1991

Suffix Trees Revisited: (Un)Expected Asymptotic Behaviors

Wojciech Szpankowski
Purdue University, spa@cs.purdue.edu

Report Number:
91-063

http://docs.lib.purdue.edu/cstech/903

This document has been made available through Purdue e-Pubs, a service of the Purdue University Libraries. Please contact epubs@purdue.edu for additional information.
SUFFIX TREES REVISITED;
(UN)EXPECTED ASYMPTOTIC BEHAVIOR

Wojciech Szpankowski

CSD-TR-91-063
August 1991
Suffix trees find several applications in computer sciences and telecommunications, most notably in algorithms on strings, data compressions and codes. Despite this, very little is known about their typical behavior. We consider in a probabilistic framework 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 \( b \to \infty \). Several parameters of \( b \)-suffix trees are of interest, namely the typical depth \( D_n^{(b)} \), the depth of insertion \( L_n^{(b)} \), the height \( H_n^{(b)} \), the external path length \( E_n^{(b)} \), and so forth. We establish several results concerning typical, that is, almost sure (a.s.), behavior of these parameters. For example, we show that \( D_n^{(b)}/\log n \) converges (a.s.) to \( 1/h \) where \( h \) is entropy of the alphabet, but not the depth of insertion for which \( L_n^{(b)}/\log n \) oscillates between \( 1/h_1 \) and \( 1/h_2^{(b)} \) (a.s.) where \( 0 < h_2^{(b)} < h \leq h_1 < \infty \) are some parameters of the underlying probabilistic model. These findings are used to obtain several insights into certain algorithms on words and universal data compression schemes. As a simple consequence of our results, we settle in the negative the conjecture of Wyner and Ziv regarding the typical length of repeated subwords; we present a new surprising results concerning the length of a block in the Lempel-Ziv parsing algorithm; and finally we demonstrate how to obtain precise asymptotic results for the average time-complexity of some algorithms on strings.
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. 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], [11], [23], [24], [41]). 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. [33]). 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 underlying sequence $X$ (cf. [25], [32], [42], [43]). At last, but not least, in comparing molecular sequences (e.g., finding homology between DNA sequences) one may search for the longest run of a given motif (pattern) (cf. [14]). These, and several other problems on words, can be efficiently solved and analyzed by a clever manipulation of a data structure known as suffix tree [2], [27], [40]). In literature another names have been also coined for this structure, and among others we mention here position trees, subword trees, directed acyclic graphs, etc. (cf. [1]).

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 ([21]). A digital tree stores $n$ strings \{$_{S_1, \ldots, S_n}$\} built over a finite alphabet $\Sigma$. In such a tree every edge is labelled by a symbol (or a set of symbols) from the alphabet $\Sigma$ 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 [12], [21]). If the strings \{$_{S_1, \ldots, S_n}$\} are statistically independent and every edge is labelled by a single symbol from $\Sigma$, then the resulting digital tree is called a regular (or independent) trie ([1], [12], [21]). If all unary nodes of a trie are eliminated, then the tree becomes PATRICIA tries (cf. [12], [21], [38]). 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 labelled by a letter from the alphabet $\Sigma$. 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 [12] coined a name PAT for such a suffix tree to resemble the name PATRICIA used for compact tries. Here, we also adopt this name. In addition, however, we introduce a family of suffix trees parametrized by an integer $b \geq 1$. 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 modify this definition and assume that external nodes of $b$-suffix trees can contain up to $b$ suffixes. Note that such a suffix tree correspond 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 ($b = 1$) on one of the extreme and compact suffix trees ($b \rightarrow \infty$) on the other extreme. This allows to assess some properties of PAT trees in a unified and substantially easier manner (e.g., compare [36] and [38] where regular tries and PATRICIA tries are analyzed respectively).

Suffix trees have found a wide variety of applications in algorithms on words including: the longest repeated substring ([40]), squares or repetitions in strings ([3]), string statistics ([3]), string matching ([9], [41]), approximate string matching ([23], [24], [9]) string comparison, compression schemes ([25]), implementation of Lempel-Ziv algorithm ([32]), genetic sequences, biologically significant motif patterns in DNA [9]), sequence assembly ([9]), approximate-overlaps ([9], and so forth. It is fair to say that suffix trees are most widely used data structure in algorithms on words. Despite this, very little is known about their behavior in a probabilistic framework. Recently, Chung and Lawler [9] used some elementary property of a typical behavior 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 suffix trees behavior in a probabilistic framework.

In this paper, we offer a characterization of suffix trees. Our probabilistic model is a very general one, namely we allow symbols of a string to be dependent. More precisely, we assume that 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 (cf. Remark 2(iv) in Section 2). Moreover, instead of concentrating on a specific algorithm we present a list of results concerning several parameters of a suffix tree namely: the typical depth $D_n^{(b)}$, depth of insertion $L_n^{(b)}$, height $H_n^{(b)}$ and the shortest path $s_n^{(b)}$. For example, the typical depth $M_n^{PAT}$ for the PAT tree built from the string $P$ then $T$ where $P$ and $T$ are the pattern and the text respectively, is used by Chung and Lawler [9] in their design of an approximate string matching algorithm. On the other hand, the depth of insertion $L_n^{(1)}$ of a noncompact suffix tree is of prime interest to the complexity of Lampel-Ziv universal compression schemes, and $L_n^{(1)}$ is responsible for a dynamic behavior of many algorithms on words. Furthermore, the
height and the shortest path indicate how balance is a typical suffix tree, that is, how much one has to worry about worst-case situations. Finally, the depth of a particular suffix can be used to unify various analyses of long runs of a pattern, say $B$ (cf. [14]). For example, if $B = 1$, then the longest run of 1’s corresponds to the depth of the right-most node in a suffix tree built over a given sequence $X$.

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 path $s_n^{(b)}/\log n$ and the normalized depth $D_n^{(b)}/\log n$ almost surely (a.s.) converge to $1/h_2^{(b)}$, $1/h_1$ and $1/h$ respectively, where for every $1 \leq b \leq \infty$ we have $h_2^{(b)} < h < h_1$. In the above, $h$ is the entropy of the alphabet $\Sigma$, while the parameters $h_1$ and $h_2^{(b)}$ depend on the underlying probabilistic model. The most interesting behavior reveals the depth of insertion $L_n^{(b)}$ which converges in probability (pr.) to $(1/h)\log n$ but not almost surely. We prove that almost surely $L_n^{(b)}/\log n$ oscillates between $1/h_1$ and $1/h_2^{(b)}$. More interestingly, almost sure behavior of the compact suffix tree can be deduced from the appropriate asymptotics of the $b$-suffix trees by taking $b \to \infty$. More precisely, if we append superindex PAT to the appropriate parameters of a compact suffix tree, then we can prove that $\lim_{n \to \infty} H_n^{PAT}/\log n = \lim_{b \to \infty} \lim_{n \to \infty} H_n^{(b)}/\log n = h^{(\infty)}$, 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 since naturally $\lim_{n \to \infty} \lim_{b \to \infty} P_n^{(b)} = 1$ where $P_n^{(b)}$ is any of the above parameters of the suffix tree. It is worth mentioning that all these results are obtained in a uniform manner by a novel technique that encompasses the so called string-ruler approach (cf. [18], [30]) and mixing condition technique. The details are discussed in Section 3. Finally, using these results we: (i) settle in the negative the conjecture of Wyner and Ziv [42] concerning the length of the repeated pattern in a universal compression scheme (cf. [39]); (ii) determine the length of the last block in the Lampel-Ziv parsing algorithm [25]; (iii) establish average complexity of some exact and approximate pattern matching algorithms (cf. [7], [22], [9]); and so forth.

