T U M

I N S T I T U T F Ü R I N F O R M A T I K

Smoothed Analysis of Trie Height

Stefan Eckhardt *Sven Kosub* *Johannes Nowak*

TUM-I0715
Juli 07

T E C H N I S C H E U N I V E R S I T Ä T M Ü N C H E N

TUM-INFO-07-I0715-0/1.-FI

Druck: Institut für Informatik der
Technischen Universität München

Smoothed Analysis of Trie Height

Stefan Eckhardt[*] *Sven Kosub*[†] *Johannes Nowak*[‡]
Institut für Informatik, Technische Universität München
Boltzmannstraße 3, D-85748 Garching, Germany

Jul 6, 2007

Abstract

Tries are very simple general purpose data structures for information retrieval. A crucial parameter of a trie is its height. In the worst case the height is unbounded when the trie is built over a set of n strings. Analytical investigations have shown that the average height under many random sources is logarithmic in n. Experimental studies of trie height suggest that this holds for non-random data. In order to give an analytical explanation to these findings we perform a smoothed analysis of trie height. Smoothed analysis combines elements from both, worst-case and average-case analysis: the paradigm assumes that inputs are chosen by an adversary and that the costs are expected costs over slight random pertubations of the chosen inputs. We consider a special class of string perturbation functions which can be modelled by probabilistic finite automata (PFAs). Those perturbation functions constitute an extension of a very natural class of *random edit perturbations*. We observe that for the case of random deletions the smoothed trie height is unbounded. We introduce read-deterministic star-like perturbation functions, which include random substitutions and insertion as a special case, and give a necessary and sufficient condition for the smoothed trie height under read-deterministic star-like perturbation functions being logarithmic. This condition is particularly appealing since it can easily be checked by looking at the transition probabilities of the corresponding probabilistic automaton.

[*]Corresponding author. Email address: eckhardt@in.tum.de
[†]Email address: kosub@in.tum.de
[‡]Email address: nowakj@in.tum.de

Contents

1 Introduction

Digital trees, for short tries, are very simple general purpose data structures for information retrieval. Also, the principle of recursive decomposition based upon successive bits of data items can be found in many other applications (see, e.g., [20]). This explains why many parameters of tries, such as height, path length, or size, have been and are still subject to extensive average-case analysis under various random string models. Usually, in analytic approaches a trie is considered to be built over a set S of n independent, infinite (binary) strings which are produced by a common random source. The height H_S of a trie built over the set S equals the length of the longest common prefix of *any* two strings in S. Clearly, in the worst-case this length is unbounded. H_S is a highly significant parameter as the running time of the retrieval operation is $\Theta(H_S)$.

Though almost all investigations of trie height using analytical methods suggest the average trie height to be logarithmic, it is not immediately clear that these results can utterly explain the fact that in many practical settings the height is in fact logarithmic in the number of strings. This holds particularly in the case of non-random data. Nilsson and Tikkanen [27] have experimentally investigated the height of PATRICIA trees, or path-compressed tries, and other search structures. There, the height of a PATRICIA tree, built over a set of 50,000 unique random uniform strings was 16 on average and 20 at most. For non-random data consisting of 19,461 strings from geometric data, of 16,542 ASCII character strings from a book, and of 38,367 strings from Internet routing tables, the height of a path-compressed trie, built over these data sets, was on average 21, 20, and 18, respectively, and at most 30, 41 and 24, respectively. These findings suggest that worst-case instances, i.e., input sets for which the height of the respective trie is unbounded, are isolated peaks in the input space. In this paper we try to give an analytical explanation of these findings. To this end, we perform a *smoothed analysis* of the trie height.

In their seminal paper [36], Spielman and Teng have introduced influential Smoothed Analysis[1], a hybrid between the two classical approaches to the analysis of algorithms: average-case analysis, which assumes that the input is random with respect to some probability distribution, and worst-case analysis, which assumes that the input is chosen by an adversary having full information and therefore is the worst possible input. In smoothed analysis, one is not interested in finding a probability distribution which models the typical input more accurately. Rather, one aims at answering the following kind of question: are worst-case inputs "isolated peaks" or "plateaus". To this aim, the smoothed complexity of an algorithm can be defined as follows: instead of measuring the running time of every possible input and then taking the maximum value, first measure for every input the expected running time of the algorithm, where the expectation is over slight random perturbations of the respective input. Second, take the maximum over all such expectations. This resembles a semi-random process where an adversary can choose the input but the running time is then the expectation over random perturbations of exactly this input. Note that the adversary is allowed to have full information on the distribution function of the random perturbation. One challenge in smoothed analysis is that of finding an *adequate* perturbation function.

Our Results. To the best of our knowledge, this is the first paper dealing with smoothed trie height. In particular we consider tries built over a set S of n infinite binary strings under perturbation functions which can be represented by probabilistic finite automata (PFAs). These perturbation functions

[1] See http://www.cs.yale.edu/homes/spielman/SmoothedAnalysis/index.html for an up-to-date list of papers concerning smoothed analysis.

constitute an extension of a very natural class of perturbation functions, i.e., random substitutions, insertions and deletions of symbols which we call *random edit perturbations*. Technically, we contribute the following results.

- We formally define smoothed trie height under arbitrary string perturbation functions $X : \{0,1\}^\omega \to \{0,1\}^\omega$ and show that it can be bounded by showing an exponential decreasing upper bound of γ^k on the probability $\mathbf{P}\big(\mathrm{lcp}\big(X(s), X(s)\big) \geq k\big)$ that the length of the longest common prefix of two independent perturbations of the same input string $s \in S$ is at least k.

- We observe that for the case of random deletions the smoothed trie height is unbounded.

- We introduce read-deterministic star-like perturbation functions which include random substitutions and insertion as a special case, and we give a necessary and sufficient condition for the smoothed trie height under read-deterministic star-like perturbation functions being logarithmic. This condition is particularly appealing since it can easily be checked by considering the transition probabilities of the corresponding probabilistic automaton.

Related Work. Based on early work on semi-random models for combinatorial objects (see, e.g., [34, 11, 19]), Spielman and Teng [36, 39] formally introduced smoothed analysis in order to explain the good practical performance of the simplex algorithm, which is not inferable from results about its average complexity and which is opposed to its poor worst-case complexity. Namely, the authors showed that the shadow-vertex simplex algorithm has polynomial smoothed complexity, i.e, its running time is polynomial in the size of the input and the standard deviation of the (Gaussian) perturbation. Thereupon, a number of different works concerning smoothed analysis of linear programming problems, numerical problems, and variants of the simplex algorithm followed (see, e.g., [3, 10, 38, 33, 12, 42] and the references therein). The smoothed analysis paradigm has also been applied to purely discrete optimization problems, particularly to ILPs (see, e.g., [8, 9, 31, 30]). Additionally, the smoothed complexity of various more specific algorithmic problems, e.g., the height of binary search trees [26], ordering problems, such as left-to-right maxima counting, shortest path, and quicksort [4], online algorithms [35], and computational geometry [5, 14, 15] has been investigated.

Random string models have frequently been used to analyze tries (see, e.g., [22, 23, 40, 17] for only few references). Clearly, in the worst-case, a trie built over n strings can have height $\Theta(n)$ even if paths consisting of internal nodes with degree one are contracted into one node (this is the principle which lies behind the more efficient version of tries, i.e., PATRICIA trees). For the average-case analysis, the expectation and variance (and more properties) of various parameters of tries (and PATRICA trees) such as size, path length, and height were intensively studied under different models of random strings. Generally, the n strings are independent. The oldest model is the memory-less source, were each string can be considered the outcome of an infinite sequence of Bernoulli trials, either biased, or unbiased, were each symbol corresponds to a possible outcome of the Bernoulli trial. Since memory-less sources can hardly account for any real-world process which generates strings, more general models were considered which could account for the undoubtedly existing dependency between consecutive symbols. Those models are the Markovian sources, where consecutive symbols are related via the transition matrix of a finite markov chain, its generalizations, i.e., ψ-mixing sources and stationary ergodic sources, and newer models, such as the density model of Devroye [16] or dynamic sources [13] (see, e.g., [40, 17] and the references therein for more information). A number of general methods for the analysis of algorithms have grown out of this line of research, particularly in the field of analytic combinatorics and generating functions (see [40, 43, 21] for a profound introduction).

Outline of the Paper. Section 2 is concerned with the smoothed analysis of trie height under arbitrary string perturbation functions. In Sections 2.2 and 2.3, we formally define the term *smoothed trie height* and show that it can be bounded by an exponential decreasing upper bound γ^k, where $0 < \gamma < 1$, on the probability $\mathbf{P}\big(\mathrm{lcp}\big(X(s), X(s)\big) \geq k\big)$ that the length of the prefix of two independent perturbations of the same input string s is at least k in Theorem 3. This constitutes our first main result.

Thereafter, we give a string perturbation model based on probabilistic automata in Subsection 3.1 and show that the natural class of edit perturbations is contained therein in Subsection 3.2. In Subsection 3.3 we state the second main result of our work, i.e., we give sufficient and necessary conditions for the smoothed trie height under read-deterministic star-like perturbation functions to be logarithmic in the number of strings. Finally, we give the proof of the main theorem in Section 4

2 Smoothed Trie Height

In this section we formally introduce smoothed trie height and establish a general sufficient criterion for logarithmic smoothed trie height for arbitrary perturbation functions.

