1991

Asymptotic Properties of Data Compressions and Suffix Trees

Wojciech Szpankowski

Purdue University, spa@cs.purdue.edu

Report Number:
91-003
ASYMPTOTIC PROPERTIES OF DATA COMPRESSION AND SUFFIX TREES

Wojciech Szpankowski

CSD-TR-91-003
January 1991
(Revised March 1992)
ASYMPTOTIC PROPERTIES OF DATA COMPRESSION AND SUFFIX TREES*

March 28, 1992

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

Abstract

Recently, Wyner and Ziv have proved that the typical length of a repeated subword found within the first $n$ positions of a stationary ergodic sequence is $(1/h) \log n$ in probability where $h$ is the entropy of the alphabet. This finding was used to obtain several insights into certain universal data compression schemes, most notably the Lempel-Ziv data compression algorithm. Wyner and Ziv have also conjectured that their result can be extended to a stronger almost sure convergence. In this paper, we settle this conjecture in the negative in the so called right domain asymptotic, that is, during a dynamic phase of expanding the data base. We prove – under an additional assumption involving mixing conditions – that the length of a typical repeated subword oscillates with probability one between $(1/h_1) \log n$ and $(1/h_2) \log n$ where $0 < h_2 < h \leq h_1 < \infty$. We also show that the length of the $n$th block in the Lempel-Ziv parsing algorithm reveals a similar behavior. We also relate our findings to another open problem in computer science, namely the asymptotic behavior of (noncompact) suffix trees which are digital trees built from suffixes of a sequence. We prove that the height and the shortest feasible path in a suffix tree are typically $(1/h_2) \log n$ (a.s.) and $(1/h_1) \log n$ (a.s.) respectively. These results were inspired by a seminal paper of Pittel who analyzed typical behavior of digital trees built from independent words (i.e., the so called independent tries).

*A preliminary version of this paper was presented at the Data Compression Conference, Snowbird, 1991.
†This research was supported in part by NSF Grants CCR-8900305 and INT-8912631, AFOSR Grant 90-0107, NATO Grant 0057/89, and Grant R01 LM05118 from the National Library of Medicine.
1. INTRODUCTION

Repeated patterns and related phenomena in words (sequences, strings) are known to play a central role in many facets of telecommunications and theoretical computer science, notably in coding theory and data compression, in the theory of formal languages, and in the design and analysis of algorithms. Several efficient algorithms have been designed to detect and to exploit the presence of repeated substrings and other kinds of regularities in words. In data compression, such a repeated subsequence can be used to compress the original sequence (cf. universal data compression schemes [6], [26], [42]). In exact string matching algorithms the longest suffix that matches a substring of the pattern string is used for "fast" shift of the pattern over a text string (cf. Knuth-Morris-Pratt and Boyer-Moore [2]; see also [10]), and so forth.

The problem of repeated patterns is studied here in a probabilistic framework. We assume that a stationary and ergodic source of information generates an infinite sequence \( \{X_k\}_{k=-\infty}^{\infty} \) over a finite alphabet \( \Sigma \) of size \( V \). This probabilistic model contains other simpler probabilistic schemes such as the Bernoulli model (i.e., symbols from the alphabet are generated independently) and the Markovian model (i.e., the next generated symbol depends in a probabilistic sense only on the previous one).

The relevance of our problem is illustrated in the following examples taken from data compression and algorithms on words. The data compression example is also used to motivate our further study. In particular, we define below some parameters of interest that are analyzed in this paper.

EXAMPLE 1.1 Data Compression

The following idea is behind most data compression schemes. Consider a "data base" sequence of length \( n \) which is known to both sender and receiver. Instead of transmitting the next \( L_n \) symbols to the other side of a communication channel, the sender can "look backward" into the data base and verify whether these \( L_n \) symbols have already occurred in the data base. If this is the case, then instead of sending \( L_n \) symbols the sender transmits only the location of these symbols in the data base and the length of \( L_n \). After identifying \( L_n \), we either append the data base with these new \( L_n \) symbols or – if the length of the data base is fixed (see the sliding window implementation in Bender and Wolf [6]) – we move the data base to the new position. This idea can be modeled mathematically in two different fashions that are discussed next.

A. Static Model – Left Domain Asymptotic

This is the model of Wyner and Ziv [40]. It is assumed that the subsequence to be
compressed \( \{X_k\}_{k=0}^{\infty} \) is always the same (by definition fixed at position \( k = 0 \)), and the data base \( \{X_k\}_{k=-n}^{-1} \) expands only to the left. Therefore, we coin the term left domain asymptotic for such a model with \( n \) tending to infinity. In practice, such a static situation occurs rather rarely since usually a new word is to be transmitted and compressed. Nevertheless, the model has some mathematical appeal and can be used to estimate the entropy. Following Wyner and Ziv [40], we define two parameters relevant to the performance of some data compression schemes. For every \( n \), let \( \tilde{L}_n \) be the smallest integer \( L > 0 \) such that

\[
X_{0}^{L-1} \neq X_{-m}^{-m+L-1} \quad \text{for all} \quad 1 \leq m \leq n , \tag{1.1a}
\]

(i.e., \( \tilde{L}_n - 1 \) is the length of the longest substring that is repeated and can be recopied from a data base of size \( n \)). In the above, we use the standard notation for subsequences, that is, \( X_i^j = (X_i, \ldots, X_j) \). In a practical implementation, the encoder observes \( X_0^{\tilde{L}_n-1} \), determines \( m_0 \) such that \( X_0^{\tilde{L}_n-2} = X_{-m_0}^{-m_0+\tilde{L}_n-2} \), and transmits \( m_0, \tilde{L}_n \) and \( X_{\tilde{L}_n-1}^{-1} \). To encode \( m_0 \) we need \( \log \nu n \) symbols, and it is known that \( \tilde{L}_n \) may be represented by \( \log \tilde{L}_n \) bits (cf. [32]). As noted by Wyner and Ziv [40], the number of encoded symbols per source symbol is asymptotically

\[
\frac{\log \nu n}{\tilde{L}_n} + \frac{\log \tilde{L}_n}{\tilde{L}_n} + \frac{\log V}{\tilde{L}_n} .
\]

Hence, the ratio \( \log \nu n / \tilde{L}_n \) determines an asymptotic efficiency of the compression scheme.

Another parameter of interest can be defined as follows. For every integer \( \ell \), let \( \tilde{N}_\ell \) be the smallest nonnegative integer \( N > 0 \) such that

\[
X_0^{\ell-1} = X_{-N}^{-N+\ell-1} , \tag{1.1b}
\]

that is, a word of length \( \ell \) is repeated for the first time in a data base of size \( \tilde{N}_\ell \). Wyner and Ziv [40] suggested the following compression scheme. The encoder sends the first \( n \) source symbols, say \( X_{-n}^{-1} \) with no compression, but the next \( \ell \) symbols \( X_0^{\ell-1} \) are encoded as follows: if \( X_0^{\ell-1} \) is a substring of \( X_{-n}^{-2} \) (i.e., \( \tilde{N}_\ell \leq n \)), then \( X_0^{\ell-1} \) is compressed by specifying only \( \tilde{N}_\ell \); otherwise \( X_0^{\ell-1} \) is not compressed. As noted in [40], in this scheme the average number of symbols required to encode \( X_0^{\ell-1} \) is \( \Pr\{\tilde{N}_\ell \leq n\} \log \nu n + \Pr\{\tilde{N}_\ell > n\} \ell + 1 \), where \( \log \nu n \) is the number of symbols required to transmit \( \tilde{N}_\ell \).

**B. Dynamic Model – Right Domain Asymptotic**

We introduce here a new model in which the next word to be compressed is not fixed, and each time after the compression the word is added to the (expanding) data base. In the analysis of such a model, it is more convenient to deal with a one-sided stationary and ergodic sequence \( \{X_k\}_{k=1}^{\infty} \). Then, the data base of length \( n \) is represented by \( \{X_k\}_{k=1}^{n} \)
and the word to be compressed starts at \( k = n + 1 \). Asymptotic analysis of such a model is carried out in the so called right domain asymptotic (since the data base is expanded to the right). This model seems to fit better to real implementation (e.g., sliding window [6]) of data compression schemes, and most of our analyses deal with this model. We can define two parameters \( L_n \) and \( N_\ell \) which correspond to \( \tilde{L}_n \) and \( \tilde{N}_\ell \) in the static model. More specifically, \( L_n \) is defined as the largest value of \( L \) such that

\[
x_{m_0 + L}^n = X_{n+1}^{n+1+L}
\]

for some \( m_0 \in \{1, \ldots, n\} \).

In a similar fashion, \( N_\ell \) is defined as the smallest \( N \) such that \( X_1^\ell = X_{N+1}^{N+\ell} \). It is easy to see that the compression schemes discussed above can be naturally expressed in terms of \( N_\ell \) and \( L_n \).