Asymptotic analyses of suffix trees are very scanty in literature, and most of them deal with noncompact suffix trees. To the best of our knowledge, there are no probabilistic results on $b$-suffix trees and compact suffix trees. This can be easily verified by checking Section 7.2 of Gonnet and Baeza-Yates' book [12] 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 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. [10], and the limiting distribution of the typical depth in a suffix tree is reported in Jacquet and Szpankowski [18]. Recently, Szpankowski [39] extended these
results to a more general probabilistic model. Finally, 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 [18]. Some related topics were discussed by Guibas and Odlyzko in [15] and [16]. Our findings were inspired by a seminal paper of Pittel [30] who considered a typical behavior of a regular trie constructed from independent words. Consequently, this paper can be viewed as a direct extension of Pittel's results to dependent tries namely suffix trees.

This paper is organized as follows. In the next section we formulate our main results and present several consequences of them. Among others, we prove some open problems on data compression schemes (cf. [42], [25]), and clarify "expected" behavior of suffix tree under "unreasonable" general probabilistic assumptions (cf. [1] page 349, [9]). We also intuitively explain why compact suffix trees can be considered as a limiting b-suffix trees as b → ∞. Section 3 contains all formal proofs. These proofs are obtained within a framework of a special methodology that encompasses a novel technique called string-ruler approach and mixing condition technique.

2. MAIN RESULTS AND THEIR CONSEQUENCES

A suffix tree is a digital tree built from suffixes of an (unbounded) sequence \{X_k\}_{k=1}^{\infty} of symbols from an alphabet \( \Sigma \) of size \( V \). More precisely, let \( X = x_1x_2x_3 \cdots \), then the \( i \)th suffix \( S_i \) of \( X \) is \( S_i = x_ix_{i+1} \cdots \). By \( S_n \) we denote a digital tree built from the set \{\( S_1, S_2, \ldots, S_n \)\} of \( n \) first 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 \( \Sigma \). Figure 1(a) shows a noncompact suffix tree built from the first six suffixes of \( X = 0101101110 \cdots \). A compact suffix tree called PAT tree (cf. [12]) 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], [40]).

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 noncompact and compact suffix trees lie on two extremes of the spectrum of b-suffix trees, namely 1-suffix trees is a noncompact suffix tree and compact suffix trees is obtained by taking \( b \rightarrow \infty \). For the purpose of our analysis, however, a modified definition of b-suffix trees is more convenient. Hereafter, by suffix tree we mean a digital tree built from \( n \) first suffixes of \( X \) that can store up to \( b \) suffixes in an external node. We denote such a suffix tree as \( S_n^{(b)} \). 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
Figure 1: Suffix trees built from the first six suffixes of $= 01010110 \cdots$
tree with $b$ nodes, then the latter definition of $b$-suffix tree corresponds to the former one. Nevertheless, we shall argue that even with such a modification we can asymptotically obtain several parameters of PAT from $b$-suffix trees by taking $b \to \infty$.

In this paper, we analyze six parameters of $b$-suffix trees $S_n^{(b)}$, namely: the $m$th depth $L_n^{(b)}(m)$, the height $H_n^{(b)}$ and the shortest path $s_n^{(b)}$, the typical depth $D_n^{(b)}$, the depth of insertion $L_n^{(b)}$ and the external path length $E_n^{(b)}$. The depth of the $m$th suffix is equal to the number of internal nodes in a path from the root to the external node containing this suffix. Then,

$$H_n^{(b)} = \max_{1 \leq m \leq n} \{L_n^{(b)}(m)\}, \quad s_n^{(b)} = \min_{1 \leq m \leq n} \{L_n^{(b)}(m)\},$$

(2.1)

that is, the height and the shortest path are the longest and the smallest paths in $S_n^{(b)}$ respectively. In the performance evaluation of algorithms on words and data compression schemes, the typical depth $D_n^{(b)}$, the depth of insertion $L_n^{(b)}$, and the external path length $E_n^{(b)}$ are even more important. The depth of insertion $L_n^{(b)}$ is the depth of the $n+1$ external node after insertion of the $(n+1)$st suffix $S_{n+1}$ into the suffix tree $S_n^{(b)}$, that is, $L_n^{(b)} = L_n^{(b)}(n+1)$. Finally, $D_n^{(b)}$ is defined as the depth of a randomly selected suffix, and the external path length $E_n^{(b)}$ is the sum of all depths $L_n^{(b)}(m)$ for $1 \leq m \leq n$. In other words,

$$E_n^{(b)} = \sum_{m=1}^{n} L_n^{(b)}(m) \quad \text{and} \quad D_n^{(b)} = \frac{E_n^{(b)}}{n}.$$  

(2.2)

Note that $D_n^{(b)}$ can be interpreted as a successful search length in a suffix tree.

It turns out that another characterization of these parameters is more useful for our analysis. For a suffix tree $S_n^{(b)}$ built from suffixes $S_1, S_2, ..., S_n$, we define the self-alignment $C_{i_1, ..., i_{b+1}}$ between $b+1$ suffixes, say $S_{i_1}, ..., S_{i_{b+1}}$, as the length of the common prefix of these $b+1$ suffixes. Then, the following is easy to establish (cf. Szpankowski [37])

$$L_n^{(b)}(i_{b+1}) = \max_{1 \leq i_1, ..., i_b \leq n} \{C_{i_1, ..., i_{b+1}}\} + 1$$

(2.3a)

$$H_n^{(b)} = \max_{1 \leq i_1, ..., i_{b+1} \leq n} \{C_{i_1, ..., i_{b+1}}\} + 1,$$

(2.3b)

$$L_n^{(b)} = \max_{1 \leq i_1, ..., i_b \leq n} \{C_{i_1, ..., i_{b+1}}\} + 1.$$  

(2.3c)

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 $i_1, ..., i_{b+1}$ but rather on the differences $d_k = i_{k+1} - i_k$. So, we also write $C_{i_1, i_1+d_1, ..., i_b+d_b}$.

