IEEE Trans. Information Theory, 39, 1993.

ASYMPTOTIC PROPERTIES OF DATA COMPRESSION AND SUFFIX TREES*

September 29, 1993

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 *almost surely* (a.s.) 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 nth block in the Lempel-Ziv parsing algorithm reveals a similar behavior. We relate our findings to some problems on digital trees, namely the asymptotic behavior of a (noncompact) suffix tree built from suffixes of a random 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).

Index Terms – universal data compression, Lempel-Ziv parsing algorithm, suffix trees, repeated subwords, asymptotic analysis.

---

*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, its extension CCR-9201078, and NCR-9206315, 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 reduce the size of the original sequence (e.g., universal data compression schemes [7], [27], [43]). 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 [11]), 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 Σ 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 (e.g., sliding window implementation in [7]) – 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 [41]. 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 [41], we define two parameters relevant to the performance of some data compression schemes. For every n, let $\widetilde{L}_n$ be the smallest integer $L > 0$ such that

$$X_0^{L-1} \neq X_{-m}^{-m+L-1} \qquad \text{for all} \qquad 1 \leq m \leq n \;, \tag{1.1a}$$

(i.e., $\widetilde{L}_n - 1$ 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 $X_0^{\widetilde{L}_n-1}$, determines m_0 such that $X_0^{\widetilde{L}_n-2} = X_{-m_0}^{-m_0+\widetilde{L}_n-2}$, and transmits m_0, $\widetilde{L}_n$ and $X_{\widetilde{L}_n-1}$. To encode m_0 we need $\log_V n$ symbols, and it is known that $\widetilde{L}_n$ may be represented by $\log \widetilde{L}_n$ bits (cf. [33]). As noted by Wyner and Ziv [41], the number of encoded symbols per source symbol is asymptotically

$$\frac{\log_V n}{\widetilde{L}_n} + \frac{\log \widetilde{L}_n}{\widetilde{L}_n} + \frac{\log V}{\widetilde{L}_n} \;.$$

Hence, the ratio $\log_V n / \widetilde{L}_n$ determines an asymptotic efficiency of the compression scheme.

Another parameter of interest can be defined as follows. For every integer ℓ, let $\widetilde{N}_\ell$ 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 ℓ is repeated for the first time in a data base of size $\widetilde{N}_\ell$. Wyner and Ziv [41] suggested the following compression scheme. The encoder sends the first n source symbols, say X_{-n}^{-1} with no compression, but the next ℓ symbols $X_0^{\ell-1}$ are encoded as follows: if $X_0^{\ell-1}$ is a substring of $X_{-n}^{\ell-2}$ (i.e., $\widetilde{N}_\ell \le n$), then $X_0^{\ell-1}$ is compressed by specifying only $\widetilde{N}_\ell$; otherwise $X_0^{\ell-1}$ is not compressed. As noted in [41], in this scheme the average number of symbols required to encode $X_0^{\ell-1}$ is $\Pr\{\widetilde{N}_\ell \le n\}\log_V n + \Pr\{\widetilde{N}_\ell > n\}\ell + 1$, where $\log_V n$ is the number of symbols required to transmit $\widetilde{N}_\ell$.

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 [7]) of data compression schemes, and most of our analyses deal with this model. We can define two parameters L_n and N_ℓ which correspond to $\widetilde{L}_n$ and $\widetilde{N}_\ell$ in the static model. More specifically, L_n is defined as the *largest* value of L such that

$$X_{m_0}^{m_0+L} = X_{n+1}^{n+1+L} \qquad \text{for some} \quad m_0 \in \{1,\ldots,n\}\ . \tag{1.2}$$

In a similar fashion, N_ℓ 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_ℓ 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 [27]. 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 $(1)(10)(10100)(11)(11\cdots)$; that is, the first block has length one, the second block length has two, the next one is of length five, and so on.[1] Observe that the third block is the longest prefix of X_2^∞ and X_4^∞. Grassberger [15] has shown how to construct such a

[1] There is another way of parsing a sequence in which phrases do not overlap. For example, our sequence

parsing by using a special data structure called *suffix tree* (cf. [1], [4], [15]). Naturally, one is interested in the length of a block in the parsing algorithm. Let l_n be the length of the nth block in the Lempel-Ziv parsing algorithm. There is a relationship between l_n and the parameter L_n defined above, namely $l_n = L_{\sum_{k=1}^{n-1} l_k}$ (cf. Section 2 for details). In view of this, the asymptotic behavior of L_n can be used to obtain the asymptotic bounds for the length l_n 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], [33], [40]). 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], [9], [14], [18], [28], [40]). In particular, recently Chang and Lawler [11] 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 [15]). 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) $\mathcal{X} = \{X(1), \ldots, X(n)\}$, each key being a sequence from a finite alphabet Σ, that is, for $1 \le \ell \le n$ 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

is parsed according to this algorithm as $(1)(10)(101)(00)(11)(11\ldots)$. In this paper, we do *not* consider such a parsing scheme.

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, $\mathcal{X} = \{S_1, S_2, ..., S_n\}$ where S_i is the ith suffix of a one-sided *single* sequence $\{X_k\}_{k=1}^{\infty}$, then the trie built from $\mathcal{X}$ 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 $X = 0\,1\,0\,1\,1\,0\,1\,1\,1\,0....$. Then the first five suffixes are $S_1 = 0\,1\,0\,1\,1\,0\,1\,1\,1\,0....$, $S_2 = 1\,0\,1\,1\,0\,1\,1\,1\,0....$, $S_3 = 0\,1\,1\,0\,1\,1\,1\,0....$, $S_4 = 1\,1\,0\,1\,1\,1\,0....$ and $S_5 = 1\,0\,1\,1\,1\,0....$. 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 $\mathcal{S}_n$ be the suffix tree constructed from the first n suffixes of a sequence $\{X_k\}_{k=1}^{\infty}$. Then, the depth of the ith suffix $L_n(i)$ in $\mathcal{S}_n$ is the length of the path from the root to this suffix. 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 $\mathcal{S}_n$. Naturally, we call this parameter the *depth of insertion.* In the next section, we will show that this L_n coincides with the $\widetilde{L}_n$ 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 [41]. In fact, the authors of [41] studied the length $\widetilde{L}_n$ in the left domain asymptotic. They established the following asymptotic result.