EXAMPLE 1.2 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 [26]. 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 \( \{X_k\} = 110101001111\cdots \) the parsing looks like \( (1)(10)(10100)(111)(1\cdots) \); that is, the first block has length one, the second block length has two, the next one is of length five, and so on. Observe that the third block is the longest prefix of \( X_2^\infty \) and \( X_4^\infty \). Grassberger [14] has shown how to construct such a parsing by using a special data structure called a suffix tree (cf. [1], [4], [14]). Naturally, of prime importance is the length of a block in the parsing algorithm. Let \( l_n \) be the length of the \( n \)th block in the Lempel-Ziv parsing algorithm. There is a simple relationship between the length \( l_n \) of the block and the parameter \( L_n \) defined above for the right domain asymptotic. (Note that we do not consider a finite sequence of length, say \( n \), as in Lempel-Ziv [26]; cf. Remark 4(i)). In our case, the underlying sequence is assumed to be unbounded, that \( \sum_{k=1}^n l_k \) is not fixed. In view of this, the asymptotic behavior of \( L_n \) in the right domain can be used to obtain the asymptotic length \( l_n \) of the last block in the Lempel-Ziv parsing algorithm.

EXAMPLE 1.3 String Matching Algorithms

Repeated substrings also arise in many algorithms on strings, notably string matching algorithms (cf. [1], [2], [32], [39]). A string matching algorithm searches for all (exact or approximate) occurrences of the pattern string \( P \) in the text string \( T \). Consider either the Knuth-Morris-Pratt algorithm or the Boyer-Moore algorithm (cf. [2]). Both algorithms rely
on an observation that in the case of a mismatch between $T$ and $P$, say at position $n+1$ of $P$, the next attempt to match depends on the internal structure (i.e., repeated substrings) of the first $n$ symbols of the pattern $P$. It turns out that this problem can be efficiently solved by means of a suffix tree (cf. [1], [4], [5], [8], [13], [19], [27], [39]). In particular, recently Chang and Lawler [10] used suffix trees to design an algorithm that on average needs $O((|T|/|P|) \log |P|)$ steps to find all occurrences of the pattern $P$ of length $|P|$ in the text $T$ of length $|T|$. □

From the above discussion, one concludes that suffix trees can be used to unify analyses of repeated patterns, and in particular to analyze $L_n$. Therefore, a short description of a suffix tree follows. An interested reader may find more on such trees and their applications in Aho et al. [1] and Apostolico [4] (cf. Grassberger [14]). A suffix tree is a digital tree built from suffixes of a string $X$. In general, a digital tree – also called a trie – stores a set of words (strings, sequences, keys) $X = \{X(1), \ldots, X(n)\}$, each key being a sequence from a finite alphabet $\Sigma$, that is, for $1 \leq \ell \leq n$ we have $\{X_k(\ell)\}_{k=1}^\infty$ with $X_k(\ell) \in \Sigma$. A trie consists of branching nodes, called also internal nodes, and external nodes that store the keys. Every external node is able to store only one key. The branching policy at any level, say $k$, is based on the $k$-th symbol of a string (key, word). For example, for a binary alphabet $\Sigma = \{0, 1\}$, if the $k$-th symbol in a key is "0", then we branch-out left in the trie, otherwise we go to the right. This process terminates when for the first time we encounter a different symbol between a key that is currently inserted into the trie and all other keys already in the trie. Then, this new key is stored in a newly generated external node. If $X(1), \ldots, X(n)$ are statistically independent sequences, then the constructed trie is called an independent trie. If, however, $X = \{S_1, S_2, \ldots, S_n\}$ where $S_i$ is the $i$th suffix of a one-sided single sequence $\{X_k\}_{k=1}^\infty$, then the trie built from $X$ is called a suffix tree. Certainly, in suffix trees the keys $X(1) = S_1, \ldots, X(n) = S_n$ are statistically dependent.

EXAMPLE 1.4 Suffix tree

Let $X = 0101101110\ldots$. Then the first five suffixes are $S_1 = 0101101110\ldots$, $S_2 = 101101110\ldots$, $S_3 = 01101110\ldots$, $S_4 = 1101110\ldots$ and $S_5 = 101110\ldots$. The suffix tree built from these five suffixes of $X$ is shown in Figure 1. □

An important parameter of a suffix tree that plays a crucial role in the analysis and design of algorithms on strings and data compression schemes is the depth of a suffix. Let $S_n$ be the suffix tree constructed from the first $n$ suffixes of a sequence $\{X_k\}_{k=1}^\infty$. Then, the depth of the $i$th suffix $L_n(i)$ in $S_n$ is the length of the path from the root to this suffix.
$S_1 = 0101101110$
$S_2 = 101101110$
$S_3 = 01101110$

$S_4 = 1101110$
$S_5 = 101110$

Figure 1: Suffix tree built from the first five suffixes of $X = 0101101110$...
We shall write \( L_n = \text{def} \ L_{n+1}(n+1) \), that is, \( L_n \) is the depth of insertion of the \( S_{n+1} \)-st suffix into the tree \( S_n \). Naturally, we call this parameter the depth of insertion. In the next section, we will show that this \( L_n \) coincides with the \( \bar{L}_n \) and \( L_n \) introduced in Example 1.1 (cf. (1.1a) and (1.2)).

From the previous discussion, it should be clear that the behavior of \( L_n \) is of considerable importance to combinatorial problems on words, in particular to data compression and string algorithms. The probabilistic behavior of repeated patterns for stationary and ergodic sequences was recently studied by Wyner and Ziv [40]. In fact, the authors of [40] studied the length \( \bar{L}_n \) in the left domain asymptotic. They established the following asymptotic result.

**Theorem 0.** (Wyner and Ziv [40]) Let \( \{X_k\}_{k=-\infty}^{\infty} \) be a stationary and ergodic sequence built over a finite alphabet \( \Sigma \). Then, in the left domain asymptotic as \( n \to \infty \)

\[
\frac{\bar{L}_n}{\log n} \to \frac{1}{h} \quad \text{in probability (pr.)} \tag{1.3a}
\]

and

\[
\frac{\log \bar{N}_\ell}{\ell} \to h \quad \text{in probability (pr.)} \tag{1.3b}
\]

where \( h \) is the entropy of \( X \). ■

This result concerns the convergence in probability (pr.) of \( \bar{L}_n \). In fact, a similar results also holds for \( L_n \) in the right domain asymptotics (cf. [33], [38]). Wyner and Ziv [40] asked whether it can be extended to a stronger almost sure (a.s.) convergence. We shall settle this question in the negative for the Markovian case in the right domain asymptotic, and show that \( L_n \) oscillates with probability one between \((1/h_1)\log n \) and \((1/h_2)\log n \) where \( h_2 < h \leq h_1 \). From this, it should be clear that the Wyner-Ziv conjecture cannot be also true in a more general than Markovian framework. In the course of the proof of our main results, we also indicate that the Wyner-Ziv conjecture concerning \( \bar{L}_n \) can be directly proved from their convergence in probability result in the left domain asymptotic for a Markovian source. This is due to the fact that \( \bar{L}_n \) is a nondecreasing sequence as opposed to \( L_n \). In the non-Markovian case, the proof for the (a.s.) convergence \( \bar{L}_n \) is more intricate and due to Ornstein and Weiss [29].

In this paper, we mainly deal with the more interesting right domain asymptotic which has also several applications in the analysis and design of algorithms on words. In particular, during the course of the proof we establish some new results regarding a typical (probabilistic) behavior of the height \( H_n \) and the shortest feasible path \( s_n \) in a suffix tree.
The height $H_n$ is the longest path in $S_n$, while the shortest feasible path is the shortest path from the root to an available (feasible) node. A node is called available if it does not belong to the tree $S_n$ but its predecessor node (either an internal or an external one) is in $S_n$. Then, under some additional assumption involving mixing conditions, we show that $H_n \sim (1/h_2) \log n \ (a.s.)$ and $s_n \sim (1/h_1) \log n \ (a.s.)$, where $h_1$ and $h_2$ will be given explicitly. This result implies that a typical suffix tree is fairly balanced. As a consequence of this, brute force (i.e., straightforward) algorithms for problems on words (e.g., construction of a suffix tree) could be a challenging competitor of more sophisticated algorithms designed to optimize the worst-case behavior (cf. Apostolico and Szpankowski [5]).

Asymptotic analyses of suffix trees and universal data compressions are rather scanty in the literature. To our best knowledge, asymptotic analysis of universal data compressions was pursued by Ziv and Lempel (cf. [42], [26]; see also [6]), Wyner and Ziv [40], [41], and Kieffer [21]. The average case analysis of suffix trees was initialized by Grassberger [14], and Apostolico and Szpankowski [5]. For the Bernoulli model, the asymptotic behavior of the height was recently obtained by Devroye, Szpankowski and Rais [13], and the limiting distribution of the depth in a suffix tree is reported in Jacquet and Szpankowski [19]. Finally, heuristic arguments were used by Blumer et al. [8] to show that the average number of internal nodes – in a slightly different model of suffix trees – is a linear function of $n$ (more precisely, the coefficient of $n$ contains an oscillating term). Jacquet and Szpankowski [19] established rigorously the latter result regarding the average size of a suffix tree. 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 trie constructed from independent words (i.e., independent tries). Pittel was the first who noticed that the depth of insertion in an independent trie does not converge almost surely but rather oscillates between the typical height and the typical shortest feasible path in a trie. Therefore, one can also consider this paper as a direct extension of Pittel's results to dependent tries such as suffix trees.

This paper is organized as follows. In the next section we formulate our main results. Namely, we settle the problem of Wyner and Ziv [40]. We also establish a new result concerning the length of a block in the Lempel-Ziv parsing algorithm. Finally, we present some new results on a typical behavior of suffix trees. We also discuss some consequences of our findings and suggest some further studies. Most proofs are delayed till Section 3.