Our purpose is to investigate the behavior of a random $b$-suffix tree in a general probabilistic framework. At the beginning, we only assume that $\{X_k\}_{k=1}^{\infty}$ is a stationary ergodic
sequence of symbols generated from a finite alphabet $\mathcal{A}$. In such a model, define a partial sequence $X_m^n$ as $X_m^n = (X_m, \ldots, X_n)$ for $m < n$, and let for every $n \geq 1$ the nth order probability distribution for $\{X_k\}$ be

$$P(X^n_k) = \Pr\{X_k = x_k, 1 \leq k \leq n, x_k \in \mathcal{A}\}.$$  \tag{2.4}

The entropy of $\{X_k\}$ is defined in a standard manner as

$$h = \lim_{n \to \infty} \frac{E \log P^{-1}(X^n_k)}{n},$$  \tag{2.5}

and we introduce three additional parameters (cf. [30]), namely

$$h_1 = \lim_{n \to \infty} \frac{\max \{\log P^{-1}(X^n_k), P(X^n_k) > 0\}}{n} = \lim_{n \to \infty} \frac{\log(1/\min \{P(X^n_k), P(X^n_k) > 0\})}{n},$$  \tag{2.6a}

$$h_2^{(b)} = \lim_{n \to \infty} \frac{\log(\sum_{i=1}^{b+1} P(X^n_k) P_{i+1}^{b+1}(X^n_k))^{1/(b+1)}}{n} = \lim_{n \to \infty} \frac{\log \left( \sum_{i=1}^{b+1} P_i^{b+1}(X^n_k) \right)^{1/(b+1)}}{n},$$  \tag{2.6b}

$$h_3 = \lim_{n \to \infty} \frac{\min \{\log P^{-1}(X^n_k), P(X^n_k) > 0\}}{n} = \lim_{n \to \infty} \frac{\log(1/\max \{P(X^n_k), P(X^n_k) > 0\})}{n}.$$  \tag{2.6c}

The existence of the limit in (2.5) is guaranteed by Kolmogorov-Sinai Theorem (cf. [5]), and the existence of $h_1$, $h_3$, and $h_2^{(1)}$ was established by Pittel [30] who also noticed that $0 \leq h_2^{(1)} \leq h \leq h_1$. A generalization to an arbitrary $b$ is easy, and left for the reader. In passing, we note that by the inequality on means [28] we have

$$\lim_{b \to \infty} h_2^{(b)} = h_3.$$  \tag{2.6d}

Remark 1.

(i) Bernoulli Model. In this commonly used model (cf. [4], [6], [9], [10], [15], [16], [21], [31], [36], [37], and [38]) symbols from the alphabet $\mathcal{A}$ are generated independently, that is, $P(X^n_k) = P^n(X^n_k)$. In particular, we assume that the ith symbol from the alphabet $\mathcal{A}$ is generated according to the probability $p_i$, where $1 \leq i \leq V$ and $\sum_{i=1}^{V} p_i = 1$. It is easy to notice that $h = \sum_{i=1}^{V} p_i \log p_i^{-1}$ ([5]), and from definition (2.3) we find that $h_1 = \log(1/p_{\min})$, $h_3 = \log(1/p_{\max})$ and $h_2^{(b)} = 1/(b+1) \log(1/P_b)$ where $p_{\min} = \min_{1 \leq i \leq V} \{p_i\}$, $p_{\max} = \max_{1 \leq i \leq V} \{p_i\}$, 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. [37]).

(ii) Markovian Model. In this model (cf. [17], [19], [30]) 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. We define a transition probability as $p_{i,j} = \Pr\{X_{k+1} = j \in \mathcal{A} | X_k = i \in \mathcal{A}\}$. 

8
The transition matrix is denoted by $P = \{p_{i,j}\}_{i,j=1}^V$. It is well known that the entropy $h$ can be computed as $h = -\sum_{i,j=1}^V \pi_i p_{i,j} \log p_{i,j}$ where $\pi_i$ is the stationary distribution of the Markov chain. The other quantities are a little harder to evaluate. Szpankowski [37] (see also Pittel [30] for $b = 1$) evaluated the height of a regular tries with Markovian dependency, and show that the parameter $h_2^{(b)}$ is a function of the largest eigenvalue $\theta_b$ of the matrix $P_{[b+1]} = P \circ P \ldots \circ P$ where $\circ$ represents the Schur product of $b+1$ matrices $P$ (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 to refer to Pittel [30] who cited a nonprobabilistic result of Romanovski who proved that $h_1 = \min_C \{\ell(C)/|C|\}$ and $h_3 = \max_C \{\ell(C)/|C|\}$ where the minimum and the maximum are taken over all simple cycles $C = \{\omega_1, \omega_2, \ldots, \omega_v, \omega_1\}$ for some $v \leq V$ such that $\omega_i \in A$, and $\ell(C) = -\sum_{i=1}^V \log p_{i,i+1 \mod V}$. 

To complete our description of the probabilistic model, we add some mixing conditions (cf. [5]) on the sequence $\{X_k\}_{k=-\infty}^\infty$. Let $F_{m+1}^m$ be a $\sigma$-field generated by $\{X_k\}_{k=m}$ for $m \leq n$. It is said that $\{X_k\}$ satisfies mixing condition if there exist two constants $c_1 \leq c_2$ and integer $d$ such that for all $-\infty \leq m \leq m + d \leq n$ the following holds

$$c_1 \Pr\{A\} \Pr\{B\} \leq \Pr\{AB\} \leq c_2 \Pr\{A\} \Pr\{B\} \quad (2.7a)$$

where $A \in F_{m+1}^m$ and $B \in F_{m+d}^m$. In some statements of our results, we need a stronger form of the above mixing condition, namely strong $\alpha$-mixing condition which becomes

$$(1 - \alpha(d))\Pr\{A\} \Pr\{B\} \leq \Pr\{AB\} \leq (1 + \alpha(d))\Pr\{A\} \Pr\{B\} \quad (2.7b)$$

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

Finally, for compact suffix trees (i.e., PAT trees) we need one more condition. Let $\omega_i \in \Sigma$ for $1 \leq i \leq n$. Define $P(\omega_1, \ldots, \omega_n) = \Pr\{X_1^n = (\omega_1, \ldots, \omega_n)\}$. Then, for PAT trees we shall require the following

$$P(\omega_1, \ldots, \omega_n) \leq \rho P(\omega_1, \ldots, \omega_{n-1}) \quad (2.7c)$$

for some $0 < \rho < 1$.

Now we ready to 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.

**Theorem 1.** Let $\{X_k\}$ be a stationary ergodic sequence satisfying the strong $\alpha$-mixing condition (2.7b) together with $h_1 < \infty$ and $h_2 > 0$. 

9
(i) \textit{b-Suffix Trees.} Fix \( b \). Then

\[
\lim_{n \to \infty} \frac{s_n^{(b)}}{\log n} = \frac{1}{h_1} \quad \text{(a.s.)}
\]  

provided

\[
\alpha(n) = O(n^\rho n^n)
\]

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

\[
\lim_{n \to \infty} \frac{H_n^{(b)}}{\log n} = \frac{1}{h_2^{(b)}} \quad \text{(a.s.)}
\]

provided the coefficient \( \alpha(d) \) in (2.7b) fulfills the following

\[
\sum_{d=0}^{\infty} \alpha^2(d) < \infty \quad .
\]

(ii) \textit{Compact Suffix Tree.} Almost sure behavior of a compact suffix tree follows from the (a.s.) behavior of \( b \)-suffix trees by taking in (2.8) and (2.9) the limit as \( b \to \infty \), that is,

\[
\lim_{n \to \infty} \frac{s_n^{\text{PAT}}}{\log n} = \frac{1}{h_1} \quad \text{(a.s.)} \quad \lim_{n \to \infty} \frac{H_n^{\text{PAT}}}{\log n} = \frac{1}{h_3^{\text{PAT}}} \quad ,
\]

provided (2.7c) holds together with condition (2.9) for \( s_n^{\text{PAT}} \) and condition (2.11) for \( H_n^{\text{PAT}} \) respectively. \( \blacksquare \)

Our next main results deal with the typical depth \( D_n^{(b)} \) 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 (2.7b), and let (2.9) hold too. Assume also that \( 1 \leq b < \infty \).

