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/

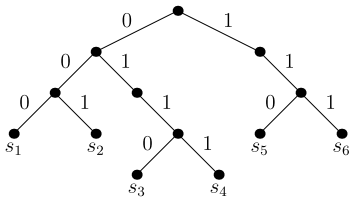

Fig. 1. The trie constructed for $\mathcal{S} = \{s_1 = 00001\ldots, s_2 = 00111\ldots, s_3 = 01100\ldots, s_4 = 01111\ldots, s_5 = 11010\ldots, s_6 = 11111\ldots\}$.

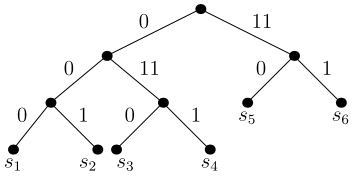

Fig. 2. The patricia trie constructed for $\mathcal{S} = \{s_1 = 00001\ldots, s_2 = 00111\ldots, s_3 = 01100\ldots, s_4 = 01111\ldots, s_5 = 11010\ldots, s_6 = 11111\ldots\}$.

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

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