Theorem 0. (Wyner and Ziv [41]) *Let $\{X_k\}_{k=-\infty}^{\infty}$ be a stationary and ergodic sequence*

$S_1 = 0\ 1\ 0\ 1\ 1\ 0\ 1\ 1\ 1\ 0$
$S_2 = 1\ 0\ 1\ 1\ 0\ 1\ 1\ 1\ 0$
$S_3 = 0\ 1\ 1\ 0\ 1\ 1\ 1\ 0$

$S_4 = 1\ 1\ 0\ 1\ 1\ 1\ 0$
$S_5 = 1\ 0\ 1\ 1\ 1\ 0$

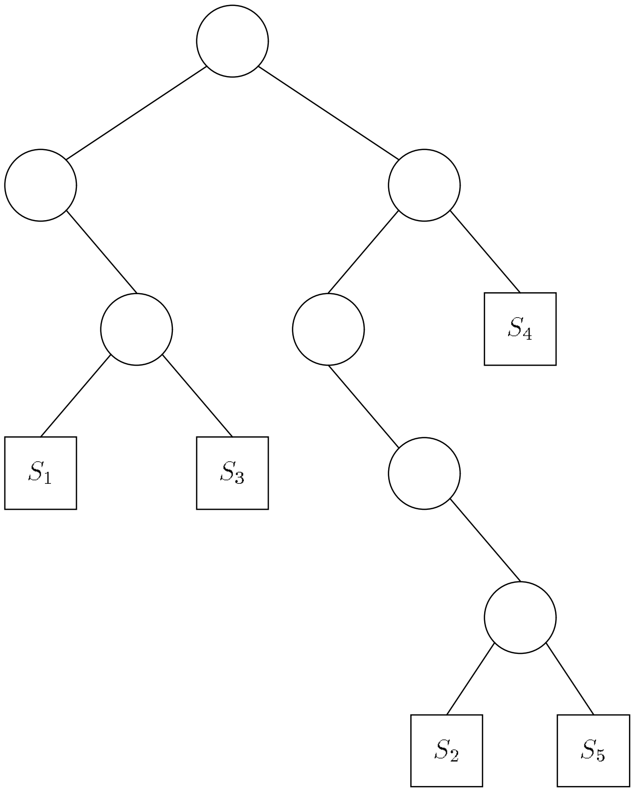

Figure 1: Suffix tree built from the first five suffixes of $X = 0\,1\,0\,1\,1\,0\,1\,1\,1\,0....$

built over a finite alphabet Σ*. Then, in the left domain asymptotic as* $n \to \infty$

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

and

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

where h *is the entropy of* X*.* ■

This result concerns the convergence *in probability* (pr.) of $\widetilde{L}_n$. In fact, a similar results also holds for L_n in the right domain asymptotics (cf. [35], [39]). Wyner and Ziv [41] asked whether it can be extended to a stronger *almost sure* (a.s.) convergence. In the right domain asymptotic, we shall settle this question in the negative for the Markovian case, and show that L_n oscillates with probability one between $(1/h_1)\log n$ and $(1/h_2)\log n$ where $h_2 < h \le h_1$. 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 $\widetilde{L}_n$ 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 $\widetilde{L}_n$ is a nondecreasing sequence as opposed to L_n. In the non-Markovian case, the proof for the (a.s.) convergence $\widetilde{L}_n$ is more intricate and due to Ornstein and Weiss [30].

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 $\mathcal{S}_n$, 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 $\mathcal{S}_n$ but its predecessor node (either an internal or an external one) is in $\mathcal{S}_n$. 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. [43], [27]; see also [7]), Wyner and Ziv [41], [42], and Kieffer [22]. The average case analysis of suffix trees was initialized by Grassberger [15], and Apostolico and Szpankowski [5]. For the Bernoulli model, the asymptotic behavior of the height was recently obtained by Devroye, Szpankowski and Rais [14], and the limiting distribution of the depth in a suffix tree is reported in Jacquet and Szpankowski [18]. Arratia and Waterman [6] investigated a related problem, namely the longest contiguous matching within a single sequence, and obtained several interesting results in this direction. Their findings are related to the hight of a suffix tree. Finally, heuristic arguments were used by Blumer *et al.* [9] 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 [18] established rigorously the latter result regarding the average size of a suffix tree. Some related topics were discussed by Guibas and Odlyzko in [16].

Our findings were inspired by the seminal paper of Pittel [31] 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, discuss some consequences of our findings, and suggest some further studies. Most proofs are delayed till Section 3 which is of its own interest.

2. MAIN RESULTS

Let $\{X_k\}_{k=-\infty}^{\infty}$ be a *stationary ergodic* sequence of symbols generated from a finite alphabet Σ of size V. Define a partial sequence X_m^n as $X_m^n = (X_m, ..., X_n)$ for $m < n$, and

k	-4	-3	-2	-1	0	1	2	3	4	5
X_k	0	1	0	1	1	0	1	1	1	0

Figure 2: A sample of data used in Example 2.1

the nth order probability distribution as follows

$$P(X_1^n) = \Pr\{X_k = x_k,\ 1 \le k \le n,\ x_k \in \Sigma\} \ . \tag{2.1}$$

The *entropy* of $\{X_k\}$ is

$$h = \lim_{n\to\infty} \frac{E \log P^{-1}(X_1^n)}{n} \ , \tag{2.2}$$

The existence of the above limit is guaranteed by Shannon's Theorem (cf. [8]). It is also known that $h \le \log V$. Hereafter, all logarithms – unless stated explicitly otherwise – are natural logarithms.

It is well known (cf. [12], [13], [22]) 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 [27], [43]. Following Wyner and Ziv [41], we defined in the Introduction two parameters to capture certain properties of repeated subsequences, namely L_n and N_ℓ (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

$$X_m^{m+L-1} \ne X_{n+1}^{n+L} \quad \text{for all} \quad 1 \le m \le n \ . \tag{2.3}$$

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 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 [41], and used in our Example 1.4. Then, in the left domain (cf. (1.1b)) one finds $\widetilde{L}_4 = 5$, and also $\widetilde{N}_3 = \widetilde{N}_4 = 3$ but $\widetilde{N}_5 > 4$.