2. MAIN RESULTS

Let $\{X_k\}_{k=-\infty}^\infty$ be a stationary ergodic sequence of symbols generated from a finite
alphabet $\Sigma$ of size $V$. Define a partial sequence $X^m_n$ as $X^m_n = (X_m, ..., X_n)$ for $m < n$, and the nth order probability distribution as follows

$$P(X^m_n) = \Pr\{X_k = x_k, 1 \leq k \leq n, x_k \in \Sigma\}.$$  (2.1)

The entropy of $\{X_k\}$ is

$$h = \lim_{n \to \infty} \frac{E \log P^{-1}(X^m_n)}{n},$$  (2.2)

The existence of the above limit is guaranteed by Shannon’s Theorem (cf. [7]). It is also known that $h \leq \log V$. Hereafter, all logarithms – unless stated explicitly otherwise – are natural logarithms.

It is well known (cf. [11], [12], [21]) that the entropy of a stationary ergodic information source is intimately related to coding and certain data compression schemes, most notably the universal compression scheme of Lempel and Ziv [26], [42]. Following Wyner and Ziv [40], we defined in the Introduction two parameters to capture certain properties of repeated subsequences, namely $L_n$ and $N_\ell$ (cf. (1.2)). Hereafter, we shall mainly deal with $L_n$, and we recall that $L_n$ is the smallest integer $L > 0$ such that

$$X^m_{n+L-1} \neq X^n_{n+1} \text{ for all } 1 \leq m \leq n.$$  (2.3)

We shall analyze $L_n$ in the right domain asymptotic, as discussed in the Introduction. However, we also show that the left domain asymptotic falls into our framework, and we provide some new results in this domain (see Remark 2(ii)). To illustrate our definition (2.3), we present one example below.

EXAMPLE 2.1 Illustration of definitions

We first discuss the left domain (cf. (1.1b)). Let $\{X_k\}$ be given in Figure 2 which is identical to the sequence discussed by Wyner and Ziv [40], and used in our Example 1.4. Then, in the left domain (cf. (1.1b)) one finds $\overline{L}_4 = 5$, and also $\overline{N}_3 = \overline{N}_4 = 3$ but $\overline{N}_5 > 4$. Of course, in the right domain we have that $L_4$ is also equal to 5. To see this, we only need to re-index the sequence in Figure 2 from 1 to 10, and apply directly definition (2.3). $\square$

Figure 2: A sample of data used in Example 2.1

<table>
<thead>
<tr>
<th></th>
<th>-4</th>
<th>-3</th>
<th>-2</th>
<th>-1</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X_k$</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>-4</th>
<th>-3</th>
<th>-2</th>
<th>-1</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X_k$</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>-4</th>
<th>-3</th>
<th>-2</th>
<th>-1</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X_k$</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
Before we present our main results, we somewhat strengthen our assumptions regarding the sequence \( \{X_k\}_{k=-\infty}^{\infty} \), namely we introduce mixing conditions (cf. [7]). Let \( \mathcal{F}_m^\infty \) be a \( \sigma \)-field generated by \( \{X_k\}_{k=m}^{n} \) for \( m \leq n \). It is said that \( \{X_k\} \) satisfies the mixing condition if there exist two positive constants \( c_1 \leq c_2 \) and an integer \( d \) such that for all \(-\infty \leq m \leq m+d \leq n \) the following holds

\[
c_1 \Pr\{A\} \Pr\{B\} \leq \Pr\{AB\} \leq c_2 \Pr\{A\} \Pr\{B\}
\]

(2.4)

where \( A \in \mathcal{F}_{-\infty}^m \) and \( B \in \mathcal{F}_{m+d}^\infty \). It is known that this condition implies ergodicity of the sequence \( \{X_k\}_{k=-\infty}^{\infty} \) (cf. [7]). In some statements of our results, we need a stronger form of the mixing condition, namely the strong \( \alpha \)-mixing condition which reads as follows

\[
(1 - \alpha(d)) \Pr\{A\} \Pr\{B\} \leq \Pr\{AB\} \leq (1 - \alpha(d)) \Pr\{A\} \Pr\{B\}
\]

(2.5)

where \( A \in \mathcal{F}_{-\infty}^m \) and \( B \in \mathcal{F}_{m+d}^\infty \) and \( \alpha(\cdot) \) is a function of \( d \) such that \( \alpha(d) \to 0 \) as \( d \to \infty \).

Following Pittel [30], we define two new parameters of \( \{X_k\} \), namely

\[
h_1 = \lim_{n \to \infty} \frac{\max\{\log P^{-1}(X^n_1), P(X^n_1) > 0\}}{n} = \lim_{n \to \infty} \frac{\log(1/\min\{P(X^n_1), P(X^n_1) > 0\})}{n},
\]

(2.6)

\[
h_2 = \lim_{n \to \infty} \frac{\log(E\{P(X^n_1)\})^{-1}}{2n} = \lim_{n \to \infty} \frac{\log \left( \sum_{i=1}^V P_i^n \right)^{-1}}{2n}.
\]

(2.7)

The existence of \( h_1 \) and \( h_2 \) was established by Pittel [30] who also noticed that \( 0 \leq h_2 \leq h \leq h_1 \).

Remark 1.

(i) **Bernoulli Model.** In this model, symbols from the alphabet \( \Sigma \) are generated independently, that is, \( P(X^n_1) = P^n(X_1) \). In particular, we assume that the \( i \)-th symbol from the alphabet \( \Sigma \) is generated according to the probability \( p_i \), where \( 1 \leq i \leq V \) and \( \sum_{i=1}^V p_i = 1 \). Thus, \( h = \sum_{i=1}^V p_i \log p_i^{-1} \) ([7]), \( h_1 = \log(1/p_{\min}) \) and \( h_2 = 2\log(1/P) \) where \( p_{\min} = \min_{1 \leq i \leq V} \{p_i\} \) and \( P = \sum_{i=1}^V p_i^2 \). The probability \( P \) can be interpreted as the probability of a match between any two symbols (cf. [37]).

(ii) **Markovian Model.** In this model, 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. Hence, the transition probability becomes \( p_{i,j} = \Pr\{X_{k+1} = j | X_k = i\} \in \Sigma \}, \) and the transition matrix is \( P = \{p_{i,j}\}_{i,j=1}^V \). It is well known [7] that the entropy \( h \) can be computed as \( h = -\sum_{i=1}^V \pi_i p_{i,j} \log p_{i,j} \) where \( \pi_i \) is the stationary distribution of the Markov chain. The other quantities, that is, \( h_1 \) and \( h_2 \), are a little harder to evaluate. Pittel [30] and
Szpankowski [37] evaluated the height of regular tries with Markovian dependency, and they showed that the parameter $h_2$ is a function of the largest eigenvalue $\theta$ of the matrix $P_2 = P \circ P$ which represents the Schur product of $P$ (i.e., elementwise product). More precisely, $h_2 = (1/2) \log \theta^{-1}$. With respect to $h_1$, we need to refer to Pittel [30] who cited a nonprobabilistic result of Romanovski who proved that $h_1 = \min_{\mathcal{C}} \{\ell(\mathcal{C})/|\mathcal{C}|\}$ where the minimum is taken over all simple cycles $\mathcal{C} = \{\omega_1, \omega_2, \ldots, \omega_v, \omega_1\}$ for some $v \leq V$ such that $\omega_i \in \Sigma$, and $\ell(\mathcal{C}) = -\sum_{i=1}^{v} \log p_{i,i+1}$. □

Now, we are ready to present our results. Our main finding of this paper is given in the following theorem which is proved after the statement of Theorem 2.

**Theorem 1.** Let the mixing condition (2.5) hold together with $h_1 < \infty$ and $h_2 > 0$. Then,

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

(2.8)

for all stationary ergodic sequences $\{X_k\}_{k=-\infty}^{\infty}$ provided that for $d \to \infty$

$$\alpha(d) = O(d^\beta \rho^d)$$

(2.9)

for some constants $0 < \rho < 1$ and $\beta$.

**Remark 2.**

(i) *How restrictive is condition (2.9)?* First of all, we note that we really need (2.9) only for establishing the lower bound in the lim inf case (see (3.22) in Section 3.2). Nevertheless, even with (2.9) we can cover many interesting cases including the Bernoulli model and the Markovian model. Naturally, in the Bernoulli model (2.9) holds since in this case $\alpha(d) = 0$. In the *Markovian model*, it is known (cf. [7]) that for a finite state Markov chain the coefficient $\alpha(d)$ decays exponentially fast; that is, for some $c > 0$ and $\rho < 1$ we have $\alpha(d) = c \rho^d$, as needed for (2.9). However, for general stationary ergodic sequences our result possibly does not hold. This is due to P. Shields [34] who recently announced that he can construct an ergodic mixing stationary sequence that does not satisfy the lim sup part of (2.8).

(ii) *Asymptotic behavior of $\tilde{L}_n$ in the left domain asymptotic.* We now show that in the left domain asymptotic the (a.s.) behavior of $\tilde{L}_n$ is the one predicted by Wyner and Ziv. From their proof of the convergence in probability (cf. [40] pp. 1253-1255), one concludes that $\Pr(|\tilde{L}_n/\log n - 1/h| > \varepsilon) = O(1/\sqrt{\log n}) + P(B_n)$, where $P(B_n)$ is the probability of "bad states" in the McMillan-Shannon theorem (cf. [7]). Assume now that $P(B_n)$ is summable, that is, $\sum_{n=1}^{\infty} P(B_n) < \infty$. This is true, for example, for the Bernoulli model,
the Markovian model, the hidden Markov model and \( m \)-dependent model (cf. [12] and [34]).

