Theoretical Computer Science 609 (2016) 620–626

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Smoothed heights of tries and patricia tries

Weitian Tong [a], Randy Goebel [a], Guohui Lin [a,b,*]

[a] Department of Computing Science, University of Alberta, Edmonton, Alberta T6G 2E8, Canada
[b] Department of Mathematics, Zhejiang Sci-Tech University, Hangzhou, Zhejiang 310018, China

ARTICLE INFO

Article history:
Received 22 September 2014
Received in revised form 15 January 2015
Accepted 8 February 2015
Available online 12 February 2015

Keywords:
Smoothed analysis
Data structure
Trie
Patricia trie

ABSTRACT

Tries and patricia tries are two popular data structures for storing strings. Let H_n denote the height of the trie (the patricia trie, respectively) on a set of n strings. Under the uniform distribution model on the strings, it is well known that $H_n/\log n \to 2$ for tries and $H_n/\log n \to 1$ for patricia tries, when n approaches infinity. Nevertheless, in the worst case, the height of a trie can be unbounded and the height of a patricia trie is in $\Theta(n)$. To better understand the practical performance of both tries and patricia tries, 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 the trie and the patricia trie are both in $\Theta(\log n)$.

1. Introduction

A *trie*, also known as a *digital tree* or a *prefix 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 Knuth [7] in 1973. Such a data structure has been 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 can be infinitely long. 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.*, labeled) 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. (These strings can be long or even infinite, but only the first 5 bits are shown, which are those used in the example construction.)

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, due to the existence of two of the strings sharing 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 the trie height H_n under the uniform distribution model had been well studied in the 1980s [3–5,8,11–13,15,16], and it is known that *asymptotically almost surely* (*a.a.s.*)

* Corresponding author at: Department of Computing Science, University of Alberta, Edmonton, Alberta T6G 2E8, Canada. Tel.: +1 (780) 492 3737.
E-mail addresses: weitian@ualberta.ca (W. Tong), rgoebel@ualberta.ca (R. Goebel), guohui@ualberta.ca (G. Lin).

http://dx.doi.org/10.1016/j.tcs.2015.02.009
0304-3975/

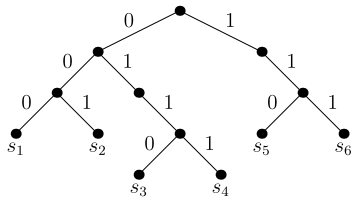

Fig. 1. The trie constructed for $\mathcal{S} = \{s_1 = 00001\ldots, s_2 = 00111\ldots, s_3 = 01100\ldots, s_4 = 01111\ldots, s_5 = 11010\ldots, s_6 = 11111\ldots\}$.

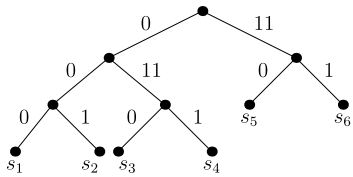

Fig. 2. The patricia trie constructed for $\mathcal{S} = \{s_1 = 00001\ldots, s_2 = 00111\ldots, s_3 = 01100\ldots, s_4 = 01111\ldots, s_5 = 11010\ldots, s_6 = 11111\ldots\}$.

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

A patricia trie, or a *compact trie*, 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 proposed 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 trie constructed for the same set of six strings used in Fig. 1. Again let H_n denote the height of the patricia trie on a set of n binary strings. In the worst case, $H_n = n - 1$, where s_i is in the form $11\ldots100\ldots$ with a prefix consisting of $i - 1$ consecutive 1's. Under the same uniform distribution model assumed for an average case analysis on the trie height, Pittel showed that *a.a.s.* the heights of patricia tries are only 50% of the heights of tries [11], that is,

