Abstract

We propose a joint source-channel coding algorithm capable of correcting some errors in the popular Lempel-Ziv'77 scheme without practically losing any compression power. This can be achieved because the LZ'77 encoder does not completely eliminate the redundancy present in the input sequence. One source of redundancy can be observed when a LZ'77 phrase has multiple matches. In this case, LZ'77 can issue a pointer to any of those matches, and a particular choice carries some additional bits of information. We call a scheme with embedded redundant information the LZS'77 algorithm. We analyze the number of longest matches in such a scheme and prove that it follows the logarithmic series distribution with mean $1/h$ (plus some fluctuations), where $h$ is the source entropy. Thus, the number of redundant bits is well concentrated around its mean, a highly desirable property for error correction. These analytic results are proved by a combination of combinatorial, probabilistic and analytic methods (e.g., Mellin transform, depoissonization, combinatorics on words). In fact, we analyze LZS'77 by studying the multiplicity matching parameter in a suffix tree, which in turn is analyzed via comparison to its independent version, also known as a trie. Finally, we present an algorithm in which a channel coder (e.g., Reed-Solomon coder) succinctly uses the inherent additional redundancy left by the LZS'77 encoder to detect and correct a limited number of errors. We call such a scheme the LZRS'77 algorithm. LZRS'77 is perfectly backward-compatible with LZ'77, that is, a file compressed with our error-resilient LZRS'77 can still be decompressed by a generic LZ'77 decoder.

Index Terms: Lempel-Ziv'77 scheme, multiple matches, joint source-channel coding, Reed-Solomon code, suffix trees, tries, Mellin transform, depoissonization, pattern matching, autocorrelation polynomial, combinatorics on words.
1 Introduction

Error-resilient adaptive lossless data compression is a particularly challenging problem because of two opposing “forces”. Source coding tries to decorrelate as much as possible the input sequence (i.e., by removing redundant information), while channel coding introduces additional correlation (i.e., by adding redundant information) in order to protect against errors. The devastating effect of errors in adaptive data compression is a long-standing open problem [22]. In fact, in many applications, a practical drawback of adaptive data compression algorithms is their lack of resilience to errors. Joint source-channel coding has emerged as a viable solution to this problem.

The separation principle formulated by Shannon divides a communication system into separate source coding and channel coding subsystems that run independently; however, in today’s communication technology this rigid separation is very limiting. In particular, this principle ignores many imperfections of real communication systems, such as the fact that channel coding is incapable of correcting all errors. Uncorrectable errors are inevitable; designing encoders while ignoring this fact simply leads to extremely fragile source codes, in which one single error can potentially yield catastrophic failures. Joint source-channel coding strikes a balance between source bits vs. channel bits, which in turn requires some adjustments in both the source coding and channel coding strategies. Our approach is somewhat orthogonal to most work in this area. We use redundancy bits left by the source coder to protect against errors without degrading the compression rate. The price we pay is that we only correct a few errors, and we do not achieve a positive error bit rate (i.e., we are unable to correct a number of errors proportional to the size of a block).

In this paper we deal with one of the best-known adaptive data compression schemes, namely that of Ziv and Lempel published in their 1977 seminal paper [29]. The popular LZ’77 compression scheme works on-line. It compresses phrases by consecutively replacing the longest prefix of the non-compressed portion of a file with a pointer and the length of the prefix. The lack of error-resilience of LZ’77 is a well-recognized problem. A few years ago we read the following posting on the comp.compression newsgroup: “...I’m a casualty of corrupt tar’d gzipped files on Solaris 8. (gzip 1.3) ... Is there a reason why there are no compression utilities that allow controlled amounts of redundancy for error correction? ... How much overhead would be needed to correct these?”

Indeed, we asked ourselves, how much overhead is needed in LZ’77 to correct errors? The surprising answer is that there is no need for additional overhead in order to correct errors in LZ’77. This seemingly impossible goal is achieved in practice thanks to the fact that the LZ’77 encoder is unable to completely decorrelate the input sequence. Some implicit redundancy, which we quantify in this paper, is still present in the compressed stream and can be exploited by the encoder. The additional redundancy derives from the encoding of phrases for which one has a choice among \( M > 1 \) possible pointers. In practice, if there are \( M \) copies of the longest prefix, we recover \( \lfloor \log_2 M \rfloor \) redundant bits by choosing one of the \( M \) pointers (see Figure 1). We call such a scheme with multiple pointers the LZS’77 algorithm.

In the first part of the paper we present an algorithm for channel coding that exploits the redundant bits identified by LZS’77. To detect and correct errors, we choose Reed-Solomon codes computed on blocks of 255 bytes of compressed data. Given the maximum number of errors \( e \) that the Reed-Solomon code can correct, the \( 2e \) parity bits of the Reed-Solomon code will be embedded in the extra redundant bits extracted from the pointer multiplicity. We should point out that if \( e \) is large then we may not always have enough redundant bits to embed the parity bits. The algorithm that incorporates the Reed-Solomon channel coding into LZS’77 is referred to throughout as the LZRS’77 scheme.
Our basic algorithm allows one to correct only a few errors, thus we set \( e = O(1) \) and \( e \) is rather small in our implementations. In fact, we prove theoretically that the average number of longest phrases is \( O(1) \) leading to \( e = O(1) \). We should observe, however, that even single errors can have devastating effects. It has been proved recently [4] that a single error in LZ’77 corrupts \( O(n^{2/3}) \) phrases, thus about \( O(n^{2/3} \log n) \) symbols, where \( n \) is the size the file to be compressed. Furthermore, a simple modification of our algorithm (e.g., instead of looking for the longest match we just consider a “long enough” match) allows \( e \) to change adaptively with the availability of redundancy bits in the stream (i.e., \( e \) will slowly grow with \( n \)) and still preserve the asymptotic optimality of the compression bit rate (see remarks after Theorem 1).

In the second part of this paper we theoretically quantify the amount of redundancy left by the LZ’77 encoder for error protection. Thus we resort to analyzing the number of pointers in the LZS’77 scheme. We let \( M_n \) denote the number of pointers (longest matches) into the database when \( n \) bits have already been compressed. We are primarily interested in precisely determining the asymptotics of the random variable \( M_n \) and its concentration around the mean. A thorough analysis of the variable \( M_n \) yields a characterization of the degree to which error correction can be performed in the scheme discussed above. We recall that \( \lceil \log_2 M_n \rceil \) bits are available for detecting and correcting errors.

Suffix trees provide a natural way to study the variable \( M_n \). A suffix tree [24] is a digital search tree (i.e., a trie [24]) built from all the suffixes of a single string (the database in our case). In a suffix tree, \( M_n \) corresponds to the number of leaves in the subtree rooted at the branching point of the \((n + 1)\)st insertion. We refer to \( M_n \) as the multiplicity matching parameter. As it turns out, strings in suffix trees are highly dependent on each other. This dependency complicates the precise analysis of \( M_n \); therefore, we also consider the analogous situation, where a trie is built over independent strings. More specifically, we study the variable \( M^I_n \) associated with the number of leaves in the subtree rooted at the branching point of the \((n + 1)\)st insertion in a trie. After determining the asymptotics of \( M^I_n \), we prove that \( M_n \) and \( M^I_n \) have asymptotically identical distributions.

The main theoretical result consists of a precise characterization of all the moments of \( M_n \) and its limiting distribution. In particular, we show that for memoryless sources the average number of pointers is \( 1/h \), where \( h \) is the entropy rate. We also show that the limiting distribution of \( M_n \) follows the logarithmic series distribution, that is, \( \Pr(M_n = k) \approx [p^k(1 - p) + (1 - p)^k p]/(kh) \) where \( p \) is the probability of generating a “1”. Thus, the number of pointers is well concentrated around the mean, which is a highly desirable property for channel coding. Still, it is more likely to have one occurrence of the longest phrase in the database than many, but the probability of seeing two longest phrases is only four times smaller than finding a single longest phrase. In practice, we usually find more than one match, as shown in Section 2.3.

In order to prove our main result we use a battery of analytic tools, including recurrence relations, analytical poissonization and depoissonization, the Mellin transform, and complex analysis. To prove that suffix trees and independent tries have similar multiplicity matching parameters, we derive bivariate generating functions for \( M_n \) and \( M^I_n \) using combinatorics on words, as recently surveyed in [14]. We compare the generating functions for \( M_n \) and \( M^I_n \) by utilizing complex asymptotics.