In order to apply the Borel-Cantelli Lemma, we use the trick suggested by Kingman [22]; that is, we construct a subsequence \( n_r \) of \( n \) for which \( O(1/\sqrt{\log n_r}) \) is summable. Fix \( s \), and define \( n_r = 2^{s^2 2^r} \). Note that \( \tilde{L}_{n_r}/\log n_r \rightarrow 1/h \) (a.s.) provided \( P(B_n) \) is summable. To prove that \( \tilde{L}_n/\log n \) converges (a.s.), we use two facts: (i) \( \tilde{L}_n \) is a nondecreasing sequence (cf. Example 2.1 below); (ii) for every \( n \) we can choose such \( r \) that \( 2^{s^2 2^r} \leq n \leq 2^{(s+1)^2 2^r} \). Then,

\[
\lim_{n \to \infty} \sup_{r \to \infty} \frac{\tilde{L}_n}{\log n} \leq \lim_{r \to \infty} \sup_{n_r} \frac{\tilde{L}_r}{\log n_r} \frac{\log 2^{(s+1)^2 2^r}}{\log 2^{s^2 2^r}} \to \frac{1}{h} \frac{(s+1)^2}{s^2} \quad (a.s.),
\]

and similarly for the \( \lim \inf \) case. Taking in the last display \( s \to \infty \), we finally prove our assertion (for more details see also the end of Section 3.1). \( \Box \)

Theorem 1 will be proved below as a simple consequence of some new results concerning a typical behavior of a suffix tree \( S_n \) built over the first \( n \) suffixes of \( \{X_k\}_{k=1}^\infty \), as discussed in the Introduction. The clue to the proof of Theorem 1 is to reformulate the definition of \( L_n \) in terms of some parameters of the associated suffix tree \( S_n \). This will also lead to some new results about suffix trees.

Define for \( S_n \) the \( m \)th depth \( L_n(m) \), the height \( H_n \) and the shortest feasible path \( s_n \) as in the Introduction. That is, the depth of the \( m \)th external node containing the \( m \)th suffix (e.g., Figs. 1 and 3) is equal to one plus the number of internal nodes in the path from the root to the \( m \)th external node. Then,

\[
H_n = \max_{1 \leq m \leq n} \{L_n(m)\} .
\]

The shortest feasible path is defined as follows. Consider a suffix tree \( S_n \), and append it with available nodes, that is, nodes that are not in the tree \( S_n \) but whose predecessors (either internal or external nodes) are in \( S_n \). Then, the shortest feasible path is the shortest path to an available node. Furthermore, we define the average depth \( D_n \) and the depth of insertion \( L_n \). The depth of insertion \( L_n \) is the depth of the \((n+1)\)st external node after insertion of the \((n+1)\)st suffix \( S_{n+1} \) into the suffix tree \( S_n \), that is, \( L_n = L_{n+1}(n+1)^1 \).

Finally, \( D_n \) is defined as the depth of a randomly selected external node, that is,

\[
D_n = \frac{1}{n} \sum_{m=1}^{n} L_n(m) .
\]

---

\(^1\)It would be more natural to denote this depth of insertion as \( L_{n+1} \), but we try to keep our notation consistent with the one introduced in [40].
This average depth is used in the analysis of the average complexity of the exact matching algorithms (cf. Example 1.3), and can be applied to assess \( N_\ell \) (cf. Remark 3 below).

For our purpose, another characterization of the above parameters is more useful. For a suffix tree \( S_n \) built from \( n \) suffixes \( S_1, S_2, \ldots, S_n \) of a stationary ergodic sequence \( \{X_k\}_{k=1}^\infty \), define the self-alignment \( C_{i,j} \) between \( S_i \) and \( S_j \) as the length of the longest common prefix of \( S_i \) and \( S_j \). Then, the following are easy to establish (cf. Szpankowski [37])

\[
L_n(m) = \max_{1 \leq k \leq n, \ k \neq m} \{ C_{k,m} \} + 1 
\]

(2.13a)

\[
H_n = \max_{1 \leq i < j \leq n} \{ C_{i,j} \} + 1 
\]

(2.13b)

and finally

\[
L_n = \max_{1 \leq m \leq n} \{ C_{m,n+1} \} + 1 
\]

(2.13c)

From the last display it is clear that \( L_n \) defined in terms of the suffix tree and \( L_n \) defined in (1.2) are the same. Therefore, we can further reason only in terms of \( L_n \) as defined in (2.13c).

In passing, we note that the second parameter defined in Example 1.1, namely \( N_\ell \), can also be re-defined in terms of the associated suffix tree. Indeed, \( N_\ell \) is the size of a suffix tree in which the depth of the first suffix in \( S_n \) is equal to \( \ell \), that is, \( L_{N_\ell}(1) = \ell \). We should also point out that in the left domain asymptotic, the analysis of \( \tilde{N}_\ell \) (cf. (1.1b)) is much easier due to the following relationship with \( \tilde{L}_n \) (cf. [40])

\[
\{ \tilde{N}_\ell > n \} = \{ X_1^\ell \neq X_m^{m+\ell-1}, \ 1 \leq m \leq n \} = \{ \tilde{L}_n \leq \ell \}.
\]

This relationship does not hold for \( N_\ell \) and \( L_n \) (e.g., consider \( X = 011010101010101 \) for which \( N_3 > 7 \) and \( L_7 > 3 \)).

**Remark 3.**

*Asymptotic Behavior of \( N_\ell \).* We can predict the (a.s.) behavior of \( N_\ell \) in the right domain asymptotic. Since the first suffix \( S_1 \) can occupy equally likely any position in the associated suffix tree \( S_n \), the quantity \( N_\ell \) can be alternatively defined with respect to the average depth \( D_n \) as \( D_{N_\ell} = \ell \). But, for the Markovian model we can prove (cf. Shields [33], Szpankowski [38]) that \( D_n / \log n \to 1/h \) (a.s.). Hence, \( \ell / \log N_\ell \to 1/h \) (a.s.). Interestingly enough, the asymptotic behavior of \( N_\ell \) is the same in the left domain asymptotic, as proved recently by Ornstein and Weiss [29] for a general probabilistic model. This is in surprising contrast to the asymptotic behavior of \( L_n \) in these two domains of asymptotics. \( \square \)

The quantity \( \tilde{L}_n \) defined by Wyner and Ziv [40], [41] (see Example 1.1) can be easily obtained from our suffix tree model, too. Indeed, in this case one has to construct the suffix
tree from the first $n$ suffixes of $\{X_k\}_{k=-n}^\infty$, and $\tilde{L}_n$ is the depth of insertion of the first suffix $S_1$ into this suffix tree. Note that in the Wyner-Ziv model, we always insert the same suffix, namely $\{X_k\}_{k=0}^\infty$, and we expand the sequence $\{X_k\}_{k=-n}^\infty$ to the left, that is, the next suffix inserted into the tree $S_n$ is $X_{-n-1}^\infty$, hence we refer to the left domain asymptotic. Finally, in Remark 2(ii) we already noted that $\tilde{L}_n$ is a nondecreasing sequence in the sense that $\tilde{L}_n \leq \tilde{L}_{n+1}$. This is illustrated below.

**EXAMPLE 2.2 Illustration of definitions**

Let $X_1^\infty = 0101101110\ldots$, as in Examples 1.4 and 2.1. Note that $L_4 = 5$. Consider now the suffix tree built from the first four suffixes of the above sequence. It is shown in Figure 3. Then, $L_4(1) = 3$, $L_4(2) = 2$, $L_4(3) = 3$, $L_4(4) = 2$. Moreover, $H_4 = 3$ and $s_4 = 2$. But the depth of insertion for the fifth suffix (see Figure 1 which represents the tree after the fifth insertion) is $L_4 = L_5(5) = 5$, as needed for (1.2) and (2.13d).

Consider now $\tilde{L}_n$. In the Wyner-Ziv model, we must re-index the original sequence to obtain $X_4^\infty = 0101101110\ldots$. Let us add some new symbols to the left of the $X_4^\infty$ keeping in mind that we always compare the depth of insertion for the zero-th suffix, $X_0^\infty = 101110\ldots$. If we add 10111 from the left, and insert $X_0^\infty$, then $\tilde{L}_n$ becomes respectively: $\tilde{L}_5 = 5$, $\tilde{L}_6 = 5$, $\tilde{L}_7 = 5$, $\tilde{L}_8 = 5$ and finally $\tilde{L}_9 = 6$. It confirms our observation that $\tilde{L}_n$ is a nondecreasing sequence. □

It turns out that the almost sure characteristic of $L_n$ depends on the almost sure behaviors of the height $H_n$ and the shortest feasible path $s_n$. This observation follows naturally
from the result of Pittel [30] on independent tries (i.e., digital trees built from statistically independent sequences) and our own experience with suffix trees. Therefore, not surprisingly, Theorem 1 is a simple consequence of the following result which we shall prove in Section 3.

**Theorem 2.** Let \( \{X_k\}_{k=1}^{\infty} \) be a stationary ergodic sequence satisfying the strong \( \alpha \)-mixing condition (2.5) together with \( h_1 < \infty \) and \( h_2 > 0 \).

(i) Asymptotically for large \( n \) we have
\[
\lim_{n \to \infty} \frac{s_n}{\log n} = \frac{1}{h_1} \quad (a.s.)
\]
provided (2.9) holds.