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

The average case analysis is intended to provide insights on the algorithm's practical performance as a string indexing structure. In 2002, Nilsson and Tikkanen [10] experimentally investigated the heights of patricia tries and other search structures. In particular, they showed that the heights of the patricia tries 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 tries 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 tries and patricia tries, under the uniform distribution model, that suggests the heights are *a.a.s.* logarithmic. Nevertheless, these average case analyses 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. Because real-world instances are not captured by a single probabilistic distribution, Spielman and Teng [14] introduced the idea of *smoothed analysis*, which can be considered as a hybrid of the worst-case and the average-case analyses, and inherits the advantages of both. Given an instance that is a set of strings, we generate the instance *neighborhood* through perturbation, by adding a slight random noise to each bit in every string of the given instance; we then evaluate the average height on this neighborhood of perturbed instances, and this *local average height* is associated with the given instance. The smoothed height is defined as the worst (largest) among all the local average heights, over all instances. One can imagine 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 for the random noise. In practice, such a magnitude is set to be small; a good smoothed analysis result under certain reasonable probabilistic distribution assumed for the random noise implies good practical performance in real world applications. One key reason underlying this hypothesis is that real world instances are often subject to some amount of noise, especially when they are obtained from measurements of real world phenomena. The classic example is the Simplex method combined with shadow pivoting rule for solving linear programming. Though it needs exponential running time to terminate in the worst case, it is good in practise, and even outperforms many other polynomial time algorithms for linear programming in the real applications. Spielman and Teng [14] showed that the Simplex method with the shadow pivoting rule has polynomial smoothed running time, which well-explained its practical performance.

Here we conduct a smoothed analysis on the heights of tries and patricia tries, to reveal certain essential properties of these two data structures. In the next section, we first introduce the string perturbation model, and 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 conclusion 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 trie on a set

of n strings, that is, $H_n \in \Theta(\log n)$, which explains the practical performance of patricia tries in the experiments conducted by Nilsson and Tikkanen [10].

2. The smoothed heights of tries

We consider an arbitrary set $\mathcal{S} = \{s_1, s_2, \ldots, s_n\}$ of n strings over the alphabet $\{0, 1\}$, where each string may be infinitely long. Let $s_i(\ell)$ denote the ℓ-th bit in the 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 $\tilde{s}_i = s_i \oplus \nu_i$, where $\oplus$ is the bitwise XOR operation, that is $\tilde{s}_i(\ell) = s_i(\ell)$ 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,

$$Pr\{\nu_i(\ell) = 1\} = \epsilon \text{ for each } \ell = 1, 2, 3, \ldots.$$

One clearly sees that the perturbation essentially 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

$$p_{ij}^{\ell} = \begin{cases} 2\epsilon(1-\epsilon) \triangleq p, & \text{if } s_i(\ell) \neq s_j(\ell), \\ \epsilon^2 + (1-\epsilon)^2 = 1 - p \triangleq q, & \text{if } s_i(\ell) = s_j(\ell). \end{cases} \tag{2.1}$$

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 $\tilde{s}_j$ has length at least k is

$$Pr\{C_{ij} \geq k\} = 1 - Pr\{C_{ij} < k\} = \prod_{\ell=1}^{k} p_{ij}^{\ell}. \tag{2.2}$$

2.1. An a.a.s. upper bound

In a slight abuse of notation, we use 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

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

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

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

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

2.2. An a.a.s. lower bound

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

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

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

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

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

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

$$Pr\{H_n > k\} = Pr\{\cup_{1\leq i<j\leq n} A_{ij}\} \geq \frac{S_1^2}{S_1 + S_2}. \tag{2.3}$$

We first derive an estimate for S_1. From Eq. (2.2), one clearly sees that

$$S_1 = \sum_{1\leq i<j\leq n} Pr\{A_{ij}\} = \sum_{1\leq i<j\leq n} \prod_{\ell=1}^{k} p_{ij}^{\ell}. \tag{2.4}$$