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). $\Box$

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. [8]). Let $\mathcal{F}_m^n$ be a σ-field generated by $\{X_k\}_{k=m}^n$ for $m \le n$. It is said that $\{X_k\}$ satisfies the mixing condition if there exist two positive constants $c_1 \le c_2$ and an integer d such that for all $-\infty \le m \le m+d \le n$ the following holds

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

where $\mathcal{A} \in \mathcal{F}_{-\infty}^m$ and $\mathcal{B} \in \mathcal{F}_{m+d}^{\infty}$. It is known that this condition implies ergodicity of the sequence $\{X_k\}_{k=-\infty}^{\infty}$ (cf. [8]). In some statements of our results, we need a stronger form of the mixing condition, namely the *strong α-mixing condition* which reads as follows

$$(1-\alpha(d))\Pr\{\mathcal{A}\}\Pr\{\mathcal{B}\} \le \Pr\{\mathcal{A}\mathcal{B}\} \le (1-\alpha(d))\Pr\{\mathcal{A}\}\Pr\{\mathcal{B}\} \tag{2.5}$$

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

Following Pittel [31], we define two new parameters of $\{X_k\}$, namely

$$h_1 = \lim_{n\to\infty} \frac{\max\{\log P^{-1}(X_1^n)\ , P(X_1^n) > 0\}}{n} = \lim_{n\to\infty} \frac{\log(1/\min\{P(X_1^n)\ , P(X_1^n) > 0\})}{n}\ , \tag{2.6}$$

$$h_2 = \lim_{n\to\infty} \frac{\log(E\{P(X_1^n)\})^{-1}}{2n} = \lim_{n\to\infty} \frac{\log\left(\sum_{X_1^n} P^2(X_1^n)\right)^{-1}}{2n}\ . \tag{2.7}$$

The existence of h_1 and h_2 was established by Pittel [31] who also noticed that $0 \le h_2 \le h \le h_1$. It should be noted that h_2 is the second-order Rényi entropy, while h_1 could be interpreted as Rényi entropy of order $-\infty$ (cf. [13]).

Remark 1.

(i) *Bernoulli Model.* In this model, symbols from the alphabet Σ are generated independently, that is, $P(X_1^n) = P^n(X_1^1)$. In particular, we assume that the ith symbol from the alphabet Σ is generated according to the probability p_i, where $1 \le i \le V$ and $\sum_{i=1}^V p_i = 1$. Thus, $h = \sum_{i=1}^V p_i \log p_i^{-1}$ ([8]), $h_1 = \log(1/p_{min})$ and $h_2 = 2\log(1/P)$ where $p_{min} = \min_{1\le i\le V}\{p_i\}$ 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.

(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 $p_{i,j} = \Pr\{X_{k+1} = j \in \Sigma | X_k = i \in \Sigma\}$, and the transition matrix is $\mathbf{P} = \{p_{i,j}\}_{i,j=1}^V$. It is well known [8] that the entropy h can be computed as $h = -\sum_{i,j=1}^V \pi_i p_{i,j} \log p_{i,j}$ where π_i is the stationary distribution of the Markov chain. The other quantities, that is, h_1 and h_2, are a little harder to evaluate. Pittel [31] and Szpankowski [38] evaluated the height of regular tries with Markovian dependency, and they showed that the parameter h_2 is a function of the largest eigenvalue θ of the matrix $\mathbf{P}_{[2]} = \mathbf{P} \circ \mathbf{P}$ which represents the Schur product of $\mathbf{P}$ (i.e., elementwise product). More precisely, $h_2 = (1/2)\log\theta^{-1}$. With respect to h_1 we need a result from digraphs (cf. Romanovski [34], Karp [21]). Consider a digraph on Σ with weights equal to $-\log p_{ij}$ where $\omega_i, \omega_j \in \Sigma$. Define a cycle $\mathcal{C} = \{\omega_1, \omega_2, ..., \omega_v, \omega_1\}$ for some $v \leq V$ such that $\omega_i \in \Sigma$, and let $\ell(\mathcal{C}) = -\sum_{i=1}^v \log(p_{\omega_i,\omega_{i+1}})$ (with $\omega_{v+1} = \omega_1$) be the total weight of the cycle $\mathcal{C}$. The quantity $\min_{\mathcal{C}}\{\ell(\mathcal{C})/v\}$ is well studied in the graph theory and it is called the minimum cycle mean. Clearly, $h_1 = \min_{\mathcal{C}}\{\ell(\mathcal{C})/|\mathcal{C}|\}$, and Karp [21] showed how to compute it efficiently. □

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,*

$$\lim_{n\to\infty} \inf_n \frac{L_n}{\log n} = \frac{1}{h_1} \qquad (a.s.) \qquad \lim_{n\to\infty} \sup_n \frac{L_n}{\log n} = \frac{1}{h_2} \tag{2.8}$$

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

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

for some constants $0 < \rho < 1$ and β. ■

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. [8]) 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 who constructed – using his approach from [35] – an ergodic mixing stationary sequence that does not satisfy the lim sup part of (2.8).

(ii) *Asymptotic behavior of* $\widetilde{L}_n$ *in the left domain asymptotic.* We now show that in the left domain asymptotic the (a.s.) behavior of $\widetilde{L}_n$ is the one predicted by Wyner and Ziv. From their proof of the convergence *in probability* (cf. [41] pp. 1253-1255), one concludes that $\Pr\{|\widetilde{L}_n/\log n - 1/h| > \varepsilon\} = O(1/\sqrt{\log n}) + P(B_n)$, where $P(B_n)$ is the probability of "bad states" in the McMillan-Shannon theorem (cf. [8]). 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. [13]). In order to apply the Borel-Cantelli Lemma, we use the trick suggested by Kingman [23]; that is, we construct a subsequence n_r of n for which $O(1/\sqrt{\log n_r})$ is summable. Fix s, and define $n_r = 2^{s^2 2^{2r}}$. Note that $\widetilde{L}_{n_r}/\log n_r \to 1/h$ (a.s.) provided $P(B_n)$ is summable. To prove that $\widetilde{L}_n/\log n$ converges (a.s.), we use two facts: (i) $\widetilde{L}_n$ is a nondecreasing sequence (cf. Example 2.1 below); (ii) for every n we can choose such r that $2^{s^2 2^{2r}} \le n \le 2^{(s+1)^2 2^{2r}}$. Then,

$$\lim_{n\to\infty} \sup \frac{\widetilde{L}_n}{\log n} \le \lim_{r\to\infty} \sup \frac{\widetilde{L}_r}{\log n_r} \frac{\log 2^{(s+1)^2 2^{2r}}}{\log 2^{s^2 2^{2r}}} \to \frac{1}{h}\frac{(s+1)^2}{s^2} \quad (a.s.)\ , \tag{2.10}$$

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). □

Theorem 1 will be proved below as a simple consequence of some new results concerning a typical behavior of a suffix tree $\mathcal{S}_n$ 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 $\mathcal{S}_n$. This will also lead to some new results about suffix trees.