(ii) For large \( n \) the height \( H_n \) satisfies
\[
\lim_{n \to \infty} \frac{H_n}{\log n} = \frac{1}{h_2} \quad (a.s.)
\]
provided the coefficient \( \alpha(d) \) satisfies
\[
\sum_{d=0}^{\infty} \alpha^2(d) < \infty,
\]
where \( h_1 \) and \( h_2 \) are defined in (2.6) and (2.7).

Provided Theorem 2 is granted, we can now give a proof of Theorem 1 along the lines suggested by Pittel in [30] for independent tries. We concentrate only on the lim sup part of Theorem 1. Note that by definition \( L_n \leq H_n \), hence \( L_n / \log n \leq H_n / \log n \), and obviously
\[
\lim_{n \to \infty} \sup L_n / \log n \leq \lim_{n \to \infty} H_n / \log n \quad (a.s.).
\]
(2.16a)
It suffices now to show that the reverse to (2.16a) holds (a.s). Note that almost surely \( L_n = H_n \) whenever \( H_{n+1} > H_n \), which happens infinitely often (i.o.) since \( H_n \to \infty \) (a.s.), and \( \{X_k\} \) is an ergodic sequence. Therefore, \( \Pr\{L_n = H_n \ i.o.\} = 1 \) implies that almost surely there exists a subsequence, say \( n_k \to \infty \), such that \( L_{n_k} = H_{n_k} \). So, \( \lim_{n_k \to \infty} L_{n_k} / \log n_k = \lim_{n_k \to \infty} H_{n_k} / \log n_k \) (a.s.), and this finally implies that
\[
\lim_{n \to \infty} \sup L_n / \log n \geq \lim_{n \to \infty} H_n / \log n \quad (a.s.);
\]
(2.16b)
that is, \( \lim_{n \to \infty} \sup L_n / \log n = \lim_{n \to \infty} H_n / \log n \) (a.s.), and by (2.15a) this proves the lim sup part of (2.8) in Theorem 1. In a similar manner we can prove the lim inf part by using \( s_n \) and (2.14) from Theorem 2 since \( s_n \) is also a nondecreasing sequence.
Finally, we apply Theorem 1 to estimate the length of the nth block in the Lempel-Ziv parsing algorithm as discussed in Example 1.2 (cf. [26]). We prove the following result.

**Corollary 3.** Let $l_n$ be the length of the nth block in the Lempel-Ziv parsing algorithm. In addition, we assume that the source is Markovian. Then, in the right domain asymptotic

$$\frac{1}{h_1} \leq \lim \inf_{n \to \infty} \frac{l_n}{\log n} \leq \lim \sup_{n \to \infty} \frac{l_n}{\log n} \leq \frac{1}{h_2} \quad (a.s.). \quad (2.17)$$

**Proof.** In Example 1.2 we noted that $l_n = L_{\sum_{k=1}^{n-1} l_k}$. This is a direct consequence of the Lempel-Ziv parsing algorithm (see also Grassberger [14]). Then,

$$\lim_{n \to \infty} \frac{l_n}{\log n} = \lim_{n \to \infty} \frac{L_{\sum_{k=1}^{n-1} l_k}}{\log \left( \sum_{k=1}^{n-1} l_k \right)} \cdot \frac{\log \left( \sum_{k=1}^{n-1} l_k \right)}{\log n}.$$

The second term of the above can be further estimated as follows

$$1 \leq \frac{\log \left( \sum_{k=0}^{n-1} l_k \right)}{\log n} \leq \frac{\log \left( \sum_{k=0}^{n-1} L_k \right)}{\log n} \to 1 \quad (a.s.),$$

where the right-hand side (RHS) of the above is a direct consequence of the fact that for Markovian models $\sum_{k=1}^{n} L_k \sim (n/h) \log n$ (a.s.) (cf. Shields [33], and Szpankowski [38] for some generalizations). Hence, (2.17) follows from Theorem 1 and the above.

**Remark 4.**

(i) **Lempel-Ziv Parsing Algorithm for Finite Strings.** In Corollary 3, we assumed an infinite length sequence $\{X_k\}_{k=1}^\infty$, and $l_n$ denoted the nth block length in such a sequence. The original parsing algorithm of Lempel and Ziv, however, postulates that the underlying sequence is finite, say of length $n$, and the number of blocks $M$ is such that $\sum_{k=1}^{M} l_k = n$. Nevertheless, Corollary 3 is valid for finite sequences too. Indeed, this follows directly from the fact $M = O(n/\log n)$ [11], [26]. (A simple proof of $M = O(n/\log n)$ works as follows: It suffices to note that $n = \sum_{k=1}^{M} l_k \leq \sum_{k=1}^{M} L_k \sim (1/h)M \log M$, where the last asymptotic is already known from Shields [33] and Szpankowski [38]).

(ii) **Behavior of $l_n$ Revisited.** Corollary 3 does not exclude the possibility that $l_n/\log n$ converges (a.s.) to a constant, however, this seems to be very unlikely. In fact, we expect that $l_n/\log n$ resembles the behavior of $L_n$. We have three reasons to believe this. First of all, $l_n$ coincides with $L_n$ approximately every $O(\log n)$ symbols, so a formal proof would only require to show that $l_n$ hits $H_n$ and $s_n$ infinitely often. Secondly, the consecutive blocks are only weakly dependent in the Markovian model. Thus, one can build a suffix tree from $M$ weakly dependent sequences (e.g., in the Bernoulli model these sequences are...
practically independent), and in the view of \( M = O(n/\log n) \), and the results of Pittel [30] for independent tries, we obtain the desired result. Finally, we can easily prove our conjecture for a modified version of the Lempel-Ziv parsing algorithm already analyzed in Aldous and Shields [3] for the symmetric Bernoulli model (i.e., all symbols occur with the same probability). For such a modified algorithm, the sequence \( \{X_k\} \) is parsed into words where each new word is the shortest consecutive sequence of symbols not seen in the past as a word. For example, the sequence 11010100111... discussed in Example 1.2 is parsed into (1)(10)(101)(0)(01)(11)... instead of (1)(10)(10100)(111)... as in the Example 1.2. For such a parsing scheme, it was already noticed by Aldous and Shields [3] that the algorithm can be modeled by another digital tree, namely the so called digital search tree (i.e., suffixes are stored in the next available node rather than in external nodes) [1], [25]. Then the length \( l_n \) of the \( n \)th block is exactly equal to the depth of the \( n \)th node in such a digital search tree. By extrapolating Pittel's result for independent digital search trees [30], we can prove under weak mixing condition that

\[
\lim_{n \to \infty} \inf \frac{l_n}{\log n} = \frac{1}{h_1} \quad (a.s.) \quad \lim_{n \to \infty} \sup \frac{l_n}{\log n} = \frac{1}{h_3} \tag{2.18}
\]

where \( h_3 \) is defined as

\[
h_3 = \lim_{n \to \infty} \frac{\log(1/\max\{P(X^n), P(X^n) > 0\})}{n}.
\]

In the Bernoulli model, we have \( h_3 = \log(1/p_{max}) \) where \( p_{max} = \max_{1 \leq i \leq V} p_i \).

(iii) Second Order Properties of \( L_n \). From our previous discussion, we know that \( L_n/\log n \to 1/h \) (pr.). We also know that the average depth \( D_n \) and the depth of insertion \( L_n \) have the same limiting distribution, however, their almost sure behaviors are different. Therefore, we can study the rate of \( L_n \) through the average depth behavior \( D_n \). Recently, Jacquet and Szpankowski [19] showed that for the Bernoulli asymmetric model the normalized depth \( (D_n - ED_n)/varD_n \) converges in distribution to the standard normal distribution \( N(0, 1) \) with mean and variance as below

\[
ED_n = \frac{1}{h} \cdot \{\log n + \gamma + \frac{h_3}{2h} \} + P_1(\log n) + O(n^{-\varepsilon}) \tag{2.19a},
\]

\[
varD_n = \frac{H_2 - h^2}{h^3} \log n + C + P_2(\log n) + O(n^{-\varepsilon}), \tag{2.19b}
\]

for some \( \varepsilon > 0 \), where \( H_2 = \sum_{i=1}^V p_i^2 \log p_i \), and \( P_1(x) \) and \( P_2(x) \) are fluctuating periodic functions with small amplitudes (an explicit formula for the constant \( C \) can be found in [36]). We conjecture that the same type of limiting distributions can be obtained for the
Markovian model. This is due to the fact that the limiting behavior of independent tries do not differ too much from asymptotics of suffix trees, and recent result of Jacquet and Szpankowski [18] who established the limiting distribution of the depth for independent tries in a Markovian framework.

(iv) Second Order Behavior for the Lempel-Ziv Parsing Scheme. The limiting distribution for the length of the $n$th block in the Lempel-Ziv parsing algorithm is much harder to study. In particular, if $\sum_{k=1}^{M} l_k = n$, then the distribution of the number of phrases $M$ depends on the external path length $E_n$ in the associated suffix tree $S_n$. The quantity $E_n$ is the sum of all depths in $S_n$, that is, $E_n = \sum_{m=1}^{n} L_n(m)$. Even in the Bernoulli model major difficulties arise in the evaluation of the limiting distribution of $E_n$. Only recently, Kirschenhofer, Prodinger and Szpankowski [23] obtained for the symmetric Bernoulli model the variance of $E_n$ which becomes $\text{var} E_n = (\alpha + P_3(\log n))n + O(\log^2 n)$ where $\alpha$ is a constant (cf. [23] for an explicit formula) and $P_3(\log n)$ is a fluctuating function. Finally, Jacquet and Régnier [17] were able to show that in the Bernoulli asymmetric model $(E_n - (n/h)\log n)/\text{var} E_n$ converges to the standard normal distribution, where $\text{var} E_n = O(n \log n)$. We conjecture that the external path length in a suffix tree has the same limiting distribution.