(i) \textit{Convergence in Probability.} For \( h < \infty \) we have

\[
\lim_{n \to \infty} \frac{L_n^{(b)}}{\log n} = \lim_{n \to \infty} \frac{P_n^{(b)}}{\log n} = \frac{1}{h} \quad \text{(pr.)} \quad .
\]

The same holds for compact suffix tree provided (2.7c) is true (i.e., we may take \( b \to \infty \) in the above but since the entropy \( h \) is independent of \( b \), (2.14) is unchanged).

(ii) \textit{Almost Sure Convergence of the Typical Depth} \( D_n \). 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,

\[
\sum_{n=1}^{\infty} P(B_n) < \infty \quad .
\]  

\[
(2.15a)
\]
Then,
\[
\lim_{n \to \infty} \frac{D_n(b)}{\log n} = \lim_{n \to \infty} \frac{E_n(b)}{n \log n} = \frac{1}{h} \quad (a.s.) \quad (2.15b)
\]

The above is true also for the compact suffix tree provided (2.7c) is satisfied.

(iii) Almost Sure Behavior of the Depth of Insertion \(L_n\). As in (ii) we assume strong mixing condition (2.7b) together with \(h_1 < \infty\) as well as \(h_2 > 0\). Then, we have the following result concerning the depth of insertion for \(b < \infty\)

\[
\lim \inf_{n \to \infty} \frac{L_n(b)}{\log n} = \frac{1}{h_1} \quad (a.s) \quad \lim \sup_{n \to \infty} \frac{L_n(b)}{\log n} = \frac{1}{h_2(b)} \quad (2.16)
\]

For the compact suffix tree (2.16) holds with \(h_2(b)\) replaced by \(h_3\), that is, we formally obtain almost sure behavior for the compact suffix tree by taking \(b \to \infty\) and assuming (2.7c).

Remark 2

(i) How restrictive are conditions (2.9) and (2.15a)? Let us first deal with (2.9). Possibly condition (2.9) is the best possible due to recent results of Paul Shields who constructed an ergodic mixing sequence for which (2.9) is not fulfilled even in probability (cf. [34]). Nevertheless, (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 \(\alpha(n) = 0\). In the Markovian model, it is known (cf. [5]) that for a finite state Markov chain the coefficient \(\alpha(n)\) decays exponentially, that is, for some \(c > 0\) and \(\rho < 1\) we have \(\alpha(n) = cp^n\), as needed for (2.9). Regarding (2.15a), we know that it holds at least for the Bernoulli and Markovian models but generally not for all ergodic stationary sequences (cf. [33], [34]). We believe that (2.15a) is included in (2.9). In [33] Shields constructed an ergodic stationary sequence for which (2.15b) does not hold. In passing, we also note that the condition (2.7c) holds in both above models. In the Markovian model, however, one needs additional assumption that all transition probabilities are positive and strictly smaller than one.

(ii) How to prove part (iii) of Theorem 2? One can view the behavior of \(D_n(b)\) and \(L_n(b)\) as a surprise. Both quantities characterize the depth of a tree, but \(L_n(b)\) is responsible for a dynamic behavior of the suffix tree while \(D_n(b)\) for the "average" one. It is also easy to notice that the main reason for \(L_n(b)\) oscillation is a "tiny" unbalance in the height and the shortest path discovered in Theorem 1 (the typical depth \(D_n(b)\) behaves nicely since these oscillations are smoothed by the sum in (2.2)). More formally, 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
For the lim inf of $L_{n}^{(b)}$ we consider the shortest path $s_{n}^{(b)}$ and repeat the above arguments (cf. [30] and [30]). Intuitively, the depth of insertion falls always between the shortest path and the height. Since both the latter quantities have almost sure limits that tend to infinity, we expect the same kind of behavior for the depth of insertion. The unbalance in the (a.s.) limits for the height and the depth (since $h_{1} > h_{2}^{(b)}$) leads to different lower and upper (a.s) behavior of the depth of the insertion $L_{n}^{(b)}$. In passing, we note that the only $b$-suffix tree that has (a.s.) limit for the depth of insertion $L_{n}^{(b)}$ is PAT tree with the symmetric alphabet (i.e., $p_{i} = 1/V$ for $i1 \leq i \leq V$). Indeed, by Theorem 2 (iii) in this case $\lim_{n \to \infty} L_{n}^{PAT} / \log n = \log V$ (a.s.).

(iii) Compact Suffix Tree as a Limit of $b$-Suffix Tree. 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 a PAT tree can be obtained as a limit when $b$ tends to infinity. However, we conjecture that there exists a sequence $a_{n}$ (e.g., in the case of parameters discussed in this paper we have $a_{n} \sim \log n$) such that for several parameters $P_{n}^{(b)}$ we have

$$\lim_{n \to \infty} P_{n}^{PAT} / a_{n} = \lim_{b \to \infty} \lim_{n \to \infty} P_{n}^{(b)} / a_{n}.$$  \hspace{1cm} (2.17a)

However, we can easily give a formal proof of this interesting fact for every parameter discussed in this section. We consider first all parameters except the height. Let us denote them as $P_{n}^{(b)}$. 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

$$\lim_{n \to \infty} \frac{P_{n}^{PAT}}{\log n} \leq \lim_{n \to \infty} \frac{P_{n}^{(1)}}{\log n}.$$  \hspace{1cm} (2.17b)

This immediately establishes the upper bound part of (2.17a) 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,

$$\lim_{n \to \infty} \frac{H_{n}^{PAT}}{\log n} \leq \lim_{b \to \infty} \lim_{n \to \infty} \frac{H_{n}^{(b)}}{\log n} = \frac{1}{h_{3}}$$  \hspace{1cm} (2.17c)

This completes the upper bound in (2.17a).

For the lower bound we use condition (2.7c). 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 (called a feasible path) in a suffix tree such that the length of it is (a.s.) asymptotically equal to $\log n / h_{3}$ and $\log n / h_{1}$ respectively. But, this is immediate from
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 (a.s.) such a path is smaller equal to $\log n/h$ which completes the lower bound arguments in the proof for the depth.