To the best of our knowledge, the scheme described here is the first joint source-channel LZ’77 algorithm. In [22], Storer and Reif address the issue of error propagation but not error recovery (cf. see [18] for an analysis of the Storer and Reif algorithm). There are, however, joint source-channel
coding algorithms for arithmetic coding and other variable length codes (see, e.g., [20]).

Regarding our theoretical results, the multiplicity matching parameter was never previously studied in tries and suffix trees. However, the methodology used here to study the matching parameter in tries is well established within the analytic algorithmic community [24]. The analysis of \( M_n \) in a suffix tree is new and quite challenging. The basic idea of comparing suffix trees to independent tries was established by Jacquet and Szpankowski [9] and recently simplified by these authors in [14]. Other aspects of suffix trees have been studied in [5, 6, 23].

The paper is organized as follows. In Section 2.1 we describe the LZS’77 encoder and present our main theoretical results. In Section 2.2 we design the encoder and decoder for the LZRS’77 scheme and in Section 2.3 discuss the experiment results. The main theoretical result is proved in Sections 3–5. In Section 3 we provide a streamlined analysis and the road-map of the proof. Independent tries are discussed in Section 4 while suffix trees are analyzed in Section 5.

2 Main Results

In this section we present our main algorithmic, theoretical, and experimental results. We first describe a modified LZ’77 scheme, called LZS’77, in which we recover redundant information by identifying multiple longest matches. In Theorem 1 we quantify the redundant information by analyzing the variable \( M_n \) associated with the number of longest matches when the database sequence is of length \( n \). Finally, the recovered redundant bits are used in a new algorithm called LZRS’77, in which \( O(1) \) number of errors is corrected at each stage of the compression. We end the section by reporting experimental results on LZRS’77.

2.1 Redundant Information in LZS’77

Let \( X \) be a text of length \( n \) over a finite alphabet \( \mathcal{A} \). We write \( X_i, 1 \leq i \leq n \) to indicate the \( i \)th symbol in \( X \). We use \( X_{i,j} \) as shorthand for the substring \( X_iX_{i+1} \ldots X_j \) where \( 1 \leq i \leq j \leq n \), with the convention that \( X_{i,i} = X_i \). Substrings of the form \( X_{i,j} \) correspond to prefixes of \( X \), and subtrngs of the form \( X_{i,n} \) correspond to the suffixes of \( X \).

The LZ’77 algorithm [29] processes the data on-line as it is read, i.e., it parses the file sequentially left to right and looks into the sequence of past symbols (called the database) to find a match with the longest prefix of the string starting at the current position. The longest prefix is replaced with a pointer, which is a triple composed of \( (\text{position, length, symbol}) \). Several variations on LZ’77 have been proposed (see, e.g., [3] and references therein), but the basic principle remains the same.

Let us suppose that the first \( i-1 \) symbols of the string \( X \) have been already parsed into \( k-1 \) phrases, i.e., \( X_{1,i-1} = y_1y_2 \ldots y_{k-1} \), where \( y_i \) is a non-empty string over \( \mathcal{A} \). In order to identify the \( k \)th phrase, LZ’77 looks for the longest prefix of \( X_{i,n} \) that matches a substring of \( X_{1,i-1} \). If \( X_{j+l-1} \) \((j < i)\) is the substring that matches the longest prefix, then the next phrase is \( y_k = X_{i+l} \). The algorithm issues the pointer \((j,l,X_{i+l})\) and updates the current position \( i \) to \( i + l + 1 \). The symbol \( X_{i+l} \) is needed to be able to advance when \( l = 0 \), which is common in the very beginning of the encoding process.

In order to recover additional bits to be used for channel coding, we slightly modify the LZ’77 scheme. The resulting algorithm, called LZS’77, allows one to embed some bits of another binary string \( K \). We define a position \( i \) corresponding to the beginning of a phrase to have multiplicity \( M \) if there exist exactly \( M \) matches for the longest prefix that starts at position \( i \) in \( X \). The positions
Figure 1: The multiplicity of the next phrase is four ($M = 4$). Choosing one of the four possible pointers recovers two redundant bits.

with multiplicity $M > 1$ are the places where we can embed some of the bits of $K$. Specifically, the next $\lfloor \log_2 M \rfloor$ bits will drive the selection of one particular pointer out of the $M$ choices (see Figure 1). These additional bits can be used for various purposes such as authentication [2] or error correction as described next.

Suppose again that the initial portion of $X$, say $X_{i-1}$, has been already parsed. Let \{$(p_0, l, X_{i+l}), (p_1, l, X_{i+l}), \ldots, (p_{M-1}, l, X_{i+l})$\}, $M \geq 1$, be the set of feasible pointers for the longest prefix of $X_{i}^n$, where $l > 1$, and $1 \leq p_l \leq i$ for all $0 \leq l \leq M - 1$. If $M = 1$ we skip to the next phrase, and no extra bits are embedded. When $M > 1$, we use the next $d = \lfloor \log_2 M \rfloor$ bits of $K$ to choose one of the $M$ pointers. Suppose that the first $r - 1$ bits of $K$ have already been embedded in previous phrases. We emit the pointer $(p_{K^{r-1}+d}, l, X_{i+l})$, we move the current position to $i + l + 1$, and we increment $r$ by $d$. The complete algorithm is summarized in Figure 2.

We want to stress that these changes do not affect the internal structure of LZ'77 encoding, other than a possible re-shuffling of the pointers. A file compressed with LZS'77 can still be decompressed by a standard LZ'77 algorithm. The fact that LZS'77 is “backward-compatible” makes it possible to deploy it gradually over the existing LZ'77 algorithm, without disrupting service.

From the above description it is clear that the size of the embedded text $K$ depends on the number of longest matches $M_n$ when the first $n$ bits of the input have already been compressed. We analyze $M_n$ for a binary memoryless source, and consider the string $X = X_1X_2X_3 \ldots$, where the $X_i$’s are i.i.d. random variables on the binary alphabet with $\Pr(X_i = 0) = p$ and $\Pr(X_i = 1) = q$. Without loss of generality, we assume throughout the discussion that $q \leq p$. Let $X^{(i)}$ denote the $i$th suffix of $X$. In other words, $X^{(i)} = X_iX_{i+1}X_{i+2} \ldots$. Consider the longest prefix $w$ of $X^{(n+1)}$ such that $X^{(i)}$ also has $w$ as a prefix, for some $i$ with $1 \leq i \leq n$. Then $M_n$ can be defined as the number of $X^{(i)}$’s (with $1 \leq i \leq n$) that also have $w$ as a prefix. We formally define the multiplicity
LZS’77 Encoder \((X, K)\)

1. let \(i, r, n, m, P \leftarrow 0, 0, |X|, |K|, []\)
2. while \(i < n\) do
3. \(\text{let } X^i_l \leftarrow \text{the longest prefix of } X_i^l \text{ that matches a substring in } X_1^{i-1}\)
4. \(\text{let } R \leftarrow \{(p_0, l, X_{i+1}), \ldots, (p_{M-1}, l, X_{i+1})\} \text{ be the set of feasible pointers for } X_i^{i+l-1}\)
5. if \(M > 1\) then
6. \(\text{let } d \leftarrow \lfloor \log_2 M \rfloor\)
7. \(\text{append } (p_{K^i+d}, l, X_{i+l}) \text{ to } P\)
8. \(\text{let } r \leftarrow r + d\)
9. else
10. \(\text{append } (p_{M-1}, l, X_{i+l}) \text{ to } P\)
11. \(\text{let } i \leftarrow i + l + 1\)
12. return \(P\)

LZS’77 Decoder \((P)\)

1. let \(D, K \leftarrow \text{empty string, empty string}\)
2. for each \((p, l, c) \in P\) do
3. \(\text{let } R \leftarrow \{p_0, \ldots, p_{M-1}\} \text{ be the set of occurrences of } D_p^{p+l-1}\)
4. \(\text{let } i \text{ be the index such that } p_i = p\)
5. \(\text{append } [\log_2 M] \text{ bits of } i \text{ to } K\)
6. \(\text{append } D_p^{p+l-1} c \text{ to } D\)
7. return \((D, K)\)

Figure 2: Recovering redundant bits \(K\) in LZ’77. Here \(X\) is the text, \(K\) represents the redundant bits, \(P\) is the compressed stream of pointers, \(D\) is the decompressed text.
matching parameter as

\[ M_n = \#\{1 \leq i \leq n \mid X^{(i)} \text{ has } w \text{ as a prefix}\}. \tag{1} \]

Our goal is to understand the probabilistic behavior of the variable \( M_n \). In particular, we compute the \( j \)th factorial moment \( \mathbf{E}[M_n^j] = \mathbf{E}[M_n(M_n - 1) \cdots (M_n - j + 1)] \), and the limiting distribution \( \Pr(M_n = k) \) for large \( n \). We accomplish this by finding the probability generating function \( \mathbf{E}[u^{M_n}] \) and extracting its asymptotic behavior for large \( n \). The main result presented next is proved in Section 3 with details explained in Sections 4 and 5.