For the modified Lempel-Ziv parsing algorithm discussed in Remark 4(ii), the situation is similar. This time, however, one needs to analyze an independent digital search tree. The independence is a consequence of nonoverlapping blocks in such an algorithm. The symmetric Bernoulli model was already analyzed in Aldous and Shields [3]. For the limiting distribution of the number of blocks $M$, one needs to evaluate the variance of the external path length. In [3] only a rough estimate was obtained. Recently, Kirschenhofer, Prodinger and Szpankowski [24], after lengthy and complicated derivations, proved that $\text{var} E_n = (\beta + P_4(\log n))n + O(\log^2 n)$ where $\beta$ is a constant with a complicated formula, and $P_4(\log n)$ is a fluctuating function. Nothing is known about the limiting distribution of the external path length $E_n$, however, we conjecture that $(E_n - (n/h)\log n)/\text{var} E_n \rightarrow N(0, 1)$ where $\text{var} E_n = O(n \log n)$.

(v) Optimal Compression Ratio. For universal data compression one may define the compression ratio $\rho$ as the ratio of the overhead information necessary to implement a data compression scheme and the length of repeated (compressed) subpatterns $L_n$. Therefore, the optimal compression ratio becomes

$$\rho_{\text{opt}} = \frac{\text{length of the minimal overhead information}}{\text{length of repeated subword}}.$$  \hspace{1cm} (2.20a)
If the length of data base is \( n \), then the length of the minimum overhead information is at least \( \log V n \) where \( V \) is the size of the alphabet used in decoding the compressed information. This estimate is a simple consequence of the fact that any overhead information must at least contain information regarding a position of the repeated pattern occurrence in the sequence of length \( n \). Therefore, we have

\[
\rho_{opt} = \frac{\log V n}{L_n}. \tag{2.20b}
\]

A probabilistic behavior of \( \rho_{opt} \) depends on the type of convergence we want to investigate. Wyner and Ziv (cf. [40], [38]) proved that \( \rho_{opt} \sim h/\log V \) in probability. Our Theorem 1 shows that almost surely the optimal compression ratio \( \rho \) "swings" between \((1/h_1)/\log V\) and \((1/h_2)\log V\). □

3. ANALYSIS

In this section we prove our main results presented in Theorem 2, that is, we establish almost sure convergence of the height \( H_n \) and the shortest feasible path \( s_n \) in a suffix tree. We prove these results separately for the height and for the shortest feasible path. In each case, we consider an upper bound and a lower bound. In both proofs we use quite often a technique that Jacquet and Szpankowski [19] named string-ruler approach, and was already in Pittel [30]. We also shall use some ideas from Szpankowski [37], and Devroye et al. [13].

In the string-ruler approach, a correlation between different (sub)strings is measured by means of another string, say \( w \), that does not necessarily have to be random. We call \( w \) a string-ruler. Its "randomness" comes from the fact that the underlying sequence \( \{X_k\} \) is random. To illustrate the technique, consider estimating the length of the longest common prefix of two independent strings, say \( \{X_k(1)\}_{k=1}^{\infty} \) and \( \{X_k(2)\}_{k=1}^{\infty} \). Let \( C_{1,2} \) be the length of such a longest prefix (the reader should recognize in \( C_{1,2} \) the alignment between \( X(1) \) and \( X(2) \)). The clue is to note that \( C_{1,2} \geq k \) implies the existence of a string \( w \) of length \( k \) such that \( X_1^k(1) = w \) and \( X_1^k(2) = w \). In fact, the reverse holds too, that is, the existence of \( w \) of length \( k \) such that \( X_1^k(1) = w \) and \( X_1^k(2) = w \) is enough for \( C_{1,2} \geq k \). We shall use this observation to estimate the self-alignment between suffixes of a single sequence \( \{X_k\} \).

We adopt the following notation. Let \( \mathcal{W}_k \) be the set of all strings \( w \) of length \( k \), that is, \( \mathcal{W}_k = \{w \in \Sigma^k : |w| = k\} \), where \( |w| \) is the length of \( w \). An element of \( \mathcal{W}_k \) will be denoted as \( w_k \), i.e., \( w_k \in \mathcal{W}_k \), and by \( w_k^\ell \) we mean a concatenation of \( \ell \) strings \( w_k \) from \( \mathcal{W}_k \). If a subsequence \( X_{m+k} \) is equal to a string ruler \( w_k \), then we write \( P(w_k) = P(X_{m+k}) \) for the probability that \( X_{m+k} = w_k \). Finally, for a function \( f(w_k) \) of \( w_k \) we write \( \sum_{w_k \in \mathcal{W}_k} f(w_k) = \sum_{w_k \in \mathcal{W}_k} f(w_k) \) for the sum over all strings \( w_k \) of length \( k \). Below, we shall reason only in
terms of suffix trees leaving other interpretations of our results (e.g., data compression) for the reader.

3.1 The Height in a Suffix Tree

In the analysis of the height $H_n$, we use the definition (2.13b), which is repeated below

$$H_n = \max_{1 \leq i \leq j \leq n} \{C_{i,j}\} + 1 ,$$

where $C_{i,j}$ is the self-alignment between subsequences $X_i^\infty$ and $X_j^\infty$. It is easy to notice that the self-alignment really depends only on the difference $d = |j - i|$ since $\{X_k\}$ is stationary. From (3.1) one concludes that the distribution of the height depends on the distribution of the self-alignments, and we express the latter by an appropriate probability on string-ruler set $\mathcal{W}_k$. Let $k \wedge d = \min\{k, d\}$ for given $d \geq 0$ and $k \geq 0$. Then for any $1 \leq i \leq n$ and $1 \leq d \leq n - i$ we have

$$\Pr\{C_{i,i+d} \geq k\} = \sum_{\mathcal{W}_{k \wedge d}} P(w_{k \wedge d}^{\lfloor k/d \rfloor+1} \bar{w}_d)$$

(3.2)

where $\bar{w}_d$ is a prefix of $w_d$ such that $|w_{k \wedge d}^{\lfloor k/d \rfloor+1} \bar{w}_d| = k + \min\{k, d\}$, and $\lfloor k/d \rfloor$ is the integer part of $k/d$. Note that for $d \geq k$ the RHS of (3.2) becomes $\sum_{\mathcal{W}_k} P(w_k^2)$. Identity (3.2) is a simple consequence of the following fact on combinatorics of words: if $Z$ is the longest common prefix of two suffixes $S_i$ and $S_j$ of a single sequence $\{X_k\}$ such that $|Z| = k$, then the string $Z$ can be represented as $Z = w_d^j \bar{w}_d$ where $d = |j - i|$. A more detailed discussion of (3.2) can be found in [5].

A. Upper Bound

We use (3.2) to prove an upper bound for the height $H_n$. We start with Boole's inequality applied to the event $\{\max_{i,d} C_{i,i+d}\}$ (we set below $i = 1$ for simplicity of notation) which leads to (cf. also [13])

$$\Pr\{H_n \geq k\} \leq n \left( \sum_{d=1}^{k-1} \Pr\{C_{1,1+d} \geq k\} + \sum_{d=k}^{n} \Pr\{C_{1,1+d} \geq k\} \right) .$$

(3.3)

We consider the above two terms separately. We first deal with the second sum, which in view of (3.2), becomes

$$\sum_{d=k}^{n} \Pr\{C_{1,1+d} \geq k\} = \sum_{d=k}^{n} \sum_{\mathcal{W}_k} P(w_k^2) \leq nc_1 \sum_{\mathcal{W}_k} P^2(w_k) = nc_1 E P(w_k) ,$$

(3.4)

where the inequality above comes from the (weak) mixing condition (2.4).
The first sum in (3.3) is more difficult to assess, but we shall also use only the weak mixing condition (2.4). We proceed as follows for \(d \leq k\)

\[
\Pr\{C_1, 1+d \geq k\} = \sum_{w_d} P(w_d^{[k/d]+1} \bar{w}_d) \leq (A) c_1 \sum_{w_d} P(w_d^{[k/d]} \bar{w}_d) P(w_d)
\]

\[
\leq (B) c_1 \sqrt{\sum_{w_d} P^2(w_d^{[k/d]} \bar{w}_d) P(w_d)} \leq c_1 \sqrt{\sum_{w_d} P^2(w_d^{[k/d]} \bar{w}_d)}
\]

\[
\leq (C) c_1 \sqrt{P^2(w_k)} = c_1 \sqrt{E P(w_k)},
\]

where the inequality (A) is due to the mixing condition (2.4), inequality (B) is a consequence of the inequality on means (cf. [28]), and the last inequality (C) follows from \(\mathcal{W}_d \subset \mathcal{W}_k\). In the above, the constant \(c_1\) may change from line to line.