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. 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 a 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 size $O(1)$ (e.g., comparing the depth of regular tries and PATRICIA we know that $ED_n^P - ED_n^T = O(1)$ [37], [38], but for the height we have $EH_n^P - EH_n^T = \log n$ [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. [2]) the string $\{X_k\}_{k=1}^n$ has finite length $n$, and it is terminated by a special symbol that does not belong to the alphabet $\Sigma$, 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 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 $D'_n/\log n \sim 1/h$ (a.s.) 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 $O(\log n)$ last 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. □

Theorem 1 and 2 find several applications in combinatorial problems on words, data compressions and molecular biology. In general, our findings are widely used in problems dealing with repeated patterns and other regularities on strings. As an illustration, we solve three problems on words using Theorem 2. Two of them deal with data compressions, and the last one discusses the average time-complexity of the exact string matching algorithm.
proposed recently by Chung and Lawler [9]. The first data compression example solves the conjecture of Wyner and Ziv [42]. The second one identifies the (a.s.) behavior of the block length in the well known parsing algorithm of Lempel and Ziv [25], which was an open problem up to now. We also point out that several other practical algorithms on words (i.e., with good average complexity) are of immediate consequence of our findings. The reader is referred to Apostolico and Szpankowski [4] for more details.

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 server and the receiver. Instead of transmitting next \( L_n \) symbols beyond the data base sequence, the sender can "look backward" into the data base and verify whether some \( L_n \) symbols have already appeared in the data base. If this is the case, then instead of sending \( L_n \) symbols the server transmits only the location of these symbols in the data base and the length of \( L_n \). More precisely, let the data base be represented by a subsequence of size \( n \) \( \{X_k\}_{k=1}^{\infty} \) of a stationary ergodic sequence \( \{X_{dk}\}_{k=1}^{\infty} \). For every \( n \) let \( L_n \) be the smallest integer \( L > 0 \) such that \( X_{m+L}^n = X_{m+1}^{n+L} \) for all \( 1 \leq m \leq n \). Wyner and Ziv [42] asked about typical behavior of \( L_n \). They have 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. In [39] we have shown 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 [39]), and we prove that \( L_n / \log n \) does not converge (a.s.) but rather oscillates between \( 1/h_1 \) and \( 1/h_2^{(1)} \).

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]. Two important features of the parsing are: (i) the blocks are pairwise distinct; (ii) each block that occurs in the parsing has already been seen somewhere to the left. For example, for \( \{X_k\} = 11010100111\cdots \) 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, and so one. In Figure 2 we show how to perform the parsing using a sequence of noncompact suffix trees (cf. [13]). Note that the length of a block is a subsequence of depth of insertions \( L_n^{(1)} \). More precisely, if \( \ell_n \) is the length of the \( n \)th block in the Lempel-Ziv parsing algorithm, then Figure 2 suggests the following
relationship \( \ell_n = L_1^{(1)} + \sum_{k=0}^{n-1} \ell_k \). For example, in Fig.1 we have \( \ell_0 = L_0^{(1)} = 1 \), \( \ell_1 = L_1^{(1)} = 2 \), \( \ell_2 = L_0^{(1)} + \ell_1 = L_3^{(1)} = 5 \), and \( \ell_3 = L_1 + 2 + 5 = 3 \), and so forth. To obtain \( \text{a.s.} \) behavior of the block length \( \ell_n \), we note that

\[
\lim_{n \to \infty} \frac{\ell_n}{\log n} = \lim_{n \to \infty} \frac{L_1^{(1)} + \sum_{k=0}^{n-1} \ell_k}{\log \left( \sum_{k=0}^{n-1} \ell_k \right)} = \frac{\log \left( \sum_{k=1}^{n-1} \ell_k \right)}{\log n}.
\]  

(2.18)

We first estimate the second term in (2.18). One immediately obtains

\[
1 \leq \frac{\log \left( \sum_{k=0}^{n-1} \ell_k \right)}{\log n} \leq \frac{\log \left( \sum_{k=0}^{n-1} L_k^{(1)} \right)}{\log n} \leq \frac{\log \left( \sum_{m=0}^{n} L_n^{(1)}(m) \right)}{\log n} \to 1 \quad \text{\( \text{a.s.} \)},
\]

where the RHS of the above is a direct consequence of Theorem 2(ii). But then, (2.18) leads to the following result.

**Corollary 3.** Let \( \{X_k\}_{k=1}^{\infty} \) be a strongly mixing stationary sequence satisfying (2.9) and (2.15a). Then

\[
\liminf_{n \to \infty} \frac{\ell_n}{\log n} = \frac{1}{h_1} \quad \text{\( \text{a.s.} \)} \quad \limsup_{n \to \infty} \frac{\ell_n}{\log n} = \frac{1}{h_2^{(1)}}.
\]  

(2.19)

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

**PROBLEM 3.** String Matching Algorithms
Recently, Chung 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 symmetric Bernoulli model. Our Theorem 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 Chung-Lawler's analysis is an observation that there exists such $d_{m,n}$ that a substring of the text $T$ of length $d_{m,n}$ is not a substring of the pattern $P$. This can be verified by building first a compact suffix tree for $P$ and then insert suffixes of $T$ (Le., constructing a compact suffix tree for $P\$T$ where $\$ is a special character). But then, on may observe that $d_{m,n}$ is equivalent to the typical depth $D\text{AT}$, and therefore $d_{m,n} \sim (1/h) \log(m+n)$ (a.s.). This further implies that the complexity $C_n$ of the algorithm becomes $C_n \sim (n/hm) \log(m+n)$ (a.s), which is a much stronger version of Chung-Lawler's result for a more general probabilistic model. In passing, we note that our findings can be directly used to estimate almost sure running time for the Knuth-Morris-Pratt algorithm [22] and the Boyer-Moore algorithm. [7] Several other approximate string matching algorithms can be analyzed in a similar manner. □

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 [18] proved that the distribution $F_{n}^{T}(x)$ of the typical depth in independent tries and the distribution $F_{n}^{S}(x)$ of the typical depth $D_{n}^{(1)}$ in suffix trees, do not differ too much. More precisely, in [18] it is proved that for large $n$ there exist such $\beta > 1$ and $\varepsilon > 0$ that

$$|F_{n}^{T}(k) - F_{n}^{S}(k)| = O \left( \frac{1}{n^{\varepsilon} \beta^{k}} \right).$$ (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 $A$, the normalized depth $(D_{n}^{(1)} - ED_{n}^{(1)})/\text{var}D_{n}^{(1)}$ converges in distribution to the

16
standard normal distribution \( \mathcal{N}(0, 1) \) with mean and variance as below

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

\[
\text{var} D_n^{(1)} = \frac{h_3 - h^2}{h^3} \log n + C + P_2(\log n) + O \left( \frac{1}{n^\epsilon} \right), \tag{2.21b}
\]

for some \( \epsilon > 0 \), where \( h_3 = \sum_{l=1}^V p_i^2 \log p_i \), 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 [36]. In the symmetric case, the variance becomes

\[
\text{var} D_n^{(1)} = \frac{\pi^2}{6 \log^2 V} + \frac{1}{12} + O \left( \frac{1}{n^\epsilon} \right), \tag{2.21c}
\]

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 uniformly in \( x \geq 0 \) we have

\[
\lim_{n \to \infty} \sup_x | \text{Pr}\{ D_n^{(1)} \leq \log_V (n) + x \} - e^{-V^{-x}} | = 0. \tag{2.21d}
\]

We conjecture that the same type of limiting distributions can be obtained for \( b \)-suffix trees and for the Markovian model. The latter is due to the fact that (2.20) seems to hold in the Markovian case, and then one can apply recent result of Jacquet and Szpankowski [17] 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. Also, the number of internal nodes in a \( b \)-suffix tree is of prime interest to the space complexity. 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. There is, however, no result regarding the limiting law of the height. These seem to be difficult problems.

