A GENERALIZED SUFFIX TREE AND ITS (UN)EXPECTED ASYMPTOTIC BEHAVIORS*

June 17, 1996

Wojciech Szpankowski†
Department of Computer Science
Purdue University
W. Lafayette, IN 47907
U.S.A.

Abstract

Suffix trees find several applications in computer science and telecommunications, most notably in algorithms on strings, data compressions and codes. Despite this, very little is known about their typical behaviors. In a probabilistic framework, we consider a family of suffix trees – further called b-suffix trees – built from the first n suffixes of a random word. In this family a noncompact suffix tree (i.e., such that every edge is labeled by a single symbol) is represented by $b = 1$, and a compact suffix tree (i.e., without unary nodes) is asymptotically equivalent to $b \to \infty$ as $n \to \infty$. We study several parameters of b-suffix trees, namely: the depth of a given suffix, the depth of insertion, the height and the shortest feasible path. Some new results concerning typical (i.e., *almost sure*) behaviors of these parameters are established. These findings are used to obtain several insights into certain algorithms on words, molecular biology and universal data compression schemes.

Key Words: generalized suffix trees, algorithms on words, data compression, hight, shortest path, typical depth and depth of insertion, probabilistic models, mixing condition, Rènyi's entropy.

AMS(MOS) Subject Classifications: 68Q25, 68P05.

*A preliminary version of this paper was presented at the *Combinatorial Pattern Matching*, Tucson, 1992.

†This research was supported in part by NSF Grants CCR-8900305, CCR-9201078, and NCR-9206315 and INT-8912631, in part by AFOSR Grant 90-0107, NATO Grant 0057/89, and Grant R01 LM05118 from the National Library of Medicine.

1. INTRODUCTION

In recent years there has been a resurgence of interest in algorithmic and combinatorial problems on words due to a number of novel applications in computer science, telecommunications, and most notably in molecular biology (cf. [40]). In computer science and molecular biology many algorithms depend on a solution to the following problem: given a word X and a set of arbitrary $b+1$ suffixes $S_1, \ldots, S_{b+1}$ of X, what is the longest common prefix of these suffixes (cf. [2], [3], [9], [12], [42]). In coding theory (e.g., prefix codes) one asks for the shortest prefix of a suffix S_i which is not a prefix of any other suffixes S_j, $1 \leq j \leq n$ of a given sequence X (cf. [34]). In data compression schemes, the following problem is of prime interest: for a given "data base" subsequence of length n, find the longest prefix of the $(n+1)$st suffix S_{n+1} which is not a prefix of any other suffixes S_i $(1 \leq i \leq n)$ of the data base sequence (cf. [25], [43], [44]). And last but not least, in molecular sequences comparison (e.g., finding homology between DNA sequences), one may search for the longest run of a given motif (cf. [16], [17], [40]), a unique sequence or the longest alignment (cf. [13], [40]). These, and several other problems on words, can be efficiently solved and analyzed by a clever manipulation of a data structure known as a *suffix tree* (cf. [2], [27], [41]). In literature other names have been also coined for this structure, and among these we mention here position trees, subword trees, directed acyclic graphs, etc. (cf. [1]).

Suffix trees find a wide variety of applications in algorithms on words including: the longest repeated substring (cf. [41]), squares or repetitions in strings (cf. [3]), string statistics (cf. [3]), string matching (cf. [9], [42]), approximate string matching (cf. [12], [9], [42]) string comparison, compression schemes (cf. [25]), implementation of the Lempel-Ziv algorithm, genetic sequences, biologically significant motif patterns in DNA (cf. [9], [40]), sequence assembly (cf. [9]), approximate-overlaps (cf. [9], [40]), and so forth. It is fair to say that suffix trees are the most widely used data structure in algorithms on words. Despite this, very little is known about their behaviors in a probabilistic framework. Recently, Chang and Lawler (cf. [9]) used some elementary property of suffix trees to design a superfast algorithm for the approximate string matching problem. In our opinion, any further development in this direction requires better understanding of the behavior of suffix trees in a probabilistic framework.

In general, a suffix tree is a digital tree built from suffixes of a given word X, and therefore it fits into the subject of digital search indexes (cf. [23]). A digital tree stores n strings $\{S_1, \ldots, S_n\}$ built over a finite alphabet Σ. In such a tree, every edge is labeled by a symbol (or a set of symbols) from the alphabet Σ and leaves (called also external nodes) contain the strings. The access path from the root to the external node is a minimal prefix information contained in the leaf (for more details see [14], [23]). If the strings $\{S_1, \ldots, S_n\}$ are statistically *independent* and every edge is labeled by a *single* symbol from Σ, then the resulting digital tree is called a regular (or independent) trie (cf. [1], [14], [23]). If all unary nodes of a trie are eliminated, then the tree becomes a PATRICIA trie (cf. [14], [23], [37]). Finally, if an external node in a regular trie can store up to b strings (keys), then such a tree is called a b-trie. As mentioned above, a suffix tree is a special trie in which the strings $\{S_1, \ldots, S_n\}$ are suffixes of a given sequence X. Note that in this case the strings are statistically dependent!

As in the case of regular tries, there are several modifications of the standard suffix tree. In a *noncompact suffix tree* – called also spread suffix tree and position tree – each edge is labeled by a letter from the alphabet Σ. If all unary nodes are eliminated in the noncompact version of the suffix tree, then the resulting tree is called *compact suffix tree* (cf. [2]). Gonnet and Baeza-Yates [14] coined a name PAT for such a suffix tree to resemble the name PATRICIA used for compact tries. Hereafter, we adopt this notation.

In this paper, we additionally introduce a family of suffix trees parametrized by an integer $b \geq 1$ such that $b = 1$ corresponds to a noncompact suffix tree and $b \to \infty$ is asymptotically equivalent (as $n \to \infty$) to PAT. A tree in such a family is constructed from the noncompact suffix tree by eliminating all unary nodes b levels above the fringe (bottom) of the tree. To simplify analysis, however, we shall modify this definition and assume that external nodes of b-suffix trees can store up to b suffixes. Note that such a suffix tree corresponds to a b-trie. Therefore, we coin a term b-suffix trees for them. These trees are useful in several applications, but more importantly b-suffix trees form a spectrum of trees with noncompact suffix trees ($b = 1$) at one extreme and compact suffix trees ($b \to \infty$ as $n \to \infty$) at the other extreme (cf. Figure 1). This allows us to assess some properties of PAT trees in a unified and substantially easier manner (e.g., compare [37] where PATRICIA tries are analyzed).

We offer a characterization of b-suffix trees in probabilistic framework, namely a word X over which the suffix tree is built represents a *stationary mixing (ergodic) sequence.* This sequence is assumed to be of infinite length (for bounded words see Remark 2(iv) in Section 2). We shall analyze the following parameters of b-suffix trees: the typical depth $D_n^{(b)}$, the depth of a particular suffix, say mth one, $L_n^{(b)}(m)$, the depth of insertion $L_n^{(b)}$, height $H_n^{(b)}$ and the shortest feasible path $s_n^{(b)}$. The typical depth represents the length of a path from the root to a *randomly* selected external node in a suffix tree; the depth of insertion is the depth of a *newly* inserted suffix; the height and the shortest feasible path are the longest and shortest path to an available node.

These parameters are most often used in the analysis and design of algorithms on words. For example, the typical depth D_n^{PAT} for the PAT tree built from the string $P\$T$ where P and T are the pattern and the text respectively, is used by Chang and Lawler [9] in their design of an approximate string matching algorithm. On the other hand, the depth of insertion $L_n^{(1)}$ and the depth of a given suffix $L_n^{(1)}(m)$ of a noncompact suffix tree are of prime interest to the complexity of the Lempel-Ziv universal compression schemes (cf. [15], [25], [43], [44]), and $L_n^{(1)}$ is responsible for a *dynamic* behavior of many algorithms on words. Furthermore, the height and the shortest path indicate how balanced a typical suffix tree is; that is, how much one has to worry about worst-case situations.

Our main results can be summarized as follows. For a b-suffix tree built over an unbounded word X, we prove that the normalized height $H_n^{(b)}/\log n$, the normalized shortest feasible path $s_n^{(b)}/\log n$, and the normalized depth of the m-th suffix (m fixed) $L_n^{(b)}(m)/\log n$. almost surely (a.s.) converge to some explicit constants that depend on characteristics of the underlying probabilistic model. The most interesting behavior reveals that the normalized depth of insertion $L_n^{(b)}/\log n$ converges *in probability* (pr.) to a constant but *not* almost surely (a similar behavior shows the typical depth $D_n^{(b)}$). More interestingly, the almost sure behavior of a compact suffix tree can be deduced from the appropriate asymptotics of b-suffix trees by taking $b \to \infty$ as $n \to \infty$. More precisely, if we append superindex PAT to the appropriate parameters of a compact suffix tree, then we can prove that $\lim_{n\to\infty} H_n^{PAT}/\log n = \lim_{b\to\infty} \lim_{n\to\infty} H_n^{(b)}/\log n$, and in a similar fashion for $s_n^{(b)}$, $D_n^{(b)}$ and $L_n^{(b)}$. Note that the iterative limit above cannot be interchanged. Indeed, for example $\lim_{n\to\infty} \lim_{b\to\infty} H_n^{(b)} = 1$ It is worth mentioning that all these results are obtained in a uniform manner by a technique that encompasses the so called string-ruler approach (cf. [19], [30]) and mixing condition technique. Our method, however, parallels on several instances Pittel's profound analysis of independent tries, and this research was inspired by the seminal paper of Pittel [30]. The details are discussed in Section 3.

Asymptotic analyses of suffix trees are very scanty in literature, and most of them deal with noncompact suffix trees, i.e., $b = 1$. To the best of our knowledge, there are no probabilistic results on b-suffix trees ($b > 1$) and compact suffix trees. This can be easily verified by checking Section 7.2 of Gonnet and Baeza-Yates' book [14] which provides an up-to-date compendium of results concerning data structures and algorithms. The average case analysis of noncompact suffix trees was initialized by Grassberger [15], and Apostolico and Szpankowski [4]. For the Bernoulli model (independent sequence of letters from a finite alphabet) the asymptotic behavior of the height was recently obtained by Devroye *et al.* [11], and the limiting distribution of the typical depth in a suffix tree is reported in Jacquet and Szpankowski

[19]. Szpankowski [38] extended some these results to a more general probabilistic model (still for $b = 1$). Heuristic arguments were used by Blumer *et al.* [6] to show that the average number of internal nodes in a suffix tree is a linear function of n, and a rigorous proof of this can be found in [19]. Finally, Shields [34] recently established the almost sure behavior of the external path length of a noncompact suffix tree in the Bernoulli model and the Markovian model. Some related topics were discussed by Guibas and Odlyzko in [16] and [17].

This paper is organized as follows. In the next section we formulate our main results and present several consequences of them. In particular, we intuitively explain why compact suffix trees can be considered as limiting b-suffix trees when $n \to \infty$. We also provide four applications of our results to data compression and pattern matching problems. Namely: (i) we settle two conjectures of Wyner and Ziv regarding the almost sure behavior of the repeated pattern and the size of the data base sequence in the universal data compression scheme (cf. [38]); (ii) we provide some information concerning the almost sure behavior of the block length in the Lempel-Ziv parsing algorithm (cf. [25], [44]); (iii) we present some complexity results regarding the Chang-Lawler pattern matching algorithm (cf. [9]); and finally (iv) we estimate the typical length of a unique subsequence of a given sequence (cf. [13]). Finally, Section 3 contains all formal proofs, and presents some new results of combinatorics on words.

2. MAIN RESULTS AND THEIR CONSEQUENCES

In this section we formally define b-suffix trees and introduce several parameters of these trees that are widely used in the complexity analysis of algorithms on words and data compression schemes. Next, we present all of our main results. We delay most of the proofs to the next section. Finally, we discuss some consequences of our findings.