2.1 Notation

Let $\mathbb{N} = \{0, 1, 2, \dots\}$ be the set of natural numbers and let $\mathbb{R}$ be the set of reals. For a finite set S, let $\|S\|$ be the cardinality of S. Given a nonempty set Σ of *symbols*, called the *alphabet*, a string s over Σ is a (possibly infinite) concatenation of symbols from Σ. The *length of* s, denoted by $|s|$, is defined as the number of symbols in s, if s is finite, and $+\infty$, otherwise. For a symbol $a \in \Sigma$ and a *finite* string s over Σ, $|s|_a$ denotes the *number of occurrences of the symbol* a *in* s. For an alphabet Σ and a number $k \in \mathbb{N}$, the set Σ^k is the set of all strings over Σ with length exactly k. For a natural number i and a string s with $|s| \geq i$, $s[i]$ denotes the i-th symbol in s. For $i, j \in \mathbb{N}$ with $i \leq j$, we abbreviate the string $s[i] \dots s[j]$ by $s[i \dots j]$. Two (equal-length) strings $s, t \in \Sigma^k$ are identical, if for all $1 \leq i \leq k$ it holds that $s[i] = t[i]$. We write $s = t$ in this case. For $l, k \in \mathbb{N} \cup \{+\infty\}$ with $0 \leq l \leq k$, we call a string $\alpha \in \Sigma^l$ a *prefix* of a string $s \in \Sigma^k$, if $\alpha = s[1 \dots l]$. We write $\alpha \sqsubseteq s$ in this case. If $l < k$ holds and α is a prefix of s, we say α is a *proper prefix* and write $\alpha \sqsubset s$. Note that by definition the *empty string*, denoted by ϵ, is a prefix of every string. Finally, for $k \in \mathbb{N}$, the set of all strings of length at most k is $\Sigma^{\leq k} = \bigcup_{i=0}^{k} \Sigma^k$. Σ^ω denotes the set of all infinite strings over Σ, $\Sigma^{<\omega}$ denotes the set of all finite strings over Σ. Let $\Sigma^{\leq\omega}$ denote the set $\Sigma^\omega \cup \Sigma^{<\omega}$.

Given two strings $s, t \in \Sigma^\omega$, the *longest common prefix* of s and t is the longest prefix of t that is also prefix of s. Further, let $\mathrm{lcp} : \Sigma^{\leq\omega} \times \Sigma^{\leq\omega} \to \mathbb{N}$ be a function measuring the length of the longest common prefix, i.e.,

$$\mathrm{lcp}(s, t) =_{\mathrm{def}} \begin{cases} \sup\{j \in \mathbb{N} \ : \ t[1 \dots j] \sqsubseteq s\} & \text{if } t[1] = s[1] \\ 0 & \text{otherwise.} \end{cases}$$

Major part of this work is concerned with binary alphabets, i.e., $\Sigma = \{0, 1\}$.

2.2 Average-Case, Worst-Case and Smoothed Trie Height

We first review the terms *average-case complexity* and *worst-case complexity* and define *smoothed complexity* more formally: Let I be the set of all inputs to an algorithm and let $I^{(n)}$ be the subset of all inputs that have size n and denote by $C(y)$ for $y \in I$ the cost of the algorithm when started with input y. The cost of an algorithm may be its running time, its number of I/O operations, its memory consumption, or any other measure, one is interested in. We at the moment assume that the cost is finite for any input. Then, the algorithm's worst-case complexity is measured by the function $\max_{y\in I^{(n)}} C(y)$ and the average-case complexity is measured by the function $\mathbf{E}_{y\in I^{(n)}}\big(C(y)\big)$, where the expectation is taken with respect to some probability measure on $I^{(n)}$.

Now, let $N : I^{(n)} \times I^{(n)} \to [0,1]$ be a function measuring similarity (that is, $N(y,x) > N(y,z)$ implies that x is more similar to y than z is to y), then the smoothed complexity is measured by

$$\max_{y\in I^{(n)}} \sum_{x\in I^{(n)}} \frac{N(y,x)}{\sum_{z\in I^{(n)}} N(y,z)} \cdot C(x).$$

Note that via the similarity measure N, we defined a probability measure on $I^{(n)}$ which depends on a specific input y. This is, we analyze for each input y the average cost of the algorithm when started with an input which is "in the neighborhood" of y and take the maximum over all y.

Another way of defining the smoothed complexity of an algorithm is the following: let $X : I^{(n)} \to I^{(n)}$ some probabilistic mapping called a *perturbation function*. [2] Here, the smoothed complexity of the algorithm is measured by the function

$$\max_{y\in I^{(n)}} \mathbf{E}\big(C(X(y))\big),$$

where the expectation is taken with respect to the perturbation function. This is, $\mathbf{E}\big(C(X(y))\big) = \sum_{x\in I^{(n)}} \mathbf{P}\big(x = X(y)\big) \cdot C(x)$. Setting $\mathbf{P}\big(x = X(y)\big) = \frac{N(x,y)}{\sum_z N(z,y)}$, one finds that both definition express essentially the same. We close this section by giving a formal definition of the smoothed trie height under arbitrary string perturbation functions. Note, that for the height $H(S)$ of the trie, built over the set S, it holds that $H(S) = \sup_{s,t\in S} \operatorname{lcp}(s,t)$ (see, e.g., [40, Theorem 1.3]).

Definition 1. *Let $S \subseteq \{0,1\}^\omega$ be some non-empty set of infinite strings. Given a perturbation function $X : \{0,1\}^\omega \to \{0,1\}^{\le\omega}$ the* smoothed trie height *over the set S under the perturbation function X, denoted by $H(S,n,X)$, is defined by*

$$H(S,n,X) =_{\text{def}} \sup_{s_1\ldots,s_n\in S} \mathbf{E}\big(\sup_{1\le i<j\le n} \operatorname{lcp}(X(s_i),X(s_j))\big)$$

2.3 Bounding Smoothed Trie Height for Arbitrary Perturbation Functions

In this section we show that smoothed analysis of trie height naturally extends the average-case analysis. Namely, we prove a lemma and a theorem which are elementary to bounding the smoothed trie height under arbitrary string perturbations.

[2] The term function" may be misleading, because classically functions map a specific input to exactly one output. Nevertheless the term perturbation function" is usually used in conjunction with smoothed analysis and one might think of X being a two-valued function $X : I^{(n)} \times [0,1] \to I^{(n)}$ such that for fixed y the interval $[0,1]$ is union of disjoint half-open intervals J_x for all x that have non-negative probability $\mathbf{P}\big(x = X(y)\big)$, i.e., $[0,1] = \cup_{x\in I^{(n)}} J_x$ with $|J_x| = \mathbf{P}\big(x = X(y)\big)$. Then let $f(y,a) = x$, if $a \in J_x$. Thus, f is a function, which is becomes a random function by letting a be a $[0,1]$-random variable.

Lemma 2 (Splitting Lemma). *Let $S \subseteq \{0,1\}^\omega$ be some non-empty set of infinite strings and let $X : \{0,1\}^\omega \to \{0,1\}^{\leq\omega}$ be any perturbation function. Then for $k' \in \mathbb{N}$ it holds that*

$$H(S,n,X) \ \leq \ k' + n^2 \sum_{k=k'+1}^{\infty} \ \sup_{s\in S} \ \sum_{|\alpha|=k} \mathbf{P}\big(\alpha \sqsubseteq X(s)\big)^2. \tag{1}$$

The Splitting Lemma states that the smoothed trie height of an arbitrary set S of n strings under an arbitrary perturbation function X can be bounded by k' plus n^2 times the probability $\sum_{|\alpha|=k} \mathbf{P}\big(\alpha \sqsubseteq X(s)\big)^2$ that the longest common prefix of two independent computations of X on the *same* input string $s \in \mathcal{S}$ is at least k'.

Proof. Let $S \subseteq \{0,1\}^\omega$ be a non-empty set and let $X : \{0,1\}^\omega \to \{0,1\}^\omega$ be some perturbation function. To bound the smoothed trie height over the set S under the perturbation function X we may write

$$\begin{aligned}
H(S,n,X) \ &= \ \sup_{\substack{A\subseteq S\\ \|A\|=n}} \mathbf{E}\Big(\sup_{s,t\in A} \operatorname{lcp}(X(s),X(t))\Big) \ = \ \sup_{\substack{A\subseteq S\\ \|A\|=n}} \sum_{k=1}^{\infty} \mathbf{P}\Big(\sup_{s,t\in A} \operatorname{lcp}(X(s),X(t)) \geq k\Big) \\
&\leq \ \sum_{k=1}^{\infty} \sup_{\substack{A\subseteq S\\ \|A\|=n}} \mathbf{P}\Big(\sup_{s,t\in A} \operatorname{lcp}(X(s),X(t)) \geq k\Big) \\
&= \ \sum_{k=1}^{k'} \sup_{\substack{A\subseteq S\\ \|A\|=n}} \mathbf{P}\Big(\sup_{s,t\in A} \operatorname{lcp}(X(s),X(t)) \geq k\Big) + \sum_{k=k'+1}^{\infty} \sup_{\substack{A\subseteq S\\ \|A\|=n}} \mathbf{P}\Big(\sup_{s,t\in A} \operatorname{lcp}(X(s),X(t)) \geq k\Big) \\
&\leq \ k' + n^2 \sum_{k=k'+1}^{\infty} \sup_{s,t\in S} \mathbf{P}\big(\operatorname{lcp}(X(s),X(t)) \geq k\big) \\
&= \ k' + n^2 \sum_{k=k'+1}^{\infty} \sup_{s,t\in S} \sum_{|\alpha|=k} \mathbf{P}\big(\alpha \sqsubseteq X(s)\big) \cdot \mathbf{P}\big(\alpha \sqsubseteq X(t)\big) \\
&\overset{(*)}{\leq} \ k' + \sum_{k=k'+1}^{\infty} \sup_{s,t\in S} \sqrt{\sum_{|\alpha|=k} \mathbf{P}\big(\alpha \sqsubseteq X(s)\big)^2} \cdot \sqrt{\sum_{|\alpha|=k} \mathbf{P}\big(\alpha \sqsubseteq X(t)\big)^2} \\
&\leq \ k' + n^2 \sum_{k=k'+1}^{\infty} \sup_{s\in S} \sum_{|\alpha|=k} \mathbf{P}\big(\alpha \sqsubseteq X(s)\big)^2
\end{aligned}$$

Here, $(*)$ follows readily from Cauchy's inequality. This proves the lemma. □

Using the Lemma 2 we are able to prove our first contribution: a tailbound for smoothed trie height.

