleq \prod_{\ell=1}^{k} p_{ij}^{\ell} + \prod_{\ell=1}^{k} p_{it}^{\ell} + \prod_{\ell=1}^{k} p_{jt}^{\ell} \\ &= p^{x_{it}+x_{jt}} q^{x_{ij}+y} + p^{x_{ij}+x_{jt}} q^{x_{it}+y} + p^{x_{ij}+x_{it}} q^{x_{jt}+y} \\ &= p^k \left[\left(\frac{q}{p}\right)^{x_{ij}+y} + \left(\frac{q}{p}\right)^{x_{it}+y} + \left(\frac{q}{p}\right)^{x_{jt}+y} \right]. \end{aligned}$$

One can check that, since $q \geq p$, the quantity in the last line reaches the minimum when $x_{ij} = x_{it} = x_{jt} = k/3$ and $y = 0$. That is,

$$S_0 \triangleq \prod_{\ell=1}^{k} p_{ij}^{\ell} + \prod_{\ell=1}^{k} p_{it}^{\ell} + \prod_{\ell=1}^{k} p_{jt}^{\ell} \geq 3p^{\frac{2}{3}k} q^{\frac{1}{3}k}.$$

This proves the lemma. □

Note that each string pair (s_i, s_j) is involved in exactly $n-2$ string triples (s_i, s_j, s_t), for $t \neq i, j$. By Lemma 3, Eq. (4) becomes

$$S_1 = \sum_{1 \leq i < j \leq n} \prod_{\ell=1}^{k} p_{ij}^{\ell} \geq \frac{1}{n-2} \binom{n}{3} 3p^{\frac{2}{3}k} q^{\frac{1}{3}k} = \binom{n}{2} p^{\frac{2}{3}k} q^{\frac{1}{3}k}. \tag{5}$$

We next estimate S_2, which is a bit harder because two events A_{ij} and A_{st} may not be independent. We split S_2 into two parts: $S_2 = S_2' + S_2''$, where

$$S_2' \triangleq \sum_{\{i,j\}\cap\{s,t\}=\emptyset} Pr\{A_{ij} \cap A_{st}\}, \text{ and}$$

$$S_2'' \triangleq \sum_{\{i,j\}\cap\{s,t\}\neq\emptyset} Pr\{A_{ij} \cap A_{st}\}.$$

Since two events C_{ij} and C_{st} are independent when $\{i,j\} \cap \{s,t\} = \emptyset$, we can estimate S_2' as follows:

$$S_2' = \sum_{\{i,j\}\cap\{s,t\}=\emptyset} \Big(Pr\{A_{ij}\}Pr\{A_{st}\}\Big) \leq \left(\sum_{\{i,j\}} Pr\{A_{ij}\}\right)^2 = S_1^2.$$

Event $\{A_{ij} \cap A_{it}\}$ is equivalent to the event in which the first k bits of all three perturbed strings $\tilde{s}_i, \tilde{s}_j$, and $\tilde{s}_t$ are identical. Using $\epsilon \leq 0.5$, we have

$$Pr\{A_{ij} \cap A_{it}\} = Pr\{\tilde{s}_i(\ell) = \tilde{s}_j(\ell) = \tilde{s}_t(\ell), 1 \leq \ell \leq k\} \leq \Big(\epsilon^3 + (1-\epsilon)^3\Big)^k.$$

It follows that

$$S_2'' = \sum_{\{i,j\}\cap\{s,t\}\neq\emptyset} Pr\{A_{ij} \cap A_{st}\} \leq 3\binom{n}{3}\Big(\epsilon^3 + (1-\epsilon)^3\Big)^k \leq 3\binom{n}{3},$$

where the factor 3 arises because a string triple $\{\tilde{s}_i, \tilde{s}_j, \tilde{s}_t\}$ gives rise to three events $\{A_{ij} \cap A_{it}\}$, $\{A_{ij} \cap A_{jt}\}$, and $\{A_{it} \cap A_{jt}\}$.

Putting S_2' and S_2'' together, we can upper bound S_2 by

$$S_2 = S_2' + S_2'' \leq S_1^2 + 3\binom{n}{3}. \tag{6}$$

Using the estimates of S_1 and S_2 in Eqs. (5) and (6) respectively, Eq. (3) becomes

$$\begin{aligned}
Pr\{H_n > k\} &\geq \frac{S_1^2}{S_1 + S_2} \\
&= \frac{1}{1/S_1 + (S_2' + S_2'')/S_1^2} \\
&\geq \frac{1}{1/S_1 + 1 + S_2''/S_1^2} \\
&\geq \frac{1}{1 + \frac{1}{\binom{n}{2}p^{\frac{2}{3}k}q^{\frac{1}{3}k}} + \frac{3\binom{n}{3}}{\left(\binom{n}{2}p^{\frac{2}{3}k}q^{\frac{1}{3}k}\right)^2}} \\
&\geq \frac{1}{1 + 4n^{-2}p^{-\frac{2}{3}k}q^{-\frac{1}{3}k} + 2n^{-1}p^{-\frac{4}{3}k}q^{-\frac{2}{3}k}}
\end{aligned}$$