Finally, putting everything together we have

\[
\Pr\{H_n \geq k\} \leq nc \left( k \sqrt{E P(w_k)} + n E P(w_k) \right).
\]

Let now \(k = (1+\varepsilon)\frac{1}{h_2} \log n\). Then, by (2.7) we have \(E P(w_k) \sim \exp(-2 \log n^{1+\varepsilon}) = n^{-2(1+\varepsilon)}\), then

\[
\Pr\{H_n \geq \left(1 + \varepsilon\right) \frac{1}{h_2} \log n\} \leq \frac{c \log n}{n^\varepsilon} \to 0 \quad (3.5)
\]

for some constant \(c\). This proves the upper bound for the convergence in probability. The almost sure convergence will follow from the above after some algebra, and we shall discuss it after deriving the lower bound for the height.

B. Lower Bound

The lower bound is more intricate, although the main idea behind the proof is quite simple. We need a sharp bound on \(\Pr\{H_n \geq k\}\) for \(k = (1-\varepsilon)\frac{1}{h_2} \log n\). We use two techniques to achieve it: first, we reduce the problem to a simpler one on tries with weaker dependency among keys, and second we apply the second moment method to evaluate the appropriate probability.

Note that \(H_n\) is stochastically non-decreasing, that is, if \(m \leq n\), then \(H_m \leq_{st} H_n\), where \(\leq_{st}\) means stochastically smaller, and hence (cf. [35])

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

with \(k = O(\log n)\). We select \(m\) in such a way that the probability of the right-hand side (RHS) of the above will be easier to evaluate than the original probability. In order to
estimate $\Pr\{H_m \geq k\}$ we use the second moment method (cf. Chung and Erdős [9]), which states that for events $A_i$
\[
\Pr\left(\bigcup_{i=1}^{m} A_i \right) \geq \frac{(\sum_{i=1}^{m} \Pr\{A_i\})^2}{\sum_{i=1}^{m} \Pr\{A_i\} + \sum_{i \neq j} \Pr\{A_i \cap A_j\}}.
\] (3.7)

In our case, we set $A_{i,j} = \{C_{i,j} \geq k\}$, and hence $\Pr\{H_m \geq k\} = \Pr\{\bigcup_{i,j=1}^{m} A_{i,j}\}$. Our aim
is to prove that for $k = (1 - \varepsilon)\frac{1}{H_2} \log n$ the probability $\Pr\{H_m \geq k\}$ tends to 1, hence also
by (3.6) we have $\Pr\{H_n \geq (1 - \varepsilon)\frac{1}{H_2} \log n\} \rightarrow 1$.

In order to fulfill this plan we must solve several problems. First of all, we introduce
a new trie with height $H_m$ such that (3.6) holds. We illustrate the idea in the case of
the Bernoulli model. We partition the string $X_1^n$ into $m = n/k$ consecutive substrings
of length $k = O(\log n)$. Naturally, the first $k$ symbols of these $m$ substrings (keys), say $Y(1), Y(2), \ldots, Y(m)$, are independent in the Bernoulli model, and we can construct a trie
from these $m$ keys. Denote such a tree as $T_m$. By the sample path comparison [35], we
 can construct such a realization of $T_m$ that its height $H_m$ is smaller (in the sample path
sense) than in our original suffix tree $S_n$, hence in particular (3.6) holds. The evaluation
of $\Pr\{H_m \geq k\}$ in $T_m$ is easy since independent tries $T_m$ were studied very extensively in
the last decade. In particular, the reader is referred to Pittel [30] and Szpankowski [37]
who proved that $\Pr\{H_m \geq (1 - \varepsilon)\frac{1}{H_2} \log n\} = 1 - O(\frac{\log n}{n^e})$. This together with (3.6) would
complete the proof of the lower bound for the Bernoulli model.

The general model is more intricate since the keys $Y(1), \ldots, Y(m)$ are not independent.
However, using our strong $\alpha$-mixing condition we can partition the original sequence $X_1^n$
such that different parts are weakly dependent. This is illustrated in Figure 4. We divide
the sequence $X_1^n$ into $m$ parts which are used to construct a new digital tree $T_m$. Each new
key $Y(i)$ consists of the first $k$ symbols at the position $(i - 1)k + (i - 1)d + 1$, and a gap
of size $d$ that is not a part of $Y(i)$. Hence, $m = n/(k + d)$, and it will be convenient to
assume that $d = k$, so $m = O(n/\log n)$. More formally, the first $k$ symbols of the $i$th key
are defined as follows $\{y(i)\}_{i=1}^{k} = \{X_{\ell}\}_{\ell=(i-1)k+(i-1)d+1}^{i(k+(i-1)d)}$, while the other symbols of $Y(i)$
are generated arbitrary (they will not be involved in any further computation), however, to establish formally (3.6) we need to assume that the symbols of $Y(i)$ after the $k$th one coincide with symbols of $X_{i(k+d)+1}^\infty$. Now, we construct the trie $T_m$ from the keys $Y(1), \ldots, Y(m)$. From the definition of $T_m$ we conclude that (3.6) holds, hence one needs only to estimate $H_m$ around $k = (1 - \varepsilon)^{1/2} \log n$ in $T_m$ (e.g., using the second moment method). We note that $T_m$ is built from weakly dependent sequences $Y(1), \ldots, Y(m)$, hence one can expect that techniques used for independent tries (cf. [13], [37]) should work in this case too.

Let $A_{i,j} = \{C_{i,j} \geq k\}$ be the event that the alignment\(^2\) $C_{i,j}$ between $Y(i)$ and $Y(j)$ keys is greater than $k = O(1 \log n)$. By the second moment method (cf. (3.7)) we have

$$\Pr\{H_m \geq k\} \geq \frac{\left(\sum_{i,j \in D} \Pr\{A_{ij}\}\right)^2}{\sum_{i,j \in D} \Pr\{A_{ij}\} + \sum_{(i,j) \neq (t,s)} \Pr\{A_{ij} \cap A_{ts}\}}$$

(3.8)

where $D = \{(i, j) : 1 \leq i \leq m, 1 \leq j \leq m$ and $i \neq j\}$. We evaluate each term in (3.8) separately. Using the strong $\alpha$-mixing condition, and arguing as in the case of the upper bound (cf. the first inequality after (3.3)) we immediately obtain for $k = O(\log n)$

$$(m^2 - o(m^2))(1 - \alpha(d_n))E P(w_k) \leq \sum_{i,j \in D} \Pr\{A_{ij}\} \leq (m^2 - o(m^2))(1 + \alpha(d_n))E P(w_k),$$

where $d_n$ is the length of the gap between the keys $Y(i)$ and $Y(i+1)$ (see Fig. 4). The second sum in the denominator of (3.8) is estimated as follows. Let $w_k g w_k$ be a concatenation of a string-ruler $w_k$, a gap-string $g$, and again the string-ruler $w_k$. Note that

$$\Pr\{A_{ij} \cap A_{ts}\} = \sum_{w_k} \sum_{w_k'} P(w_k g_1 w_k \cap w_k' g_2 w_k').$$

(3.9)

In order to estimate the RHS of (3.9), we consider two cases:

**Case A. The gaps $g_1$ and $g_2$ do not overlap.**

In this case the events $A_{ij}$ and $A_{st}$ are separated by a gap of length at least $d_n$ (cf. Fig. 4), hence

$$\sum_{w_k} \sum_{w_k'} P(w_k g_1 w_k \cap w_k' g_2 w_k') \leq (1 + \alpha(d_n)) \sum_{w_k} \sum_{w_k'} P(w_k g_1 w_k) P(w_k' g_2 w_k')$$

$$\leq (1 + \alpha(d_n))^3 \sum_{w_k} \sum_{w_k'} P^2(w_k) P^2(w_k') = (1 + \alpha(d_n))^3 E^2 P(w_k).$$

\(^2\)We now refer to $C_{i,j}$ as an alignment instead of the self-alignment since the trie $T_m$ is built from weakly dependent ("almost independent") sequences $Y(1), \ldots, Y(m)$. 23
Case B. The gaps $g_1$ and $g_2$ overlap.

Let $g$ be the overlapping string of $g_1$ and $g_2$, that is, $g_1 = g_1'g$ and $g_2 = gg_2'$. We consider two subcases:

(B1) Neither $g_1'$ nor $g_2'$ is null. Then, it is easy to see that we can reduce this case to the previous case A with $g_1$ and $g_2$ replaced by $g_1'$ and $g_2'$.

(B2) One of the strings $g_1'$ and $g_2'$ is empty, that is, two keys out of the following four strings $Y(i), Y(j), Y(s)$ and $Y(t)$ are the same, say the $i$th one and the $t$-th one. Then,

$$
\Pr\{A_{ij} \cap A_{it}\} = (A) \sum_{W_k} P(w_k^3) \leq c_1 \sum_{W_k} P^3(w_k)
\leq (B) c_1 \left( \sum_{W_k} P^2(w_k) \right)^{3/2}
= c_1 (EP(w_k))^{3/2},
$$

where (A) follows directly from (3.9) (by setting $w_k = w_k'$), and (B) is a consequence of the following inequality, which can be found in Karlin and Ost [20] and Szpankowski [37],

$$\ell \geq r \quad \Rightarrow \quad \left( \sum_{W_k} P^\ell(w_k) \right)^{1/\ell} \leq \left( \sum_{W_k} P^r(w_k) \right)^{1/r}.
$$

Finally, the above implies the following estimate for the second sum in the denominator of (3.8)

$$
\sum_{(i,j) \neq (i,s)} \Pr\{A_{ij} \cap A_{is}\} \leq m^4(1 + \alpha(d_n))^3E^2P(w_k) + m^3c(EP(w_k))^{3/2}
$$

where $c$ is a constant.