2.1 Definitions and Probabilistic Models

A suffix tree is a trie built from suffixes of an (unbounded) sequence $\{X_k\}_{k=1}^{\infty}$ of symbols from an alphabet Σ of size V. More precisely, let $X = x_1 x_2 x_3 \ldots$, then the ith suffix S_i of X is $S_i = x_i x_{i+1} \ldots$. By $\mathcal{S}_n$ we denote a digital tree built from the set $\{S_1, S_2, \ldots, S_n\}$ of the first n suffixes of X. In such a tree – which we further call a *noncompact suffix tree* – every edge is labeled by a single symbol from the alphabet Σ. Figure 1(a) shows a noncompact suffix tree built from the first six suffixes of $X = 0101101110\ldots$. A *compact suffix tree* called PAT tree (cf. [14]) is constructed from the noncompact version by eliminating all unary nodes (cf. Fig. 1(d)). It is characterized by the fact that an edge in such a tree is labeled by a substring of X (cf. [2], [27], [41]).

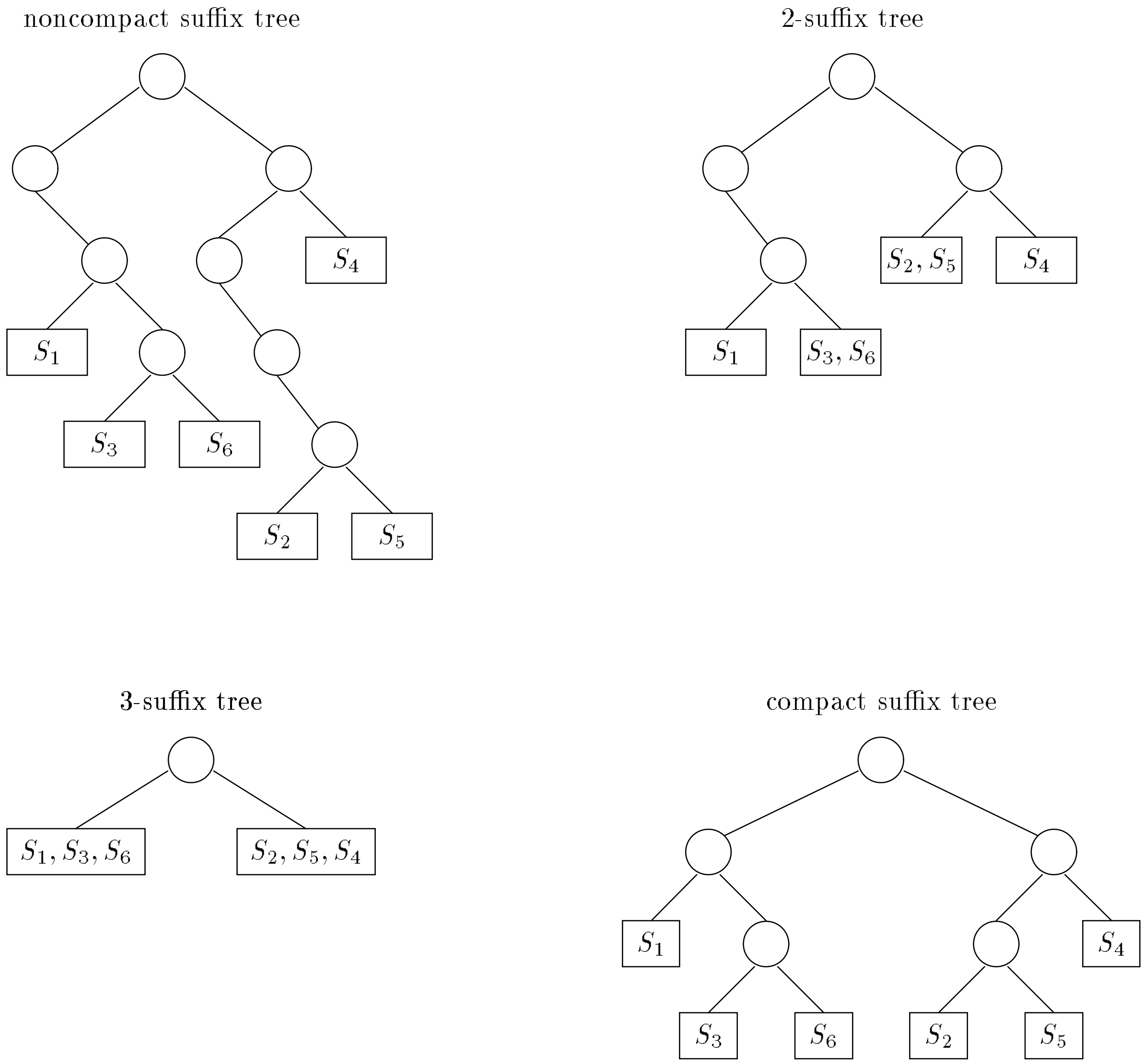

Figure 1: Suffix trees built from the first six suffixes of = $0101101110\cdots$

In this paper, we consider a family of suffix trees called b-suffix trees. A tree in such a family has no unary nodes in all b levels above the fringe level of the corresponding noncompact suffix tree. Note that a noncompact suffix tree coincides with 1-suffix tree, and a compact suffix tree corresponds to $b \to \infty$ as $n \to \infty$. For the purpose of our analysis, however, a modified definition of b-suffix trees is more convenient. Hereafter, *by a b-suffix tree we mean a trie built from the first n suffixes of X that can store up to b suffixes in an external node.* We denote such a suffix tree as $\mathcal{S}_n^{(b)}$. This definition is illustrated in Figure 1(b) and 1(c). It is easy to note that if in a b-suffix tree we replace every external node by a complete binary tree with b nodes, then the latter definition of b-suffix tree corresponds to the former one.

Hereafter, we analyze several parameters of b-suffix trees that are formally defined below. The relevance of these parameters to the analysis and design of algorithms on words and data compression schemes was already discussed in the Introduction.

Definition 1. TREES PARAMETERS

(i) The *mth depth* $L_n^{(b)}(m)$ is the length of a path from the root of the b-suffix tree $\mathcal{S}_n^{(b)}$ to the external node containing the mth suffix.

(ii) The *typical depth* $D_n^{(b)}$ is the depth of a randomly selected suffix, that is,

$$\Pr\{D_n^{(b)} \le k\} = \frac{1}{n} \sum_{m=1}^{n} \Pr\{L_n^{(b)}(m) \le k\} \; . \tag{2.1}$$

(iii) The *height* $H_n^{(b)}$ is the length of the longest depth, that is,

$$H_n^{(b)} = \max_{1 \le m \le n} \{L_n^{(b)}(m)\} \; . \tag{2.2}$$

(iv) The *shortest feasible path* $s_n^{(b)}$ is the length of the shortest path from the root to an *available* (feasible) node, that is, a node that is *not* in the tree $\mathcal{S}_n^{(b)}$ but whose predecessor node (either an internal node or an external node) is in $\mathcal{S}_n^{(b)}$. In other words, $s_n^{(b)}$ is the shortest path to all external nodes and all internal nodes that have degree *smaller* than the size V of the underlying alphabet.

(v) The *shortest depth* $\tilde{s}_n^{(b)}$ is the length of the shortest path from the root to an *external node*. (Clearly, $s_n^{(b)} \le \tilde{s}_n^{(b)}$.) $\square$

(Another parameter of interest not studied in this paper is the external path length $E_n^{(b)}$ which is the sum of all depths, that is, $E_n^{(b)} = \sum_{m=1}^{n} L_n^{(b)}(m)$.)

For the purpose of this analysis, we present below another representation of the above tree parameters. We start with the following definition.

Definition 2. SELF-ALIGNMENTS

For suffixes $S_1, S_2, ..., S_n$, the *self-alignment* $C_{i_1...i_{b+1}}$ between $b+1$ suffixes, say $S_{i_1}, \ldots, S_{i_{b+1}}$, is the length of the longest common prefix of all these $b+1$ suffixes. $\Box$

Then, the following relationships are easy to establish (cf. [4], [36])

$$L_n^{(b)}(i_{b+1}) = \max_{1 \le i_1, \ldots, i_b \le n} \{C_{i_1...i_{b+1}}\} + 1 \tag{2.3a}$$

$$H_n^{(b)} = \max_{1 \le i_1, \ldots, i_{b+1} \le n} \{C_{i_1...i_{b+1}}\} + 1 \ , \tag{2.3b}$$

$$L_n^{(b)} = \max_{1 \le i_1, \ldots, i_b \le n} \{C_{i_1...i_b, n+1}\} + 1 \ . \tag{2.3c}$$

$$\widetilde{s}_n^{(b)} = \min_{1 \le i_{b+1} \le n} \{L_n^{(b)}(i_{b+1})\} = \min_{1 \le i_{b+1} \le n} \max_{1 \le i_1, \ldots, i_b \le n} \{C_{i_1...i_{b+1}}\} + 1 \tag{2.3d}$$

In passing, we note that for a stationary (infinite) ergodic sequence $\{X_k\}$, the self-alignment $C_{i_1...i_{b+1}}$ does not depend explicitly on $i_1, \ldots, i_{b+1}$ but rather on the differences $d_k = i_{k+1} - i_k$. So, we also write $C_{1,1+d_1,\ldots,1+d_1+\cdots+d_b}$.

Our plan is to investigate the behavior of a random b-suffix tree in as general probabilistic framework as possible. We could only assume that $\{X_k\}_{k=1}^{\infty}$ is a *stationary ergodic* sequence of symbols generated from a finite alphabet Σ, but this is too strong for our purpose. Therefore, we adopt the following two weaker probabilistic models.

(A1) MIXING MODEL

The sequence $\{X_k\}_{k=-\infty}^{\infty}$ satisfies the so called mixing condition [5], that is, there exist two constants $0 < c_1 \le c_2$ and an integer d such that for all $-\infty \le m \le m + d \le n$ the following holds

$$c_1 \Pr\{\mathcal{A}\}\Pr\{\mathcal{B}\} \le \Pr\{\mathcal{A}\mathcal{B}\} \le c_2 \Pr\{\mathcal{A}\}\Pr\{\mathcal{B}\} \tag{2.4a}$$

with $\mathcal{A} \in \mathcal{F}_{-\infty}^{m}$ and $\mathcal{B} \in \mathcal{F}_{m+d}^{\infty}$ where $\mathcal{F}_m^n$ is a σ-field generated by $\{X_k\}_{k=m}^{n}$ for $m \le n$.

In some statements of our results, we need a stronger form of the above mixing condition which are defined in sequel.

(A2) STRONG MIXING MODEL

The sequence $\{X_k\}_{k=-\infty}^{\infty}$ satisfies the so called *strong α-mixing condition* if (2.4a) is replaced by

$$(1 - \alpha(d))\Pr\{\mathcal{A}\}\Pr\{\mathcal{B}\} \le \Pr\{\mathcal{A}\mathcal{B}\} \le (1 + \alpha(d))\Pr\{\mathcal{A}\}\Pr\{\mathcal{B}\} \tag{2.4b}$$

where the function $\alpha(d)$ is such that $\alpha(d) \to 0$ as $d \to \infty$.

In words, model (A1) says that the dependency between $\{X_k\}_{k=-\infty}^{m}$ and $\{X_k\}_{k=m+d}^{\infty}$ is rather weak (note that when the sequence $\{X_k\}$ is i.i.d., then $\Pr\{\mathcal{AB}\} = \Pr\{\mathcal{A}\}\Pr\{\mathcal{B}\}$). Assumption (A2) says that this dependency is weaker and weaker as d becomes larger. The "quantity" of dependency is characterized by $\alpha(d)$.

Finally, for compact suffix trees (i.e., PAT trees) we need one more assumption which strengthens (A2).

(P) Contractive Mixing Model

Let $\omega_i \in \Sigma$ for $1 \le i \le n$, and define $P(\omega_1, \ldots, \omega_n) = \Pr\{X_1 = \omega_1, \ldots, X_n = \omega_n)\}$. Then, for PAT trees we shall require the following condition

$$P(\omega_1, \ldots, \omega_n) \le \rho P(\omega_1, \ldots, \omega_{n-1}) \tag{2.5}$$

for some $0 < \rho < 1$.

Under (A1) – which is stronger than plain stationarity and ergodicity of $\{X_k\}$ – we can define some parameters needed for the formulation of our results. First of all, let $X_m^n = (X_m, ..., X_n)$ for $m < n$, and let for every $n \ge 1$ the nth order probability distribution for $\{X_k\}$ be

$$P(X_1^n) = \Pr\{X_k = x_k,\ 1 \le k \le n,\ x_k \in \mathcal{A}\}\ .$$

Then, the *entropy* of $\{X_k\}$ is defined in a standard manner as (cf. [5])

$$h = \lim_{n\to\infty} \frac{E \log P^{-1}(X_1^n)}{n}\ .$$

The next three parameters are well defined under our assumption (A1) (cf. [10], [30]).