Define for $\mathcal{S}_n$ the mth *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 mth external node containing the mth 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 mth external node. Then,

$$H_n = \max_{1 \le m \le n} \{L_n(m)\} \ . \tag{2.11}$$

The shortest feasible path is defined as follows. Consider a suffix tree $\mathcal{S}_n$, and append it with *available nodes*, that is, nodes that are not in the tree $\mathcal{S}_n$ but whose predecessors (either internal or external nodes) are in $\mathcal{S}_n$. 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 $\mathcal{S}_n$, that is, $L_n = L_{n+1}(n+1)$[2]. Finally, D_n is defined as the depth of a *randomly* selected external node, that is,

$$D_n = \frac{1}{n} \sum_{m=1}^{n} L_n(m) \ . \tag{2.12}$$

This typical depth depth D_n is used in the analysis of the average complexity of the exact matching algorithms (cf. Example 1.3)).

For our purpose, another characterization of the above parameters is more useful. For a suffix tree $\mathcal{S}_n$ built from n suffixes $S_1, S_2, ..., S_n$ 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. [38])

$$L_n(m) = \max_{1 \le k \le n \ , \ k \ne m} \{C_{k,m}\} + 1 \tag{2.13a}$$

$$H_n = \max_{1 \le i < j \le n} \{C_{i,j}\} + 1 \tag{2.13b}$$

and finally

$$L_n = \max_{1 \le m \le n} \{C_{m,n+1}\} + 1 \ . \tag{2.13c}$$

¿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).

[2]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 [41].

In passing, we note that the second parameter defined in Example 1.1, namely N_ℓ, can also be re-defined in terms of the associated suffix tree. Indeed, N_ℓ is the *size* of a suffix tree in which the depth of the first suffix in $\mathcal{S}_n$ is equal to ℓ, that is, $L_{N_\ell}(1) = \ell$. We should also point out that in the left domain asymptotic, the analysis of $\widetilde{N}_\ell$ (cf. (1.1b)) is much easier due to the following relationship with $\widetilde{L}_n$ (cf. [41])

$$\{\widetilde{N}_\ell > n\} = \{X_1^\ell \neq X_m^{m+\ell-1} \ , \quad 1 \leq m \leq n\} = \{\widetilde{L}_n \leq \ell\} \ .$$

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

Remark 3.

Asymptotic Behavior of N_ℓ. We can predict the (a.s.) behavior of N_ℓ in the right domain asymptotic since we have already observed that the quantity N_ℓ can be alternatively defined as $L_{N_\ell}(1) = \ell$. But, for the Markovian model it is known (cf. Szpankowski [39]) that $L_n(1)/\log n \to 1/h$ (a.s.). Hence, $\ell/\log N_\ell \to 1/h$ (a.s.). Interestingly enough, the asymptotic behavior of N_ℓ is the same in the left domain asymptotic, as proved recently by Ornstein and Weiss [30] for a general probabilistic model. This is in surprising contrast to the asymptotic behavior of L_n in these two domains of asymptotics. $\Box$

The quantity $\widetilde{L}_n$ defined by Wyner and Ziv [41], [42] (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 $\widetilde{L}_n$ 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 $\mathcal{S}_n$ is X_{-n-1}^{∞}, hence we refer to the left domain asymptotic. Finally, in Remark 2(ii) we already noted that $\widetilde{L}_n$ is a nondecreasing sequence in the sense that $\widetilde{L}_n \leq \widetilde{L}_{n+1}$. This is illustrated below.

EXAMPLE 2.2 *Illustration of definitions*

Let $X_1^\infty = 0\,1\,0\,1\,1\,0\,1\,1\,1\,0...$, 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

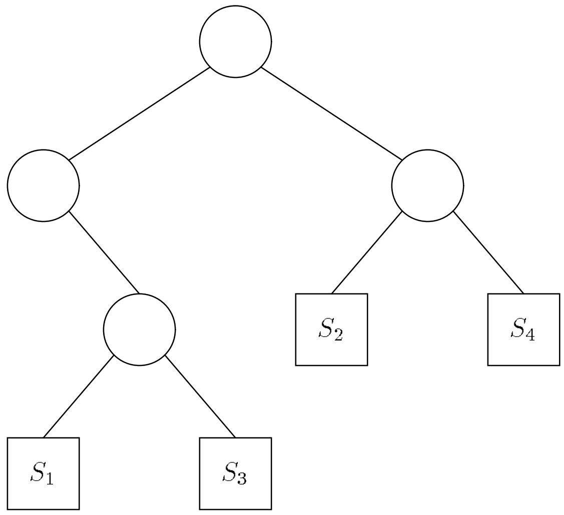

Figure 3: Suffix tree built from the first four suffixes of $X = 0\,1\,0\,1\,1\,0\,1\,1\,1\,0...$

$s_4 = 2$. 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 $\widetilde{L}_n$. In the Wyner-Ziv model, we must re-index the original sequence to obtain $X_{-4}^{\infty} = 0\,1\,0\,1\,1\,0\,1\,1\,1\,0...$. Let us add some new symbols to the left of the X_{-4}^{∞} keeping in mind that we always compare the depth of insertion for the zero-th suffix, $X_0^{\infty} = 1\,0\,1\,1\,1\,0...$. If we add $1\,0\,1\,1\,1$ from the left, and insert X_0^{∞}, then $\widetilde{L}_n$ becomes respectively: $\widetilde{L}_5 = 5$, $\widetilde{L}_6 = 5$, $\widetilde{L}_7 = 5$, $\widetilde{L}_8 = 5$ and finally $\widetilde{L}_9 = 6$. It confirms our observation that $\widetilde{L}_n$ is a nondecreasing sequence. $\Box$

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. Therefore, not surprisingly, Theorem 1 is a simple consequence of the following result which we prove in Section 3.

Theorem 2. *Let $\{X_k\}_{k=1}^{\infty}$ be a stationary ergodic sequence satisfying the strong α-mixing condition (2.5) together with $h_1 < \infty$ and $h_2 > 0$.*
(i) *Asymptotically for large n we have*

$$\lim_{n\to\infty} \frac{s_n}{\log n} = \frac{1}{h_1} \qquad (a.s.) \tag{2.14}$$

provided (2.9) holds.