Theorem 3 (Tailbound for Smoothed Trie Height). *Let $m \in \mathbb{N}$ and let $\gamma \in \mathbb{R}$ satisfy $0 < \gamma < 1$. Let $X : \{0,1\}^\omega \to \{0,1\}^{\leq\omega}$ be some perturbation function and let $S \subseteq \{0,1\}^\omega$ be a non-empty set of strings. Further, let $n > \gamma^{-\frac{m}{2}}$. If for all $s \in S$ and all $k \geq m$ it holds that*

$$\sum_{|\alpha|=k} \mathbf{P}\big(\alpha \sqsubseteq X(s)\big)^2 \leq \gamma^k,$$

then

$$H(\mathcal{S}, n, X) \le 2\lceil \log_{\frac{1}{\gamma}} n \rceil + \frac{\gamma}{1-\gamma}.$$

Further, if there are $c, d > 0$ *such that for all* $s \in S$ *and for all* $k \ge m$ *it holds that*

$$\sum_{|\alpha|=k} \mathbf{P}\big(\alpha \sqsubseteq X(s)\big)^2 \le d(k+c) \cdot \gamma^k,$$

then

$$H(S, n, X) \;\le\; 2\lceil \log_{\frac{1}{\gamma}} n \rceil \Big(1 + \frac{d\gamma}{1-\gamma}\Big) + d \cdot \gamma \cdot \frac{c(1-\gamma)+1}{(1-\gamma)^2}.$$

Proof. For the first part, assume that $\sum_{|\alpha|=k} \mathbf{P}\big(\alpha \sqsubseteq X(s)\big)^2 \le \gamma^k$ holds for $k \ge m$. To apply the Splitting Lemma (Lemma 2) choose $k' = 2\lceil \log_{\frac{1}{\gamma}} n \rceil = \lceil -\log_\gamma n \rceil \ge m$. Then

$$\begin{aligned}
H(\mathcal{S}, n, X) &\le k' + n^2 \sum_{k=k'+1}^{\infty} \sup_{s \in \mathcal{S}} \sum_{|\alpha|=k} \mathbf{P}\big(\alpha \sqsubseteq X(s)\big)^2 \\
&\le 2\lceil \log_{\frac{1}{\gamma}} n \rceil + \sum_{k=2\lceil \log_{\frac{1}{\gamma}} n \rceil + 1}^{\infty} n^2 \cdot \gamma^k \\
&= 2\lceil \log_{\frac{1}{\gamma}} n \rceil + \sum_{k=1}^{\infty} n^2 \cdot \gamma^{2\lceil -\log_\gamma n \rceil} \cdot \gamma^k \\
&= 2\lceil \log_{\frac{1}{\gamma}} n \rceil + \frac{\gamma}{1-\gamma}
\end{aligned}$$

Assume that $\sum_{|\alpha|=k} \mathbf{P}\big(X(s)[1 \dots k] = \alpha\big)^2 \le d(k+c) \cdot \gamma^k$ holds for $k \ge m$ and appropriate $c, d > 0$. To apply the Splitting Lemma (Lemma 2) choose $k' = 2\lceil \log_{\frac{1}{\gamma}} n \rceil = \lceil -\log_\gamma n \rceil \ge m$. We obtain from the Splitting Lemma

$$\begin{aligned}
H(S, n, X) &\le k' + n^2 \sum_{k=k'+1}^{\infty} \sup_{s \in S} \sum_{|\alpha|=k} \mathbf{P}\big(\alpha \sqsubseteq X(s)\big)^2 \\
&\le 2\lceil \log_{\frac{1}{\gamma}} n \rceil + \sum_{k=2\lceil \log_{\frac{1}{\gamma}} n \rceil + 1}^{\infty} d \cdot (k+c) \cdot n^2 \cdot \gamma^k \\
&= 2\lceil \log_{\frac{1}{\gamma}} n \rceil + \sum_{k=1}^{\infty} d \cdot (2\lceil \log_{\frac{1}{\gamma}} n \rceil + k + c) \cdot n^2 \cdot \gamma^{2\lceil -\log_\gamma n \rceil} \cdot \gamma^k \\
&= 2\lceil \log_{\frac{1}{\gamma}} n \rceil + d \cdot (2\lceil \log_{\frac{1}{\gamma}} n \rceil + c) \cdot \sum_{k=1}^{\infty} \gamma^k + d \cdot \sum_{k=1}^{\infty} k \cdot \gamma^k \\
&= 2\lceil \log_{\frac{1}{\gamma}} n \rceil \Big(1 + \frac{d\gamma}{1-\gamma}\Big) + d \cdot \gamma \cdot \frac{c(1-\gamma)+1}{(1-\gamma)^2}
\end{aligned}$$

This proves the theorem. $\square$

3 Generalized Edit Perturbations

3.1 Mealy-type Probabilistic Finite Automata

A probabilistic automaton [29, 28] is a standard way to model a noisy communication channel or other unreliable deterministic systems. We suggest to consider random perturbation functions representable by probabilistic automata. It is not the aim of this paper to develop a general theory of automata-based perturbation models. Rather we use probabilistic automata as precise, compact and nevertheless fairly general specifications of random perturbations of strings. It is convenient for our purposes to define probabilistic automata in a slightly non-standard way, by separating states where an input symbol is read from states where an output symbol is written. This provides an easy way to describe automata computing non-length-respecting input-output relations.

A (Mealy-type) probabilistic finite automaton (PFA) over a finite alphabet Σ is a tuple $X = (R, W, \mu_R, \mu_W, \sigma)$ where:

- R is a non-empty, finite set of *input states*.
- W is a non-empty, finite set of *output states*.
- $\mu_R : R \times \Sigma \times (W \cup R) \to [0, 1]$ is the transition probability function for reading states such that
$$(\forall q \in R)(\forall a \in \Sigma) \sum_{p \in R \cup W} \mu_R(q, a, p) = 1.$$
A function value $\mu_R(q, a, p)$ should be read as follows: when X is in state q and when reading symbol a, move into state p with probability $\mu_R(q, a, p)$. Note that possibly $\mu_R(q, a, q) > 0$.
- $\mu_W : W \times \Sigma \times (W \cup R) \to [0, 1]$ is the transition probability function for writing states such that
$$(\forall q \in W)(\forall a \in \Sigma) \sum_{p \in R \cup W} \mu_W(q, a, p) = 1.$$
A function value $\mu_W(q, a, p)$ should be read as follows: when X is in state q, with probability $\mu_W(q, a, p)$, write symbol a and move into state p. Note that possibly $\mu_W(q, a, q) > 0$.
- $\sigma : W \cup R \to [0, 1]$ is the initial probability distribution.

A computation by X on input string t stops when X reaches a reading state after the last symbol of t is read. Concatenating all output symbols X describes a random function mapping finite or infinite words to finite or infinite words. By a slight abuse of notation we will identify the random mapping $X : \Sigma^{\leq\omega} \to \Sigma^{\leq\omega}$ with its defining PFA X.

3.2 Edit Perturbations as PFAs

Edit operations, i.e., *substituting*, *deleting* or *inserting* symbols, are among the most fundamental operations for locally manipulating strings. We say that a perturbation function on strings is an *edit perturbation* if it perturbs the input by randomly substituting, inserting or deleting symbols. The PFAs which we introduced above are on the one hand a compact representation for edit perturbations,

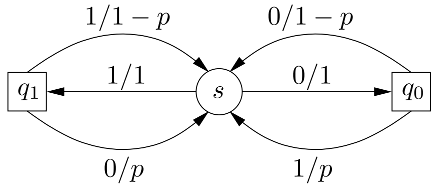

Figure 1: Substitution PFA with initial state distribution $\sigma(s) = 1$ and $\sigma(q_0) = \sigma(q_1) = 0$.

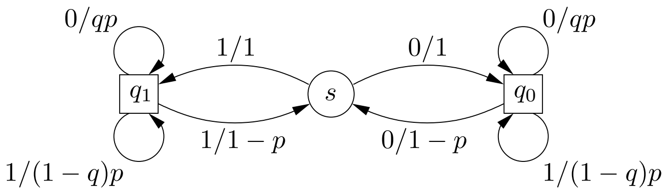

Figure 2: Insertion PFA with initial state distribution $\sigma(s) = 1-p$, $\sigma(q_0) = qp$, and $\sigma(q_1) = (1-q)p$.

but on the other hand they also provide the opportunity to generalize edit perturbations. We next give a PFA for each of the edit perturbations.

In all figures to follow states in circles are input states and states in boxes are output states. As usual transitions are only drawn if their probability is strictly positive. Transitions are labeled by a tuples x/p where $x \in \Sigma$ and $0 \leq p \leq 1$. The semantics is as follows: for a reading transition, x/p means "if we read symbol x, then we move along the respective transition with probability p"; for a writing transition, x/p means "with probability p, we move along the respective transition and write x".

Substitutions. Let $p \in (0,1)$ and SUB_p be the PFA visualized in Figure 1. If we are in input state s and read a symbol $a \in \{0,1\}$, then we move into writing state q_a and move back to state s writing a with probability $1-p$ and a symbol other than a with probability p. The automaton perturbs an input string by performing random substitutions on it. As a consequence of our main theorem (Theorem 8) which will be presented in the next subsection the following result can be obtained.

Theorem 4. *Let $p \in (0,1)$ and let SUB_p be the PFA depicted in Figure 1. Then for $n \geq 1$, the smoothed trie height $H(\{0,1\}^\omega, n, \mathrm{SUB}_p)$ over a set of n arbitrary binary strings under substitutions is $O(\log n)$.*

Though the result itself is not very surprising it is nevertheless worth being mentioned since it nicely coincides with the average-case prefix length under a memory-less random source (see [40] for a detailed proof that the average-case prefix length is logarithmic in this case). Also it shows that our approach of analysis is an extension of the average-case analysis for memory-less random sources.