Recall the definition of p_{ij}^{ℓ} and its value in Eq. (2.1). The following Lemma 2 is then straight-forward:

Lemma 2. *For any $\ell \geq 1$ 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. (2.1),

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

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 $\{(s_i(\ell), s_j(\ell), s_t(\ell)), 1 \leq \ell \leq k\}$; 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:

$$Z_{ij} = x_{it} + x_{jt},$$
$$Z_{it} = x_{ij} + x_{jt},$$
$$Z_{jt} = x_{ij} + x_{it},$$
$$k = x_{ij} + x_{it} + x_{jt} + y.$$

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. (2.4) becomes

$$\begin{aligned} S_1 &= \sum_{1 \leq i < j \leq n} \prod_{\ell=1}^{k} p_{ij}^{\ell} \\ &\geq \frac{1}{n-2} \binom{n}{3} 3p^{\frac{2}{3}k} q^{\frac{1}{3}k} \\ &= \binom{n}{2} p^{\frac{2}{3}k} q^{\frac{1}{3}k}. \end{aligned} \tag{2.5}$$

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:

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

The 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

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

It follows that

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

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

$$S_2 = S_2' + S_2'' \leq S_1^2 + 3\binom{n}{3}. \tag{2.6}$$

Using the estimates of S_1 and S_2 in Eqs. (2.5) and (2.6) respectively, Eq. (2.3) becomes

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

$$\begin{aligned}
&\geq \frac{1}{1+4n^{-2}p^{-\frac{2}{3}k}q^{-\frac{1}{3}k}+2n^{-1}p^{-\frac{4}{3}k}q^{-\frac{2}{3}k}} \\
&\geq \frac{1}{1+4n^{-2}n^{\frac{1}{2}(1-\delta)}+2n^{-1}n^{1-\delta}} \\
&= \frac{1}{1+4n^{-\frac{3}{2}-\frac{1}{2}\delta}+2n^{-\delta}} \\
&\geq 1-O(n^{-\delta}) \to 1,
\end{aligned} \tag{2.7}$$

where the inequality Eq. (2.7) is achieved by setting

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

for a constant $\delta > 0$. Therefore, H_n is larger than $2\log_{p^{-2/3}q^{-1/3}} n$ with 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 heights of patricia tries

Here we briefly do the smoothed analysis on the height of the patricia trie on a set of n binary strings, since much of the detail is similar to that for tries. 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 heights.

3.1. An a.a.s. upper bound

Following the work by 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 has length at least k (denoted as $C_{i_1 i_2 \ldots i_b} \geq k$). The correctness of the above claim follows because, in a patricia trie, there are no degree-2 nodes (except for the root), and thus a path of length $k+b-1$ suggests at least b leaves in the subtree rooted at the node at distance k from the patricia trie root.

Similar to the definition of p^{ℓ}_{ij} in Eq. (2.1), $p^{\ell}_{i_1 i_2 \ldots i_b}$ denotes the probability of the event $\{\tilde{s}_{i_1}(\ell) = \tilde{s}_{i_2}(\ell) = \ldots = \tilde{s}_{i_b}(\ell)\}$, 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 numbers of 0's and 1's among the b bit values $s_{i_1}(\ell), s_{i_2}(\ell), \ldots, s_{i_b}(\ell)$, respectively. By a similar argument as presented for $Pr\{C_{ij} \geq k\}$ in Eq. (2.2) 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

$$\begin{aligned}
k &= b(1+\delta/2)\log_{1/q_b} n \\
&= (1+\delta/2)\frac{\ln n}{\ln q_b^{-1/b}} \\
&= (1+\delta/2)\frac{\ln n}{\ln\left(\epsilon^b + (1-\epsilon)^b\right)^{-1/b}} \\
&\leq (1+\delta/2)\frac{\ln n}{\ln\left(\epsilon^2 + (1-\epsilon)^2\right)^{-1/2}} \\
&= 2(1+\delta/2)\log_{1/q} n,
\end{aligned} \tag{3.1}$$