Definition 3. Renyi's order entropy

For $-\infty \le b \le \infty$, define the bth *order Rényi entropy* as

$$h_2^{(b)} = \lim_{n\to\infty} \frac{\log(E\{P^b(X_1^n)\})^{-1}}{(b+1)n} = \lim_{n\to\infty} \frac{\log\left(\sum_{X_1^n} P^{b+1}(X_1^n)\right)^{-1/(b+1)}}{n}\ . \tag{2.6}$$

In particular, we set

$$h_1 = \lim_{b\to-\infty} h_2^{(b)} \qquad \text{and} \qquad h_3 = \lim_{b\to\infty} h_2^{(b)}\ .$$

Note that by the *inequality on means* [28], we can equivalently express the last two parameters as follows

$$h_1 = \lim_{n\to\infty} \frac{\max\{\log P^{-1}(X_1^n)\ , P(X_1^n) > 0\}}{n} = \lim_{n\to\infty} \frac{\log(1/\min\{P(X_1^n)\ , P(X_1^n) > 0\}}{n}\ , \tag{2.7a}$$

$$h_3 = \lim_{n\to\infty} \frac{\min\{\log P^{-1}(X_1^n)\ , P(X_1^n) > 0\}}{n} = \lim_{n\to\infty} \frac{\log(1/\max\{P(X_1^n)\ , P(X_1^n) > 0\}}{n}\ , \tag{2.7b}$$

as already defined in Pittel [30]. Note also that $h_3 \le h_2^{(b)} \le h \le h_1$. □

Remark 1.

(i) *Bernoulli Model.* In this widely used model (cf. [4], [6], [9], [11], [16], [17], [23], [31], [36], and [37]) symbols from the alphabet Σ are generated independently, that is, $P(X_1^n) = P^n(X_1^1)$. In particular, the ith symbol from the alphabet Σ is generated according to the probability p_i, where $1 \le i \le V$ and $\sum_{i=1}^V p_i = 1$. Then, $h = \sum_{i=1}^V p_i \log p_i^{-1}$ ([5]), and the Rényi entropies become $h_1 = \log(1/p_{min})$, $h_3 = \log(1/p_{max})$, and $h_2^{(b)} = 1/(b+1)\log(1/P_b)$ where $p_{min} = \min_{1\le i\le V}\{p_i\}$, $p_{max} = \max_{1\le i\le V}\{p_i\}$, and $P_b = \sum_{i=1}^V p_i^{b+1}$. The probability P_b can be interpreted as the probability of a match of $b+1$ strings in a given position (cf. [36]).

(ii) *Markovian Model.* In this model (cf. [18], [21], [30], [34]) the sequence $\{X_k\}$ forms a stationary Markov chain, that is, the $(k+1)$st symbol in $\{X_k\}$ depends on the previously selected symbol, and the transition probability becomes $p_{i,j} = \Pr\{X_{k+1} = j \in \Sigma | X_k = i \in \Sigma\}$. Clearly, $P(X_1^n) = P(X_1)\Pr\{X_2|X_1\}\cdots\Pr\{X_n|X_{n-1}\}$. It is well known (cf. [5]) that the entropy h can be computed as $h = -\sum_{i,j=1}^V \pi_i p_{i,j} \log p_{i,j}$ where π_i is the stationary distribution of the Markov chain. The other quantities are a little harder to evaluate. Szpankowski [36], and Pittel [30] for $b = 1$, evaluated the height of a regular tries with Markovian dependency, and show that the bth order Rényi entropy $h_2^{(b)}$ is a function of the largest eigenvalue θ_b of the matrix $\mathbf{P}_{[b+1]} = \mathbf{P}\circ\mathbf{P}\ldots\circ\mathbf{P}$ where $\mathbf{P} = \{p_{i,j}\}_{i,j=1}^V$ is the transition matrix of the underlying Markov chain, and $\circ$ represents the Schur product of $b+1$ matrices $\mathbf{P}$ (i.e., elementwise product). More precisely, $h_2^{(b)} = 1/(b+1)\cdot\log\theta_b^{-1}$. With respect to h_1 and h_3 we need a result from digraphs (cf. Romanovski [33], Karp [22]). Consider a digraph on $\Sigma = \{1,\ldots,V\}$ with weights equal to $-\log p_{ij}$ where $i,j \in \Sigma$. Define a cycle $\mathcal{C} = \{\omega_1, \omega_2, ..., \omega_v, \omega_1\}$ for some $v \le V$ such that $\omega_i \in \Sigma$, and let $\ell(\mathcal{C}) = -\sum_{i=1}^v \log(p_{\omega_i,\omega_{i+1}})$ (with $\omega_{v+1} = \omega_1$) be the total weight of the cycle $\mathcal{C}$. The quantities $\min_{\mathcal{C}}\{\ell(\mathcal{C})/v\}$ and $\max_{\mathcal{C}}\{\ell(\mathcal{C})/v\}$ are known as the minimum and maximum cycle mean, respectively. Karp [22] showed how to compute them efficiently. Clearly, $h_1 = \min_{\mathcal{C}}\{\ell(\mathcal{C})/|\mathcal{C}|\}$ and $h_3 = \max_{\mathcal{C}}\{\ell(\mathcal{C})/|\mathcal{C}|\}$. □

2.2 Formulation of Main Results

Now, we present our first main result concerning the typical height and the shortest path, which is further used to prove our next findings. The proof of the below theorem is delayed till Section 3, except part (ii) regarding PAT trees which is a simple consequence of part (i), and it is proved in Remark 2 (iii) below. For the reader's convenience, we recall that we write $X_n \to \beta$ (pr.) for a sequence of random variables X_n and a constant β if for every $\varepsilon > 0$ the following holds: $\lim_{n\to\infty} \Pr\{|X_n/\beta - 1| > \varepsilon\} = 0$; and similarly $X_n \to \beta$ (a.s.) if for every $\varepsilon > 0$ we have $\lim_{N\to\infty} \Pr\{\sup_{n>N} |X_n/\beta - 1| > \varepsilon\} = 0$. A sufficient condition for the almost sure convergence can be obtained from the Borel-Cantelli Lemma (cf. [5]). In particular, the following suffices for (a.s.): $\sum_{n=0}^{\infty} \Pr\{|X_n/a - 1| > \varepsilon\} < \infty$.

Theorem 1. *Let $\{X_k\}$ be a stationary ergodic sequence satisfying the strong α-mixing condition as in (A2) together with $h_1 < \infty$ and $h_2 > 0$.*

(i) b-Suffix Trees. *Fix b. Then*

$$\lim_{n\to\infty} \frac{s_n^{(b)}}{\log n} = \frac{1}{h_1} \qquad \text{(a.s.)} \tag{2.8a}$$

$$\lim_{n\to\infty} \frac{\tilde{s}_n^{(b)}}{\log n} = \frac{1}{h_1} \qquad \text{(pr.)} \tag{2.8b}$$

provided

$$\alpha(d) = O(n^\beta \rho^d) \tag{2.9}$$

for some constants $0 < \rho < 1$ and $\beta > 0$. For the height $H_n^{(b)}$ we have

$$\lim_{n\to\infty} \frac{H_n^{(b)}}{\log n} = \frac{1}{h_2^{(b)}} \qquad \text{(a.s.)} \tag{2.10}$$

provided the mixing coefficient $\alpha(d)$ fulfills the following

$$\sum_{d=0}^{\infty} \alpha^2(d) < \infty \ . \tag{2.11}$$

(ii) Compact Suffix Tree. *Almost sure behavior of PAT follows from the (a.s.) behavior of b-suffix trees by taking in (2.8) and (2.10) the limit as $b \to \infty$, that is,*

$$\lim_{n\to\infty} \frac{s_n^{PAT}}{\log n} = \frac{1}{h_1} \qquad \text{(a.s.)} \qquad \lim_{n\to\infty} \frac{H_n^{PAT}}{\log n} = \frac{1}{h_3} \ , \tag{2.12}$$

provided (P) holds together with condition (2.9) for s_n^{PAT} and condition (2.11) for H_n^{PAT} respectively. ■

Our next main results deal with the depths $D_n^{(b)}$, $M_n^{(b)}$, $L_n^{(b)}(m)$ (for fixed m), and the depth of insertion $L_n^{(b)}$. The proof of Theorem 2 is presented in Section 3.3 except part (iii) which is discussed in Remark 2 (ii).

Theorem 2. *Let $\{X_k\}$ be a stationary ergodic and mixing sequence in the strong sense of (A2), and let (2.9) hold too. Assume also that $1 \leq b < \infty$.*

(i) Convergence in Probability. *For $h < \infty$ and m fixed, we have*

$$\lim_{n\to\infty} \frac{L_n^{(b)}(m)}{\log n} = \lim_{n\to\infty} \frac{L_n^{(b)}}{\log n} = \lim_{n\to\infty} \frac{D_n^{(b)}}{\log n} = \frac{1}{h} \qquad (pr.)\ . \tag{2.14}$$

The same holds for the compact suffix tree provided (2.5) in (P) is fulfilled (i.e., we may take $b \to \infty$ in the above).

(ii) Almost Sure Convergence . *Let, in addition, the probability $P(B_n)$ of "bad states" in the Shannon-McMillan-Breiman Theorem (more precisely: in the so called Asymptotic Equipartition Property) [5] be summable (cf. Sec. 3.3), that is,*

$$\sum_{n=1}^{\infty} P(B_n) < \infty\ . \tag{2.15}$$

Then, for fixed m

$$\lim_{n\to\infty} \frac{L_n^{(b)}(m)}{\log n} = \frac{1}{h} \qquad (a.s.)\ . \tag{2.16}$$

The above is true also for the compact suffix tree provided (2.5) in (P) is satisfied.

(iii) Almost Sure Oscillations. *As in (ii) we assume strong mixing condition (2.4b) together with $h_1 < \infty$ and $h_2 > 0$. Then, for $b < \infty$ we have the following result concerning the depth of insertion and the typical depth*

$$\lim_{n\to\infty} \inf_n \frac{L_n^{(b)}}{\log n} = \frac{1}{h_1} \qquad (a.s) \qquad \lim_{n\to\infty} \sup_n \frac{L_n^{(b)}}{\log n} = \frac{1}{h_2^{(b)}}. \tag{2.17a}$$

$$\lim_{n\to\infty} \inf_n \frac{D_n^{(b)}}{\log n} = \frac{1}{h_1} \qquad (a.s) \qquad \lim_{n\to\infty} \sup_n \frac{D_n^{(b)}}{\log n} = \frac{1}{h_2^{(b)}}. \tag{2.17b}$$

For the compact suffix tree, (2.17a) and (2.17b) hold with $h_2^{(b)}$ replaced by h_3, that is, we formally obtain almost sure behavior for PAT by taking $b \to \infty$ and assuming (2.5) of (P). ■

Remark 2

(i) *How restrictive are conditions (2.9) and (2.15) ?* Let us first deal with (2.9). Note that (2.9) holds in many interesting cases including the Bernoulli model and the Markovian model. Naturally, (2.9) is true for the Bernoulli model since in this case $\alpha(d) = 0$. In the Markovian model, it is known (cf. [5]) that for a finite state Markov chain the coefficient $\alpha(d)$ decays exponentially, that is, for some $c > 0$ and $\rho < 1$ we have $\alpha(d) = c\rho^d$, as needed for (2.9). Regarding (2.15), we know that it holds at least for the Bernoulli and Markovian models but generally not for all ergodic stationary sequences (cf. [10]). We believe that (2.15) is included in (2.9). In passing, we also note that condition (2.5) holds in both above models. In the Markovian model, however, one needs the additional assumption that all transition probabilities are positive and strictly smaller than one.

It should be mentioned, however, that condition (2.9) probably cannot be improved. This is due to recent results of Paul Shields [34] who proved that the normalized external path length $E_n^{(1)}/n\log n$ converges almost surely to $1/h$ in the Bernoulli and Markovian models. But, the author of [34] also constructed an ergodic stationary (non-Markovian) sequence for which the external path length $E_n^{(b)}$ does not converge even in probability. The same construction can be used to show nonconvergence results for other tree parameters considered in this paper. Hence, some kind of restrictions for the mixing coefficient $\alpha(d)$ is necessary.