Insertions. Let $p, q \in (0,1)$. Let $\mathrm{INS}_{p,q}$ be the automaton depicted in Figure 2. It scans an input string $s \in \{0,1\}^\omega$ and randomly inserts symbols with probability p. The inserted symbol is 0 with probability q and 1 with probability $1-q$. Note that the specified initial distribution enables insertions before the first letter is read.

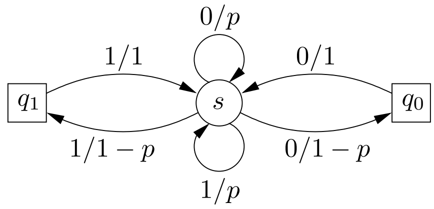

Figure 3: Deletion PFA with initial state distribution $\sigma(s) = 1$ and $\sigma(q_0) = \sigma(q_1) = 0$.

Theorem 5. *Let $p, q \in (0,1)$ and let $\mathrm{INS}_{p,q}$ be the PFA depicted in Figure 2. Then for $n \geq 1$ the smoothed trie height $H(\{0,1\}^\omega, n, \mathrm{INS}_{p,q})$ over a set of n arbitrary binary strings under insertions is $O(\log n)$.*

Again the above theorem turns out to be a direct corollary of our main theorem.

Deletions. For $p \in (0,1)$ let DEL_p be the automaton depicted in Figure 3 that while scanning an infinite binary string deletes the symbol at a certain position with probability p independently. There is no general positive result in the case of deletions, i.e., $H(\{0,1\}^\omega, n, \mathrm{DEL}_p)$ is unbounded. This becomes clear by the following argument: consider the infinite string consisting of only one kind of symbol, i.e., $s = 111\ldots$. Then $\mathrm{DEL}_p(s) = 111\ldots$ (and being finite with probability zero). Thus, $\sup_{s,t\in\{0,1\}^\omega} \mathrm{lcp}(X(s), X(t))$ is unbounded.

3.3 Read-deterministic star-like string perturbation functions

In this subsection we specify a class of PFAs capturing typical properties of edit perturbations—the star-like automata—and a further restricted class reasonably large and allowing a characterization of logarithmic smoothed trie height.

Definition 6. *Let Σ be a finite alphabet and let $X = (R, W, \mu_R, \mu_W, \sigma)$ be a PFA. X is said to be* star-like *if and only if*

1. *$\|R\| = 1$, i.e., $R = \{s\}$, and*
2. *the function μ_W is such that*

$$(\forall q, q' \in W, q \neq q')(\forall a \in \Sigma) \quad \mu_W(q, a, q') = 0,$$

i.e., the graph induced by the vertex set W and the edge set $\{(q,q') : \mu_W(q,a,q') > 0\}$ consists of a number of connected components each of which is a single vertex, and

3. *for all $q \in W$ it holds that $\sum_{a\in\Sigma} \mu_W(q,a,q) < 1$, i.e, the probability that X loops at q is strictly less than one.*

We say that a perturbation function is star-like if it can be represented by a star-like automaton. The class of star-like perturbation functions is a natural extension of the edit perturbations from Section 3.2: all of them have already been depicted by star-like automata (see Figures 1, 2 and 3 above).

Definition 7. *A PFA $X = (\{s\}, W, \mu_R, \mu_W, \sigma)$ is said to be* read-deterministic *if and only if for all $a \in \Sigma$ it holds that $\mu_R(s, a, s) = 0$, i.e., μ_R has no loops at s, and $\|\{q \in W : \mu_R(s, a, q) > 0\}\| = 1$, i.e., for each symbol a there is one state $q_a \in W$ such that, when X is in input state s and reads a, then it moves into state q_a with probability one.*

We say that a perturbation function is read-deterministic star-like if it can be represented by a read-deterministic star-like automaton. The class of read-deterministic star-like perturbation functions includes the functions SUB_p and $\mathrm{INS}_{p,q}$.

Now we are ready to state the main result of the paper.

Theorem 8. *Let X be a read-deterministic star-like PFA over the binary alphabet $\{0, 1\}$. Then the following statements are equivalent.*

1. $(\forall q \in W)(\forall a \in \{0, 1\}) \quad \mu_W(q, a, q) + \mu_W(q, a, s) < 1.$
2. $(\exists \varepsilon \in \mathbb{R}, c, m' \in \mathbb{N})(\forall t \in \{0, 1\}^\omega)(\forall m \geq m') \quad \sum_{|\alpha|=m} \mathbf{P}\big(\alpha \sqsubseteq X(t)\big)^2 \leq cm \cdot \big(1 - \varepsilon\big)^m.$
3. $H(\{0, 1\}^\omega, n, X) \in O(\log n).$

The meaning of the theorem is the following: it states that the smoothed trie height over a set of n infinite strings under a read-deterministic star-like perturbation function is logarithmic iff the corresponding automaton is such that condition (1) is satisfied, i.e., if there is no input symbol $a \in \{0, 1\}$ such that if X reads a then it deterministically writes a symbol $b \in \{0, 1\}$. Thus, Theorems 4 and 5 are corollaries of Theorem 8 because both perturbation functions are read-deterministic star-like ones and $p \in (0, 1)$ for SUB_p and $p, q \in (0, 1)$ for INS_{pq} imply that condition (1) is satisfied.

4 Proof of Theorem 8

4.1 The Proof at a Glance

We will prove Theorem 8 via a series of Lemmas. Recall that, according to Theorem 3, the smoothed prefix length $H(\{0, 1\}, n, X)$ under an arbitrary perturbation function X can be bounded by showing that there exists an $m \in \mathbb{N}$ such that for all $k \geq m$ it holds that

$$\sum_{|\alpha|=k} \mathbf{P}\big(\alpha \sqsubseteq X(t)\big)^2 \leq d \cdot (k + c) \cdot \gamma^k$$

for $\gamma < 1$ and appropriate constants $c, d > 0$. Technically, we show that statement (1) in Theorem 8 is a sufficient condition for a read-deterministic star-like perturbation function X such that there exist $\varepsilon, \varepsilon' \in (0, 1)$ and constants $d, c' > 0$ such that the following chain of inequalities holds

$$\sum_{|\alpha|=m} \mathbf{P}\big(\alpha \sqsubseteq X(t)\big)^2 \leq c' \cdot m \sum_{|\alpha|=m} \sum_{l=1}^{m} (1 - \varepsilon)^m \rho(\alpha, t, 1, l) \leq d \cdot m(1 - \varepsilon')^m,$$

where for $1 \leq l \leq m$, $\rho(\alpha, t, 1, l)$ is roughly the probability that α is the output of the perturbation function on input $t[1 \ldots l]$. The first inequality will be shown by Lemmas 9 and 10 and the second inequality by expanding the sum $\sum_{|\alpha|=m} \sum_{l=1}^{m} (1 - \varepsilon)^m \rho(\alpha, t, 1, l)$ and showing its exponentially decreasing upper bound using generating functions in Lemma 11. This gives $(1) \Rightarrow (2)$ in the theorem. Then, applying Theorem 3, $(2) \Rightarrow (3)$ follows. Finally, $(3) \Rightarrow (1)$ will be shown by contraposition.

4.2 Proving the Tailbound

In the following we are interested in the quantity $\sum_{|\alpha|=m} \mathbf{P}(\alpha \sqsubseteq X(t))^2$ for an infinite binary string t and a read-deterministic star-like perturbation function represented by the PFA $X = (\{s\}, W, \mu_R, \mu_W, \sigma)$. Let $q \in W$ be a writing state and let βa be a finite binary string. Assume that the computation moves into state q at some point. Then it can easily been checked that the probability that X writes the output string βa *without reading any symbol* and moves into the reading state s thereafter, an event denoted by $q \hookleftarrow \beta a$, is

$$\mathbf{P}(q \hookleftarrow \beta a) =_{\text{def}} \mu_W(q,0,q)^{|\beta|_0} \mu_W(q,1,q)^{|\beta|_1} \mu_W(q,a,s)$$

and the probability that X writes the output string βa *without reading any symbol* and *either* moves into state s thereafter *or* "stays" in state q, an event denoted by $q \to \beta a$, is

$$\mathbf{P}(q \to \beta a) =_{\text{def}} \mu_W(q,0,q)^{|\beta|_0} \mu_W(q,1,q)^{|\beta|_1} (\mu_W(q,a,q) + \mu_W(q,a,s)).$$

It is clear that in both cases, X must move along a transition that writes 0 exactly $|\beta|_0$ times and along a transition writing 1 exactly $|\beta|_1$ times. This event has the probability $\mu_W(q,0,q)^{|\beta|_0} \cdot \mu_W(q,1,q)^{|\beta|_1}$. Then it either moves back into s and writes a, which happens with probability $\mu_W(q,a,s)$, or loops into q and writes a, which has probability $\mu_W(q,a,q)$. Now, for a fixed non-empty string α, the probability that α is a prefix of $X(t)$ is

$$\mathbf{P}(\alpha \sqsubseteq X(t)) = \sum_{l=1}^{|\alpha|} \sum_{\alpha=\beta_1\ldots\beta_l} \prod_{j=1}^{l-1} \mathbf{P}\big(q_{t[j]} \hookleftarrow \beta_j\big) \cdot \mathbf{P}\big(q_{t[l]} \to \beta_l\big) \tag{2}$$

This can be seen as follows: each computation on t which produces the prefix α must read the prefix $t[1\ldots l]$ of t where $1 \le l \le |\alpha|$. The upper bound of $|\alpha|$ in this case holds because read-deterministic star-like perturbation functions are *non-abridging* perturbation functions, i.e., for all $t \in \{0,1\}^+$ it holds that $|X(t)| \ge |t|$. Now, for a prefix $t[1\ldots l]$, there are $\binom{m-1}{l-1}$ possible decompositions $\alpha = \beta_1\beta_2\ldots\beta_l$ such that β_j is written *after* reading $t[j]$ and *before* reading $t[j+1]$ for $1 \le j < l$. The probability for a fixed decomposition is $\prod_{j=1}^{l-1} \mathbf{P}\big(q_{t[j]} \hookleftarrow \beta_j\big) \cdot \mathbf{P}\big(q_{t[l]} \to \beta_l\big)$. Together this gives Equation (2). For the sake of readability we define

$$\rho(\alpha,t,i,j) =_{\text{def}} \begin{cases} 1 & \text{if } |\alpha| = 0 \\ 0 & \text{if } |\alpha| > 0 \\ & \quad \text{and } (i > j \text{ or } |\alpha| < j-i+1) \\ \displaystyle\sum_{\alpha=\beta_1\ldots\beta_{j-i+1}} \prod_{k=1}^{j-i+1} \mathbf{P}\big(q_{t[i+k-1]} \hookleftarrow \beta_k\big) & \text{otherwise} \end{cases}$$