where the inequality in Eq. (3.1) holds for any $b \geq 2$. Setting $b = \delta \log_{1/q} n$, it follows that

$$\begin{aligned}
Pr\{H_n \geq 2(1+\delta)\log_{1/q} n\} &\leq Pr\{H_n \geq k+b-1\} \\
&\leq Pr\{\max_{i_1, i_2, \ldots, i_b} C_{i_1 i_2 \ldots i_b} \geq k\}
\end{aligned}$$

$$\begin{aligned}
&\le n^b \prod_{\ell=1}^{k} p^{\ell}_{i_1 i_2 \ldots i_b} \\
&\le n^b q_b^k \\
&\in O(n^{-b\delta}) \to 0,
\end{aligned}$$

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 the node labeled $\tilde{s}_i$ in the patricia trie.

Clearly, $H_n = \max_{i=1}^{n} D_i$ and the node at 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 trie on n strings is in* $\Theta(\log n)$, *where the bit perturbation model is i.i.d. Bernoulli distribution.*

4. Conclusions

Under the *i.i.d.* Bernoulli bit perturbation model, we have shown that the smoothed heights of both tries and patricia tries 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.

Acknowledgements

This research was supported in part by NSERC and AITF. G.L. was also supported by the Science Foundation of Zhejiang Sci-Tech University (ZSTU) Grant No. 14062170-Y. G.L.'s work was partially done during his visit to the ZSTU.

References

[1] K.L. Chung, P. Erdös, On the application of the Borel–Cantelli lemma, Trans. Amer. Math. Soc. 72 (1952) 179–186.
[2] L. Comtet, Advanced Combinatorics: The Art of Finite and Infinite Expansions, Springer, 1974.
[3] L. Devroye, A probabilistic analysis of the height of tries and of the complexity of triesort, Acta Inform. 21 (1984) 229–237.
[4] P. Flajolet, On the performance evaluation of extendible hashing and trie search, Acta Inform. 20 (1983) 345–369.
[5] P. Flajolet, J.M. Steyaert, A branching process arising in dynamic hashing, trie searching and polynomial factorization, in: Proceedings of the Ninth International Colloquium on Automata, Languages and Programming, ICALP, in: LNCS, vol. 140, 1982, pp. 239–251.
[6] E. Fredkin, Trie memory, Commun. ACM 3 (1960) 490–499.
[7] D.E. Knuth, The Art of Computer Programming, Volume III: Sorting and Searching, Addison-Wesley, 1973.
[8] H. Mendelson, Analysis of extendible hashing, IEEE Trans. Softw. Eng. 8 (1982) 611–619.
[9] D.R. Morrison, Patricia — practical algorithm to retrieve information coded in alphanumeric, J. ACM 15 (1968) 514–534.
[10] S. Nilsson, M. Tikkanen, An experimental study of compression methods for dynamic tries, Algorithmica 33 (2002) 19–33.
[11] B. Pittel, Asymptotical growth of a class of random trees, Ann. Probab. 13 (1985) 414–427.
[12] B. Pittel, Path in a random digital tree: limiting distributions, Adv. in Appl. Probab. 18 (1986) 139–155.
[13] M. Régnier, On the average height of trees in digital searching and dynamic hashing, Inform. Process. Lett. 13 (1981) 64–66.
[14] D.A. Spielman, S.-H. Teng, Smoothed analysis of algorithms: why the simplex algorithm usually takes polynomial time, J. ACM 51 (2004) 385–463.
[15] W. Szpankowski, Some results on V-ary asymmetric tries, J. Algorithms 9 (1988) 224–244.
[16] W. Szpankowski, Digital data structures and order statistics, in: Proceedings of the 1989 Workshop on Algorithms and Data Structures, WADS, in: LNCS, vol. 382, 1989, pp. 206–217.