$$\begin{aligned} &\geq \frac{1}{1+4n^{-2}n^{2(1-\delta)}+2n^{-1}n^{1-\delta}} \qquad (7)\\ &= \frac{1}{1+4n^{-2\delta}+2n^{-\delta}}\\ &\geq 1-O(n^{-\delta}) \to 1, \end{aligned}$$

where the inequality Eq. (7) is achieved by setting

$$k = 2(1-\delta)\log_{p^{-2/3}q^{-1/3}} n, \text{ that is, } p^{-\frac{2}{3}k}q^{-\frac{1}{3}k} = n^{2(1-\delta)},$$

for a constant $\delta > 0$. Therefore, H_n is larger than $2\log_{p^{-2/3}q^{-1/3}} n$ with a high probability when n approaches infinity.

Theorem 1. *The smoothed height of the Trie on n strings is in $\Theta(\log n)$, where the bit perturbation model is i.i.d. Bernoulli distribution.*

3 The Smoothed Height of Patricia

Here we briefly do the smoothed analysis on the height of the Patricia tree on a set of n binary strings. We adopt the same *i.i.d.* Bernoulli bit perturbation model as in the last section. Again, we present an a.a.s. upper bound and an a.a.s. lower bound for the smoothed height.

3.1 An a.a.s. Upper Bound

Following Pittel [11], on the set of n perturbed strings $\tilde{\mathcal{S}} = \{\tilde{s}_1, \tilde{s}_2, \ldots, \tilde{s}_n\}$, we claim that for any fixed integers $k \geq 0$ and $b \geq 2$, the event $\{H_n \geq k+b-1\}$ implies the event that there exist b strings $\tilde{s}_{i_1}, \tilde{s}_{i_2}, \ldots, \tilde{s}_{i_b}$ such that their common prefix is of length at least k (denoted as $C_{i_1 i_2 \ldots i_b} \geq k$). The correctness of the above claim follows from because, in Patricia trees, there are no degree-2 nodes (except for the root), and thus a path of length $k+b-1$ hints at least b leaves in the subtree rooted at the node at distance k from the Patricia root.

Similar to the definition of p^{ℓ}_{ij} in Eq. (1), $p^{\ell}_{i_1 i_2 \ldots i_b}$ denotes the probability of the event $\{\tilde{s}^{\ell}_{i_1} = \tilde{s}^{\ell}_{i_2} = \ldots = \tilde{s}^{\ell}_{i_b}\}$, for any $b \geq 2$, which is calculated as follows:

$$p^{\ell}_{i_1 i_2 \ldots i_b} = (1-\epsilon)^{k_0}\epsilon^{k_1} + (1-\epsilon)^{k_1}\epsilon^{k_0},$$

where k_0 and k_1 are the number of 0's and 1's among the b bit values $\tilde{s}_{i_1}(\ell), \tilde{s}_{i_2}(\ell), \ldots, \tilde{s}_{i_b}(\ell)$, respectively. By a similar argument as presented for $Pr\{A_{ij}\}$ in Section 2, we have

$$Pr\{C_{i_1 i_2 \ldots i_b} \geq k\} = \prod_{\ell=1}^{k} p^{\ell}_{i_1 i_2 \ldots i_b}.$$

For a fixed $b \geq 2$, let $q_b = \epsilon^b + (1-\epsilon)^b$ and $k = k_b = b(1+\delta/2)\log_{1/q_b} n$. We have

$$k = b(1+\delta/2)\log_{1/q_b} n$$

$$
\begin{aligned}
&= (1+\delta/2)\frac{\ln n}{\ln q_b^{-1/b}} \\
&= (1+\delta/2)\frac{\ln n}{\ln\left(\epsilon^b + (1-\epsilon)^b\right)^{-1/b}} \\
&\le (1+\delta/2)\frac{\ln n}{\ln\left(\epsilon^2 + (1-\epsilon)^2\right)^{-1/2}} \qquad (8) \\
&= 2(1+\delta/2)\log_{1/q} n,
\end{aligned}
$$

where the inequality in Eq. (8) holds for any $b \ge 2$. Setting $b = \delta \log_{1/q} n$, it follows that

$$
\begin{aligned}
Pr\{H_n \ge 2(1+\delta)\log_{1/q} n\} &\le Pr\{H_n \ge k+b-1\} \\
&\le Pr\{\max_{i_1,i_2,\ldots,i_b} C_{i_1 i_2 \ldots i_b} \ge k\} \\
&\le n^b \prod_{\ell=1}^{k} p^{\ell}_{i_1 i_2 \ldots i_b} \\
&\le n^b q_b^k \\
&\in O(n^{-b\delta}) \to 0,
\end{aligned}
$$

when $n \to \infty$.