(ii) *How to prove part (iii) of Theorem 2 ?* One can view the behavior of $L_n^{(b)}(m)$ and $L_n^{(b)}$ as a surprise. The main reason for the oscillation of $L_n^{(b)}$ is a "tiny" unbalance in the height and the shortest feasible path discovered in Theorem 1. The almost sure behavior of $L_n^{(b)}(m)$ is guaranteed by the fact that it is a nondecreasing sequence. In passing, we note that the only b-suffix tree that has (a.s.) limit for the depth of insertion $L_n^{(b)}$ is the PAT tree with the *symmetric* alphabet (i.e., $p_i = 1/V$ for $1 \le i \le V$). Indeed, in this case by Theorem 2 (iii) $\lim_{n\to\infty} L_n^{PAT}/\log n = \log V$ (a.s.).

To prove formally Theorem 2(iii) for $L_n^{(b)}$ we argue as in Pittel [30] (cf. [38]). Provided Theorem 1 *is granted*, we note that almost surely $L_n^{(b)} = H_n^{(b)}$ whenever $H_{n+1}^{(b)} > H_n^{(b)}$, which happen infinitely often (a.s.) since $H_n^{(b)} \to \infty$ (a.s.), and $\{X_k\}$ is an ergodic sequence. This establishes the lim sup part of $L_n^{(b)}$. For the lim inf of $L_n^{(b)}$ we consider the shortest feasible path $s_n^{(b)}$ and repeat the above arguments. The same is true for the typical depth $D_n^{(b)}$ since it represents the length of a *randomly* selected external node, so $\tilde{s}_n^{(b)} \le D_n^{(b)} \le H_n^{(b)}$. But, $s_n^{(b)} = \tilde{s}_n^{(b)}$ infinitely often, hence (2.16) follows from Theorem 1, too. Note that $\tilde{s}_n^{(b)}$ is *not* a monotone sequence, the property needed to estimate the almost sure convergence of $s_n^{(b)}$ (for more details see Section 3.1).

(iii) *Compact Suffix Tree as a Limit of b-Suffix Tree.* We prove now results for PAT trees provided the corresponding results for b-suffix trees are true (see Section 3). We are not able to prove in general that for any parameter (appropriately normalized) of b-suffix tree, say $P_n^{(b)}$, its corresponding parameter P_n^{PAT} of the PAT tree can be obtained as a limit when b tends to infinity as $n \to \infty$. However, we conjecture that there exists a sequence $a_n = o(n)$

(e.g., in the case of parameters discussed in this paper we have $a_n \sim \log n$) such that

$$\lim_{n\to\infty} P_n^{PAT}/a_n = \lim_{b\to\infty}\lim_{n\to\infty} P_n^{(b)}/a_n \ . \tag{2.18a}$$

(The condition $a_n = o(n)$ seems to be important since the above does not hold for the size of b-suffix trees, i.e., number of internal nodes, which grows asymptotically as n/h while for the PAT tree the size is equal to $n - V + 1$.) However, we can easily give a formal proof of this fact for every parameter discussed in this section. We first consider all parameters except the height. Using the *Sample Path Theorem* of the stochastic dominance relationship [35], we can easily prove that $P_n^{(b)}$ is a decreasing sequence with respect to b. Hence, in particular

$$\lim_{n\to\infty} \frac{P_n^{PAT}}{\log n} \le \lim_{n\to\infty} \frac{P_n^{(1)}}{\log n} \ . \tag{2.18b}$$

This immediately establishes the upper bound part of (2.18a) for the above parameters (excluding the height). For the height H_n^{PAT}, following Pittel [30] we note that the event $\{H_n^{PAT} > k + b\}$ implies that there exists a set of b suffixes such that all of them share the same first k symbols. In other words, the event $\{H_n^{PAT} > k + b\}$ implies $\{H_n^{(b)} > k\}$. Therefore,

$$\lim_{n\to\infty} \frac{H_n^{PAT}}{\log n} \le \lim_{b\to\infty}\lim_{n\to\infty} \frac{H_n^{(b)}}{\log n} = \frac{1}{h_3} \tag{2.18c}$$

This completes the upper bound in (2.18a).

For the lower bound we use condition (2.5) of (P). We need a separate discussion for every parameter. Following Pittel [30], for the height and the shortest path we argue as follows. We try to find a path in a suffix tree such that its length is (a.s.) asymptotically equal to $\log n/h_3$ and $\log n/h_1$ respectively. But, this is immediate from (2.7a) and (2.7b), and Pittel's [30] Lemma 2. For the depth, we consider a path for which the initial segment of length $O(\log n)$ is such that all nodes are branching (i.e., no unary nodes occurs in it). Naturally, such a path after compression will not change, and the depth in the compact suffix tree is at least as large as the length of this path. Copying our arguments from Section 3 and using Pittel [30], we easily establish that almost surely such a path is smaller or equal to $\log n/h$ which completes the lower bound arguments in the proof for the depth. The details are left for the interested reader.

Despite our formal proof, it is important to understand intuitively why a compact suffix tree can be considered as a limit of b-suffix trees as $n \to \infty$. There are at least three reasons supporting this claim: (1) b-suffix trees do not possess unary nodes in any place that is b levels above the fringe of the noncompact suffix tree (cf. Figure 1); (2) unary nodes tend to occur more likely at the bottom of a suffix tree, and it is very unlikely in a typical suffix tree to have unary nodes close to the root (e.g., in the Bernoulli model the probability that the root is unary node is equal to $\sum_{i=1}^{V} p_i^n$); (3) on a typical path the compression is of order of $O(1)$. For example, comparing the depth of regular tries and PATRICIA we know that $ED_n^P - ED_n^T = O(1)$ [36], [37], but for the height we have $EH_n^P - EH_n^T \sim \log n$ [30], and therefore, we can expect troubles only with the height. This is in fact confirmed by our analysis.

(iv) *Finite Strings.* In several computer science applications (cf. [1], [2], [3], [9], [12]) the string $\{X_k\}_{k=1}^n$ has a finite length n, and it is terminated by a special symbol that does not belong to the alphabet Σ, e.g., $X\$$ with $\$ \notin \Sigma$. Most of our results, however, can be directly applied to such strings. Let s'_n, H'_n and D'_n denote the shortest feasible path, the height and the depth in a suffix tree (b-suffix tree or compact suffix tree) built over such a finite word respectively. Then, it is easy to see that $s'_n = 1$, but the other two parameters have exactly the same asymptotics as for the infinite string case, that is, $H'_n/\log n \sim 1/h_2^{(b)}$ (a.s.) and $D'_n/\log n \sim 1/h$ (pr.) under hypotheses of Theorems 1 and 2. Indeed, assume for simplicity $b = 1$, and define new self-alignments C'_{ij} as $C'_{ij} = \min\{C_{ij},\ n-i,\ n-j\}$, where C_{ij} is the self-alignment between the i and j suffixes for the *infinite* string $\{X_k\}_{k=1}^{\infty}$. But, our analysis reveals that only the last $O(\log n)$ suffixes may have any impact on the self-alignments C'_{ij}. Hence, building a suffix tree from the first $n' = n - O(\log n)$ suffixes will lead to the same asymptotics as for an infinite string. Details are left to the interested reader. □

2.3 Applications and Further Discussions

Theorem 1 and 2 find several applications in combinatorial problems on words, data compression and molecular biology. In general, our findings can be widely used in problems dealing with repeated patterns and other regularities on strings. As an illustration, we solve some problems on words using Theorem 2. Two of them deal with data compression, and the others concern pattern matching. The first data compression example solves the conjecture of Wyner and Ziv [43], and was already reported in Szpankowski [38] while here we present some further extensions. The second one identifies the (a.s.) behavior of the block length in the well known parsing algorithm of Lempel and Ziv [25].

PROBLEM 1. *Wyner-Ziv Conjecture for Data Compression Schemes*

The following idea is behind most data compression schemes. Consider a "data base" sequence of length n which is known to both the sender and the receiver. Instead of transmitting the next L_n symbols (not in the data base), the sender can "look backward" into the data base and verify whether these L_n symbols have already appeared in the data base. If this is the case, then the sender transmits only the location of these L_n symbols in the data base and the length of L_n. More precisely, let the data base be represented by a subsequence $\{X_k\}_{k=1}^n$ of a stationary ergodic sequence $\{X_k\}_{k=1}^\infty$. For every n, let L_n be the smallest integer $L > 0$ such that $X_m^{m+L} \neq X_{n+1}^{n+1+L}$ for all $1 \leq m \leq n$. Wyner and Ziv [43] asked about almost sure behavior of L_n. The authors of [43] proved that $L_n \sim \log n/h$ in probability (pr.), and they conjectured that this can be extended to the almost sure (a.s.) convergence. Szpankowski in [38] showed that the parameter L_n is equal to the depth of insertion $L_n^{(1)}$ in a noncompact suffix tree ($b = 1$). Hence, the convergence in probability of $L_n/\log n$ is demonstrated in Theorem 1(i). But our Theorem 2(iii) settles the Wyner-Ziv conjecture in the negative (in the so called right domain asymptotics; see for details [38]), and we know that $L_n/\log n$ does not converge (a.s.) but rather oscillates between $1/h_1$ and $1/h_2^{(1)}$.

In the same paper Wyner and Ziv [43] considered another quantity, namely N_ℓ that can be defined as the smallest N such that $X_1^\ell = X_N^{N+\ell-1}$ (i.e., the word of length ℓ is found for the first time in a data base of size N_ℓ). Using the suffix tree representation of the sequence $\{X_k\}_{k=1}^{N_\ell}$ one can express N_ℓ in terms of the depth of the associated suffix tree. Indeed, N_ℓ is the size of a suffix tree for which the depth of the first suffix is equal to ℓ, that is, in our notation $L_{N_\ell}^{(1)}(1) = \ell$. But, by (2.16) of Theorem 2(ii) we have $\ell/\log N_\ell \to 1/h$ (a.s.), hence

$$\lim_{\ell \to \infty} \frac{\log N_\ell}{\ell} = h \qquad (a.s.)$$

provided (2.9) holds. This settles in the positive the second conjecture of Wyner and Ziv [43] for the Markovian model.

PROBLEM 2. *Block Length in the Lempel-Ziv Parsing Algorithm*

The heart of the Lempel-Ziv compression scheme is a method of parsing a string $\{X_k\}_{k=1}^n$ into blocks of different words. The precise scheme of parsing the first n symbols of a sequence $\{X_k\}_{k=1}^\infty$ is complicated and can be found in [25]. The main idea of the parsing is to divide the sequence into pairwise distinct blocks such that each block that occurs in the parsing has already been seen somewhere to the left (overlapping is allowed as in Grassberger [15]). For example, for $\{X_k\} = 110101001111\cdots$ the parsing looks like $(1)(10)(10100)(111)(1\cdots)$, that is, the first block has length one, the second block length is two, the next one is of length five since $X_2^5 = X_4^7$, and so on. In Figure 2 we show how to perform the parsing using a sequence of noncompact suffix trees (cf. [15]). Note that the length of a block is a subsequence of depth of insertions $L_{n_k}^{(1)}$. More precisely, if ℓ_n is the length of the nth block in the Lempel-Ziv parsing algorithm, then Figure 2 suggests the following relationship $\ell_n = L^{(1)}_{\sum_{k=0}^{n-1} \ell_k}$. For example, in Fig.1 we have $\ell_0 = L_0^{(1)} = 1$, $\ell_1 = L_1^{(1)} = 2$, $l_2 = L_{\ell_0+\ell_1}^{(1)} = L_3^{(1)} = 5$, and $\ell_3 = L_{1+2+5} = 3$,