(ii) *For large n the height H_n satisfies*

$$\lim_{n\to\infty} \frac{H_n}{\log n} = \frac{1}{h_2} \qquad (a.s.) \tag{2.15a}$$

provided the coefficient $\alpha(d)$ satisfies

$$\sum_{d=0}^{\infty} \alpha^2(d) < \infty \ , \tag{2.15b}$$

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 [31] for independent tries. We concentrate only on the lim sup part of Theorem 1. Note that by definition $L_n \le H_n$, hence $L_n/\log n \le H_n/\log n$, and obviously

$$\lim_{n\to\infty} \sup \frac{L_n}{\log n} \le \lim_{n\to\infty} \frac{H_n}{\log n} \qquad (a.s.)\ . \tag{2.16a}$$

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, $\Pr\{L_n = H_n \text{ i.o.}\} = 1$ 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

$$\lim_{n\to\infty} \sup \frac{L_n}{\log n} \ge \lim_{n\to\infty} \frac{H_n}{\log n} \qquad (a.s.)\ ; \tag{2.16b}$$

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 and unbounded (a.s.) 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. [27]). 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*

$$\frac{1}{h_1} \le \lim_{n\to\infty} \inf_n \frac{l_n}{\log n} \le \lim_{n\to\infty} \sup_n \frac{l_n}{\log n} \le \frac{1}{h_2} \qquad (a.s.)\ . \tag{2.17}$$

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 [15]). 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

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

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 [35], and Szpankowski [39]). 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_n is such that $\sum_{k=1}^{M_n} l_k = n$. Nevertheless, Corollary 3 is valid for finite sequences too. Indeed, this follows directly from the fact $M_n = O(n/\log n)$ [12], [27]. (A simple argument for $M_n = O(n/\log n)$ works as follows: It suffices to note that $n = \sum_{k=1}^{M_n} l_k \le \sum_{k=1}^{M_n} L_k \sim (1/h) M_n \log M_n$, where the last asymptotic is already known from Shields [35]).

(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 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 n weakly dependent sequences (e.g., in the Bernoulli model these sequences are practically independent), and in the view of $M_n = O(n/\log n)$, and the results of Pittel [31] 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 that we already discussed in

Example 1.2 (cf. [3]) in which the phrases do not overlap. Such a parsing scheme can be conveniently modeled by another digital tree, namely the so called *digital search tree* (cf. [3], [1], [26]). Then the length l_n of the nth block is exactly equal to the depth of the nth node in such a digital search tree. Applying Pittel's result for independent digital search trees [31], we can prove under weak mixing condition that (cf. [19])

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

where h_3 is defined as

$$h_3 = \lim_{n\to\infty} \frac{\log(1/\max\{P(X_1^n)\ ,\ P(X_1^n) > 0\})}{n} \ .$$

In the Bernoulli model, we have $h_3 = \log(1/p_{max})$ where $p_{max} = \max_{1\le i\le V} p_i$. Again, h_3 is the Rényi entropy of order $+\infty$.

(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 typical depth D_n and the depth of insertion L_n have the same limiting distribution, however, their almost sure behaviors are different. Recently, Jacquet and Szpankowski [18] showed that for the Bernoulli asymmetric model the normalized depth $(D_n - ED_n)/varD_n$ converges *in distribution* to the standard normal distribution $\mathcal{N}(0,1)$ with mean and variance as below

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

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

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 [37]). We conjecture that the same type of limiting distributions can be obtained for the Markovian model. This is due to two facts: (i) the limiting behavior of independent tries do not differ too much from asymptotics of suffix trees (cf. [18]); (ii) Jacquet and Szpankowski [17] 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 of a randomly selected phrase in the Lempel-Ziv parsing algorithm is related to the typical depth in a sequence of dynamically built suffix trees. This is due to the relationship $l_n = L_{\sum_{k=1}^{n-1} l_k}$, as already noted by us. More interestingly, the number of phrases M_n defined as $\sum_{k=1}^{M_n} l_k = n$, is related – but not in a simple manner – to the external path length E_n defined as $E_n = \sum_{m=1}^{n} L_n(m)$. Even in the Bernoulli model major difficulties arise in the evaluation of the limiting distribution of E_n in a suffix tree. For *independent* tries, Kirschenhofer, Prodinger and Szpankowski [24] recently obtained for the symmetric alphabet the variance of E_n which is $varE_n = (\alpha + P_3(\log n))n + O(\log^2 n)$ where $\alpha \approx 4.35\ldots$ (explicit formula for α can be found in [24]) and $P_3(\log n)$ is a fluctuating function. We conjecture that the limiting distribution of the external path length in a suffix tree converges to a normal distribution, but this might not be too easy to prove.

For the modified Lempel-Ziv parsing algorithm as discussed in Example 1.2 and Remark 4(ii) (without phrase overlapping), the situation is similar. This time, however, one needs to analyze an independent digital search tree, and this simplifies the problem. The independence is a consequence of nonoverlapping blocks in such an algorithm. The symmetric Bernoulli model was already analyzed in Aldous and Shields [3] who obtained the limiting distribution for the number of pharses M_n and the internal path length of the associated digital tree, however, without explicit formula for the variance. This was recently rectified by Jacquet and Szpankowski [19] who used a result of Kirschenhofer, Prodinger and Szpankowski [25] to show that $var\ M_n \sim (\beta + P_4(\log n))n/\log^3 n$ where $\beta \approx 0.26600\ldots$ (explict but complicated formula on β can be found in [25]), and $P_4(\log n)$ is a fluctuating function. The extension to the asymmetric model seems to be a challenging problem. Nevertheless, it was recently shown (cf. [19]) that the internal path length E_n and the number of phrases M_n are related as follows $\Pr\{M_n > m\} = \Pr\{E_m \leq n\}$s, so one needs the limiting distribution for E_n. Nothing is still known about the limiting distribution of E_n in the asymmetric model, however, we conjecture that $(E_n - (n/h)\log n)/var\ E_n \to N(0,1)$ where $var\ E_n = O(n \log n)$. □

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 [18] named *string-ruler* approach, and was already used by Pittel [31]. We also shall use some ideas from Szpankowski [38], and Devroye *et al.* [14].