**Theorem 1.** Consider a binary memoryless source, and let \( h = -p \log p - q \log q \) be its entropy rate.

(i) There exists \( \delta > 0 \) depending on \( p \) such that the \( j \)th factorial moment of \( M_n \) is

\[
\mathbf{E}[M_n^j] = \Gamma(j) \left( \frac{q(p/q)^j + p(q/p)^j}{h} \right) + \gamma_j(\log_{1/p} n) + O(n^{-\delta}),
\tag{2}
\]

where \( \Gamma \) is the Euler gamma function, and \( \gamma_j(\cdot) \) is a periodic function with mean 0 and small modulus for \( \log p/\log q \) rational, and asymptotically zero for \( \log p/\log q \) irrational.

(ii) The probability generating function \( \mathbf{E}[u^{M_n}] = \sum_{k \geq 0} \Pr(M_n = k) u^k \) is for some \( \epsilon > 0 \)

\[
\mathbf{E}[u^{M_n}] = -\frac{q \ln(1 - pu) + p \ln(1 - qu)}{h} + \gamma(\log_{1/p} n, u) + O(n^{-\epsilon}),
\tag{3}
\]

where \( \gamma(\cdot, u) \) is a periodic function with mean 0 and small modulus for \( \log p/\log q \) rational and asymptotically zero otherwise. More precisely,

\[
\mathbf{E}[u^{M_n}] = \sum_{j=1}^{\infty} \left[ \frac{p^j q + q^j p}{jh} + \sum_{k \in \mathbb{Z} \setminus \{0\}} \left( -\frac{e^{2ki\pi \log_{1/p} n} \Gamma(z_k)(p^j q + q^j p)(z_k)^j}{j!(p^{z_k+1} \ln p + q^{-z_k+1} \ln q)} \right) u^j \right] + O(n^{-\epsilon})
\tag{4}
\]

where for \( \ln p/\ln q = r/t \) and some integers \( r, t \in \mathbb{Z} \) we have \( z_k = \frac{2kr\pi i}{\ln p} \). The above translates into

\[
\Pr(M_n = j) = \frac{p^j q + q^j p}{jh} + \sum_{k \neq 0} \left( -\frac{e^{2ki\pi \log_{1/p} n} \Gamma(z_k)(p^j q + q^j p)(z_k)^j}{j!(p^{z_k+1} \ln p + q^{-z_k+1} \ln q)} \right) + O(n^{-\epsilon})
\tag{5}
\]

for some \( \epsilon > 0 \).

A few remarks are in order. We first comment on the behavior of the function \( \gamma_j(t) \). For instance, if we set \( p = 1/2 \), then

\[
|\gamma_j(t)| \leq \frac{1}{\ln^2 2} \sum_{k \neq 0} \left| \Gamma \left( j - \frac{2ki\pi}{\ln 2} \right) \right|.
\]

The approximate values of \( \frac{1}{\ln^2 2} \sum_{k \neq 0} \left| \Gamma \left( j - \frac{2ki\pi}{\ln 2} \right) \right| \) are given below for the first ten values of \( j \).
\begin{align*}
\begin{array}{|c|c|}
\hline
j & \frac{1}{\ln 2} \sum_{k \neq 0} \left| \Gamma \left( j - \frac{2k\pi}{\ln 2} \right) \right| \\
\hline
1 & 1.4260 \times 10^{-5} \\
2 & 1.3005 \times 10^{-4} \\
3 & 1.2072 \times 10^{-3} \\
4 & 1.1527 \times 10^{-2} \\
5 & 1.1421 \times 10^{-1} \\
6 & 1.1823 \times 10^{0} \\
7 & 1.2853 \times 10^{1} \\
8 & 1.4721 \times 10^{2} \\
9 & 1.7798 \times 10^{3} \\
10 & 2.2737 \times 10^{4} \\
\hline
\end{array}
\end{align*}

We note that, if \( \ln p / \ln q \) is irrational, then \( \gamma_j(x) \to 0 \) as \( x \to \infty \). So \( \gamma_j \) does not exhibit fluctuation when \( \ln p / \ln q \) is irrational.

For large \( n \) we conclude that on average there are \( 1/h \) eligible pointers and that \( M_n \) follows the logarithmic series distribution, i.e.,

\[
\Pr(M_n = j) \approx \frac{p^j q + q^j p}{jh}
\]

plus some small fluctuations. Observe that the probability is maximal for \( j = 1 \), but \( \Pr(M_n = 2) \) is only four times smaller; for \( p \gg q \) we also have \( \Pr(M_n = j + 1) / \Pr(M_n = j) \approx p j / (j + 1) \), thus the distribution is rather “flat”. This bears some immediate consequences for the LZRS’77 scheme since the number of corrected errors depends on \( \log M_n \). Knowing that \( M_n \) is well concentrated around its mean is quite reassuring and contributes to a good behavior of the algorithm in practice. In fact, experimental results presented in the next section show that there are sufficiently many redundant bits to warrant the use of the LZRS’77 error correction scheme.

In passing we should point out that there is an easy way to extend our scheme to recover more than a constant number of redundant bits. One just has to give up the idea of always looking for the longest match and instead agree to use “long enough” matches. More precisely, we propose to investigate a scheme which is still asymptotically optimal with the (compression) bit rate \( 1/h + O(\log \log n / \log n) \) and with \( M_n \) growing slowly with \( n \). For example, instead of using the longest match we look for the \( r \)th longest match. We expect that if \( r \) grows with \( n \) in such a way that the \( r \)th longest match is of order \( (\log n - \log \log n) / h \), then \( M_n \) grows with \( n \) (possibly \( M_n = O(\log n) \)?) in this case, only the constant of the asymptotic redundancy \( O(\log \log n / \log n) \) is affected, but the leading term \( 1/h \) remains the same.

### 2.2 Error Resilient LZRS’77 Scheme

We now describe how to use the extra redundant bits to achieve error-resilience. Recall that we are protecting the stream of pointers, which is represented by a sequence of bytes. We chose Reed-Solomon (RS) codes \([16]\), which are block-based error correcting codes widely used in digital communications and storage.

Reed-Solomon codes belong to the family of BCH codes (see, e.g., \([15]\)). A Reed-Solomon code is specified as RS(\( a, b \)), where \( a \) is the size of the block and \( b \) is the size of the payload. Let the datum be a symbol drawn from an alphabet of cardinality \( 2^a \). The encoder collects \( b \) symbols and
Figure 3: The right-to-left sequence of operations on the compressed blocks as processed by the LZRS’77 encoder.

adds $a - b$ parity symbols to make a block of length $a$. A Reed-Solomon decoder can correct up to $e$ errors in a block, where $e = (a - b)/2$. One symbol error occurs if one or more of the bits of the symbol (up to $s$) is wrong.

Given a symbol size $s$, the maximum block length $a$ for a Reed-Solomon code is $a = 2^s - 1$. For example, the maximum length of a code with 8-bit symbols ($s = 8$) is 255 bytes. The family of Reed-Solomon codes for $s = 8$ is therefore RS(255, 255 − 2e). Each block contains 255 bytes, of which 255 − 2e are data and 2e are parity. Errors up to $e$ bytes anywhere in the block can be automatically detected and corrected.

We can use the extra redundancy bits of LZS’77 to embed 2e extra bytes, as described in the following. The encoder, called LZRS’77, first compresses $X$ using the standard LZ’77. The data is broken into blocks of size 255 − 2e. Then, blocks are processed in reverse order, beginning with the very last. When processing block $i$, the encoder computes first the Reed-Solomon parity bits for the block $i + 1$ and then it embeds the extra bits in the pointers of block $i$ using the method described in Section 2.1. The sequence of operations of the encoder is illustrated in Figure 3. If one wants to protect the first block as well, then the parity bits of the first block are not embedded, but saved at the beginning of the compressed file. Note that if we store these extra bits at the beginning of the file, the compressed file is not backward-compatible anymore with the standard LZ’77 decoder. To keep the file backward-compatible one must forgo protecting the first block of the compressed data.