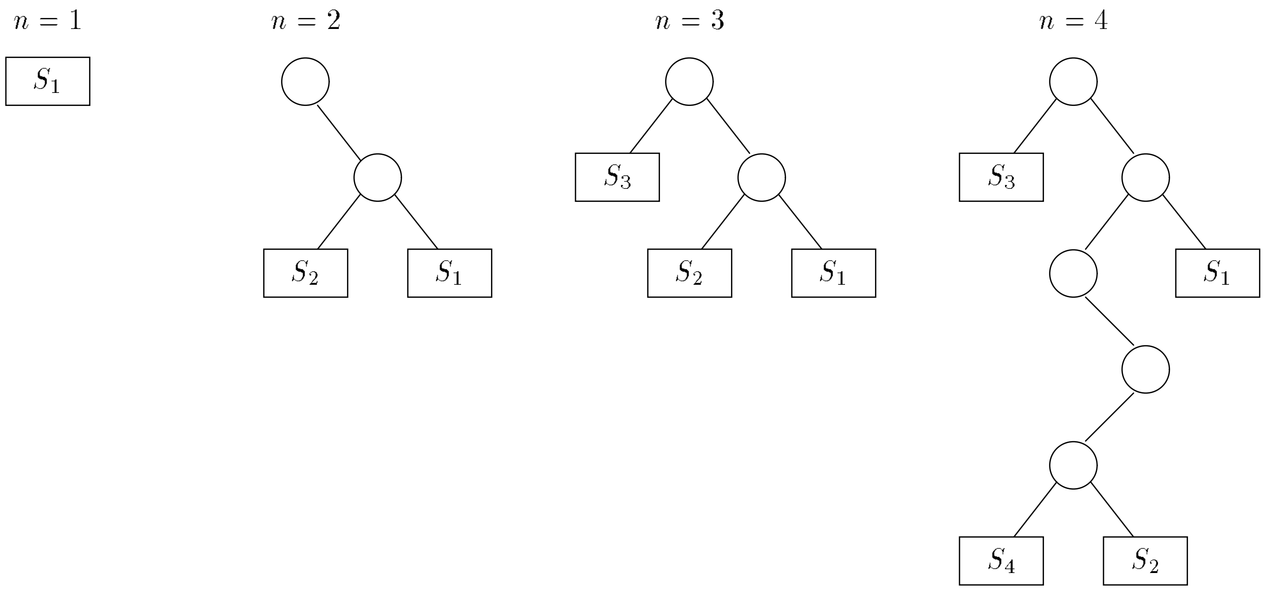

Figure 2: First four suffix trees used to parse the sequence $X = 110101001111\cdots$

and so forth. To obtain (a.s.) behavior of the block length ℓ_n, we note that

$$\lim_{n\to\infty} \frac{\ell_n}{\log n} = \lim_{n\to\infty} \frac{L_{\sum_{k=0}^{n-1}\ell_k}}{\log\left(\sum_{k=0}^{n-1}\ell_k\right)} \cdot \frac{\log\left(\sum_{k=1}^{n-1}\ell_k\right)}{\log n} . \tag{2.19}$$

We first estimate the second term in (2.19). One immediately obtains

$$1 \le \frac{\log\left(\sum_{k=0}^{n-1}\ell_k\right)}{\log n} \le \frac{\log\left(\sum_{m=0}^{n} L_n^{(1)}(m)\right)}{\log n} \to 1 \qquad (a.s) ,$$

where the RHS of the above is a direct consequence of our result concerning the external path length $E_n^{(1)}$ proved in Shields [34] (in fact, a slight extension of our proof of Theorem 2 (ii) leads to the same result). Then, by (2.19) (cf. also [38]) we obtain the following corollary.

Corollary 3. *Let $\{X_k\}_{k=1}^{\infty}$ be a strongly mixing stationary sequence as in (A2) with the mixing coefficient $\alpha(d)$ satisfying (2.9) and (2.15). Then almost surely*

$$\frac{1}{h_1} \le \lim_{n\to\infty} \inf_n \frac{\ell_n}{\log n} \le \lim_{n\to\infty} \sup_n \frac{\ell_n}{\log n} \le \frac{1}{h_2^{(1)}}$$

provided $h_1 < \infty$ and $h_2^{(1)} > 0$. ■

We conjecture that the lim sup and lim inf are attained at $1/h_2^{(1)}$ and $1/h_1$ (a.s.), respectively.

PROBLEM 3. *String Matching Algorithms*

Recently, Chang and Lawler [9] demonstrated how to use PAT trees to design *practical* and still efficient algorithms for approximate string matching algorithms. They formulated several conclusions based on a heuristic analysis of PAT trees under the *symmetric* Bernoulli model. Our Theorems 1 and 2 immediately generalize results of [9] to a more general probabilistic model, and additionally provide stronger results. For example, consider the exact string matching algorithm (cf. Section 2.3 in [9]) in which we search for all occurrences of the pattern string P of length m in the text string T of length n. The heart of the Chang-Lawler algorithm is an analysis of $d_{m,n}$ that is the length of a substring of the text T which is not a substring of the pattern P. This can be verified by building a compact suffix tree for P, and then comparing substrings of T with suffixes of P. But then, one may observe that $d_{m,n}$ is equivalent to the typical depth D_n^{PAT} in such a suffix tree, and therefore $d_{m,n} \sim (1/h)\log m$ (pr.). This further implies that the complexity C_n of the algorithm becomes $C_n \sim O(n/(hm)\log m)$ (pr.), which is a stronger version of the Chang-Lawler result for a more general probabilistic model. In passing, we note that our findings can be also used to estimate the time-complexity for the Knuth-Morris-Pratt algorithm [24] and the Boyer-Moore algorithm [7]. Several other approximate string matching algorithms can be analyzed in a similar manner. The reader is referred to Apostolico and Szpankowski [4] for more algorithmic examples.

PROBLEM 4. *A Molecular Biology Problem: Rare Subsequences*

Biologists often need a (shortest) subsequence of a sequence $\{X_k\}_{k=1}^n$ (e.g., DNA or RNA) that determines (identifies) uniquely this sequence or that occurs very rarely in the underlying sequence. Sometimes, they also need to find the shortest subsequence which does not occur in the sequence (cf. [13]). (In fact, biologist allow gaps, but we will not treat this case here.) These problems can be solved with a clever use of the suffix tree data structure (no gaps are allowed!). We here illustrate only how our result can be used to solve the latter problem. Call the shortest subsequence that does not occur in the underlying sequence as U_n. A question arises what is the typical length of U_n, and how to construct U_n. It should be clear that the length of U_n cannot be too short (e.g., single characters or pairs of characters occur too often!). If one builds a noncompact suffix tree of $\{X_k\}_{k=1}^n$, then certainly all subsequences up to level $s_n^{(1)}$ occur in $\{X_k\}$ since the suffix tree is a complete tree up to this level. Hence, the length of U_n should be equal to $s_n^{(1)}$, and (a.s.) its length is asymptotically equal to $(1/h_1)\log n$. Moreover, U_n can be discovered by taking any subsequence that leads to the closest 'hole" in the associated suffix tree. $\Box$

Finally, we would like to offer some remarks regarding further implications and generalizations of our results.

Remark 3.

(i) *Convergence in Distributions.* In this paper we deal only with the almost sure convergence. One may ask about the limiting distribution of $L_n^{(b)}$, $H_n^{(b)}$, and so forth. At this time, we have very limited knowledge about the limiting distribution of the above parameters. In fact, only the typical depth in the Bernoulli model of noncompact suffix tree ($b = 1$) was analyzed in the past. Jacquet and Szpankowski [19] proved that the distribution $F_n^{\mathcal{T}}(x)$ of the typical depth in *independent tries* and the distribution $F_n^{\mathcal{S}}(x)$ of the typical depth $D_n^{(1)}$ in suffix trees, do not differ too much. More precisely, in [19] it is proved that for large n there exist such $\beta > 1$ and $\varepsilon > 0$ that

$$|F_n^T(k) - F_n^S(k)| = O\left(\frac{1}{n^\epsilon \beta^k}\right) . \tag{2.20}$$

This establishes similarities between a trie and a noncompact suffix tree. Therefore, using well known results for independent tries (cf. [31]) it is easy to show that for an asymmetric alphabet Σ, the normalized depth $(D_n^{(1)} - ED_n^{(1)})/varD_n^{(1)}$ converges *in distribution* to the standard normal distribution $\mathcal{N}(0,1)$ with mean and variance as below

$$ED_n^{(1)} = \frac{1}{h} \cdot \{\log n + \gamma + \frac{h_3}{2h}\} + P_1(\log n) + O\left(\frac{1}{n^\epsilon}\right) , \tag{2.21a}$$

$$varD_n^{(1)} = \frac{h_3 - h^2}{h^3} \log n + C + P_2(\log n) + O\left(\frac{1}{n^\epsilon}\right), \tag{2.21b}$$

for some $\varepsilon > 0$, where $h_3 = \sum_{i=1}^V p_i^2 \log p_i$, and $P_1(x), P_2(x)$ are fluctuating periodic functions with small amplitudes, and an explicit formula for the constant C can be found in [37]. In the symmetric case, the variance becomes

$$varD_n^{(1)} = \frac{\pi^2}{6\log^2 V} + \frac{1}{12} + O\left(\frac{1}{n^\epsilon}\right), \tag{2.21c}$$

Moreover, in the symmetric case the distribution of $D_n^{(1)}$ is not any longer asymptotically normal, but rather resembles one of the extreme distribution. More precisely, in this case we have (cf. [31])

$$\lim_{n\to\infty} \sup_x \mid \Pr\{D_n^{(1)} \le x\} - \exp(-nV^{-x}) \mid = 0 . \tag{2.21d}$$

We conjecture that the same type of limiting distributions can be obtained in the Markovian model. The is due to the fact that (2.20) seems to hold in the Markovian case. If so, we can apply the recent result of Jacquet and Szpankowski [18] regarding the limiting distribution of the depth for the Markovian model of independent tries. Furthermore, one may investigate the limiting distribution for the height and the external path length. We conjecture that b-suffix trees do not differ too much from b-tries in the sense of (2.20), and therefore, the limiting law for the height can be obtained from the one for b-tries (cf. [31]), and so on.

The compact suffix tree is more intricate. Only very recently some results regarding limiting law for the depth in PATRICIA have been obtained (cf. [32]). Using this result, Jacquet, Rais and Szpankowski [20] proved that the limiting distribution for the depth in PAT tree under an *asymmetric* Bernoulli model is asymptotically normal. There is, however, no result regarding the limiting law of the height. These seem to be difficult problems.

(ii) *How well is a suffix tree balanced?* In the worst case a suffix tree may degenerate, and the worst case height can be as much as n. But, our analysis indicates that this happens very, very rarely. In fact, our Theorem 2 shows that the typical depth of a suffix is equal to $(1/h)\log n$ (pr.). The best balanced tree built over n external nodes is a complete tree (cf. [1]), and the depth for every external node in such a complete tree is equal to $\log_V n$. We note that for the symmetric alphabet a typical shape of suffix tree resembles that of a complete tree since the depth $D_n^{(b)}$ with high probability is equal to $\log_V n$, and almost surely is not greater than $H_n^{(b)} \sim (1+1/b)\log_V n$ but not smaller than $s_n \sim \log_V n$. Such a tree can be called highly balanced (in a probability sense), and – as our analysis shows – there is no need – in most practical cases – for additional rebalancing of this tree in order to assure a nice behavior in the worst case, as is done in AVL-tree and other balanced trees. $\Box$

3. ANALYSIS AND PROOFS

We first present a formal proof of Theorem 1 concerning the height $H_n^{(b)}$ and the shortest feasible path $s_n^{(b)}$. Then, we establish parts (i) and (ii) of Theorem 2 for the typical depth $D_n^{(b)}$, and the depth of the mth suffix $L_n^{(b)}(m)$. We remind the reader that Theorem 2 (iii) was already proved in Remark 2 (ii), and the compact suffix tree was discussed in Remark 2 (iii). Therefore, hereafter we fix $b<\infty$. Also, for simplicity of presentation we drop the upper index b in the notation of the tree parameters (e.g., we write H_n instead of $H_n^{(b)}$).

Throughout the proof we use a novel technique that encompasses the mixing condition and another technique called the *string-ruler approach* that was already applied by Pittel in his seminal paper [30], and extended by Jacquet and Szpankowski [19] (cf. [38]). The idea of the string-ruler approach is to measure a correlation between words by another *nonrandom* word w belonging to a set of words $\mathcal{W}$. Usually, we deal with fixed length rulers w_k where k is the length of the string-ruler. Let $\mathcal{W}_k$ be the set of all strings w_k, that is, $\mathcal{W}_k = \{w \in \Sigma^k :$

$|w| = k\}$, where $|w|$ is the length of w. We write w_k^ℓ to mean a concatenation of ℓ strings w_k from $\mathcal{W}_k$, and if $X_m^{m+k} = w_k$, then we denote $P(w_k) = P(X_m^{m+k})$. Finally, we adopt the following rule regarding sums over a set of string-rulers: if $f(w_k)$ is a function of w_k, then $\sum_{\mathcal{W}_k} f(w_k) = \sum_{w_k \in \mathcal{W}_k} f(w_k)$ where the sum is over all strings w_k of length k.