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^ℓ we mean a concatenation of ℓ strings w_k from $\mathcal{W}_k$. If a subsequence X_m^{m+k} is equal to a string ruler w_k, then we write $P(w_k) = P(X_m^{m+k})$ for the probability that $X_m^{m+k} = w_k$. Finally, for a function $f(w_k)$ of w_k we write $\sum_{\mathcal{W}_k} f(w_k) = \sum_{w_k \in \mathcal{W}_k} f(w_k)$ 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

$$H_n = \max_{1 \leq i < j \leq n} \{C_{i,j}\} + 1 , \tag{3.1}$$

where $C_{i,j}$ is the self-alignment between subsequences X_i^∞ and X_j^∞. 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

$$\Pr\{C_{i,i+d} \geq k\} = \sum_{\mathcal{W}_{k\wedge d}} P(w_{k\wedge d}^{\lfloor \frac{k}{d} \rfloor + 1} \overline{w}_d) \tag{3.2}$$

where $\overline{w}_d$ is a prefix of w_d such that $|w_{k\wedge d}^{\lfloor k/d \rfloor + 1} \overline{w}_d| = k + \min\{k, d\}$, and $\lfloor k/d \rfloor$ is the integer part of k/d. Note that for $d \geq k$ the right-hand side (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 $Z = w_d^\ell \overline{w}_d$ 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 [14])

$$\Pr\{H_n \geq k\} \leq n \left(\sum_{d=1}^{k-1} \Pr\{C_{1,1+d} \geq k\} + \sum_{d=k}^{n} \Pr\{C_{1,1+d} \geq k\} \right) . \tag{3.3}$$

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

$$\sum_{d=k}^{n} \Pr\{C_{1,1+d} \geq k\} = \sum_{d=k}^{n} \sum_{\mathcal{W}_k} P(w_k^2) \leq nc_1 \sum_{\mathcal{W}_k} P^2(w_k) = nc_1 EP(w_k) , \tag{3.4}$$

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 $d \leq k$

$$\begin{aligned}
\Pr\{C_{1,1+d} \geq k\} &= \sum_{\mathcal{W}_d} P(w_d^{\lfloor k/d \rfloor + 1} \overline{w}_d) \leq^{(A)} c_1 \sum_{\mathcal{W}_d} P(w_d^{\lfloor k/d \rfloor} \overline{w}_d) P(w_d) \\
&\leq^{(B)} c_1 \sqrt{\sum_{\mathcal{W}_d} P^2(w_d^{\lfloor k/d \rfloor} \overline{w}_d) P(w_d)} \leq c_1 \sqrt{\sum_{\mathcal{W}_d} P^2(w_d^{\lfloor k/d \rfloor} \overline{w}_d)} \\
&\leq^{(C)} c_1 \sqrt{\sum_{\mathcal{W}_k} P^2(w_k)} = c_1 \sqrt{EP(w_k)} ,
\end{aligned}$$

where the inequality (A) is due to the mixing condition (2.4), inequality (B) is a consequence of the *inequality on means* (cf. [29]), 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

$$\Pr\{H_n \geq k\} \leq nc\left(k\sqrt{EP(w_k)} + nEP(w_k)\right) \ .$$

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

$$\Pr\{H_n \geq (1+\varepsilon)\frac{1}{h_2}\log n\} \leq \frac{c\log n}{n^{\varepsilon}} \to 0 \tag{3.5}$$

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 $H_m \leq_{\rm st} H_n$, where $\leq_{\rm st}$ means *stochastically smaller*, and hence (cf. [36])

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

with $k = O(\log n)$. We select m in such a way that the probability of the 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 Erdös [10]), which states that for events A_i

$$\Pr\{\bigcup_{i=1}^{m} A_i\} \geq \frac{(\sum_{i=1}^{m}\Pr\{A_i\})^2}{\sum_{i=1}^{m}\Pr\{A_i\} + \sum_{i\neq j}\Pr\{A_i\cap A_j\}} \ . \tag{3.7}$$

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 $\Pr\{H_n \geq (1-\varepsilon)\frac{1}{h_2}\log n\} \to 1$.

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

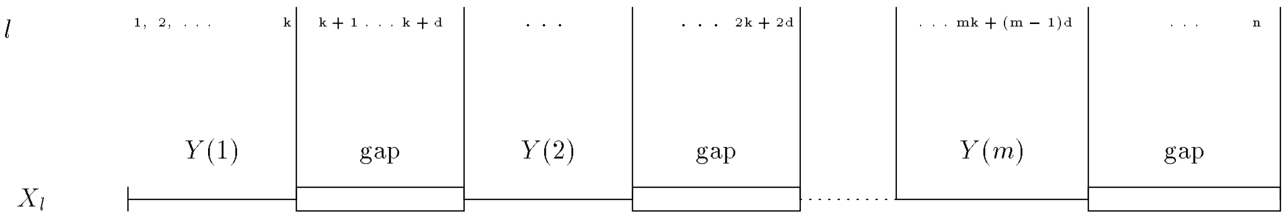

Figure 4: Illustration for the construction of the suffix tree $\mathcal{T}_m$

the Bernoulli model. We partition the string X_1^n into $m = \lfloor n/k \rfloor$ consecutive substrings of length $k = O(\log n)$. Naturally, the first k symbols of these m substrings (keys), say $Y(1), Y(2), ..., Y(m)$, are independent in the Bernoulli model, and we can construct a trie from these m keys. Denote such a tree as $\mathcal{T}_m$. By the *sample path comparison* [36], we can construct such a realization of $\mathcal{T}_m$ that its height H_m is smaller (in the sample path sense) than in our original suffix tree $\mathcal{S}_n$, hence in particular (3.6) holds. The evaluation of $\Pr\{H_m \geq k\}$ in $\mathcal{T}_m$ is easy since independent tries $\mathcal{T}_m$ were studied very extensively in the last decade. In particular, the reader is referred to Pittel [31] and Szpankowski [38] who proved that $\Pr\{H_m \geq (1-\varepsilon)\frac{1}{h_2}\log n\} = 1 - O(\frac{\log n}{n^\varepsilon})$. 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 $Y(1), .. , Y(m)$ are not independent. However, using our strong α-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 $\mathcal{T}_m$. 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, $m = \lfloor n/(k+d) \rfloor$, and it will be convenient to assume that $d = k$, so $m = O(n/\log n)$. More formally, the first k symbols of the ith key are defined as follows $\{Y_\ell(i)\}_{\ell=1}^k = \{X_\ell\}_{\ell=(i-1)k+(i-1)d+1}^{ik+(i-1)d}$, 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 kth one coincide with symbols of $X_{i(k+d)+1}^\infty$. Now, we construct the trie $\mathcal{T}_m$ from the keys $Y(1)$, .. , $Y(m)$. ¿From the definition of $\mathcal{T}_m$ we conclude that (3.6) holds, hence one needs only to estimate H_m around $k = (1-\varepsilon)\frac{1}{h_2}\log n$ in $\mathcal{T}_m$ (e.g., using the second moment method). We note that $\mathcal{T}_m$ is built from weakly dependent sequences $Y(1)$, ..., $Y(m)$, hence one can expect that techniques used for independent tries (cf. [14], [38]) should work in this case too.