In summary, for any $\delta > 0$, we have

$$
Pr\{H_n \ge 2(1+\delta)\log_{1/q} n\} \in O(n^{-b\delta}) \to 0,
$$

when n approaches infinity, and thus a.a.s. $H_n \le 2(1+\delta)\log_{1/q} n$.

3.2 An a.a.s. Lower Bound

Let D_i be the depth of node labelled $\tilde{s}_i$ in the Patricia tree.

Clearly, $H_n = \max_{i=1}^{n} D_i$ and the $\tilde{s}_{i^*}$ reaching the maximum depth must be a leaf node. It follows that if $H_n < k$, then at least one of the 2^k possible length-k strings does not appear as a prefix of any perturbed strings $\tilde{s}_1, \tilde{s}_2, \ldots, \tilde{s}_n$.

Let $\mathbb{L}n = \log_{1/\epsilon} n$ and $k = \mathbb{L}\frac{n}{\mathbb{L}\ln n}$. We have

$$
\begin{aligned}
Pr\{H_n < k\} &\le 2^k Pr\{\text{no } \tilde{s}_i \text{ starts with } k \text{ 0's}\} \\
&\le 2^k (1-\epsilon^k)^n \\
&\le 2^k e^{-\epsilon^k n} \\
&= \exp\{k\ln 2 - \epsilon^k n\} \\
&= \exp\{\ln 2 \cdot \mathbb{L}\frac{n}{\mathbb{L}\ln n} - \mathbb{L}\ln n\} \to 0,
\end{aligned}
$$

when n approaches infinity, and thus a.a.s. $H_n \ge \mathbb{L}\frac{n}{\mathbb{L}\ln n}$.

In summary, we have the following theorem.

Theorem 2. *The smoothed height of the Patricia on n strings is in $\Theta(\log n)$, where the bit perturbation model is i.i.d. Bernoulli distribution.*

4 Conclusion

Under the *i.i.d.* Bernoulli bit perturbation model, we have shown that the smoothed heights of both Trie and Patricia index trees on n strings are in the order of $\log n$. These theoretical results explain the typical probabilistic behavior of these two important data structures on real-world applications.

Acknowledgement. This research was supported in part by NSERC, AITF and iCORE.

References

1. Chung, K.L., Erdös, P.: On the application of the Borel-Cantelli lemma. Transactions of the American Mathematical Society 72, 179–186 (1952)
2. Comtet, L.: Advanced Combinatorics: The Art of Finite and Infinite Expansions. Springer (1974)
3. Devroye, L.: A probabilistic analysis of the height of tries and of the complexity of triesort. Acta Informatica 21, 229–237 (1984)
4. Flajolet, P.: On the performance evaluation of extendible hashing and trie search. Acta Informatica 20, 345–369 (1983)
5. Flajolet, P., Steyaert, J.M.: A branching process arising in dynamic hashing, trie searching and polynomial factorization. In: Nielsen, M., Schmidt, E.M. (eds.) ICALP 1982. LNCS, vol. 140, pp. 239–251. Springer, Heidelberg (1982)
6. Fredkin, E.: Trie memory. Communications of the ACM 3, 490–499 (1960)
7. Knuth, D.E.: The Art of Computer Programming. Sorting and Searching, vol. III. Addison-Wesley (1973)
8. Mendelson, H.: Analysis of extendible hashing. IEEE Transactions on Software Engineering 8, 611–619 (1982)
9. Morrison, D.R.: Patricia — practical algorithm to retrieve information coded in alphanumeric. Journal of the ACM 15, 514–534 (1968)
10. Nilsson, S., Tikkanen, M.: An experimental study of compression methods for dynamic tries. Algorithmica 33, 19–33 (2002)
11. Pittel, B.: Asymptotical growth of a class of random trees. Annals of Probability 13, 414–427 (1985)
12. Pittel, B.: Path in a random digital tree: limiting distributions. Advances in Applied Probability 18, 139–155 (1986)
13. Régnier, M.: On the average height of trees in digital searching and dynamic hashing. Information Processing Letters 13, 64–66 (1981)
14. Spielman, D.A., Teng, S.-H.: Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time. Journal of the ACM 51, 385–463 (2004)
15. Szpankowski, W.: Some results on V-ary asymmetric tries. Journal of Algorithms 9, 224–244 (1988)
16. Szpankowski, W.: Digital data structures and order statistics. In: Dehne, F., Santoro, N., Sack, J.-R. (eds.) WADS 1989. LNCS, vol. 382, pp. 206–217. Springer, Heidelberg (1989)