The decoder receives a sequence of pointers, preceded by the parity bits of the first block. It first breaks the rest of the input stream into blocks of size 255 − 2e. Then it uses the parity bits to correct the first block. Once block $B_1$ is correct, it decompresses $B_1$ using LZS’77. This not only reconstructs the initial portion of the original text, but it also recovers the bits stored in those particular choices for the pointers. These extra bits are collected, and they become the parity bits for the second block. The decoder can therefore detect and correct errors in $B_2$. Block $B_2$ is then decompressed, and the parity bits for $B_3$ are recovered. This process continues until all blocks have been decompressed. A high-level description of the encoder and the decoder is shown in Figure 4.

The reason the encoder needs to process the blocks in reverse order should now be apparent. The encoder cannot compute the RS parity bits before the pointers are finalized. We embed the RS bits for the current block in the previous block, because the decoder needs to know the parity bits of a block before it attempts to decompress it. This has the unfortunate effect of making the encoder off-line, since it requires the encoder to keep the entire set of buffers in primary memory. The problem can be alleviated by breaking up large inputs in chunks of a size that could be easily stored and processed in main memory.
LZRS'77 Encoder $(X, e)$
1. let $b, j, n \leftarrow 1, 1, |X|$
2. while $j < n$ do
3. append $LZ'77$ Compress $(X_j)$ to $B_b$
4. if $|B_b| = 255 - 2e$ then
5. let $b \leftarrow b + 1$
6. for $i \leftarrow b, \ldots, 2$ do
7. let $RS_i \leftarrow REED\_SOLOMON\_ENCODER(B_i, e)$
8. embed in the block $B_{i-1}$ the bits $RS_i$ using LZS'77
9. let $RS_1 \leftarrow REED\_SOLOMON\_ENCODER(B_1, e)$
10. return $RS_1, B_1, B_2, \ldots, B_b$

LZRS'77 Decoder $(RS_1, B_1, B_2, \ldots, B_b, e)$
1. $D \leftarrow$ empty string
2. if REED\_SOLOMON\_DECODER($B_1 + RS_1, e$) = errors then correct $B_1$
3. append $LZ'77$ Decompress($B_i$) to $D$
4. recover $RS_2$ from the pointers used in $B_1$ using LZS'77
5. for $i \leftarrow 2, \ldots, b$ do
6. if REED\_SOLOMON\_DECODER($B_i + RS_i, e$) = errors then correct $B_i$
7. append $LZ$ Decompress($B_i$) to $D$
8. recover $RS_{i+1}$ from the pointers used in $B_i$ using LZS'77
9. return $D$

Figure 4: Error-resilient LZ'77 algorithm. Here $X$ is the text, $e$ is the maximum number of errors that can be corrected in each block of $255 - 2e$ bytes.

2.3 Experimental Results

In order to validate our theoretical studies presented in Theorem 1 and test the correctness of our scheme of LZS'77, we instrumented several implementations. In the first one, we designed an implementation of LZ'77 based on suffix trees [19], and we kept track of the multiplicity $M$ for each phrase of the LZ'77 parsing, when the length of the phrase is greater than two. The average value of $M$ is shown in Figure 5, for increasing lengths of the prefixes. Note that for both graphs, the average for $M$ appears to converge asymptotically to some constant, as Theorem 1 suggests.

In the second, we modified the code of gzip-1.2.4 to evaluate the impact of our method on compression performance. The tool gzip is an implementation of the sliding window variant of LZ'77, that issues pointers in a fixed-size window preceding the current position.

The modified gzip, called gzipS, implements directly LZS'77 as described in Section 2.1. It allows the user to specify a second file, which contains the text to be embedded in the pointers. The compression performance of the gzipS with respect to the original gzip was measured, and it is illustrated in Table 1 on the Calgary corpus dataset. The embedding of the message slightly degrades the compression performance, on the order of 1%–2% on average for the files in the Calgary corpus. A file compressed with gzipS can still be decompressed by the original gzip, and therefore is backward-compatible.

Finally, in the last implementation we coded the error-resilient LZRS'77. The prototype implementation is written in Python, with calls to C public-domain code that implements the Reed-
Figure 5: The average value of the pointer multiplicity $M$ for increasing portions of paper2 (left), and news (right) of the Calgary corpus.

<table>
<thead>
<tr>
<th>file size</th>
<th>gzip</th>
<th>gzipS</th>
<th>file</th>
<th>redundant bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>111,261</td>
<td>39,473</td>
<td>39,511</td>
<td>bib</td>
<td>1,721</td>
</tr>
<tr>
<td>768,771</td>
<td>333,776</td>
<td>336,256</td>
<td>book1</td>
<td>14,524</td>
</tr>
<tr>
<td>610,856</td>
<td>228,321</td>
<td>228,242</td>
<td>book2</td>
<td>10,361</td>
</tr>
<tr>
<td>102,400</td>
<td>69,478</td>
<td>71,168</td>
<td>geo</td>
<td>4,101</td>
</tr>
<tr>
<td>377,109</td>
<td>155,290</td>
<td>156,150</td>
<td>news</td>
<td>5,956</td>
</tr>
<tr>
<td>21,504</td>
<td>10,584</td>
<td>10,783</td>
<td>obj1</td>
<td>353</td>
</tr>
<tr>
<td>246,814</td>
<td>89,467</td>
<td>89,757</td>
<td>obj2</td>
<td>3,628</td>
</tr>
<tr>
<td>53,161</td>
<td>20,110</td>
<td>20,204</td>
<td>paper1</td>
<td>937</td>
</tr>
<tr>
<td>82,199</td>
<td>32,529</td>
<td>32,507</td>
<td>paper2</td>
<td>1,551</td>
</tr>
<tr>
<td>46,526</td>
<td>19,450</td>
<td>19,567</td>
<td>paper3</td>
<td>893</td>
</tr>
<tr>
<td>13,286</td>
<td>5,853</td>
<td>5,898</td>
<td>paper4</td>
<td>249</td>
</tr>
<tr>
<td>11,954</td>
<td>5,252</td>
<td>5,294</td>
<td>paper5</td>
<td>210</td>
</tr>
<tr>
<td>38,105</td>
<td>14,433</td>
<td>14,506</td>
<td>paper6</td>
<td>738</td>
</tr>
<tr>
<td>513,216</td>
<td>62,357</td>
<td>61,259</td>
<td>pic</td>
<td>3,025</td>
</tr>
<tr>
<td>39,611</td>
<td>14,510</td>
<td>14,660</td>
<td>progc</td>
<td>736</td>
</tr>
<tr>
<td>71,446</td>
<td>18,310</td>
<td>18,407</td>
<td>progl</td>
<td>1,106</td>
</tr>
<tr>
<td>49,378</td>
<td>12,532</td>
<td>12,572</td>
<td>progp</td>
<td>741</td>
</tr>
<tr>
<td>93,695</td>
<td>22,178</td>
<td>22,098</td>
<td>trans</td>
<td>1,201</td>
</tr>
</tbody>
</table>

Table 1: The compression of “gzip -3” versus “gzipS -3” for the files of the Calgary corpus; the last column shows the total number of available bits for error correction.
Figure 6: The probability that a file of \( b \) blocks could not be recovered correctly, for increasing number of errors uniformly distributed over the blocks. Top-left: \( e = 1 \) and \( b = 10 \), top-right: \( e = 1 \) and \( b = 100 \), lower-left: \( e = 2 \) and \( b = 10 \), lower-right: \( e = 2 \) and \( b = 100 \).

Solomon encoder/decoder [11]. Based on the considerations mentioned in introduction, we initially choose \( e = 1 \) and \( e = 2 \) which require respectively at least 2 and 4 parity bytes on a block of data of size \( 255 - 2e \). We experimented with the resilience to errors by introducing a controlled number of errors uniformly distributed over the \( b \) blocks of the compressed file. The graphs in Figure 6 show the probability that the file did not unpack correctly for increasing numbers of errors for different choices of \( e \) and \( b \). For example, using \( e = 2 \) over 100 blocks, LZRS’77 is able to unpack the file correctly with 20 uniformly distributed errors, 90% of the time.

3 Streamlined Analysis

In this section we guide the reader through the main ideas of the proof of Theorem 1 with details explained in the last two sections.

We recall the definition of the multiplicity matching parameter. The variable \( M_n \) represents the number of longest matches within the first \( n \) symbols of the database as formally expressed in (1). We now provide an alternative definition of \( M_n \) via suffix trees. A suffix tree is a trie built from suffixes of a single string. A trie is a digital tree built over, say \( n \), strings (the reader is referred to [12, 21, 24] for an in-depth discussion of digital trees). A string is stored in an external node of a trie; the path length to such a node is the shortest prefix of the string that is not a prefix of any
other strings (cf. Figure 7). For a binary alphabet, each branching node in a trie is a binary node. A special case of a trie structure is a suffix trie (tree) which is a trie built over suffixes of a single string.