That is, $\rho(\alpha,t,i,j)$ equals the probability that the perturbation-automaton X starts in reading-state s, reads the string $t[i\ldots j]$, writes the string α, and after having written the last symbol of α moves into the reading-state s again. For $\alpha \in \{0,1\}^m$ it holds that

$$\rho(\alpha,t,i,j) = \sum_{k=j-i+1}^{m-1} \rho(\alpha[1\ldots k],t,i,j-1) \cdot \rho(\alpha[k+1\ldots m],t,j,j)$$

and we can express

$$\mathbf{P}(\alpha \sqsubseteq X(t)) = \sum_{l=1}^{m} \sum_{k=l}^{m} \rho(\alpha[1..k-1],t,1,l-1) \cdot \mathbf{P}\big(q_{t[l]} \to \alpha[k..m]\big) \tag{3}$$

The following lemma shows that we can treat prefixes of t, separated by their length and that we can further restrict ourselves to those computations which can be expressed in terms of ρ, i.e., which moved back into the reading state after having written the last symbol of α.

Lemma 9. *For $m \geq 1$ and X a read-deterministic star-like PFA X there exists a constant $c > 0$ which depends only on X such that*

$$\sum_{|\alpha|=m} \mathbf{P}(\alpha \sqsubseteq X(t))^2 \leq cm \cdot \sum_{|\alpha|=m} \sum_{l=1}^{m} \rho(\alpha, t, 1, l)^2.$$

Proof. Choose c as reciprocal to the minimum non-zero "write and return to s" transition probability in X, i.e.,

$$c =_{\text{def}} \max\{\mu_W(q, a, s)^{-1} : q \in W, a \in \{0, 1\}, \text{ and } \mu_W(q, a, s) > 0\}.$$

Note that c is always defined since X is a star-like automaton. Now,

$$\begin{aligned} &\sum_{|\alpha|=m} \mathbf{P}(\alpha \sqsubseteq X(t))^2 \\ &= \sum_{|\alpha|=m} \left(\sum_{l=1}^{m} \sum_{k=l}^{m} \rho(\alpha[1 \ldots k-1], t, 1, l-1) \cdot \mathbf{P}\big(q_{t[l]} \to \alpha[k \ldots m]\big) \right)^2 \qquad \text{(by Equation (3))} \\ &\leq m \cdot \sum_{|\alpha|=m} \sum_{l=1}^{m} \left(\sum_{k=l}^{m} \rho(\alpha[1 \ldots k-1], t, 1, l-1) \cdot \mathbf{P}\big(q_{t[l]} \to \alpha[k \ldots m]\big) \right)^2 \qquad (4) \end{aligned}$$

where (4) follows from Cauchy's inequality. We consider for each pair (α, l), where $|\alpha| = m$ and $1 \leq l \leq m$, the addend in (4): if $\mu_W(q_{t[l]}, \alpha[m], s) > 0$, then it holds that $\mu_W(q_{t[l]}, \alpha[m], q_{t[l]}) \leq c \cdot \mu_W(q_{t[l]}, \alpha[m], s)$ and

$$\begin{aligned} &\sum_{k=l}^{m} \rho(\alpha[1 \ldots k-1], t, 1, l-1) \cdot \mathbf{P}\big(q_{t[l]} \to \alpha[k \ldots m]\big) \\ &\quad \leq \sum_{k=l}^{m} \rho(\alpha[1 \ldots k-1], t, 1, l-1) \cdot (1+c) \cdot \mathbf{P}\big(q_{t[l]} \hookrightarrow \alpha[k \ldots m]\big) = (1+c) \cdot \rho(\alpha, t, 1, l). \end{aligned}$$

Now, assume that $\mu_W(q_{t[l]}, \alpha[m], s) = 0$. Then we know from Property (3) in Definition 6 that there must be some symbol $b \neq \alpha[m]$ such that $\mu_W(q_{t[l]}, b, s) > 0$. That is $\mu_W(q_{t[l]}, \alpha[m], s) \leq c \cdot \mu_W(q_{t[l]}, b, s)$ and for all k, $1 \leq k \leq m-1$, it holds that

$$\mathbf{P}\big(q_{t[l]} \to \alpha[k \ldots m]\big) \leq c \cdot \mathbf{P}\big(q_{t[l]} \hookrightarrow \alpha[k \ldots m-1]b\big)$$

Thus, we obtain

$$\begin{aligned} &\sum_{k=l}^{m} \rho(\alpha[1 \ldots k-1], t, 1, l-1) \cdot \mathbf{P}\big(q_{t[l]} \to \alpha[k \ldots m]\big) \\ &\quad \leq \sum_{k=l}^{m-1} \rho(\alpha[1 \ldots k-1], t, 1, l-1) \cdot c \cdot \mathbf{P}\big(q_{t[l]} \hookrightarrow \alpha[k \ldots m-1]b\big) = c \cdot \rho(\beta, t, 1, l) \end{aligned}$$

where $\beta = \alpha[1\dots m-1]b$. That is, we bound the addend, corresponding to the pair (α, l) in (4), by the estimate $c \cdot \rho(\beta, t, 1, l)$ for the addend corresponding to the pair (β, l). What is left is to show that we do not account an estimate for too many non-corresponding addends. However, since the alphabet is binary there is *at most one* α such that $\alpha[1\dots m] = \beta[1\dots m]$ for each β, i.e, each estimate is accounted for at most twice, and we conclude as follows

$$\begin{aligned}
&\sum_{|\alpha|=m} \mathbf{P}(\alpha \sqsubseteq X(t))^2 \\
&\quad \le \; m \cdot \sum_{|\alpha|=m} \sum_{l=1}^{m} \left(\sum_{k=l}^{m} \rho(\alpha[1\dots k-1], t, 1, l-1) \cdot \mathbf{P}\big(q_{t[l]} \to \alpha[k\dots m]\big) \right)^2 \text{ (by Inequation (4))} \\
&\quad \le \; 2 \cdot (1+c)^2 \cdot m \cdot \sum_{|\alpha|=m} \sum_{l=1}^{m} \rho(\alpha, t, 1, l)^2.
\end{aligned}$$

This proves the lemma. ☐

Lemma 10. *Let $X = (\{s\}, W, \mu_R, \mu_W, \sigma)$be a read-deterministic star-like PFA and assume that there exists a constant $\varepsilon > 0$ depending only on X such that for all $q \in W$ condition (1) of Theorem 8 is satisfied, i.e., it holds that*

$$(\forall q \in W)(\forall a \in \{0,1\}) \quad \mu_W(q, a, q) + \mu_W(q, a, s) \le 1 - \varepsilon.$$

Then there is a constant $c > 0$ such that for all $l \ge 1$ and for all input strings $t \in \{0,1\}^\omega$ and for all $m \ge l$ it holds that

$$\sum_{|\alpha|=m} \rho(\alpha, t, 1, l)^2 \le c \cdot (1-\varepsilon)^l \cdot \sum_{|\alpha|=m} \rho(\alpha, t, 1, l).$$

Proof. Assume that the condition holds for some $\varepsilon > 0$. Let $1 \le j \le l$ and $q = q_{t[j]}$. For a fixed $\alpha \in \{0,1\}^m$, where $m \ge 1$, it holds that

$$\begin{aligned}
\sum_{i=1}^{m} \rho(\alpha[1\dots i], t, j, j) &= \mu_W(q, \alpha[1], s) + \mu_W(q, \alpha[1], q) \cdot \underbrace{\sum_{i=1}^{m-1} \rho(\alpha[2\dots i], t, j, j)}_{\le 1} \\
&\le \mu_W(q, \alpha[1], s) + \mu_W(q, \alpha[1], q) \\
&\le 1 - \varepsilon.
\end{aligned} \tag{5}$$

Now, we prove the rest of the lemma by induction on the *length* l of the substring of t, which is read. Let $c = \frac{1}{1-\varepsilon}$. For $l = 1$, it holds that

$$\sum_{|\alpha|=m} \rho(\alpha, t, 1, 1)^2 \le c \cdot (1-\varepsilon) \cdot \sum_{|\alpha|=m} \rho(\alpha, t, 1, 1).$$

Now, for $l > 1$, assume that the inequality holds for all strings of length up to $l-1$. Then,

$$\sum_{|\alpha|=m} \rho(\alpha, t, 1, l)^2$$

$$
\begin{aligned}
&= \sum_{|\alpha|=m}\left(\sum_{k=1}^{m-l+1}\rho(\alpha[1\ldots k],t,1,1)\rho(\alpha[k+1\ldots m],t,2,l)\right)^2\\
&= \sum_{|\alpha|=m}\left(\sum_{k=1}^{m-l+1}\frac{\sum_{i=1}^{m-l+1}\rho(\alpha[1\ldots i],t,1,1)}{\sum_{i=1}^{m-l+1}\rho(\alpha[1\ldots i],t,1,1)}\rho(\alpha[1\ldots k],t,1,1)\rho(\alpha[k+1\ldots m],t,2,l)\right)^2\\
&= \sum_{|\alpha|=m}\left(\sum_{i=1}^{m-l+1}\rho(\alpha[1\ldots i],t,1,1)\right)^2\left(\sum_{k=1}^{m-l+1}\frac{\rho(\alpha[1\ldots k],t,1,1)}{\sum_{i=1}^{m-l+1}\rho(\alpha[1\ldots i],t,1,1)}\rho(\alpha[k+1\ldots m],t,2,l)\right)^2\\
&\stackrel{(*)}{\le} \sum_{|\alpha|=m}\left(\sum_{i=1}^{m-l+1}\rho(\alpha[1\ldots i],t,1,1)\right)\left(\sum_{k=1}^{m-l+1}\rho(\alpha[1\ldots k],t,1,1)\rho(\alpha[k+1\ldots m],t,2,l)^2\right)\\
&\le (1-\varepsilon)\sum_{|\alpha|=m}\left(\sum_{k=1}^{m-l+1}\rho(\alpha[1\ldots k],t,1,1)\rho(\alpha[k+1\ldots m],t,2,l)^2\right) \qquad \text{(by Inequation (5))}
\end{aligned}
$$