(ii) **How well is suffix tree balanced?** In the worst case suffix tree may be degenerate and the worst case height can be as much as \( n \). But our analysis indicates that this happen very, very rarely. In fact, our Theorem 2, show that the typical depth of a suffix is equal to \((1/h) \log n \) (a.s.). 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 the one of a complete tree since the depth \( D_n^{(0)} \) with high probability is equal to \( \log_V n \), and almost surely
is not heigher than $H_n^{(b)} \sim (1 + 1/b) \log n$ but not smaller than $s_n \sim \log n$. Such a tree can be called highly balanced (in a probability sense), and as our analysis shows there is no "real" justification for additional rebalancing of the tree in order to assure a good worst case behavior, as it is done in AVL-tree and other (artificially) balanced trees (cf. [1]). \(\Box\)

3. ANALYSIS AND PROOFS

We present now a formal proof of Theorem 1 concerning a typical behavior of the height $H_n^{(b)}$ and the shortest path $s_n^{(b)}$. Then, we establish parts (i) and (ii) of Theorem 2 for the typical depth $D_n^{(b)}$. We remind the reader that Theorem 2 (iii) was proved already in Remark 2 (ii), and 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 notations of the height $H_n$, the shortest path $s_n$ and the typical path $D_n$.

Throughout the proof we use a novel technique that encompasses the mixing condition and another new technique called the string-ruler approach that was already applied by Pittel [30] and extended by Jacquet and Szpankowski [18]. 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 $W$. Usually we deal with fixed length rulers $w_k$ where $k$ is the length of the string-ruler. Let $W_k$ be the set of all strings $w_k$, that is, $W_k = \{ w \in A^k : |w| = k \}$, where $|w|$ is the length of $w$. We write $w_k^\ell$ to mean a concatenation of $\ell$ strings $w_k$ from $W_k$, 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 we write $\sum_{W_k} f(w_k) = \sum_{w_k \in W_k} f(w_k)$ 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 analysis of suffix trees and other combinatorial structures on words.

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

$$d_0 = 0 \leq d_1 \leq \cdots \leq d_i \leq k \leq d_i+1 \leq \cdots \leq d_b .$$

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

$$\Pr\{C_{1,1+d_1,\ldots,1+d_1+\cdots+d_b} \geq k\} = \sum_{W_d} P\left(\frac{w_d^{k+\frac{b+1}{2}+\cdots+d_i}}{w_d^{(b+1)-i}}\right)$$

and

$$= \sum_{W_d} P\left(\frac{w_d^{k+1-i}}{w_d^{(b+1)-i}}\right) ,$$

$$= \sum_{W_d} P\left(\frac{w_d^{(b+1)-i}}{w_d^{(b+1)-i}}\right) ,$$

$$= \sum_{W_d} P\left(\frac{w_d^{(b+1)-i}}{w_d^{(b+1)-i}}\right) ,$$

$$= \sum_{W_d} P\left(\frac{w_d^{(b+1)-i}}{w_d^{(b+1)-i}}\right) ,$$

$$= \sum_{W_d} P\left(\frac{w_d^{(b+1)-i}}{w_d^{(b+1)-i}}\right) .$$
where $w_d$ is a prefix of $w_d$, and $[x]$ is the floor function. Two cases are of particular interest, namely: (i) if $k \leq d_1 \leq \cdots \leq d_b$, then

$$\Pr\{C_{1,1+d_1,\ldots,1+d_1+\cdots+d_b} \geq k\} = \sum_{w_k} P(w_k^{b+1});$$  \hspace{1cm} (3.2b)

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

$$\Pr\{C_{1,1+d_1,\ldots,1+d_1+\cdots+d_b} \geq k\} = \sum_{w_d} P \left( w_d^{\frac{k}{d_1}+\frac{d_1}{d_1}+\cdots+\frac{d_b}{d_b} w_d \right).$$  \hspace{1cm} (3.2c)

\textbf{Proof.} It is illustrative to start with $b = 1$. In such a 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 $Z_k = w_d^{\frac{k}{d_1} w_d}$. Then, (3.2) (in fact (3.2c)) is a simple consequence of this. 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 $k \leq d_1 \leq \cdots \leq d_b$, 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 $d_1 \leq \cdots \leq d_b \leq k$, that is, there are mutual overlaps between any two consecutive suffixes. Then, there must exist a word $w_d$ of length $d = \text{gcd}(d_1, \ldots, d_b)$ such that $Z_k = w_d^{\frac{k}{d_1}+\frac{d_1}{d_1}+\cdots+\frac{d_b}{d_b} w_d}$, which leads to (3.2c). Finally, the general solution (3.2a) is a combination of the two above cases. \[\square\]

Finally, we point out that there is another definition of the parameter $h_2$ which is more useful for our purpose. We have

$$\lim_{n \to \infty} \frac{\log \left( \sum_{w_n} P(w_n^{b+1}) \right)}{(b + 1)n} = \lim_{n \to \infty} \frac{\log \left( \sum_{w_n} P^{b+1}(w_n) \right)}{(b + 1)n}. \hspace{1cm} (3.3)$$

Indeed, the above is a simple consequence of the weak mixing condition (2.7a) and the fact that $b$ is fixed.

\subsection*{3.1 Height of b-Suffix Trees}

We prove now 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.

\textit{Upper Bound}

We start with the representation (2.3b) for the height $H_n$. Hence by Boole's inequality the distribution of the height can be bounded from the above by a sum of marginal distributions
of the self-alignments. In other words, by (3.2)

\[ \Pr\{H_n \geq k\} = \Pr\left\{ \max_{1 \leq d_1, \ldots, d_b \leq n} \{C_1, 1+d_1, \ldots, 1+d_1+\cdots+d_b \} \geq k \right\} \]

\[ \leq \sum_{d_1=1}^{n} \cdots \sum_{d_b=1}^{n} \Pr\{C_1, 1+d_1, \ldots, 1+d_1+\cdots+d_b \geq k \} \]

\[ \leq n \sum_{i=1}^{b} \binom{b}{i} k^i n^{b-i} \sum_{W_d} P \left( w_d^{[\frac{1}{d}]} + \frac{d_1+\cdots+d_i}{d} w_d^{b-i} \right). \]

The last sum can be estimated as follows

\[ \sum_{W_d} P \left( w_d^{[\frac{1}{d}]} + \frac{d_1+\cdots+d_i}{d} w_d^{b-i} \right) \leq (A) \ c^{b-i} \sum_{W_k} P(w_k) P^{b+1-i}(w_k) \]

\[ \leq (B) \ c^{b-i} \left( \sum_{W_k} P^{b+1}(w_k) \right)^{(b+1-i)/(b+1)} \]

where the first inequality (A) comes from the strong mixing condition and the fact that the set of words of the form \( w_d^{[k/d]} \) is a subset of all words of length \( k \) (i.e., \( W_k \)). The inequality (B) is a consequence of the well known inequality on means (cf. [28]). So, finally we have