Let $A_{i,j} = \{C_{i,j} \geq k\}$ be the event that the alignment[3] $C_{i,j}$ between $Y(i)$ and $Y(j)$ keys is greater than $k = O(\log n)$. By the second moment method (cf. (3.7)) we have

$$\Pr\{H_m \geq k\} \geq \frac{\left(\sum_{i,j\in D} \Pr\{A_{ij}\}\right)^2}{\sum_{i,j\in D} \Pr\{A_{ij}\} + \sum_{(i,j)\neq(t,s)} \Pr\{A_{ij} \cap A_{ts}\}} \ , \tag{3.8}$$

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

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

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

$$\Pr\{A_{ij} \cap A_{ts}\} = \sum_{\mathcal{W}_k} \sum_{\mathcal{W}'_k} P(w_k g_1 w_k \cap w'_k g_2 w'_k) \ . \tag{3.9}$$

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

$$\begin{aligned}
\sum_{\mathcal{W}_k} \sum_{\mathcal{W}'_k} P(w_k g_1 w_k \cap w'_k g_2 w'_k) &\leq (1 + \alpha(d_n)) \sum_{\mathcal{W}_k} \sum_{\mathcal{W}'_k} P(w_k g_1 w_k) P(w'_k g_2 w'_k) \\
&\leq (1 + \alpha(d_n))^3 \sum_{\mathcal{W}_k} \sum_{\mathcal{W}'_k} P^2(w_k) P^2(w'_k) = (1 + \alpha(d_n))^3 E^2 P(w_k) \ .
\end{aligned}$$

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 $g_2 = g g'_2$. We consider two subcases:

[3] We now refer to $C_{i,j}$ as an alignment instead of the self-alignment since the trie $\mathcal{T}_m$ is built from weakly dependent ("almost independent") sequences $Y(1), \ldots, Y(m)$.

(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 ith one and the t-th one. Then,

$$\begin{aligned}\Pr\{A_{ij} \cap A_{it}\} &=^{(A)} \sum_{\mathcal{W}_k} P(w_k^3) \le c_1 \sum_{\mathcal{W}_k} P^3(w_k) \\ &\le^{(B)} c_1 \left(\sum_{\mathcal{W}_k} P^2(w_k)\right)^{3/2} = c_1 \left(EP(w_k)\right)^{3/2} ,\end{aligned}$$

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 [38],

$$\ell \ge r \quad \Rightarrow \quad \left(\sum_{\mathcal{W}_k} P^\ell(w_k)\right)^{1/\ell} \le \left(\sum_{\mathcal{W}_k} P^r(w_k)\right)^{1/r} . \tag{3.10}$$

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

$$\sum_{(i,j)\ne(t,s)} \Pr\{A_{ij} \cap A_{ts}\} \le m^4(1+\alpha(d_n))^3 E^2 P(w_k) + m^3 c \left(EP(w_k)\right)^{3/2}$$

where c is a constant.

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

$$\begin{aligned}\Pr\{H_m \ge (1-\varepsilon)\frac{1}{h_2}\log n\} &\ge \left(\frac{n^{2(1-\varepsilon)}(1+\alpha(d_n))}{m^2(1-\alpha(d_n))^2} + (1-o(1))(1+\alpha^2(d_n) + o(\alpha^2(d_n)) + c\frac{n^{1-\varepsilon}}{m}\right)^{-1} \\ &\ge 1 - c_1\frac{n^{2(1-\varepsilon)}}{m^2} - c_2\frac{n^{1-\varepsilon}}{m} - c_3\alpha^2(d_n)\end{aligned}$$

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

$$\Pr\{H_m \le (1-\varepsilon)\frac{1}{h_2}\log n\} \le c_1\frac{\log^2 n}{n^{2\varepsilon}} + c_2\frac{\log n}{n^\varepsilon} + c_3\alpha^2(\log n) , \tag{3.11}$$

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

$$\Pr\{\mid \frac{H_n}{\log n} - \frac{1}{h_2} \mid\ge \varepsilon\} \le c_1\frac{\log n}{n^\varepsilon} + c_2\alpha^2(\log n) \to 0 \tag{3.12}$$

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 [23] (cf. also [31]). 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

$$\sum_{r=0}^{\infty} \Pr\{\mid \frac{H_{s2^r}}{\log(s2^r)} - \frac{1}{h_2} \mid \geq \varepsilon\} < \infty \tag{3.13}$$

provided

$$\sum_{r=0}^{\infty} \alpha^2(r) < \infty \ , \tag{3.14}$$

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

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

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

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

This leads to

$$\lim_{n\to\infty} \frac{H_n}{\log n} = \frac{1}{h_2} \qquad (a.s.) \tag{3.15}$$

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 $\mathcal{S}_n$. Our proof resembles the derivation used by Pittel [31] for independent tries (cf. also [18]). We need some more notation. We write $\langle X_1^n, w_k\rangle$ to denote the set of positions on which X_1^n and w_k agree, that is, $i \in \langle X_1^n, w_k\rangle$ if the ith suffix X_i^∞ 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, $C(X, w_k) \le k$. Finally, we define

$$p_{\min}(k) = \min_{w_k \in \mathcal{W}_k} \{P(w_k)\} \ . \tag{3.16}$$

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 $\mathcal{S}_n$ 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_1^∞ whose prefix of length k agrees with w_k. Therefore, the set $\langle X_1^n, w_k\rangle$ is nonempty, i.e., $|\langle X_1^n, w_k\rangle| \ge 1$. This is particularly true for the word w_{min}. Hence

$$\Pr\{s_n > k\} \le \Pr\{|\ \langle X_1^n, w_{\min}\rangle\ | \ge 1\} \ . \tag{3.17}$$