The inequality $(*)$ follows from Jensen's inequality for convex functions, i.e., from $\left(\sum_{i=1}^n a_i b_i\right)^2 \le \left(\sum_{i=1}^n a_i\right)\left(\sum_{i=1}^n b_i^2 a_i\right)$. Now,

$$
\begin{aligned}
&(1-\varepsilon)\sum_{|\alpha|=m}\sum_{k=1}^{m-l+1}\rho(\alpha[1\ldots k],t,1,1)\rho(\alpha[k+1\ldots m],t,2,l)^2\\
&\quad\le (1-\varepsilon)\sum_{k=1}^{m-l+1}\left(\sum_{|\alpha|=k}\rho(\alpha,t,1,1)\right)\left(\sum_{|\alpha|=m-k}\rho(\alpha,t,2,l)^2\right)\\
&\quad\stackrel{(**)}{\le} (1-\varepsilon)\sum_{k=1}^{m-l+1}\left(\sum_{|\alpha|=k}\rho(\alpha,t,1,1)\right)\left(\sum_{|\alpha|=m-k}c\cdot(1-\varepsilon)^{l-1}\rho(\alpha,t,2,l)\right)\\
&\quad= c\cdot(1-\varepsilon)^l\sum_{|\alpha|=m}\rho(\alpha,t,1,l)
\end{aligned}
$$

where inequality $(**)$ follows from the induction hypothesis since $|t[2\ldots l]| = l-1$. Here, it does not matter whether we start reading at the first or second position of t. This proves the lemma. $\square$

Now, we have arrived at

$$\sum_{|\alpha|=m}\mathbf{P}\big(\alpha\sqsubseteq X(t)\big)^2 \le c\cdot m\cdot\sum_{l=1}^{m}(1-\varepsilon)^l\sum_{|\alpha|=m}\rho(\alpha,t,1,l).$$

The rest is dedicated to bounding the right-hand side of the above inequality, i.e., to bound

$$\upsilon(m) =_{\text{def}} \sum_{l=1}^{m}(1-\varepsilon)^l\sum_{|\alpha|=m}\rho(\alpha,t,1,l).$$

We start by expanding its last component for fixed l. For $a\in\{0,1\}$, let $p_a = \mu_W(q_a,0,s)+\mu_W(q_a,1,s)$ be the *return probability* for q_a. Thus, we get

$$\sum_{|\alpha|=m}\rho(\alpha,t,1,l)$$

$$
\begin{aligned}
&= \sum_{|\alpha|=m} \sum_{\alpha=\beta_1 a_1 \dots \beta_l a_l} \prod_{k=1}^{l} \rho(\beta_k a_k, t, k, k) \\
&= \sum_{|\alpha|=m} \sum_{\alpha=\beta_1 a_1 \dots \beta_l a_l} \prod_{k=1}^{l} \mu_W(q_{t[k]},0,q_{t[k]})^{|\beta_k|_0} \mu_W(q_{t[k]},1,q_{t[k]})^{|\beta_k|_1} \mu_W(q_{t[k]},a_k,s) \\
&\overset{(*)}{=} \sum_{\Delta_1+\dots+\Delta_l=m-l} \prod_{k=1}^{l} \Big(\sum_{a_k \in \{0,1\}} \mu_W(q_{t[k]},a_k,s) \Big) \Big(\sum_{\substack{\beta_k \in \{0,1\} \\ |\beta_k|=\Delta_k}} \mu_W(q_{t[k]},0,q_{t[k]})^{|\beta_k|_0} \mu_W(q_{t[k]},1,q_{t[k]})^{|\beta_k|_1} \Big) \\
&= \sum_{\Delta_1+\dots+\Delta_l=m-l} \prod_{k=1}^{l} p_{t[k]} (1-p_{t[k]})^{\Delta_k}, \qquad (6)
\end{aligned}
$$

where the last equality follows from the Binomial Theorem. For $(*)$, note that there are exactly $\binom{m-1}{l-1}$ possibilities to decompose α in $\beta_1 a_1 \dots \beta_l a_l$. The rest follows by factoring out the respective terms. For $p_0 = p_1$, we get

$$
\upsilon(m) = (1-\varepsilon)p_0 \sum_{l=0}^{m-1} \binom{m-1}{l} \big((1-\varepsilon)p_0\big)^l \big(1-p_0\big)^{m-1-l} = (1-\varepsilon)p_0 \big(1-\varepsilon p_0\big)^{m-1}.
$$

Therefore, in the following, without loss of generality, we assume that $p_0 > p_1$. Now, we can expand the whole l-th addend of $\upsilon(m)$: for l fixed, let $l_0 = |t[1 \dots l]|_0$ be the number of zeros in $t[1 \dots l]$ and assume that $0 < l_0 < l$ holds (the special case of input strings consisting of only one kind of symbol will be consider thereafter). Then

$$
\begin{aligned}
&(1-\varepsilon)^l \sum_{|\alpha|=m} \rho(\alpha, t, 1, l) \\
&= (1-\varepsilon)^l \cdot \sum_{\Delta_1+\dots+\Delta_l=m-l} \prod_{k=1}^{l} p_{q_{t[l]}} (1-p_{q_{t[k]}})^{\Delta_k} \qquad \text{(by Equation (6))} \\
&\overset{(*)}{=} (1-\varepsilon)^l \cdot \sum_{i=l_0}^{m-l+l_0} \binom{i-1}{l_0-1} p_0^{l_0} (1-p_0)^{i-l_0} \binom{m-i-1}{l-l_0-1} p_1^{l-l_0} (1-p_1)^{m-l+l_0-i} \\
&= \sum_{i=l_0}^{m-l+l_0} \binom{i-1}{l_0-1} \big((1-\varepsilon)p_0\big)^{l_0} (1-p_0)^{i-l_0} \binom{m-i-1}{l-l_0-1} \big((1-\varepsilon)p_1\big)^{l-l_0} (1-p_1)^{m-l+l_0-i} \qquad (7)
\end{aligned}
$$

Here, $(*)$ holds, because every fixed α is written out of writing states q_0 and q_1. At least l_0 symbols of it must be written while returning from q_0 to s and also at least $l - l_0$ symbols must be written while returning from q_1 to s. The other $m - l$ symbols of α can be written out of either state q_0 or state q_1. Assume that i symbols, where $l_0 \le i \le m - l + l_0$, are written out of state q_0. Then there are exactly $\binom{i-1}{l_0-1}\binom{m-i-1}{l-l_0-1}$ possibilities and each has probability $p_0^{l_0}(1-p_0)^{i-l_0} p_1^{l-l_0} (1-p_0)^{m-i-l+l_0}$. Summing of all such i, $(*)$ follows. For the special case of $l_0 \in \{0, l\}$, Equation (6) simplifies to

$$
(1-\varepsilon)^l \sum_{|\alpha|=m} \rho(\alpha, t, 1, l) = \binom{m-1}{l-1} \cdot \big((1-\varepsilon)p_i\big)^l \cdot (1-p_i)^{m-l},
$$

where $i = 0$, if $l_0 = l$ and $i = 1$, otherwise. Since we do not know the number of 0's and 1's in t and in order to show that $v(m)$ decreases exponentially in m for all t, we make use of the following trick: As shown above each addend in $v(m)$ depends only on the *number* of 0's and 1's in t, but *not* on the order in which they appear in t. Thus, we have only $l+1$ different classes of input strings with respect to the size of the l-th addend in $v(m)$ and since we aim at showing an exponential decrease in m, counting over each addend at most m times should not matter at all. This is, we bound

$$\begin{aligned} v(m) \;\le\;& \overbrace{\sum_{l=1}^{m}\binom{m-1}{l-1}\cdot\Big(\big((1-\varepsilon)p_0\big)^l(1-p_0)^{m-l}+\big((1-\varepsilon)p_1\big)^l(1-p_1)^{m-l}\Big)}^{(1)\ \text{(the case that } t \text{ consists of only one kind of symbol)}}+\\ &+\underbrace{\sum_{l=2}^{m}\sum_{l_0=1}^{l-1}\sum_{i=l_0}^{m-l+l_0}\binom{i-1}{l_0-1}\big((1-\varepsilon)p_0\big)^{l_0}(1-p_0)^{i-l_0}\binom{m-i-1}{l-l_0-1}\big((1-\varepsilon)p_1\big)^{l-l_0}(1-p_1)^{m-l+l_0-i}}_{(2)\text{(each string } t \text{ consisting of more than one kind of symbol is counted at least once)}} \qquad (8)\end{aligned}$$

The sum (1) can be upper-bounded by $(1-\varepsilon p_0)^{m-1}+(1-\varepsilon p_1)^{m-1}$ by the Binomial Formula. To give an upper bound on (2), we use generating functions.