Now we can re-define $M_n$ via suffix trees. First, build a suffix tree from the first $n+1$ suffixes of $X$. Consider the insertion point of the $(n+1)$st suffix. A well-known property of suffix trees is that $M_n$ is exactly equal to the number of leaves in the subtree rooted at the branching point of the $(n+1)$st insertion. For instance, suppose that the $(n+1)$st suffix starts with $w\beta$ for some $\beta \in \mathcal{A} := \{0,1\}$, and some $w \in \mathcal{A}^*$. Then, examining the first $n$ suffixes, if there are exactly $k$ suffixes that begin with $w\alpha$ (where $\alpha = 1 \oplus \beta$ where $\oplus$ is addition modulo 2), and the other $n-k$ suffixes do not begin with $w$, we conclude that $M_n = k$. Figure 7 illustrates this scenario.

Our goal is to study $M_n$ in a suffix tree built from a string $X$ generated by a binary memoryless source. Unfortunately, the strings in a suffix tree are highly dependent on each other; thus, a precise analysis of $M_n$ is quite difficult. For this reason, we first analyze the analogous situation in a trie built over independent strings. Specifically, in Section 4 we analyze the distribution and moments of a random variable with similar properties, namely $M'_n$, via the analysis of independent tries, using recurrence relations, analytical poissonization and depoissonization, the Mellin transform, and complex analysis (cf. [24]). To define $M'_n$, we consider the situation described above, but we build a trie from $n+1$ independent strings from $\mathcal{A}^*$. So we consider independent $X(i)$'s (more specifically, $X(i) = X_1(i)X_2(i)X_3(i)\ldots$, where the $X_j(i)$'s are i.i.d. random variables). We let $w$ denote the longest prefix of $X(n+1)$ such that $X(i)$ also has $w$ as a prefix, for some $i$ with $1 \leq i \leq n$. Then $M'_n$ is defined as the number of $X(i)$'s (with $1 \leq i \leq n$) that also have $w$ as a prefix, that is,

$$M'_n = \#\{1 \leq i \leq n \mid X(i) \text{ has } w \text{ as a prefix}\}. \quad (6)$$

In order to analyze $M'_n$, we define the alignment $C_{j_1,\ldots,j_k}$ among $k$ strings $X(j_1),\ldots,X(j_k)$ as the length of the longest common prefix of the $k$ strings. The $k$th depth $D_{n+1}(k)$ in a trie built over $n+1$ strings is the length of the path from the root of the trie to the leaf containing the $k$th string. Note $D_{n+1}(n+1) = \max_{1 \leq j \leq n} C_{j,n+1} + 1$. Thus, in the context of tries,

$$M'_n = \#\{j \mid 1 \leq j \leq n, C_{j,n+1} + 1 = D_{n+1}(n+1)\}.$$
Then for some $c > 0$, we analyze $M_n^j$ through generating functions. Define the exponential generating functions

$$G(z, u) = \sum_{n \geq 0} E[u M_n^j] \frac{z^n}{n!}, \quad W_j(z) = \sum_{n \geq 0} E[(M_n^j)^j] \frac{z^n}{n!}$$

for complex $u \in \mathbb{C}$ and $j \in \mathbb{N}$. If $f : \mathbb{C} \rightarrow \mathbb{C}$, then the recurrence relation

$$E[f(M_n^j)] = p^n(qf(n) + pE[f(M_n^j)]) + q^n(pf(n) + qE[f(M_n^j)])$$

$$+ \sum_{k=1}^{n-1} \binom{n}{k} p^k q^{n-k} (pE[f(M_k^j)]) + qE[f(M_{n-k}^j)])$$

(7)

holds for all natural $n \in \mathbb{N}$. If $f(0) = 0$, then the recurrence also holds when $n = 0$. To verify (7), just consider the possible values of $X_1(j)$ for $1 \leq j \leq n + 1$. Two useful facts follow immediately from this recurrence relation. First, if $n \in \mathbb{N}$, then

$$E[u M_n^j] = p^n(qu^n + pE[u M_n^j]) + q^n(pu^n + qE[u M_n^j]) + \sum_{k=1}^{n-1} \binom{n}{k} p^k q^{n-k} (pE[u M_k^j] + qE[u M_{n-k}^j]).$$

(8)

Also, if $j \in \mathbb{N}$ and $n \geq 0$ then

$$E[(M_n^j)^j] = p^n(qn^j + pE[(M_n^j)^j]) + q^n(pn^j + qE[(M_n^j)^j])$$

$$+ \sum_{k=1}^{n-1} \binom{n}{k} p^k q^{n-j} (pE[(M_k^j)^j] + qE[(M_{n-k}^j)^j]).$$

(9)

We derive in Section 4 an asymptotic solution for these recurrence relations using poissonization, the Mellin transform, and depoissonization. These methods yield the following two theorems.

**Theorem 2.** Let $z_k = \frac{2kr\pi i}{\ln p}$, where $\ln p / \ln q = r/s$ for some relatively prime $r, s \in \mathbb{Z}$. Then for some $c > 0$

$$E[(M_n^j)^j] = \Gamma(j) \frac{q(p/q)^j + p(q/p)^j}{h} + \gamma_j(\log_{1/p} n) + O(n^{-c}),$$

where

$$\gamma_j(t) = -\sum_{k \neq 0} e^{2k\pi it} \frac{\Gamma(z_k + j) (p^j q^{-z_k-j+1} + q^j p^{-z_k-j+1})}{p^{-z_k+1} \ln p + q^{-z_k+1} \ln q},$$

$\Gamma$ is the Euler gamma function, and $\gamma_j$ is a periodic function that has small magnitude and exhibits fluctuation. For $\ln p / \ln q$ irrational, we have $\gamma_j(n) \rightarrow 0$ as $n \rightarrow \infty$.

The next result describes the asymptotic distribution of $M_n^j$.

**Theorem 3.** Let $z_k = \frac{2kr\pi i}{\ln p}$, where $\ln p / \ln q = r/s$ for some relatively prime $r, s \in \mathbb{Z}$. Then for some $c > 0$

$$E[u M_n^j] = -\frac{q \ln (1 - pu) + p \ln (1 - qu)}{h} + \gamma(\log_{1/p} n, u) + O(n^{-c}),$$

(10)
\[
\gamma(t, u) = \sum_{k \neq 0} \frac{e^{2kr\pi i t} \Gamma(z_k) (q(1-pu)^{-z_k} + p(1-qu)^{-z_k} - p^{-z_k+1} - q^{-z_k+1})}{p^{-z_k+1} \ln p + q^{-z_k+1} \ln q},
\]

and \( \Gamma \) is the Euler gamma function. It follows immediately that

\[
E[u^m I_n] = \sum_{j=1}^{\infty} \frac{p^j q + q^j p}{j h} u^j + \sum_{j=1}^{\infty} \sum_{k \neq 0} \frac{e^{2kr\pi i \log_1/p} \Gamma(z_k) (p^j q + q^j p)(z_k)^j}{j!(p^{-z_k+1} \ln p + q^{-z_k+1} \ln q)} u^j + O(n^{-c}) \tag{11}
\]

and

\[
Pr(M_n^I = j) = \frac{p^j q + q^j p}{j h} + \sum_{k \neq 0} \frac{e^{2kr\pi i \log_1/p} \Gamma(z_k) (p^j q + q^j p)(z_k)^j}{j!(p^{-z_k+1} \ln p + q^{-z_k+1} \ln q)} + O(n^{-c}) \tag{12}
\]

for some \( c > 0 \). If \( \ln p/\ln q \) is irrational and \( u \) is fixed, then \( \gamma(x, u) \to 0 \) as \( x \to \infty \).

Once we have established the probabilistic properties of \( M_n^I \), we can deal with the more difficult problem, namely the multiplicity matching parameter \( M_n \) in a suffix tree. It remains to be shown that \( M_n \) has a similar asymptotic distribution as \( M_n^I \). To prove this, we compare the distribution of \( M_n \) in suffix trees versus the distribution of \( M_n^I \) in independent tries.