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

Let now \( k = (1+\epsilon) \log n/h_2 \), so that \( EP^b(w_k) \sim 1/n^{(1+\epsilon)(b+1)} \). Then, (3.3) and the above proves the following upper bound

\[ \Pr\{H_n \geq (1+\epsilon) \frac{\log n}{h_2}\} \leq c \sum_{i=1}^{b} \binom{b}{i} k^i n^{b-i} n^{b+1-i} \frac{1}{n^{(1+\epsilon)(b+1-i)}} \leq c \frac{\log^b n}{n^\epsilon}. \]  \hfill (3.5)

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

**Lower Bound**

The lower bound is more intricate. The idea, however, is quite simple. Firstly, we construct another \( b \)-suffix tree with height smaller than the in original \( b \)-suffix tree, but which is more similar to 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 [37] to the modified suffix tree to prove the lower bound. The second moment method gives a sharp lower bound for \( \Pr\{H_n > k\} \). In particular, using this method we prove that \( \Pr\{H_n > k\} \rightarrow 1 \) for \( k = (1-\epsilon) \frac{1}{h_2} \log n \).
To fulfill this plan, we start with a construction of a modified b-suffix tree. We partition the sequence $X^n_1$ 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 $m = \lceil n/(k + d) \rceil$. 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 $Y(i) = X^n_{(i-1)(k+d)+1} - X^n_{ik+(i-1)d+1}$ where $-$ means deletion, that is, $Y(i)$ is the $(i - 1)(k + d) + 1$st suffix of $X_k$ with the first gap following this suffix (i.e., the gap between the $ik + (i - 1)d + 1$st symbol and the $i(k + d)$th symbol eliminated). 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 $k + d + 1$ continue for the next $k$ symbols after which next $d$ symbols of the second gap are eliminated, and then the strings expends up to infinity. We built a b-suffix tree out of these $m$ strings $Y(1), \ldots, Y(m)$. We denote such a b-suffix tree as $T_m$ 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 in the modified b-suffix tree $T_m$. 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

$$\Pr\{H_n \geq k\} \geq \Pr\{H_m \geq k\} \quad \text{for} \quad m \leq n, \quad (3.6)$$

We estimate the probability $\{H_m \geq k\}$ by the second moment method (cf. [8], [37]). We need some additional notation. Let $i = (i_1, i_2, \ldots, i_{b+1})$ be a $b + 1$ dimensional vector, and define a set $D$ as $D = \{i : 1 \leq i_j \leq m \text{ for } 1 \leq j \leq b + 1\}$. Let also $D^2 = D \times D$ which we additionally partition into two sets $D^2_1$ and $D^2_2$ such that

$$D^2_1 = \{(i, j) : (i_1, \ldots, i_{b+1}) \cap (j_1, \ldots, j_{b+1}) = \emptyset\},$$

and $D^2_2$ contains the other pairs $(i, j)$ of $D^2$. Now, let us define an event $A_i = \{C_i \geq k\}$ where we use $C_i$ as a short notation for the self-alignment $C_{i_1, \ldots, i_{b+1}}$. Note that $\Pr\{H_m \geq k\} = \Pr\{\bigcup_{i \in D} A_i\}$. Then, the second moment method asserts that (cf. [8])

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

We will show that for $k = (1 - \varepsilon)\frac{1}{h^2} \log n$ the right-hand side (RHS) of (3.7) tends to one, hence also by (3.6) $\Pr\{H_n \geq (1 - \varepsilon)\frac{1}{h^2} \log n\} \to 1$ as $n \to \infty$, which is the desired inequality.

We must now evaluate the terms in the RHS of (3.7). Using the strong $\alpha$-mixing condition,
and arguing as in the upper bound case, we immediately show that for \( k = O(\log n) \) (cf. [39])

\[
(m^b - o(m^b))(1 - \alpha(d_n))^b E P^b(w_k) \leq \sum_{i \in D} \Pr\{A_i\} \leq (m^b - o(m^b))(1 + \alpha(d_n))^b E P^b(w_k),
\]

where the length of the gap \( d_n \) is explicitly shown to be a function on \( n \). The probability \( \Pr\{A_i \cap A_j\} \) is more difficult to estimate. However, on the set \( D_2^2 \) the suffixes of \( T_m \) do not coincide, hence easily analysis leads to (cf. [39])

\[
\sum_{(i,j) \in D_2^2} \Pr\{A_i \cap A_j\} \leq (1 + \alpha(d_n))^{2b+1} E^2 P^b(w_k).
\]

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

\[
\Pr\{A_i \cap A_j\} = \sum_{w_k} P(w_k^{2b+1}) \leq (A) c_1 \sum_{w_k} P^{2b+1}(w_k) - (B) c_1 \left( \sum_{w_k} P^{b+1}(w_k) \right)^{(2b+1)/(b+1)} = c_1 \left( E P^b(w_k) \right)^{2-1/(b+1)},
\]

where (A) follows from the strong mixing conditions, and (B) is a consequence of the following known inequality (cf. [19], [37])

\[
\ell \geq r \implies \left( \sum_{w_k} P^\ell(w_k) \right)^{1/\ell} \leq \left( \sum_{w_k} P^r(w_k) \right)^{1/r}.
\]

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

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

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

\[
\Pr\{H_m \leq (1 - \epsilon)\frac{1}{h_2} \log n\} \leq c_1 \frac{\log^{b+1} n}{n^{(b+1)\epsilon}} + c_2 \frac{\log n}{n^\epsilon} + c_3 (2b + 1) \alpha^2 (\log n) + O(\alpha^3(d_n)),
\]

which proves the lower bound for \( H_m \), and hence by (3.6) also for our original \( b \)-suffix tree. In summary, the upper bound (3.5) and the above lead to the following

\[
\Pr\{|\frac{H_n}{\log n} - \frac{1}{h_2}| \geq \epsilon\} \leq c_1 \frac{\log^b n}{n^\epsilon} + c_2 \alpha^2 (\log n) \to 0
\]

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 guarantee the almost sure convergence. But due to the fact that \( H_n \) is a 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 [10] (cf. also [20], [39]) 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

\[
\sum_{r=0}^{\infty} \Pr\left\{ \frac{H_{2^r+1}}{\log(2^{r+1})} - \frac{1}{h_2} \geq \varepsilon \right\} < \infty \quad (3.10)
\]

provided

\[
\sum_{r=0}^{\infty} \alpha^2(r) < \infty ,
\]

which holds for example for \( \alpha(n) = O(n^{-1/2-\delta}) \) for some \( \delta > 0 \). This completes the proof of Theorem 1 (i) concerning the height \( H_n^{(b)} \) of a \( b \)-suffix tree.