The usefulness of the string-ruler approach stems from the fact that we can express the self-alignment $C_{i_1,\ldots,i_{b+1}}$ in terms of w_k. The following lemma is of prime importance to the analysis of suffix trees and other combinatorial structures on words.

Lemma 4. *Let $d_1, \ldots, d_b$ and k be such that*

$$d_0 = 0 \le d_1 \le \cdots \le d_i \le k \le d_{i+1} \le \cdots \le d_b \ . \tag{3.1}$$

Define d as the greatest common divisor of $\{d_i\}_{i=1}^b$, that is, $d = \gcd(d_1, \ldots, d_b)$. Then, the self-alignment $C_{1,1+d_1,\ldots,1+d_1+\cdots+d_b}$ satisfies

$$\begin{aligned}
\Pr\{C_{1,1+d_1,\ldots,1+d_1+\cdots+d_b} \ge k\} &= \sum_{\mathcal{W}_d} P\left(w_d^{\lfloor \frac{k}{d} \rfloor + \frac{d_1+\cdots+d_i}{d}} \overline{w}_d (w_d^{\lfloor \frac{k}{d} \rfloor} \overline{w}_d)^{b-i} \right) \\
&= \sum_{\mathcal{W}_d} P\left(w_d^{(b+1-i)\lfloor \frac{k}{d} \rfloor + \frac{d_1+\cdots+d_i}{d}} \overline{w}_d^{b+1-i} \right) \ ,
\end{aligned} \tag{3.2a}$$

where $\overline{w}_d$ is a prefix of w_d, and $\lfloor x \rfloor$ is the floor function. Two cases are of particular interest, namely: (i) if $k \le d_1 \le \cdots \le d_b$, then

$$\Pr\{C_{1,1+d_1,\ldots,1+d_1+\cdots+d_b} \ge k\} = \sum_{\mathcal{W}_k} P(w_k^{b+1}) \ ; \tag{3.2b}$$

(ii) if $d_1 \le \cdots \le d_b \le k$, then

$$\Pr\{C_{1,1+d_1,\ldots,1+d_1+\cdots+d_b} \ge k\} = \sum_{\mathcal{W}_d} P\left(w_d^{\lfloor \frac{k}{d} \rfloor + \frac{d_1+\cdots+d_i}{d}} \overline{w}_d \right) \ . \tag{3.2c}$$

Proof. It is illustrative to start with $b = 1$. In this case, it is well known [26] that for any pair of suffixes S_1 and S_{1+d} there exists a word w_d such that the common prefix Z_k of length k of S_1 and S_{1+d} can be represented as $Z_k = w_d^{\lfloor \frac{k}{d} \rfloor} \overline{w}_d$. Then, (3.2) (in fact (3.2c)) is a simple consequence of the above. The above rule is easy to extend to b suffixes. Let Z_k be the common prefix of length k of the following b suffixes $\{S_1, S_{1+d_1}, \ldots, S_{1+d_1+\cdots+d_b}\}$. To avoid heavy notation, we consider three cases separately. If $k \le d_1 \le \cdots \le d_b$, then all suffixes are separated by more than k symbols, so certainly there exists a word w_k such that $Z_k = w_k^{b+1}$, which further implies (3.2b). Let us now consider the case $d_1 \le \cdots \le d_b \le k$, that is, there are mutual overlaps between any two consecutive suffixes. Then, there must exist a word w_d

of length $d = \gcd(d_1, \ldots, d_b)$ such that $Z_k = w_d^{\lfloor \frac{k}{d} \rfloor + \frac{d_1 + \cdots + d_i}{d}} \overline{w}_d$, which leads to (3.2c). The general solution (3.2a) is a combination of the two above cases. ∎

Finally, in the proof below we often use another representation for the b-th order Rényi entropy, namely (we drop the index b according to our convention) under assumption (A1) we have

$$h_2 = \lim_{n\to\infty} \frac{\log\left(\sum_{\mathcal{W}_n} P^{b+1}(w_n)\right)}{(b+1)n} . \tag{3.3}$$

Indeed, the above is a simple consequence of the weak mixing condition (2.4a) of (A1), and the fact that b is fixed.

3.1 Height of b-Suffix Trees

We now prove Theorem 1 (i) formula (2.10) concerning the (a.s.) behavior of the height. We discuss separately the upper and the lower bounds for the convergence in probability. Finally, (a.s.) convergence is established.

Upper Bound

We start with the representation (2.3b) for the height H_n. By Boole's inequality the tail of the height distribution can be bounded from above by a sum of marginal distributions of the self-alignments. In other words, using Lemma 4 we have

$$\begin{aligned}
\Pr\{H_n \geq k\} &= \Pr\{\max_{1\leq d_1,\ldots,d_b \leq n} \{C_{1,1+d_1,\ldots,1+d_1+\cdots+d_b}\} \geq k\} \\
&\leq \sum_{d_1=1}^{n} \cdots \sum_{d_b=1}^{n} \Pr\{C_{1,1+d_1,\ldots,1+d_1+\cdots+d_b} \geq k\} \\
&\leq n \sum_{i=1}^{b} \binom{b}{i} k^i n^{b-i} \sum_{\mathcal{W}_d} P\left(w_d^{\lfloor \frac{k}{d} \rfloor + \frac{d_1+\cdots+d_i}{d}} \overline{w}_d w_k^{b-i}\right) .
\end{aligned}$$

The last sum can be estimated as follows

$$\begin{aligned}
\sum_{\mathcal{W}_d} P\left(w_d^{\lfloor \frac{k}{d} \rfloor + \frac{d_1+\cdots+d_i}{d}} \overline{w}_d w_k^{b-i}\right) &\leq^{(A)} c^{b-i} \sum_{\mathcal{W}_k} P(w_k) P^{b+1-i}(w_k) \\
&\leq^{(B)} c^{b-i} \left(\sum_{\mathcal{W}_k} P^{b+1}(w_k)\right)^{(b+1-i)/(b+1)}
\end{aligned}$$

where the first inequality (A) comes from the strong mixing condition of assumption (A1), and the fact that the set of words of the form $w_d^{\lfloor k/d \rfloor} \overline{w}_d$ is a subset of all words of length k

(i.e., $\mathcal{W}_k$). The inequality (B) is a consequence of the the well known *inequality on means* (cf. [28]). So, finally for some constant c, we have

$$\Pr\{H_n \geq k\} \leq c \sum_{i=1}^{b} \binom{b}{i} k^i n^{b-i+1} \left(EP^b(w_k)\right)^{(b+1-i)/(b+1)} . \tag{3.4}$$

Let now $k = \lfloor (1+\varepsilon) \log n / h_2 \rfloor$. Then, definition (2.6), identity (3.3), and the above prove the following upper bound

$$\begin{aligned} \Pr\{H_n \geq (1+\varepsilon)\frac{\log n}{h_2}\} &\leq c \sum_{i=1}^{b} \binom{b}{i} k^i n^{b+1-i} \frac{1}{n^{(1+\varepsilon)(b+1-i)}} \\ &\leq \frac{c}{n^{\varepsilon}}(k+n^{-\varepsilon})^b \sim \frac{c(1+\varepsilon)^b \log^b n}{n^{\varepsilon} h_2^b} . \end{aligned}$$

This completes our arguments for the upper bound of the height for the convergence in probability. The (a.s.) convergence will be established after the proof of the lower bound.

Lower Bound

The lower bound is more intricate. The idea, however, is quite simple. At first, we construct another b-suffix tree with height that is smaller than in the original b-suffix tree, but which resembles independent tries (i.e., strings stored in such a suffix tree are less correlated than in the original b-suffix tree). Secondly, we apply the *second moment method* [36] to the modified suffix tree. The second moment method gives a sharp lower bound for $\Pr\{H_n > k\}$. In particular, using this method we prove that $\Pr\{H_n > k\} \to 1$ for $k = \lfloor (1-\varepsilon)\frac{1}{h_2} \log n \rfloor$.

To fulfill this plan, we start with a construction of the modified b-suffix tree. We partition the sequence X_1^n into m parts each composed of k consecutive symbols followed by a gap of size d. Therefore, the size of each part is $k+d$ and $m = \lfloor n/(k+d) \rfloor$. In the following we assume that $k = O(\log n)$ as well as $d = O(\log n)$, hence $m = O(n/\log n)$. We define new strings $Y(1), \ldots, Y(m)$ as $Y(i) = X_{(i-1)(k+d)+1}^{\infty} - X_{ik+(i-1)d+1}^{i(k+d)}$ where "$-$" means deletion, that is, $Y(i)$ is the $((i-1)(k+d)+1)$st suffix of $\{X_k\}$ with a gap of length d between the $(ik+(i-1)d+1)$st symbol and the $(i(k+d))$th symbol deleted. For example, the first string $Y(1)$ consists of the first k symbols followed by all symbols after the $(k+d)$th symbol (the first gap between $k+1$ and $k+d$ is omitted). The second string $Y(2)$ starts at position $k+d+1$ and continues for the next k symbols after which the next d symbols of the second gap are eliminated, and then the strings expands up to infinity. We build a b-suffix tree out of these m strings $Y(1), \ldots, Y(m)$. We denote such a b-suffix tree as $\mathcal{T}_m$ since for a typical sequence $\{X_k\}$ these m strings resemble independent keys in a b-trie, that is, they are weakly dependent on their first k symbols.

We denote by H_m the height of the modified b-suffix tree $\mathcal{T}_m$. Certainly, this height is stochastically smaller than the height H_n in the original tree. This can be proved formally by the *Sample Path Theorem* [35]. As a simple consequence of this fact, we have

$$\Pr\{H_n \geq k\} \geq \Pr\{H_m \geq k\} \qquad \text{for} \quad m \leq n \ , \tag{3.5}$$

We estimate the probability $\{H_m \geq k\}$ by the second moment method (cf. [8], [36]). We need some additional notation. Let $\mathbf{i} = (i_1, i_2, \ldots, i_{b+1})$ be a $b+1$ dimensional vector, and define a set D as $D = \{\mathbf{i}: \ 1 \leq i_j \leq m \ \text{ for } \ 1 \leq j \leq b+1\}$. Let also $D^2 = D \times D$ which we additionally partition into two sets D_1^2 and D_2^2 such that

$$D_1^2 = \{(\mathbf{i}, \mathbf{j}): \ (i_1, \ldots, i_{b+1}) \cap (j_1, \ldots, j_{b+1}) = \emptyset\} \ , \tag{3.6}$$

and D_2^2 contains the other pairs $(\mathbf{i}, \mathbf{j})$ of D^2. Now, let us define an event $A_{\mathbf{i}} = \{C_{\mathbf{i}} \geq k\}$ where we use $C_{\mathbf{i}}$ as a short notation for the self-alignment $C_{i_1,\ldots,i_{b+1}}$. Note that $\Pr\{H_m \geq k\} = \Pr\{\bigcup_{\mathbf{i}\in D} A_{\mathbf{i}}\}$. Then, the second moment method asserts that (cf. [8])

$$\Pr\{H_m \geq k\} = \Pr\{\bigcup_{\mathbf{i}\in D} A_{\mathbf{i}}\} \geq \frac{(\sum_{\mathbf{i}\in D} \Pr\{A_{\mathbf{i}}\})^2}{\sum_{\mathbf{i}\in D} \Pr\{A_{\mathbf{i}}\} + \sum_{(\mathbf{i},\mathbf{j})\in D^2} \Pr\{A_{\mathbf{i}} \cap A_{\mathbf{j}}\}} \ . \tag{3.7}$$

We will show that for $k = \lfloor (1-\varepsilon)\frac{1}{h_2} \log n \rfloor$ the right-hand side (RHS) of (3.7) tends to one, hence also by (3.5) $\Pr\{H_n \geq (1-\varepsilon)\frac{1}{h_2} \log n\} \to 1$ as $n \to \infty$, which is the desired inequality.