Lemma 11. *Let a_m denote the value of (2) as a function of m. Then the univariate ordinary generating function (OGF) corresponding to the sequence $\{a_m\}_{m\ge 2}$ is*

$$A(z) = z^2\cdot(1-\varepsilon)\frac{p_0p_1}{p_0-p_1}\cdot\left(\frac{1-(1-\varepsilon)p_0+\varepsilon^2}{1-(1-\varepsilon p_0)z}-\frac{1-(1-\varepsilon)p_1+\varepsilon^2}{1-(1-\varepsilon p_1)z}\right).$$

Proof. Consider the following bivariate ordinary generating function (BGF) in the variables z and u, where we substitute $\delta = 1-\varepsilon$ for the sake of readability.

$$B(z,u) = \sum_{m,l\ge 2}\sum_{l_0=1}^{l-1}\sum_{i=l_0}^{m-l+l_0}\binom{i-1}{l_0-1}\big(\delta p_0\big)^{l_0}(1-p_0)^{i-l_0}\binom{m-i-1}{l-l_0-1}\big(\delta p_1\big)^{l-l_0}(1-p_1)^{m-l+l_0-i}z^m u^l.$$

Then the corresponding *vertical* generating function is

$$\begin{aligned} &B^{\langle l\rangle}(z)\\ &= u^l\sum_{m\ge l}\sum_{l_0=1}^{l-1}\sum_{i=l_0}^{m-l+l_0}\binom{i-1}{l_0-1}\big(\delta p_0\big)^{l_0}(1-p_0)^{i-l_0}\binom{m-i-1}{l-l_0-1}\big(\delta p_1\big)^{l-l_0}(1-p_1)^{m-l+l_0-i}z^m\\ &= u^l\sum_{l_0=1}^{l-1}\sum_{m\ge l}\sum_{i=0}^{m-l}\binom{i+l_0-1}{l_0-1}\big(\delta p_0\big)^{l_0}(1-p_0)^{i}\binom{m-i-l_0-1}{l-l_0-1}\big(\delta p_1\big)^{l-l_0}(1-p_1)^{m-l-i}z^m\\ &= u^lz^l\sum_{l_0=1}^{l-1}\sum_{m\ge 0}\sum_{i=0}^{m}\binom{i+l_0-1}{l_0-1}\big(\delta p_0\big)^{l_0}(1-p_0)^{i}\binom{m-i+l-l_0-1}{l-l_0-1}\big(\delta p_1\big)^{l-l_0}(1-p_1)^{m-i}z^m\\ &= u^lz^l\sum_{l_0=1}^{l-1}\big(\delta p_0\big)^{l_0}\big(\delta p_1\big)^{l-l_0}\sum_{m\ge 0}\sum_{i=0}^{m}\binom{i+l_0-1}{l_0-1}\big((1-p_0)z\big)^{i}\binom{m-i+l-l_0-1}{l-l_0-1}\big((1-p_1)z\big)^{m-i}\\ &= u^lz^l\sum_{l_0=1}^{l-1}\big(\delta p_0\big)^{l_0}\big(\delta p_1\big)^{l-l_0}\cdot\frac{1}{\big(1-(1-p_0)z\big)^{l_0}}\cdot\frac{1}{\big(1-(1-p_1)z\big)^{l-l_0}}\end{aligned}$$

$$
\begin{aligned}
&= u^l z^l \frac{\delta p_0}{1-(1-p_0)z} \cdot \frac{\delta p_1}{1-(1-p_1)z} \sum_{l_0=0}^{l-2} \frac{\left(\delta p_0\right)^{l_0}}{\left(1-(1-p_0)z\right)^{l_0}} \cdot \frac{\left(\delta p_1\right)^{l-2-l_0}}{\left(1-(1-p_1)z\right)^{l-2-l_0}} \\
&= u^l z^l \frac{\delta p_0}{1-(1-p_0)z} \cdot \frac{\delta p_1}{1-(1-p_1)z} \frac{1}{\frac{\delta p_0}{1-(1-p_0)z} - \frac{\delta p_1}{1-(1-p_1)z}} \left(\left(\frac{\delta p_0}{1-(1-p_0)z} \right)^{l-1} - \left(\frac{\delta p_1}{1-(1-p_1)z} \right)^{l-1} \right) \\
&= \delta \frac{p_0 p_1}{p_0 - p_1} \frac{uz}{1-z} \left(\left(\frac{\delta p_0 uz}{1-(1-p_0)z} \right)^{l-1} - \left(\frac{\delta p_1 uz}{1-(1-p_1)z} \right)^{l-1} \right).
\end{aligned}
$$

Now, since $B(z,u) = \sum_{l\geq 2} B^{\langle l\rangle}(z)$, we arrive at

$$
\begin{aligned}
B(z,u) &= \sum_{l\geq 2} B^{\langle l\rangle}(z) \\
&= \delta \frac{p_0 p_1}{p_0 - p_1} \frac{uz}{1-z} \sum_{l\geq 2} \left(\left(\frac{\delta p_0 uz}{1-(1-p_0)z} \right)^{l-1} - \left(\frac{\delta p_1 uz}{1-(1-p_1)z} \right)^{l-1} \right) \\
&= \delta \frac{p_0 p_1}{p_0 - p_1} \frac{uz}{1-z} \sum_{l\geq 0} \left(\left(\frac{\delta p_0 uz}{1-(1-p_0)z} \right)^{l} - \left(\frac{\delta p_1 uz}{1-(1-p_1)z} \right)^{l} \right) \\
&= \delta \frac{p_0 p_1}{p_0 - p_1} \frac{uz}{1-z} \left(\frac{1}{1 - \frac{\delta p_0 uz}{1-(1-p_0)z}} - \frac{1}{1 - \frac{\delta p_1 uz}{1-(1-p_1)z}} \right)
\end{aligned}
$$

Now, the sought-after univariate OGF equals the counting version of the above BGF, i.e., $A(z) = B(z,1)$. Resubstituting $1-\varepsilon$ for δ, we get

$$
\begin{aligned}
A(z) = B(z,1) &= \delta z \frac{p_0 p_1}{(p_0 - p_1)(1-z)} \left(\frac{1-(1-p_0)z}{1-(1-\varepsilon p_0)z} - \frac{1-(1-p_1)z}{1-(1-\varepsilon p_1)z} \right) \\
&= \delta z \frac{p_0 p_1}{p_0 - p_1} \cdot \left(\frac{(1-\varepsilon)p_0 z - (1-\varepsilon)p_1 z}{\left(1-(1-\varepsilon p_0)z\right)\left(1-(1-\varepsilon p_1)z\right)} \right) \\
&= (1-\varepsilon) z^2 \frac{p_0 p_1}{p_0 - p_1} \cdot \left(\frac{1-(1-\varepsilon)p_1 + \varepsilon^2}{1-(1-\varepsilon p_1)z} - \frac{1-(1-\varepsilon)p_0 + \varepsilon^2}{1-(1-\varepsilon p_0)z} \right).
\end{aligned}
$$

This ends the proof □

W.l.o.g., assume that $p_0 > p_1$. We get a sufficiently exact bound on the growth rate of $a_m = [z^m]A(z)$, where $[z^m]A(z)$ denotes the coefficient of z^m in $A(z)$. Thus

$$
\begin{aligned}
a_m = [z^m]A(z) &= (1-\varepsilon) \frac{p_0 p_1}{p_0 - p_1} \left(\frac{(1-\varepsilon p_1)^{m-2}}{(1-(1-\varepsilon)p_1 + \varepsilon^2)^{-1}} + \frac{(1-\varepsilon p_0)^{m-2}}{(1-(1-\varepsilon)p_0 + \varepsilon^2)^{-1}} \right) \\
&\leq 2 \cdot \frac{(1-\varepsilon) p_0 p_1}{(p_0 - p_1)(1-\varepsilon p_1)^2} \cdot (1-\varepsilon p_1)^m. \qquad (9)
\end{aligned}
$$

Now, we are ready to formally prove the theorem.

4.3 Composing the big Picture

In the following we complete the proof of Theorem 8.

Proof. We claim that $(1) \Rightarrow (2)$, $(2) \Rightarrow (3)$ and $(3) \Rightarrow (1)$. This proves the theorem. For the individual claims:

- $(1) \Rightarrow (2)$: We show directly that (1) implies (2). To this aim assume that (1) holds for $\varepsilon > 0$ and, without loss of generality, let $p_0 > p_1$. Then, for all input strings $t \in \{0,1\}^\omega$ it holds that

$$\begin{aligned}
&\sum_{|\alpha|=m} \mathbf{P}\big(\alpha \sqsubseteq X(t)\big) \\
&\quad \leq c_9 \cdot m \cdot \sum_{|\alpha|=m} \sum_{l=1}^{m} \rho(\alpha, t, 1, l)^2 && \text{(by Lemma 9)} \\
&\quad \leq \frac{c_9}{1-\varepsilon} \cdot m \cdot \sum_{|\alpha|=m} \sum_{l=1}^{m} (1-\varepsilon)^l \rho(\alpha, t, 1, l) && \text{(by Lemma 10)} \\
&\quad \leq \frac{c_9}{1-\varepsilon} \cdot m \cdot \Big((1-\varepsilon p_0)^{m-1} + (1-\varepsilon p_1)^{m-1} + [z^m]A(z)\Big) \\
&\qquad && \text{(by Equation (8) and Lemma 11)} \\
&\quad \leq 2\frac{c_9 p_0 p_1}{(p_0 - p_1)(1-\varepsilon p_1)^2} \cdot m \cdot (1-\varepsilon p_1)^m && \text{(by Equation (9))}
\end{aligned}$$

  Here, c_9 denotes the constant from Lemma 9. Thus, the claim follows.

- $(2) \Rightarrow (3)$: follows directly from Lemma 3, where $\gamma = 1 - \varepsilon p_1$, $c = 0$, and $d = 2 \cdot c_9 \cdot p_0 \cdot p_1 \cdot (p_0 - p_1)^{-1}(1 - \varepsilon p_1)^{-2}$ as given above.

- $(3) \Rightarrow (1)$: We show the contraposition. To this end, assume that there exists a symbol $a \in \{0,1\}$ and a writing state $q \in W$ such that $\mu_W(q,a,q) + \mu_W(q,a,s) = 1$. Since X is read-deterministic there must be some symbol $b \in \{0,1\}$ such that $\mu_R(s,b,q) = 1$. But then, X maps the string $t = bbb\ldots$ deterministically to the string $X(t) = aaa\ldots$. Hence, the claim follows.