Putting everything together, inequality (3.8) becomes for $k = (1 - \varepsilon) \frac{1}{h_2} \log n$ with $EP(w_k) \sim n^{-2(1-\varepsilon)}$

$$
\Pr\{H_m \geq (1 - \varepsilon) \frac{1}{h_2} \log n\} \geq \frac{1}{n^{2(1-\varepsilon)} \frac{1 + \alpha(d_n)}{(1 - \alpha(d_n))}^2 + (1 - o(1))(1 + \alpha^2(d_n)) + o(\alpha^2(d_n)) + c_1 \frac{n^1 - \varepsilon}{m^2}}
\geq 1 - c_1 \frac{n^2(1-\varepsilon)}{m^2} - c_2 \frac{n^{1-\varepsilon}}{m} - c_3 \alpha^2(d_n)
$$

Now, setting $n^{1-\varepsilon}/m \to 0$ (e.g., $m = n/\log n$ and $d_n = \Theta(\log n)$), we finally obtain

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

which shows the lower bound for $H_m$, and hence by (3.6) also for $H_n$ in our original suffix tree $S_n$. In summary, (3.5) and (3.11) lead to the following

$$
\Pr\{\left| \frac{H_n}{\log n} - \frac{1}{h_2} \right| \geq \varepsilon\} \leq c_1 \frac{\log n}{n^{\varepsilon}} + c_2 \alpha^2(\log n) \to 0
$$

(3.12)
for some constants $c_1$ and $c_2$. This proves $H_n/\log n \to 1/h_2$ (pr.). To establish Theorem 2(i) we must extend the above to the almost sure convergence, which is discussed below.

C. Almost Sure Convergence

The estimate in (3.12) does not yet allow us to use the Borel-Cantelli Lemma to prove the almost sure convergence. But, the fact that $H_n$ is nondecreasing and the fact that $\log n$ is slowly varying function will allow to show the almost sure convergence (see also Remark 2(ii)). To do so we apply the trick suggested by Kesten and reported by Kingman in [22] (cf. also [30]). The idea is to replace $n$ by $s2^r$ for some integers $s$ and $r$. Note then for $n = s2^r$ the estimate (3.12) implies that

$$\sum_{r=0}^{\infty} \Pr\{|H_{s2^r}/\log(s2^r) - 1/h_2| \geq \varepsilon\} < \infty$$

provided

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

which holds for example for $\alpha(n) = O(n^{-1/2-\delta})$ for some $\delta > 0$.

To finish the proof we need to translate (3.13) for every $n$. Fix $s$. Naturally, for every $n$ we find such $r$ that

$$s2^r \leq n \leq (s+1)2^r.$$

By the above and (3.13), together with the fact that logarithm is a slowly varying function, we have

$$\lim\sup_{n \to \infty} \frac{H_n}{\log n} \leq \lim\sup_{r \to \infty} \frac{H_{(s+1)2^r}}{\log(s+1)2^r} \frac{\log(s+1)2^r}{\log s2^r} = \frac{1}{h_2} \quad (a.s.)$$

for $s \to \infty$. In a similar manner, we prove the lim inf, and after some algebra we get

$$\lim\inf_{n \to \infty} \frac{H_n}{\log n} \geq \lim_{s,r \to \infty} \inf \frac{H_{s2^r}}{\log(s+1)2^r} = \frac{1}{h_2} \quad (a.s.).$$

This leads to

$$\lim_{n \to \infty} \frac{H_n}{\log n} = \frac{1}{h_2} \quad (a.s.)$$

which proves our Theorem 2(i).

3.2 The Shortest Feasible Path in a Suffix Tree

Now we concentrate on establishing the almost sure convergence for the shortest feasible path length $s_n$ in a suffix tree $S_n$. Our proof resembles the derivation used by Pittel [30] for independent tries (cf. also [19]). We need some more notation. We write $(X^n_k, w_k)$ to
denote the set of positions on which $X^n$ and $w_k$ agree, that is, $i \in (X^n, w_k)$ if the $i$th suffix $X^{i\infty}$ agrees entirely with $w_k$ (i.e., on $k$ positions). Moreover, let $C(X, w)$ be the alignment between $X$ and $w$, that is, the length of the longest common prefix of $X$ and $w$. Clearly, $C(X, w_k) \leq k$. Finally, we define
\[
p_{\min}(k) = \min_{w_k \in \mathcal{W}_k} \{P(w_k)\}
\] (3.16)
Note that according to our definition (2.6) we have $p_{\min}(k) \sim e^{-h_1 k}$ for large $k$.

A. Upper Bound

Define $w_{\min}$ as $P(w_{\min}) = p_{\min}(k)$, so $w_{\min} \in \mathcal{W}_k$. Let $\{s_n > k\}$. Then, up to the level $k$, the suffix tree $S_n$ can be modeled as a complete tree, that is, all nodes of the depth not higher than $k$ have the maximum degree equal to $V$. This implies that for every word $w_k \in \mathcal{W}_k$ there must exist at least one suffix of $X^n$ whose prefix of length $k$ agreee with $w_k$. Therefore, the set $\langle X^n, w_k \rangle$ is nonempty, i.e., $|\langle X^n, w_k \rangle| \geq 1$. This is particularly true for the word $w_{\min}$. Hence
\[
Pr\{s_n > k\} \leq Pr\{|\langle X^n, w_{\min} \rangle| \geq 1\} .
\] (3.17)
But $|\langle X^n, w_{\min} \rangle| \geq 1$ implies that at least one depth is greater than $k$ which further implies that there exists an index, say $1 \leq i \leq n$, such that $C(X^n, w_{\min}) = k$. Then, by Boole's inequality
\[
Pr\{s_n > k\} \leq nPr\{C(X^n, w_{\min}) = k\} = np_{\min}(k) .
\] (3.18)
The rest is easy. Let $k = (1 + \varepsilon)\frac{1}{h_1} \log n$. Then, using the definition of $h_1$ and (3.18) we finally obtain
\[
Pr\{s_n > (1 + \varepsilon)\frac{1}{h_1} \log n\} \leq \frac{c}{n^\varepsilon} ,
\] (3.19)
which proves the desired upper bound.

B. Lower Bound

The lower bound is more intricate. Fortunately, we can use the same trick as in the case of the lower bound for the height. We partition $X^n$ into $m$ keys $Y(1), \ldots, Y(m)$ separated by $m$ gaps of size $d$ (cf. Fig. 4). We construct a trie $T_m$ from those $m$ weakly dependent keys (for detailed construction see previous subsection). Let $s_m$ be the shortest feasible path in $T_m$. Then, due to the fact that $s_m \leq s_n$, we have
\[
Pr\{s_n < k\} \leq Pr\{s_m < k\} .
\] (3.20)
Now, we need only to investigate the reduced tree $T_m$. But the event $\{s_n < k\}$ implies that there exists a word $w_k \in W_k$ such that longest common prefixes between $w_k$ and the keys $Y(1), \ldots, Y(m)$ are of lengths smaller than $k$. That is,

$$\{s_n < k\} \Rightarrow \exists w_k \in W_k \{C(Y(1), w_k) < k, \ldots, C(Y(m), w_k) < k\}.$$  (3.21)

This is the same as in Pittel [30] since the condition (3.21) is naturally also true for independent tries.

By the strong $\alpha$-mixing condition (2.5), and the above we have

$$\Pr\{s_n < k\} \leq \sum_{W_k} \Pr\{C(Y(1), w_k) < k, \ldots, C(Y(m), w_k) < k\} \leq \sum_{W_k} (1 + \alpha(d_n))^m (1 - P(w_k))^m \leq (1 + \alpha(d_n))^m \sum_{W_k} (1 - p_{\min}(k))^m \leq V_k^k (1 + \alpha(d_n))^m (1 - p_{\min}(k))^m.$$  

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

$$\Pr\{s_n < (1 - \varepsilon)\frac{1}{h_1} \log n\} \leq (1 + \alpha(\log n))^m \exp(-n^{\varepsilon/2}/\log n).$$  (3.22)

To complete our derivation we recall the condition (2.9) which implies that $(1 + \alpha(\log n))^m \leq cn^\beta$ for some constants $c$ and $\beta$. Then, (3.22) becomes

$$\Pr\{s_n < (1 - \varepsilon)\frac{1}{h_1} \log n\} \leq cn^\beta \exp(-n^{\varepsilon/2}/\log n).$$  (3.23)

which completes the proof of the convergence in probability.

For the almost sure convergence, we apply the same arguments as in the case of the height since $s_n$ is also a nondecreasing sequence. Note, however, that in this case we need only to re-consider the upper bound since the Borel-Cantelli Lemma can be directly applied to (3.23). That is, we set $n = s2^r$ and due to the the monotonicity property of $s_n$, we finally prove

$$\lim_{n \to \infty} \frac{s_n}{\log n} = \frac{1}{h_1} \quad (a.s.)$$  (3.24)

as needed for Theorem 2(ii).

ACKNOWLEDGMENT

It is my pleasure to acknowledge numerous enlightening discussions that I have had with Professor Boris Pittel and Professor Luc Devroye on the subject of this paper and related topics. The author is particularly obliged to Professor Paul Shields for bringing to his attention the existence of what is called in this paper the right and the left domain asymptotics. Finally, comments of two conscientious referees led to the improved presentation and elimination of some slips in the first draft of this paper.
References


[34] P. Shields, Private Communication.