But $|\ \langle X_1^n, w_{\min}\rangle\ | \ge 1$ implies that at least one depth is greater than k which further implies that there exists an index, say $1 \le i \le n$, such that $C(X_i^\infty, w_{\min}) = k$. Then, by Boole's inequality

$$\Pr\{s_n > k\} \le n\Pr\{C(X_i^\infty, w_{\min}) = k\} = np_{\min}(k) \ . \tag{3.18}$$

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

$$\Pr\{s_n > (1+\varepsilon)\frac{1}{h_1}\log n\} \le \frac{c}{n^\varepsilon} \ , \tag{3.19}$$

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_1^n into m keys $Y(1), \dots , Y(m)$ separated by m gaps of size d (cf. Fig. 4). We construct a trie $\mathcal{T}_m$ from those m *weakly dependent* keys (for detailed construction see previous subsection). Let s_m be the shortest feasible path in $\mathcal{T}_m$. Then, due to the fact that $s_m \le_{\rm st} s_n$, we have

$$\Pr\{s_n < k\} \le \Pr\{s_m < k\} \ . \tag{3.20}$$

Now, we need only to investigate the reduced tree $\mathcal{T}_m$. But the event $\{s_n < k\}$ implies that there *exists* a word $w_k \in \mathcal{W}_k$ such that longest common prefixes between w_k and the keys $Y(1), \dots, Y(m)$ are of lengths smaller than k. That is,

$$\{s_m < k\} \quad \Rightarrow \quad \exists_{w_k \in \mathcal{W}_k} \ \{C(Y(1), w_k) < k, \dots, C(Y(m), w_k) < k\} \ . \tag{3.21}$$

This is the same as in Pittel [31] since the condition (3.21) is naturally also true for independent tries.

By the strong α-mixing condition (2.5), and the above we have

$$\begin{aligned}
\Pr\{s_m < k\} &\le \sum_{\mathcal{W}_k} \Pr\{C(Y(1), w_k) < k \ , \ \dots , C(Y(m), w_k) < k\} \\
&\le \sum_{\mathcal{W}_k} (1+\alpha(d_n))^m (1 - P(w_k))^m \le (1+\alpha(d_n))^m \sum_{\mathcal{W}_k} (1 - p_{\min}(k))^m \\
&\le V^k (1+\alpha(d_n))^m (1 - p_{\min}(k))^m \ .
\end{aligned}$$

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

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

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

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

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

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

as needed for Theorem 2(ii).

ACKNOWLEDGMENT

It is my pleasure to acknowledge numerous enlightning 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

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

[2] A.V. Aho, Algorithms for Finding Patterns in Strings, in *Handbook of Theoretical Computer Science. Volume A: Algorithms and Complexity* (ed. J. van Leeuwen), 255-300, The MIT Press, Cambridge (1990).

[3] D. Aldous and P. Shields, A Diffusion Limit for a Class of Random-Growing Binary Trees, *Probab. Th. Rel. Fields*, 79, 509-542 (1988).

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

[5] A. Apostolico and W. Szpankowski, Self-alignments in Words and Their Applications, *J. of Algorithms*, 13 , 446-467 (1992).

[6] R. Arratia and M. Waterman, The Erdös-Rényi Strong Law for pattern Matching with Given proportion of Mismatches, *Annals of Probability*, 17, 1152-1169 (1989).

[7] P. Bender and J.K. Wolf, New Asymptotic Bounds and Improvements on the Lempel-Ziv Data Compression Algorithm, *IEEE Trans. Information Theory*, 37, 721-734 (1991).

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

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

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

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

[12] T.M. Cover and J.A. Thomas, *Elements of Information Theory*, John Wiley&Sons, New York (1991).

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

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

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

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

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

[18] P. Jacquet and W. Szpankowski, Autocorrelation on Words and Its Applications. Analysis of Suffix Tree by String-Ruler Approach, *J. Combinatorial Theory. Ser. A*, to appear.

[19] P. Jacquet and W. Szpankowski, On the Lempel-Ziv Parsing Algorithm and Its Digital Tree Representation, INRIA Rapport de Recherche, 1992.

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

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

[22] J. C. Kieffer, Sample Converses in Source Coding Theory, *IEEE Trans. Information Theory*, 37, 263-268 (1991).

[23] J.F.C. Kingman, *Subadditive Processes*, in Ecole d'Eté de Probabilités de Saint-Flour V-1975, Lecture Notes in Mathematics, 539, Springer-Verlag, Berlin 1976.

[24] P. Kirschenhofer, H. Prodinger and W. Szpankowski, On the Variance of the External Path in a Symmetric Digital Trie *Discrete Applied Mathematics*, 25, 129–143 (1989).

[25] P. Kirschenhofer, H. Prodinger and W. Szpankowski, Digital Search Trees Again Revisited: The Internal Path Length Perspective, Purdue University, CSD TR-989, 1990.

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

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

[28] E.M. McCreight, A Space Economical Suffix Tree Construction Algorithm, *JACM*, 23, 262272 (1976).

[29] G. Hardy, J.E. Littlewood and G. Pólya, *Inequalities*, Cambridge University Press, Cambridge (1952).

[30] D. Ornstein and B. Weiss, Entropy and Data Compression Schemes, *IEEE Information Theory*, to appear.

[31] B. Pittel, Asymptotic Growth of a Class of random Trees, *Annals of Probability*, 13, 414 - 427 (1985).

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

[33] M. Rodeh, V. Pratt and S. Even, Linear Algorithm for Data Compression via String Matching, *Journal of the ACM*, 28, 16-24 (1981).

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

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

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

[37] W. Szpankowski, Some Results on V-ary Asymmetric Tries, *Journal of Algorithms,* 9, 224-244 (1988).

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

[39] W. Szpankowski, A Generalized Suffix Tree and Its (Un)Expected Asymptotic Behaviors, *SIAM J. Computing*, to appear.

[40] P. Weiner, Linear Pattern Matching Algorithms, *Proc. of the 14-th Annual Symposium on Switching and Automata Theory,* 111 (1973).

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

[42] A. Wyner and J. Ziv, Fixed Data Base Version of the Lempel-Ziv Data Compression Algorithm, *IEEE Trans. Information Theory*, 37, 878-880 (1991).

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