A detailed analysis of \( M_n \) is presented in Section 5. Briefly, our proof technique follows these lines. We let \( M(z, u) = \sum_{1 \leq k, n} Pr(M_n = k) u^k z^n \) and \( M^I(z, u) = \sum_{1 \leq k, n} Pr(M_n^I = k) u^k z^n \) denote the bivariate generating functions for \( M_n \) and \( M_n^I \), respectively. To study these generating functions, we consider the \( w \)'s defined above. Specifically, for \( M(z, u) \), we recall from (1) that if \( w \) denotes the longest prefix of \( X^{(n+1)} = X_{n+1}X_{n+2}X_{n+3} \ldots \) that appears as a prefix of any \( X(i) = X_iX_{i+1}X_{i+2} \ldots \), then \( M_n \) enumerates the number of such occurrences of \( w \). This approach to \( M(z, u) \) allows us to sum over all \( w \in \mathcal{A}^* \) instead of summing over \( k, n \in \mathbb{N} \). Similarly, for \( M^I(z, u) \), we utilize (6) to determine that if \( w \) denotes the longest prefix of \( X(n+1) = X_1(n+1)X_2(n+1)X_3(n+1) \ldots \) that appears as a prefix of any \( X_1(i)X_2(i)X_3(i) \ldots \), then \( M_n^I \) is precisely the number of such occurrences of \( w \). Therefore, to evaluate \( M^I(z, u) \), we can sum over all \( w \in \mathcal{A}^* \) instead of summing over the integers \( k \) and \( n \).

We note that the \( X(i) \)'s are highly dependent on each other. In fact, if \( i \geq j \), then \( X(i) = X_iX_{i+1}X_{i+2} \ldots \) is a substring of \( X(j) = X_jX_{j+1}X_{j+2} \ldots \). This dependency makes the derivation of the bivariate generating function \( M(z, u) \) quite difficult. We overcome this hurdle by succinctly describing the degree to which a suffix of \( X \) can overlap with itself. We accomplish this by utilizing the autocorrelation polynomial \( S_w(z) \) of a word \( w \), which measures the amount of overlap of a word \( w \) with itself. The autocorrelation polynomial is defined as (cf. [8, 14, 17])

\[
S_w(z) = \sum_{k \in \mathcal{P}(w)} Pr(w_k^m) z^{m-k} \tag{13}
\]

where \( \mathcal{P}(w) \) denotes the set of positions \( k \) of \( w \) satisfying \( w_1 \ldots w_k = w_{m-k+1} \ldots w_m \), that is, \( w \)'s prefix of length \( k \) is equal to \( w \)'s suffix of length \( k \). Via the autocorrelation polynomial, we are able to surmount the difficulties inherent in the overlapping suffixes. Thus, using \( S_w(z) \), we obtain a succinct description of the bivariate generating function \( M(z, u) \). The autocorrelation polynomial is well-understood; we utilize several results about \( S_w(z) \) from [14] and [17]. In particular, when comparing \( M(z, u) \) and \( M^I(z, u) \), it is extremely useful to note that the autocorrelation polynomial
$S_w(z)$ is close to 1 with high probability (for $|w|$ large), that is, for a random string $w$ there is not much overlap.

In order to obtain information about the difference of the above two random variables, we analyze $Q(z,u) = M(z,u) - M^I(z,u)$ using residue analysis. We make a comparison of the poles of $M(z,u)$ and $M^I(z,u)$ using Cauchy’s theorem (integrating with respect to $z$). As a result, we prove that $Q_n(u) := [z^n]Q(z,u) = O(n^{-\epsilon})$ uniformly for $|u| \leq p^{-1/2}$ as $n \to \infty$. Then we use another application of Cauchy’s theorem (integrating with respect to $u$). Specifically, we extract the coefficient $Pr(M_n = k) - Pr(M^I_n = k) = [u^k]Q(z,u)$ in order to obtain our main result that we prove in Section 5.

**Theorem 4.** There exists $\epsilon > 0$ such that for $|u| < 1 + \delta$ and some $\delta > 0$

$$|M_n(u) - M^I_n(u)| = O(n^{-\epsilon}).$$

(14)

As a consequence, there exists $b > 1$ such that

$$Pr(M_n = k) - Pr(M^I_n = k) = O(n^{-\epsilon}b^{-k})$$

(15)

for large $n$.

### 4 Analysis of Independent Tries

Now we present our analytical approach for proving Theorems 2 and 3 of Section 3. Our first strategy is to **poissonize** the problem. Then we utilize the Mellin transform and complex analysis; thus we obtain asymptotic descriptions of the distribution and factorial moments of $M^I_n$. Since these results are valid for the poissonized model of the problem, we must depoissonize our results in order to find the asymptotic distribution and factorial moments of $M^I_n$ in the original model.

#### 4.1 Poissonization

We first utilize analytical poissonization. The idea is to replace the fixed-size population model (i.e., the model in which the number of strings $n$ is fixed) by a poissonized model in which the number of strings is a Poisson random variable with mean $n$. This is affectionately referred to as “poissonizing” the problem. So we let the number of strings be $N$, a random variable that has Poisson distribution and mean $n$ (i.e., $Pr(N = j) = e^{-n}n^j/j! \forall j \geq 0$). We apply the Poisson transform to the exponential generating functions $G(z,u)$ and $W_j(z)$, which yields:

$$\tilde{G}(z,u) = \sum_{n \geq 0} E[u^{M_n}] \frac{z^n}{n!} e^{-z}, \quad \tilde{W}_j(z) = \sum_{n \geq 0} E[(M^I_n)^j] \frac{z^n}{n!} e^{-z}.$$ 

By using (8) to expand the coefficients of $z^n$ in $G(z,u)$ for $n \geq 1$, we observe

$$G(z,u) = qe^{puz} + pe^{quz} - pe^{qz} + qG(pz,u)e^{qz} + qG(qz,u)e^{pz}.$$ 

(16)

Similarly, we apply (9) to the coefficients of $z^n$ in $W_j(z)$ for $n \geq 1$ to see that

$$W_j(z) = q(pz)^je^{pz} + p(qz)^je^{qz} + pW_j(pz)e^{qz} + qW_j(qz)e^{pz}.$$ 

(17)
for all \( j \in \mathbb{N} \).

We observe that \( \widetilde{G}(z, u) = G(z, u)e^{-z} \). If we multiply by \( e^{-z} \) throughout (16) and then simplify, we obtain

\[
\widetilde{G}(z, u) = q e^{(pu-1)z} + pe^{(qu-1)z} - pe^{-pz} - q e^{-qz} + p \widetilde{G}(pz, u) + q \widetilde{G}(qz, u).
\] (18)

Similarly, from (17) we know that if \( j \in \mathbb{N} \) then

\[
\widetilde{W}_j(z) = q(pz)^j e^{-qz} + p(qz)^j e^{-pz} + p \widetilde{W}_j(pz) + q \widetilde{W}_j(qz).
\] (19)

Note that the functional equations (18) and (19) for the poissonized versions of \( G(z, u) \) and \( W_j(z) \) are simpler than the corresponding equations (16) and (17) from the original (Bernoulli) model. We solve (18) and (19) asymptotically for large \( z \in \mathbb{R} \).

### 4.2 Mellin Transform

If \( f \) is a complex-valued function which is continuous on \((0, \infty)\) and is locally integrable, then the Mellin transform of \( f \) is defined as

\[
\mathcal{M}[f(x); s] = f^*(s) = \int_0^\infty f(x)x^{s-1} \, dx
\]

(see [7] and page 400 of [24]). Three basic properties of the Mellin transform are useful in proving the next two results. We observe that

\[
\mathcal{M}[x^j f(x); s] = \int_0^\infty x^j f(x)x^{s-1} \, dx = \int_0^\infty f(x)x^{s+j-1} \, dx = f^*(s+j) = \mathcal{M}[f(x); s+j].
\]

If \( \mu > 0 \) we also notice

\[
\mathcal{M}[f(\mu x); s] = \int_0^\infty f(\mu x)x^{s-1} \, dx = \mu^{-s} \int_0^\infty f(x)x^{s-1} \, dx = \mu^{-s} f^*(s) = \mu^{-s} \mathcal{M}[f(x); s].
\]

Also

\[
\mathcal{M}[e^{-x}; s] = \int_0^\infty e^{-x}x^{-s} \, dx = \Gamma(s).
\]

We first find the fundamental strip of the Mellin of \( \widetilde{G}(x, u) \), that is, those complex \( s \) where the Mellin is defined. By (18), we observe that

\[
\widetilde{G}(x, u) = q e^{(pu-1)x} + pe^{(qu-1)x} - pe^{-px} - q e^{-qx} + p \widetilde{G}(px, u) + q \widetilde{G}(qx, u)
\]

\[
= q \sum_{k \geq 0} \frac{(pu-1)x)^k}{k!} + p \sum_{k \geq 0} \frac{(qu-1)x)^k}{k!} - p \sum_{k \geq 0} \frac{(-px)^k}{k!} - q \sum_{k \geq 0} \frac{(-qx)^k}{k!}
\]

\[
+ p \sum_{n \geq 0} \mathbb{E}[u^{M_n^1}] \frac{(px)^n}{n!} \sum_{k \geq 0} \frac{(-px)^k}{k!} + q \sum_{n \geq 0} \mathbb{E}[u^{M_n^1}] \frac{(qx)^n}{n!} \sum_{k \geq 0} \frac{(-qx)^k}{k!}
\]

\[
\rightarrow q + q(pu-1)x + p + p(qu-1)x - p + p^2 x - q + q^2 x
\]

\[
+ p + p^2 w - p^2 x + q + q^2 w - q^2 x \quad \text{as } x \to 0
\]

\[
= (u - 1)x + 1
\]

17
We notice that $\tilde{G}(x, u) \to 1$ as $x \to 0$, but we want instead to have $\tilde{G}(x, u) = O(x)$ as $x \to 0$. So we replace $\tilde{G}(x, u)$ by writing $\tilde{G}(x, u) = \tilde{G}(x, u) - 1$. We expect $\tilde{G}(x, u) = O(1) = O(x^0)$ as $x \to \infty$.