□

5 Conclusion

Smoothed analysis of trie height and related parameters is a further step in the quest for rigorous mathematical arguments for explaining observed data. For instance, suffi x trees (or tries) built over human DNA sequences have logarithmic height. These sequences are in particular non-random; huge stretches of the DNA are highly repetitive. However, it seems that small mutations, i.e., random perturbations, in such sequences are enough to smooth out the worst case, i.e., linear tree height. Thus, future work might be directed to extending the analysis given in this paper to tries built over the set of suffi xes of a string.

Technically, the counter-example for the case of random perturbations gives rise to two different directions for further research of smoothed trie height under star-like perturbation functions. On the one hand, one might argue, that the counter-example is quite artificial and thus might search for subsets $S \subseteq \{0,1\}^\omega$, such that $H(S, n, \mathrm{DEL}_p)$ is logarithmic. Clearly such sets S must satisfy $t, s \notin S$ for the bad strings $s = 0^\omega$ and $t = 1^\omega$, but it is not totally clear, if there are more bad strings. One the other hand, one might ask, if there are conditions for general, i.e., non-read-deterministic, star-like perturbation functions such that the smoothed trie height becomes logarithmic. We consider this the main open question posed by this paper.

References

[1] Heiner Ackermann, Alantha Newman, Heiko Röglin, and Berthold Vöcking. Decision making based on approximate and smoothed pareto curves. In *Proceedings of the 16th International Symposium on Algorithms and Computation (ISAAC'05), Porceedings*, volume 3827 of *Lecture Notes in Computer Science*, pages 675–684, 2005.

[2] Noga Alon and Joel. H. Spencer. *The Probabilistice Method, Second Edition*. Wiley-Interscience Series in Discrete Mathematics and Optimization. John Wiley, New York, NY, 2000.

[3] David Arthur and Sergei Vassilvitskii. Worst-case and smoothed analysis of the icp algorithm, with an application to the k-means method. In *Proceedings of the 47rd Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, pages 153–164. IEEE Computer Society, 2006.

[4] Cyril Banderier, René Beier, and Kurt Mehlhorn. Smoothed analysis of three combinatorial problems. In *Proceedings of the 28th International Symposium on Mathematical Foundations of Computer Science*, 2003.

[5] Vikas Bansal and Friedhelm Meyer auf der Heide und Christian Sohler. Labeling smart dust. In *Proceedings of the 12th European Symposium on Algorithms (ESA'04)*, volume 3221 of *Lecture Notes in Computer Science*, pages 77–88. Springer-Verlag, Berlin, 2004.

[6] Luca Becchetti, Stefano Leonardi, Alberto Marchetti-Spaccamela, Guido Schäfer, and Tjark Vredeveld. Average case and smoothed competitive analysis of the multi-level feedback algorithm. In *Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science (FOCS'03)*. IEEE Computer Society, 2003.

[7] René Beier and Berthold Vöcking. Random knapsack in expected polynomial time. In *Proceedings of the 35th Annual ACM Symposium on Theory of Computing (STOC'03)*, pages 232–241. ACM, 2003.

[8] René Beier and Berthold Vöcking. Typical properties of winners and losers in discrete optimization. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing (STOC'04)*, pages 343–352. ACM, 2004.

[9] René Beier and Berthold Vöcking. Typical properties of winners and losers in dicrete optimization. *SIAM Journal on Computing*, 34(4):855–881, 2006.

[10] Avrim Blum and John Dunagan. Smoothed analysis of the perceptron algorithm for linear programming. In *Proceedings of the 13th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'02)*, pages 905 – 914. ACM Press, New York, NY, 2002.

[11] Avrim Blum and Joel Spencer. Coloring random and semi-random k-colorable graphs. *Journal of Algorithms*, 19:204–234, 1995.

[12] Peter Bürgisser, Felipe Cucker, and Martin Lotz. General formulas for the smoothed analysis of condition numbers. *C. R. Acad. Sci. Paris, Ser. I*, 343:145–150, 2006.

[13] Julien Clément, Philippe Flajolet, and Brigitte Vallé. Dynamical sources in information theory: A general analysis of trie structures. *Algorithmica*, 29(1-2):307–369, 2001.

[14] Valentina Damerow, Friedhelm Meyer auf der Heide, Harald R¨acke, Christian Scheideler, and Christian Sohler. Smoothed motion complexity. In *Proceedings of the 11th European Symposium on Algorithms (ESA'03)*, volume 2832 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin, 2003.

[15] Valentina Damerow and Christian Sohler. Extreme points under random noise. In *Proceedings of the 12th European Symposium on Algorithms (ESA'04)*, volume 3221 of *Lecture Notes in Computer Science*. Springer-Verlag, Berlin, 2004.

[16] Luc Devroye. A study of trie-like structures under the density model. *Annals of Applied Probability*, 2:402–434, 1992.

[17] Luc Devroye. Universal asymptotics for random tries and PATRICIA trees. *Algorithmica*, 42:11–29, 2005.

[18] Stefan Eckhardt, Sven Kosub, and Johannes Nowak. Smoothed analysis of trie height. Technical Report TUM-I0715, Technische Universi¨at M¨unchen, 2007.

[19] Uriel Feige and Joe Kilian. Heuristice for semirandom graph problems. *Journal of Computer and System Sciences*, 63:639–671, 2001.

[20] Philippe Flajolet. The ubiquitous digital tree. In *Proceedings of the 23rd International Symposium on Theoretical Aspects of Computer Science (STACS'06)*, volume 3884 of *Lecture Notes in Computer Science,* pages 1–22. Springer-Verlag, Berlin, 2006.

[21] Philippe Flajolet and Robert Sedgewick. *Analytic Combinatorics*. Web edition, 9th edition, 2007.

[22] Edward Fredkin. Trie memory. *Communication of the ACM*, 3(9), 1960.

[23] Donald E. Knuth. *The Art of Computer Programming, Vol 3 : Searching and Sorting, Second Edition*. Addison-Wesley Publishing Co., Reading, MA, 1998.

[24] Ming Li and Paul M. B. Vitányi. *An Introduction to Kolmogoriv Complexity and Its Applications*. Graduate Texts in Computer Science. Springer-Verlag, New York, NY, 2nd edition, 1997.

[25] M. Lothaire, editor. *Combinatorics on Words,* volume 17 of *Encyclopedia of Mathematics and Its Applications*. Cambridge University Press, Cambridge, 2nd edition, 2003.

[26] Bodo Manthey and R¨udiger Reischuk. Smoothed analysis of binary search trees. In *Proceedings of the 16th International Symposium on Algorithms and Computation (ISAAC'05), Porceedings*, volume 3827 of *Lecture Notes in Computer Science*, pages 483–492. Springer-Verlag, Berlin, 2005.

[27] Stefan Nilsson and Matti Tikkanen. An experimental study of compression methods for dynamic tries. *Algorithmica*, 33(1):19–33, 2002.

[28] Azaria Paz. *Introduction to Probabilistic Automata*. Computer Science and Applied Mathematics. A Series of Monographs and Textbooks. Academic Press, 1971.

[29] Michael O. Rabin. Probabilistic automata. *Information and Control*, 6(3):230–245, 1963.

[30] Heiko Röglin and Berthold Vöcking. Smoothed analysis of integer programming. In *Proceedings of the 11th Conference on Integer Programming and Combinatorial Optimization (IPCO'05)*, volume 3509 of *Lecture Notes in Computer Science*, pages 276–290. Springer-Verlag, Berlin, 2005.

[31] Heiko Röglin and Berthold Vöcking. Smoothed analysis of integer programming. *Mathematical Programming*, 110(1):21–56, 2007.

[32] Jacque Sakarovitch and Imre Simon. *Combinatorics on Words*, chapter 6, pages 121–134. Cambridge University Press, Cambridge, 1997.

[33] Arvind Sankar, Daniel A. Spielman, and Shang-Hua Teng. Smoothed analysis of the condition numbers and growth factors of matrices. *SIAM Journal on Matrix Analysis and Applications*, 28(2):446–476, 2006.

[34] Miklos Santha and Umesh V. Vazirani. Generating quasi-random sequences from semi-random sources. *Journal of Computer and System Sciences*, 33:75–87, 1986.

[35] Guido Schäfer and Naveen Sivadasan. Topology matters: Smoothed competitiveness of metrical task systems. *Theoretical Computer Science*, 341(1-3):216–246, 2005.

[36] Daniel A. Spielman and Shang-Hua Teng. Smoothed analysis of algorithms: why the simplex algorithm usually takes polynomial time. In *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing (STOC'01)*, pages 296–305. ACM, 2001.

[37] Daniel A. Spielman and Shang-Hua Teng. Smoothed analysis (motivation and discrete models). In *Proceedings of the 8th International Workshop on Algorithms and Data Structures (WADS'03)*, volume 2748 of *Lecture Notes in Computer Science*, pages 256–270. Springer-Verlag, Berlin, 2003.

[38] Daniel A. Spielman and Shang-Hua Teng. Smoothed analysis of termination of linear programming algorithms. *Mathematical Programming, Series B*, 97(1-2):237–404, 2003.

[39] Daniel A. Spielman and Shang-Hua Teng. Smoothed analysis of algorithms: why the simplex algorithm usually takes polynomial time. *Journal of the ACM*, 2004.

[40] Wojciech Szpankowski. *Average Case Analysis of Algorithms on Sequences*. Wiley-Interscience Series in Discrete Mathematics and Optimization. John Wiley, New York, NY, 2001.

[41] Brigitte Vallé. Dynamical sources in information theory : fundamental intervals and word prefixes. *Algorithmica*, 29(1-2):262–306, 2001.

[42] Roman Vershynin. Beyond hirsch conjecture: Walks on random polytopes and smoothed complexity of the simplex method. In *Proceedings of the 47rd Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, pages 133–142. IEEE Computer Society Press, Washington, D.C., 2006.

[43] Herbert S. Wilf. *Generatingfunctionology*. Academic Press, 2nd edition, 1994.