3.2 The Shortest Path of \( b \)-Suffix Trees

For the upper bound we use the fact that \( s_n^{(b)} \) is nonincreasing in terms of \( b \), that is, \( s_n^{(b)} \leq s_n^{(1)} \). But for noncompact suffix trees we have proved in [39] that

\[
\Pr\{s_n^{(1)} > (1 + \varepsilon) \frac{1}{h_1} \log n\} \leq \frac{c}{n^\varepsilon} . \quad (3.11)
\]

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 path. We proceed as in the case of the lower bound for the height, that is, we define the suffix tree \( T_m \) 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 path \( s_m \) in \( T_m \) is stochastically smaller than the shortest path \( s_n \) in the original \( b \)-suffix tree, which implies the following

\[
\Pr\{s_n < k\} \leq \Pr\{s_m < k\} . \quad (3.12)
\]

To estimate the probability \( \Pr\{s_m < k\} \) in the modified tree \( T_m \) we need some more notation. Let \( p_{\min}(k) = \min_{w_k \in W_k} \{P(w_k)\} \), and \( C_i(w_k) \) be the length of the longest prefix of the word \( w_k \) and \( b + 1 \) suffixes belonging to \( i = (i_1, \ldots, i_{b+1}) \). We assume that \( i \in D \) 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 \( i \in D \) the self-alignment \( C_i \) is smaller than \( k \), that is, \( C_i < k \). Using the strong \( \alpha \)-mixing condition we have

\[
\Pr\{s_m < k\} \leq \sum_{w_k} \Pr\{\bigcap_{i \in D} [C_i(w_k) < k]\} \leq \sum_{w_k} (1 + \alpha(d_n))^{m^b} (1 - P(w_k))^{m^b}
\]

\[
\leq (1 + \alpha(d_n))^{m^b} \sum_{w_k} (1 - c_{\min}(k))^{m^b} \leq V^k (1 + \alpha(d_n))^{m^b} (1 - P_{\min}(k))^{m^b}.
\]

Let now \( k = (1 - \varepsilon) \frac{1}{h_1} \log n \) and \( m = n / \log n \) while \( d_n = \log n \). Then, the above implies

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

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

\[
\Pr\{s_n < (1 - \varepsilon) \frac{1}{h_1} \log n\} \leq cn^b \exp(-n^{eb/2}/ \log^b n). \tag{3.14}
\]

The upper bound (3.11) and the lower bound (3.14) establish the convergence in probability of the shortest 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 Borel-Cantelli lemma (cf. also [20], [39]).

### 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 \( D_n \) since both quantities are asymptotically equally distributed. The last assertion is easy to prove. Roughly speaking, it must hold in the suffix tree \( T_m \) defined in Section 3.1, at least when (2.9) takes place. Indeed, in \( T_m \) the next inserted suffix is "almost" independent from the previous suffixes stored already in \( T_m \). Hence, it randomly selects an external node which implies that \( L_m \) and \( D_m \) are distributed in a similar manner. But, as easy to see the typical depths and depths of insertion in \( T_m \) and \( T_n \) are asymptotically equally distributed. Details are left to the interested reader.

The idea of the proof in this section is quite different from the ones 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 information theory community. For the convenience of the reader, we shortly restate Asymptotic Equipartition Property (AEP) [5] [42] which is a direct consequence of Shannon-McMillan-Breiman Theorem [5]. For a stationary and ergodic sequence \( \{X_k\}_{k=1}^\infty \), the state space \( \mathcal{A}^n \) 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 $P(X^n) \geq 1 - \varepsilon$ for any $\varepsilon > 0$, and $P(B_n) \leq \varepsilon$. Moreover, a typical $X^n$ (i.e., $X^n \in G_n$) has probability of occurrence estimated as $e^{-n(h+\varepsilon)} \leq P(X^n) \leq e^{-n(h+\varepsilon)}$ where $h$ is the entropy.

We concentrate on $L_n$, but the proof for $D_n$ follows the same steps. Define an event $A_n$ such that

$$A_n = \{X^n : |L_n/\log n - 1/h| \geq \varepsilon/h\}. \quad (3.15a)$$

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")

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

By the total probability formula we have

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

where $\delta > 1/h_2$ and

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

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 the 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

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

where

$$C_n = \{r : r/\log n - 1/h |> \varepsilon/h \text{ and } r \leq \delta \log n\}. \quad (3.17b)$$

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

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

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 $i = (i_1, \ldots, i_n)$ and $w_{r-1} = w_{r-1}$ (call this event $P_{n}^1$); and (ii) for all other $j = (j_1, \ldots, j_n) \neq i$, and all $w_r$ we have $C_j \neq w_r$ (call this event $P_{n}^2$). Then,

$$Pr\{L_n = r\} \leq cn^k \sum_{w_r} P(F_{n}^1 \cap F_{n}^2). \quad (3.19)$$
Now, we are in position to prove Theorem 2(i). We first establish the upper bound. Set $r \geq (1 + \varepsilon)\log n$. Hence, by the RHS of (3.18) we have $P(w^b_{r-1}) \leq 1/n$.

But, using mixing conditions we have $P(F^1_n) \leq cP(w_r)P(w^b_{r-1})$, and this together with the above lead to

$$P^{(r)} \leq \frac{c}{n^\varepsilon},$$

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. As in the case of the height and the shortest path, it is more intricate. Fortunately, we can apply here only a slightly modified approach discussed in the previous lower bounds. So, let $T_m$ be the suffix tree built from the strings $Y(1), \ldots, Y(m)$ as defined before. In particular, the typical depth $D_m$ in $T_m$ is bounded from the above by the typical depth $D_n$ in the original $b$-suffix tree (in a formal proof it is better to think in terms of the external path length $E_n$). Then,

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

since $L_n$ and $D_n$ 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 $P(F^1_n \cap F^2_n)$ as follows. Using the strong $\alpha$-mixing condition, we have

$$P(F^1_n \cap F^2_n) \leq cP(w_r)P(w^b_{r-1})(1 + \alpha(d_n))^{m^b} (1 - P(w^b_r))^{m^b}.$$

Let now $r \leq (1 - \varepsilon)\log n$, 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 path, we finally obtain in (3.17a) for $m = n/\log n$

$$P^{(r)} \leq c\exp(\frac{n^{b\varepsilon/2}}{\log n}) \cdot$$

provided (2.9) holds.

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

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

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 typical depth $D_n$. Note that the depth of insertion $L_n$ does not converge (a.s.) – as part (iii) asserts – since $L_n$ oscillates between the height $H_n$ and the shortest depth $s_n$. But, the typical depth $D_n$ may be not a nondecreasing sequence with respect to $n$ (i.e.,
the property we need to extend the convergence in probability (2.14) to (a.s.) convergence result (2.15)). Fortunately, the external path length $E_n$ (i.e., the sum of all typical depths) is a nondecreasing sequence. Naturally, Theorem 2 (i) formula (2.14) immediately implies that $\lim_{n \to \infty} E_n/(n \log n) = 1/h \, (pr.)$. This together with (3.23) and condition (2.15a) lead directly to (a.s.) convergence result (2.15b) by the same line of arguments as in Section 3.1. This completes the proof of Theorem 2, and the entire analysis.

References