Therefore, the Mellin of $\tilde{G}(x, u)$ exists for $\Re(s) \in (-1, 0)$.

We next determine the fundamental strip of $W_j(x)$. By (19), we know

$$
\tilde{W}_j(x) = q(px)^j e^{-qx} + p(px)^j e^{-px} + p\tilde{W}_j(px) + q\tilde{W}_j(qx)
$$

$$
= q(px)^j \sum_{k \geq 0} \frac{(-qx)^k}{k!} + p(px)^j \sum_{k \geq 0} \frac{(-px)^k}{k!}
$$

$$
+ p \sum_{n \geq 0} E[(M_n^I)x] \frac{(px)^n}{n!} \sum_{k \geq 0} \frac{(-px)^k}{k!} + q \sum_{n \geq 0} E[M_n^I] \frac{(qx)^n}{n!} \sum_{k \geq 0} \frac{(-qx)^k}{k!}
$$

$$
\to q(px)^j + p(px)^j + pE[M_n^I] \frac{(px)^j}{j!} + qE[M_n^I] \frac{(qx)^j}{j!}, \quad \text{as } x \to 0
$$

$$
= O(x^j)
$$

We expect $\tilde{W}_j(x) = O(1) = O(x^0)$ as $x \to \infty$. So the Mellin exists for $\Re(s) \in (-j, 0)$.

If $u \in \mathbb{R}$ with $u < \min \{1/p, 1/q\}$ and if $\Re(s) \in (-1, 0)$ then it follows from (18) and the properties of the Mellin transform given above that

$$
\tilde{G}^s(s, u) = \frac{\Gamma(s) \left( q(1-pu)^{-s} + p(1-qu)^{-s} - p^{s+1} - q^{s+1} \right)}{1 - p^{s+1} - q^{s+1}}.
$$

If $j \in \mathbb{N}$ and $\Re(s) \in (-j, 0)$, then by (19) and the properties of the Mellin transform we mentioned, we see that

$$
\tilde{W}_j^s(s) = \frac{\Gamma(s+j) \left( p^j q^{-s-j+1} + q^j p^{-s-j+1} \right)}{1 - p^{s+1} - q^{s+1}}.
$$

We note that the Mellin transform is a special case of the Fourier transform. So there is an inverse Mellin transform. Since $\tilde{W}_j$ is continuous on $(0, \infty)$, then

$$
\tilde{W}_j(x) = \frac{1}{2\pi i} \int_{c-i\infty}^{c+i\infty} \tilde{W}_j(s)x^{-s} \, ds
$$

if $c \in (-a, -b)$, where $(-a, -b)$ is the fundamental strip of $\tilde{W}_j$. Thus

$$
\tilde{W}_j(x) = \frac{1}{2\pi i} \int_{-1/2-i\infty}^{-1/2+i\infty} \tilde{W}_j(s)x^{-s} \, ds
$$

since $c = -1/2$ is in the fundamental strip of $\tilde{W}_j(x)$ $\forall j \in \mathbb{N}$.

Similarly

$$
\tilde{G}(x, u) = \frac{1}{2\pi i} \int_{-1/2-i\infty}^{-1/2+i\infty} \tilde{G}(s, u)x^{-s} \, ds
$$

since $c = -1/2$ is in the fundamental strip of $\tilde{G}(x, u)$. 

18
4.3 Results for the Poisson Model

We restrict our attention to the case where \( \ln p / \ln q = r/t \) for some relatively prime \( r, t \in \mathbb{Z} \). Then, by a theorem of Jacquet and Schachinger (see page 356 of [24]), we know that the set of poles of \( \tilde{W}_j^* (s) x^{-s} \) is exactly \( \{ z_k = \frac{2k \pi i}{\ln p} \mid k \in \mathbb{Z} \} \). We also observe that \( \tilde{W}_j^* (s) x^{-s} \) has simple poles at each \( z_k \). Now we assume that \( u \neq 1 \). Then \( \tilde{G}^* (s, u)x^{-s} \) has the same set of poles as \( \tilde{W}_j^* (s) x^{-s} \), each of which is a simple pole.

Let \( T_1 \) denote the line segment from \( -\frac{1}{2} - iA \) to \( -\frac{1}{2} + iA \) in the complex plane, where \( A \) is a large real number. Let \( T_2 \) denote the line segment from \( -\frac{1}{2} + iA \) to \( L + iA \). Let \( T_3 \) denote the line segment from \( L + iA \) to \( L - iA \). Let \( T_4 \) denote the line segment from \( L - iA \) to \( -\frac{1}{2} - iA \). Now we claim that, if \( j \in \mathbb{N} \) and \( z_k = \frac{2k \pi i}{\ln p} \), then

\[
\tilde{W}_j (x) = \sum_{k \in \mathbb{Z}} -\text{Res}(\tilde{W}_j^* (s)x^{-s}; z_k) + O(x^{-L}).
\]

Using the Cauchy residue theorem [1], integrating clockwise around the curve described by \( T_1, T_2, T_3, T_4 \), we have

\[
\tilde{W}_j (x) = \frac{1}{2\pi i} \int_{-\frac{1}{2} + i\infty}^{\frac{1}{2} + i\infty} \tilde{W}_j^* (s) x^{-s} \, ds
\]

\[
= \lim_{A \to \infty} \frac{1}{2\pi i} \int_{T_1} \tilde{W}_j^* (s) x^{-s} \, ds
\]

\[
= \lim_{A \to \infty} \left( \sum_{a_i} -\text{Res}(\tilde{W}_j^* (s)x^{-s}; z = a_i) - \frac{1}{2\pi i} \left( \int_{T_2} + \int_{T_3} + \int_{T_4} \right) \tilde{W}_j^* (s) x^{-s} \, ds \right)
\]

where the sum is taken over all poles \( a_i \) of \( \tilde{W}_j^* (s)x^{-s} \) in the region bounded by \( T_1, T_2, T_3, T_4 \).

By the smallness property of the Mellin transform (see page 402 of [24]), we observe that

\[
\frac{1}{2\pi i} \left( \int_{T_2} + \int_{T_4} \right) \tilde{W}_j^* (s) x^{-s} \, ds = O(A^{-1}).
\]

We also observe (see page 408 of [24]) that

\[
\left| \frac{1}{2\pi i} \int_{T_3} \tilde{W}_j^* (s) x^{-s} \, ds \right| = \left| \frac{1}{2\pi i} \int_{L+i\infty}^{L-i\infty} \tilde{W}_j^* (s) x^{-s} \, ds \right|
\]

\[
= \left| \frac{1}{2\pi i} \int_{-\infty}^{\infty} \tilde{W}_j^* (L + it) x^{-L - it} \, ds \right|
\]

\[
\leq \frac{1}{2} |x^{-L}/2\pi| \int_{-\infty}^{\infty} \left| \tilde{W}_j^* (L + it) \right| \, ds
\]

\[
= O(x^{-L}).
\]

Combining these results proves the claim made in (20). The same reasoning shows that

\[
\tilde{G}(x, u) = \sum_{k \in \mathbb{Z}} -\text{Res}(\tilde{G}^* (s, u)x^{-s}; z_k) + O(x^{-L}).
\]