We must now evaluate the terms in the RHS of (3.7). Using the strong α-mixing condition of (A2), and arguing as in the upper bound case, we immediately show that for $k = O(\log n)$ (cf. [38])

$$(m^b - o(m^b))(1-\alpha(d_n))^b EP^b(w_k) \leq \sum_{\mathbf{i}\in D} \Pr\{A_{\mathbf{i}}\} \leq (m^b - o(m^b))(1+\alpha(d_n))^b EP^b(w_k) \ ,$$

where the length of the gap d_n is explicitly shown to be a function on n. The probability $\Pr\{A_{\mathbf{i}} \cap A_{\mathbf{j}}\}$ is more difficult to estimate. However, on the set D_1^2, suffixes of $\mathcal{T}_m$ do not coincide, hence we have (cf. [38])

$$\sum_{(\mathbf{i},\mathbf{j})\in D_1^2} \Pr\{A_{\mathbf{i}} \cap A_{\mathbf{j}}\} \leq (1+\alpha(d_n))^{2b+1} E^2 P^b(w_k) \ .$$

In the set D_2^2, there exists at least one pair of suffixes that is the same for $\mathbf{i}$ and $\mathbf{j}$. For example, if $\mathbf{i} = (1,5)$ and $\mathbf{j} = (1,6)$, then $\Pr\{A_{\mathbf{i}} \cap A_{\mathbf{j}}\} = \sum_{\mathcal{W}_k} P(w_k^3)$, since the first suffix is common to $\mathbf{i}$ and $\mathbf{j}$. In general, the following is true

$$\begin{aligned}
\Pr\{A_{\mathbf{i}} \cap A_{\mathbf{j}}\} \quad &= \quad \sum_{\mathcal{W}_k} P(w_k^{2b+1}) \leq^{(A)} c_1 \sum_{\mathcal{W}_k} P^{2b+1}(w_k) \\
&\leq^{(B)} \ c_1 \left(\sum_{\mathcal{W}_k} P^{b+1}(w_k) \right)^{(2b+1)/(b+1)} \quad = c_1 \left(EP^b(w_k) \right)^{2-1/(b+1)} \ ,
\end{aligned}$$

where (A) follows from the strong mixing conditions, and (B) is a consequence of the following known inequality (cf. [21], [36])

$$\ell \geq r \qquad \Rightarrow \qquad \left(\sum_{\mathcal{W}_k} P^\ell(w_k)\right)^{1/\ell} \leq \left(\sum_{\mathcal{W}_k} P^r(w_k)\right)^{1/r} .$$

Putting everything together, the inequality (3.7) becomes for $k = \lfloor (1-\varepsilon)\frac{1}{h_2}\log n \rfloor$

$$\Pr\{H_m \geq k\} \geq \left(\frac{n^{(b+1)(1-\varepsilon)}}{m^{b+1}(1-\alpha(d_n))^b} + [1 - O(m^{-1})]\frac{(1+\alpha(d_n))^{2b+1}}{(1-\alpha(d_n))^b} + c\frac{n^{1-\varepsilon}}{m} \right)^{-1} .$$

Substituting $m = \Theta(n/\log n)$ and $d_n = \Theta(\log n)$, we finally obtain

$$\Pr\{H_m \leq (1-\varepsilon)\frac{1}{h_2}\log n\} \leq c_1\frac{\log^{b+1} n}{n^{(b+1)\varepsilon}} + c_2\frac{\log n}{n^\varepsilon} + c_3(2b+1)b\alpha^2(\log n) + O(\alpha^3(d_n)) \ , \quad (3.8)$$

which proves the lower bound for H_m, and hence by (3.5) also for our original b-suffix tree. In summary, the upper bound (3.5) and the above lead to the following

$$\Pr\{\mid \frac{H_n}{\log n} - \frac{1}{h_2} \mid \geq \varepsilon\} \leq c_1\frac{\log^b n}{n^\varepsilon} + c_2\alpha^2(\log n) \to 0 \quad (3.9)$$

for some constants c_1 and c_2. This proves $H_n/\log n \to 1/h_2$ (pr.).

Almost Sure Convergence

The rate of convergence in (3.9) does not yet warrant the application of the Borel-Cantelli Lemma to prove almost sure convergence. But due to the fact that H_n is nondecreasing in n and $a_n = \frac{1}{h_2}\log n$ is a slowly increasing function of n, we can establish (a.s.) convergence for the height. Indeed, as in [11] (cf. also [38]) we note that $H_n > a_n$ infinitely often (i.o.) if $H_{2^r} > a_{2^{r-1}}$ (i.o.) in r, and similarly $H_n < a_n$ (i.o.) if $H_{2^r} < a_{2^{r+1}}$ (i.o.). But the latter events holds indeed infinitely often due to (3.9) and the Borel-Cantelli lemma since

$$\sum_{r=0}^{\infty} \Pr\{\mid \frac{H_{2^{r\pm1}}}{\log(2^{r\pm1})} - \frac{1}{h_2} \mid \geq \varepsilon\} < \infty \quad (3.10)$$

provided

$$\sum_{r=0}^{\infty} \alpha^2(r) < \infty .$$

This completes the proof of Theorem 1 (i) concerning the height $H_n^{(b)}$ of a b-suffix tree.

3.2 The Shortest Feasible Path of b-Suffix Trees

For the upper bound we use the fact that $s_n^{(b)}$ is nonincreasing in b, that is, $s_n^{(b)} \leq s_n^{(1)}$. Note that the first s_n levels of any suffix tree are "filled" with internal nodes (i.e., there is no "hole" in the tree up to this level). In other words, up to the level s_n a suffix tree resembles a complete tree. This fact was used in [38] (cf. [30]) to establish the following bound

$$\Pr\{s_n^{(1)} > (1+\varepsilon)\frac{1}{h_1}\log n\} \le \frac{c}{n^\varepsilon} . \tag{3.11}$$

This upper bound holds also for b-suffix trees since the parameter h_1 does not depend on b.

The rest of this section is devoted to the lower bound for $s_n^{(b)}$. As in the case of the height, we drop hereafter the upper index b in the notation of the shortest feasible path. We proceed as in the case of the lower bound for the height, that is, we define the modified suffix tree $\mathcal{T}_m$ composed of m weakly dependent strings $Y(1), \ldots, Y(m)$ which are defined precisely in Section 3.1. Again by the *Sample Path Theorem* we conclude that the shortest feasible path s_m in $\mathcal{T}_m$ is stochastically smaller than the shortest feasible path s_n in the original b-suffix tree, which implies the following

$$\Pr\{s_n < k\} \le \Pr\{s_m < k\} . \tag{3.12}$$

To estimate the probability $\Pr\{s_m < k\}$ in the modified tree $\mathcal{T}_m$ we need some more notation. Let $p_{\min}(k) = \min_{w_k \in \mathcal{W}_k}\{P(w_k)\}$, and $C_{\mathbf{i}}(w_k)$ be the length of the longest prefix of the word w_k and the $b+1$ suffixes belonging to $\mathbf{i} = (i_1, \ldots, i_{b+1})$. We assume that $\mathbf{i} \in D$ where D is the set of all $(b+1)$-tuples from the set $\{1, \ldots, m\}$. Note now that $\{s_m < k\}$ implies that there must exist a word $w_k \in \mathcal{W}_k$ such that for all $\mathbf{i} \in D$ the self-alignment $C_{\mathbf{i}}$ is smaller than k, that is, $C_{\mathbf{i}} < k$. Using the strong α-mixing condition of (A2) we have

$$\begin{aligned}
\Pr\{s_m < k\} &\le \sum_{\mathcal{W}_k} \Pr\{\bigcap_{\mathbf{i}\in D}[C_{\mathbf{i}}(w_k) < k]\} \le \sum_{\mathcal{W}_k}(1+\alpha(d_n))^{m^b}(1-P(w_k^b))^{m^b} \\
&\le (1+\alpha(d_n))^{m^b}\sum_{\mathcal{W}_k}(1-cp_{\min}^b(k))^{m^b} \le V^k(1+\alpha(d_n))^{m^b}(1-p_{\min}^b(k))^{m^b} .
\end{aligned}$$

Let now $k = \lfloor(1-\varepsilon)\frac{1}{h_1}\log n\rfloor$ and $m = \Theta(n/\log n)$ while $d_n = \Theta(\log n)$. Then,

$$\Pr\{s_n < (1-\varepsilon)\frac{1}{h_1}\log n\} \le (1+\alpha(\log n))^{m^b}\exp(-n^{\varepsilon b/2}/\log^b n) ,$$

and therefore, together with condition (2.9), this leads to the lower bound of the form

$$\Pr\{s_n < (1-\varepsilon)\frac{1}{h_1}\log n\} \le cn^\beta \exp(-n^{\varepsilon b/2}/\log^b n) . \tag{3.14}$$

The upper bound (3.11) and the lower bound (3.14) establish the convergence in probability of the shortest feasible path s_n in a b-suffix tree. The almost sure convergence can be derived in an identical manner as for the height since s_n is nondecreasing in n, and for $n = s2^r$ with some fixed s we can apply the Borel-Cantelli lemma (cf. also [38]).

The proof for the shortest depth $\widetilde{s}_n$ is simple. Since $s_n \le \widetilde{s}_n$, we need only an upper bound. Clearly, a result for $b = 1$ suffices for the proof. Note that either $\widetilde{s}_n = s_n$ or, in the same branch where s_n is located, there are two suffixes, say number one and two, with common prefix of length greater than $\widetilde{s}_n$. Hence,

$$\Pr\{\widetilde{s}_n > k\} \le \Pr\{s_n > k\} + \Pr\{C_{1,2} > k\} \ .$$

Then, (3.12) and the bound for the self-alignment derived in Section 3.1 (just above (3.4)), for $k = \lfloor \frac{1}{h_1} \log n \rfloor$ lead to the following estimate

$$\Pr\{\widetilde{s}_n > k\} \le \frac{1}{n^{\varepsilon}} + \frac{1}{n^{h_2(1+\varepsilon)/h_1}}$$

which completes the proof of Theorem 1.

3.3 The Typical Depth in b-Suffix Trees

In this section we prove Theorem 2(i) and 2(ii). We start with the convergence in probability (pr.) for the depth of insertion L_n. This will also prove the convergence in probability for the typical depth D_n and the depth of a given suffix $L_n(m)$, since all of these quantities are asymptotically equally distributed. The last assertion is easy to prove. Roughly speaking, it must hold in the suffix tree $\mathcal{T}_m$ defined in Section 3.1, at least when (2.9) takes place. Indeed, consider for example D_n and L_n. In $\mathcal{T}_m$ the next inserted suffix is "almost" independent of the previous suffixes stored already in $\mathcal{T}_m$. Hence, it randomly selects an external node which implies that L_m and D_m are distributed in a similar manner. But, as it is easy to see, the typical depths and depths of insertion in $\mathcal{T}_m$ and $\mathcal{T}_n$ are asymptotically equally distributed. Details are left to the interested reader.

The idea of the proof in this section is quite different from the one discussed before, and it resembles Pittel's proof [30] of the convergence in probability of the depth in an independent trie. It is based on counting, and it is quite typical for the information theory community. For the convenience of the reader, we shortly review the *Asymptotic Equipartition Property* (AEP) [5] [43] which is a direct consequence of the Shannon-McMillan-Breiman Theorem [5]: *For a stationary and ergodic sequence $\{X_k\}_{k=1}^n$, the state space Σ^n can be partitioned into two sets, namely "good states" set G_n and "bad states" set B_n such that for $X_1^n \in G_n$ and for sufficiently large n we have $P(X_1^n) \ge 1 - \varepsilon$ for any $\varepsilon > 0$, and $P(B_n) \le \varepsilon$.* Moreover, the nth order probability distribution of $X_1^n \in G_n$ is bounded as $e^{-n(h+\varepsilon)} \le P(X_1^n) \le e^{-n(h+\varepsilon)}$ where h is the entropy.

We concentrate on L_n. Define an event A_n such that

$$A_n = \{X_1^{\infty} : \mid L_n/\log n - 1/h \mid \ge \varepsilon/h\} \ . \tag{3.15a}$$

For Theorem 2 (i) it suffices to prove that $\Pr\{A_n\} \to 0$ as $n \to \infty$. Also, for some $\varepsilon_1 > 0$ and $n_0 \geq n$ we define another event (i.e., set of "good states")

$$G_{n_0} = \{X_1^\infty : \mid n^{-1} \log P^{-1}(X_1^n) - h \mid < \varepsilon_1 h \quad , \quad n > n_0\} \ . \tag{3.15b}$$

We partition A_n to obtain

$$P(A_n) \leq \Pr\{A_n \textbf{ and } G_{n_0} \textbf{ and } L_n < \delta \log n\} + \Pr\{L_n \geq \delta \log n\} + P(B_{n_0}) \tag{3.16a}$$

where $\delta > 1/h_2$ and

$$B_{n_0} = \sup_{n \geq n_0} \{X_1^\infty : \ |n^{-1} \log P^{-1}(X_1^n) - h| \geq \varepsilon_1 h \quad , \quad n > n_0\} \ . \tag{3.16b}$$

By AEP, we have $\lim_{n_0 \to \infty} P(B_{n_0}) = 0$. In addition, from the proof of the upper bound for the height H_n we know that $\Pr\{L_n \geq \delta \log n\} \leq c/n^{\delta - 1/h_2}$ for $\delta > 1/h_2$, hence the second probability in the above also tends to zero.

In view of the above, we can now deal only with the first term in (3.16a) which we denote for simplicity by $P_1(A_n G_n)$. This probability can be estimated as follows

$$P_1(A_n G_n) \leq \sum_{r \in C_n} \Pr\{L_n = r \ ; \ |\log(P^{-1}(X_1^r))/r - h| < \varepsilon_1 h \ , \ r \geq n_0\} = \sum_{r \in C_n} P_n^{(r)} \ , \tag{3.17a}$$

where

$$C_n = \{r : \mid r/\log n - 1/h \mid \geq \varepsilon/h \quad \textbf{and} \quad r \leq \delta \log n\} \ . \tag{3.17b}$$

Note that in (3.17) we restrict the summation only to "good states" represented by G_n. Therefore, for a word $w_r \in G_n$ we have with high probability

$$c_1 \exp\{-(1-\varepsilon_1)hr\} \leq P(w_r) \leq c_2 \exp\{-(1+\varepsilon_1)hr\} \ . \tag{3.18}$$

The next step is to estimate the probability $\Pr\{L_n = r\}$. But the event $\{L_n = r\}$ takes place if: (i) there exists an $\mathbf{i} = (i_1, \ldots, i_b, n)$ and w_{r-1} such that $C_{\mathbf{i}} = w_{r-1}$ (call this event F_n^1); and (ii) for all other $\mathbf{j} = (j_1, \ldots, j_b, n) \neq \mathbf{i}$, and all w_r we have $C_{\mathbf{j}} \neq w_r$ (call this event F_n^2). Then,

$$\Pr\{L_n = r\} \leq cn^b \sum_{\mathcal{W}_r} P(F_n^1 \cap F_n^2) \ . \tag{3.19}$$

Now, we are in position to prove Theorem 2(i). We first establish the upper bound. Set $r \geq (1+\varepsilon)\frac{\log n}{h}$. Hence, by the RHS of (3.18) we have $P(w_{r-1}^b) \leq 1/n^{b(1+\varepsilon)}$. But, using the mixing conditions of (A1) we have $P(F_n^1) \leq cP(w_r)P(w_{r-1}^b)$, and this together with the above leads to

$$P_n^{(r)} \leq \frac{c}{n^\varepsilon} \ , \tag{3.20}$$

and therefore by (3.17), and the fact that the cardinality of C_n is smaller than $\log n$, we have $P(A_n) \leq c \log n / n^{\varepsilon}$, as needed for the upper bound.

Now we consider the lower bound. We apply here the same approach as adopted in the previous lower bounds. So, let $\mathcal{T}_m$ be the suffix tree built from the strings $Y(1), \ldots, Y(m)$ as defined before. In particular, the depth of a given suffix, say the first one, $L_m(1)$ in $\mathcal{T}_m$ is bounded from above by the depth $L_n(1)$ in the original b-suffix tree. Then,

$$\Pr\{L_n \leq (1-\varepsilon)\frac{\log n}{h}\} \leq \Pr\{L_m \leq (1-\varepsilon)\frac{\log n}{h}\} \ , \tag{3.21}$$

since L_n and $L_n(1)$ have asymptotically the same distribution.

Now, we pick up the derivation at (3.19) in which the first n^b should be replaced by m^b. We estimate the probability $P(F_n^1 \cap F_n^2)$ as follows. Using the strong α-mixing condition of (A2), we have

$$P(F_m^1 \cap F_m^2) \leq cP(w_r)P(w_{r-1}^b)(1+\alpha(d_n))^{m^b}(1-P(w_r^b))^{m^b} \ .$$

Let now $r \leq (1-\varepsilon)\frac{\log n}{h}$, hence by the left-hand side (LHS) of (3.18) and (3.19), and the same argument as in the lower bound for the shortest feasible path, we finally obtain in (3.17a) for $m = O(n/\log n)$

$$P_n^{(r)} \leq c \exp(-n^{b\varepsilon/2}/\log^b n) \ . \tag{3.22}$$

provided (2.9) holds.

Putting everything together, we note that the cardinality of the set C_n in (3.17b) is bounded from above by $\delta \log n$, hence by (3.20) and (3.22) our estimate (3.16) becomes

$$P(A_n) \leq c \log n \left(\exp(-n^{b\varepsilon/2}/\log^b n) + n^{-\varepsilon}\right) + P(B_{n_0}) \ , \tag{3.23}$$

which suffices for the proof of Theorem 2 (i).

To complete the proof of Theorem 2, we need to establish the almost sure convergence for the depth $L_n(m)$. But this is an immediate consequence of the fact that the depth $L_n(m)$ is a nondecreasing sequence in n. The formal proof is along the same lines as for the height, and is omitted. This completes the proof of Theorem 2, and the entire analysis. In passing, we note that a slight extension of the above proof will directly lead to Shields results concerning the external path length, namely $E_n^{(b)}/n \log n \to 1/h$ (a.s.).

ACKNOWLEDGMENT

I thank two refrees for detailed comments that led to an improvement of the presentation in this paper. I also thank Dr. Pavel Pevzner for pointing out the reference [22]. And last but not least, I am grateful to Professor Boris Pittel for numerous discussions concerning this research.

References

[1] A.V. Aho, J.E. Hopcroft and J.D. Ullman, *The Design and Analysis of Computer Algorithms*, Addison-Wesley (1974).

[2] A. Apostolico, The Myriad Virtues of Suffix Trees, *Combinatorial Algorithms on Words*, 85-96, Springer-Verlag, ASI F12 (1985).

[3] A. Apostolico and F.P. Preparata, Optimal Off-Line Detection of Repetitions in a String, *Theoretical Computer Science*, 22, 297-315 (1983).

[4] A. Apostolico and W. Szpankowski, Self-alignments in Words and Their Applications, *J. of Algorithms*, 13, (1992).

[5] P. Billingsley, *Ergodic Theory and Information*, John Wiley & Sons, New York (1965).

[6] A. Blumer, A. Ehrenfeucht and D. Haussler, Average Size of Suffix Trees and DAWGS, *Discrete Applied Mathematics*, 24, 37-45 (1989).

[7] R. Boyer and J. Moore, A Fast String Searching Algorithm, *Communication of the ACM*, 20, 762-772 (1977).

[8] K.L. Chung and P. Erdös, On the Application of the Borel-Cantelli Lemma, *Trans. of the American Math. Soc.*, 72, 179-186 (1952).

[9] W. Chang and E. Lawler, Approximate String Matching in Sublinear Expected Time, *Proc. of 1990 FOCS*, 116-124 (1990).

[10] I. Csiszár and J. Körner, *Information Theory: Coding Theorems for Discrete Memoryless Systems*, Academic Press, New York (1981).

[11] L. Devroye, W. Szpankowski and B. Rais, A Note on the Height of Suffix Trees, *SIAM J. Computing* 21, 48-53 (1992).

[12] Z. Galil and K. Park, An Improved Algorithm for Approximate String Matching, *SIAM J. Computing*, 19, 989-999 (1990)

[13] P. Gilham and H.L. Weith, Personal Communications.

[14] G.H. Gonnet and R. Baeza-Yates, *Handbook of Algorithms and Data Structures*, Addison-Wesley, Workingham (1991).

[15] P. Grassberger, Estimating the Information Content of Symbol Sequences and Efficient Codes, *IEEE Trans. Information Theory*, 35, 669-675 (1991).

[16] L. Guibas and A. Odlyzko, Periods in Strings, *Journal of Combinatorial Theory*, Series A, 30, 19-43 (1981).

[17] L. Guibas and A. W. Odlyzko, String Overlaps, Pattern Matching, and Nontransitive Games, *Journal of Combinatorial Theory,* Series A, 30, 183-208 (1981).

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

[19] P. Jacquet and W. Szpankowski, What Can We Learn About Suffix Trees From Independent Tries?, *Proc. 2nd Workshop on Algorithms and Data Structures*, Springer-Verlag LNCS, 519, pp. 228-239, (1991).

[20] P. Jacquet, B. Rais and W. Szpankowski, Compact Suffix Trees Resemble PATRICIA Tries: Limiting Distribution of Depth, in preparation.

[21] S. Karlin and F. Ost, Counts of Long Aligned Word Matches Among Random Letter Sequences, *Adv. Appl. Prob.*, 19, 293-351 (1987).

[22] R. Karp, A Characterization of the Minimum Cycle Mean in a Digraph, *Discrete Mathematics*, 23, 309-311 (1978).

[23] D. Knuth, *The Art of Computer Programming. Sorting and Searching* , vol. III, Addison-Wesley (1973).

[24] D. Knuth, J. Morris and V. Pratt, Fast Pattern Matching in Strings, *SIAM J. Computing*, 6, 323-350 (1977).

[25] A. Lempel and J. Ziv, On the Complexity of Finite Sequences, *IEEE Information Theory* 22, 1, 75-81 (1976).

[26] M. Lothaire, *Combinatorics on Words*, Addison-Wesley (1982).

[27] E.M. McCreight, A Space Economical Suffix Tree Construction Algorithm, *Journal of the ACM*, 23, 262-272 (1976).

[28] G. Hardy, J.E. Littlewood and G. Pólya, *Inequalities*, Cambridge University Press, Cambridge (1989)

[29] D. Ornstein and B. Weiss, Entropy and Data Compression Schemes, preprint.

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

[31] B. Pittel, Paths in a Random Digital Tree: Limiting Distributions, *Adv. Appl. Prob.*, 18, 139-155 (1986).

[32] B. Rais, P. Jacquet and W. Szpankowski, Limiting Distribution for the Depth of PATRICIA Tries, *SIAM J. Discrete Mathematics*, (1992), to appear.

[33] L.V. Romanovski, Optimization of Stationary Control of a Discrete Deterministic Process, *Cybertentics*, 3, 52-62 (1967).

[34] P. Shields, Entropy and Prefixes, *Annals of Probability*, 20, 403-409 (1992).

[35] D. Stoyan, *Comparison Methods for Queues and Other Stochastic Models*, John Wiley & Sons, Chichester (1983).

[36] W. Szpankowski, On the Height of Digital Trees and Related Problems, *Algorithmica*, 6, 256-277 (1991).

[37] W. Szpankowski, Patricia tries again revisited, *Journal of the ACM*, 37, 691-711 (1991).

[38] W. Szpankowski, Asymptotic Properties of Data Compression and Suffix Trees, *IEEE Trans. Information Theory*, submitted.

[39] W. Szpankowski, (Un)Expected behavior of typical suffix trees, *Third Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 422-431, Orlando (1992).

[40] M. Waterman, *Mathematical Methods for DNA Sequences*, CRC Press Inc., Boca Raton, (1991).

[41] P. Weiner, Linear Pattern Matching Algorithms, *Proc. of the 14-th Annual Symposium on Switching and Automata Theory*, 1-11 (1973).

[42] U. Vishkin, Deterministic Sampling – A New Technique for Fast Pattern Matching, *SIAM J. Computing*, 20, 22-40 (1991).

[43] A. Wyner and J. Ziv, Some Asymptotic Properties of the Entropy of a Stationary Ergodic Data Source with Applications to Data Compression, *IEEE Trans. Information Theory*, 35, 1250-1258 (1989).

[44] J. Ziv and A. Lempel, A Universal Algorithm for Sequential Data Compression, *IEEE Trans. Information Theory*, 23, 3, 337-343 (1977).