We make the observation that

\[
x^{-z_k} = x^{-2k \pi i / \ln p} = \exp \ln x^{2k \pi i / \ln (1/p)} = \exp \left( \ln x^{2k \pi i / \ln (1/p)} \right) = e^{2k \pi i \log_1 / p} x.
\]
Using this observation, we claim that if \( j \in \mathbb{N} \), then
\[
\tilde{W}_j(x) = \Gamma(j) \frac{q(p/q)^j + p(q/p)^j}{h} + \delta_j (\log_{1/p} x) + O(x^{-L})
\]
where \( h = -p \ln p - q \ln q \) denotes entropy and where
\[
\gamma_j(t) = \sum_{k \neq 0} \frac{e^{2kr \pi i} \Gamma(z_k + j) (p^j q^{-z_k - j + 1} + q^j p^{-z_k - j + 1})}{p^{-z_k + 1} \ln p + q^{-z_k + 1} \ln q}.
\]

To prove the claim, we first observe that, if \( k \in \mathbb{Z} \), then
\[
\text{Res}[\tilde{W}_j(s)x^{-s}; z_k] = x^{-z_k} \text{Res}[\tilde{W}_j(s); z_k] = e^{2kr \pi i} \frac{\Gamma(z_k + j) (p^j q^{-z_k - j + 1} + q^j p^{-z_k - j + 1})}{p^{-z_k + 1} \ln p + q^{-z_k + 1} \ln q}.
\]

Now the claim made in (22) follows immediately from (20).

We also observe that
\[
\tilde{G}(x, u) = -q \ln (1 - pu) + p \ln (1 - qu) - 1 + \gamma(\log_{1/p} x, u) + O(x^{-L})
\]
where \( h = -p \ln p - q \ln q \) denotes entropy and where
\[
\gamma(t, u) = \sum_{k \neq 0} \frac{e^{2kr \pi i} \Gamma(z_k) (q(1 - pu)^{-z_k} + p(1 - qu)^{-z_k} - p^{-z_k + 1} - q^{-z_k + 1})}{p^{-z_k + 1} \ln p + q^{-z_k + 1} \ln q}.
\]

Indeed, the proof is similar to the proof of (22). If \( k \neq 0 \) then
\[
\text{Res}[\tilde{G}^*(s, u)x^{-s}; z_k] = x^{-z_k} \text{Res}[\tilde{G}^*(s, u); z_k] = e^{2kr \pi i} \frac{\Gamma(z_k)}{p^{-z_k + 1} \ln p + q^{-z_k + 1} \ln q} 
\times (q(1 - pu)^{-z_k} + p(1 - qu)^{-z_k} - p^{-z_k + 1} - q^{-z_k + 1}).
\]

Now we compute \( \text{Res}[\tilde{G}^*(s, u)x^{-s}; z_0] \). We first observe that
\[
\Gamma(s) (q(1 - pu)^{-s} + p(1 - qu)^{-s} - p^{-s+1} - q^{-s+1}) = (s^{-1} + O(1))(-q \ln (1 - pu)s - p \ln (1 - qu)s + p \ln (p)s + q \ln (q)s + O(s^2))
\]
\[
= -q \ln (1 - pu) - p \ln (1 - qu) + q \ln p + q \ln q + O(s).
\]

It follows that
\[
\text{Res}[\tilde{G}^*(s, u)x^{-s}; z_0] = x^{-z_0} \text{Res}[\tilde{G}^*(s, u); z_0] = \frac{q \ln (1 - pu) + p \ln (1 - qu)}{h} + 1.
\]

Combining these results, the claim given in (23) now follows from (21).

As an immediate corollary of (23), we see that
\[
\tilde{G}(x, u) = -q \ln (1 - pu) + p \ln (1 - qu) + \gamma(\log_{1/p} x, u) + O(x^{-L}).
\]

We note that, if \( \ln p/\ln q \) is irrational and \( u \) is fixed, then \( \gamma_j(x) \to 0 \) and \( \gamma(x, u) \to 0 \) as \( x \to \infty \). Thus \( \gamma_j \) and \( \gamma(\cdot, u) \) do not exhibit fluctuation when \( \ln p/\ln q \) is irrational.
4.4 Depoissonization

Recall that, in the original problem statement, \( n \) is a large, fixed integer. Most of our analysis has utilized a model where \( n \) is a Poisson random variable. Therefore, to obtain results about the problem we originally stated, it is necessary to depoissonize our results. We utilize the depoissonization techniques discussed in \([10]\) and Chapter 10 of \([24]\), especially the Depoissonization Lemma, to prove Theorems 2 and 3.

For the reader’s convenience we recall here some depoissonization results of \([10]\). Recall that a measurable function \( \psi : (0, \infty) \rightarrow (0, \infty) \) is slowly varying if \( \psi(tx)/\psi(x) \rightarrow 1 \) as \( x \rightarrow \infty \) for every fixed \( t > 0 \).

**Theorem 5.** Assume that \( \tilde{G}(z) = \sum_{n=0}^{\infty} g_n \frac{n^z}{n!} e^{-z} \) is a Poisson transform of a sequence \( g_n \) which is an entire function of a complex variable \( z \). Suppose that there exist real constants \( a < 1 \), \( b \), \( \theta \in (0, \pi/2) \), \( c_1 \), \( c_2 \), and \( z_0 \), and a slowly varying function \( \psi \) such that the following conditions hold, where \( S_\theta \) is the cone \( S_\theta = \{ z : |\arg(z)| \leq \theta \} \):

(I) For all \( z \in S_\theta \) with \( |z| \geq z_0 \),

\[
\tilde{G}(z) \leq c_1|z|^b \psi(|z|) \tag{24}
\]

(O) For all \( z \notin S_\theta \) with \( |z| \geq z_0 \),

\[
|\tilde{G}(z)e^z| \leq c_2e^{a|z|} \tag{25}
\]

Then for \( n \geq 1 \),

\[
g_n = \tilde{G}(n) + O(n^{b-1}\psi(n)) \tag{26}
\]

More precisely,

\[
g_n = \tilde{G}(n) - \frac{1}{2}n\tilde{G}'(n) + O(n^{b-2}\psi(n)) \tag{27}
\]

The “Big-Oh” terms in (26) and (27) are uniform for any family of entire functions \( \tilde{G} \) that satisfy the conditions with the same \( a \), \( b \), \( \theta \), \( c_1 \), \( c_2 \), \( z_0 \) and \( \psi \).

Now, we are in a position to depoissonize our results. By (22), it follows that

\[
|\tilde{W}_j(z)| = \left| \Gamma(j) \frac{q(p/q)^j + p(q/p)^j}{h} + \gamma_j(\log_{1/p} z) + O(z^{-L}) \right|
\]

\[
\leq \left| \Gamma(j) \frac{q(p/q)^j + p(q/p)^j}{h} \right| + \left| \gamma_j(\log_{1/p} z) \right| + O(|z|^{-L})
\]

\[= O(1) \]

since \( |\delta_j| \) is uniformly bounded on \( \mathbb{C} \).

By (23), we see that

\[
|\tilde{G}(z, u)| = \left| -\frac{q \ln (1 - pu) + p \ln (1 - qu)}{h} + \gamma(\log_{1/p} z, u) + O(z^{-L}) \right|
\]

\[
\leq \left| -\frac{q \ln (1 - pu) + p \ln (1 - qu)}{h} \right| + \left| \gamma(\log_{1/p} z, u) \right| + O(|z|^{-L})
\]

\[= O(1) \]

when \( u \) is fixed since \( |\gamma| \) is uniformly bounded on \( \mathbb{C} \).
We define \( \psi(z) = 1 \forall z \) and note that \( \psi \) is a slowing varying function (i.e., \( \psi : (0, \infty) \rightarrow (0, \infty) \) and \( \psi(tx)/\psi(x) \rightarrow 1 \) as \( x \rightarrow \infty \) for every fixed \( t > 0 \)). Also there exist real-valued constants \( c_L, c_{j,L}, z_L, z_{j,L} \) such that

\[
|\tilde{W}_j(z)| \leq c_{j,L}|z|^0\psi(|z|),
\]

and there exists \( z \in S_{\pi/4} = \{ z : |\arg(z)| \leq \pi/4 \} \) with \( |z| \geq z_{j,L} \). Also,

\[
|\tilde{G}(z,u)| \leq c_L|z|^0\psi(|z|),
\]

and there exists \( z \in S_{\pi/4} = \{ z : |\arg(z)| \leq \pi/4 \} \) with \( |z| \geq z_L \). So condition (I) of Theorem 5 is satisfied. It follows immediately by Theorem 10.4 of [24] that condition (O) of [24] (see page 456) is also satisfied. So by Theorem 5 it follows that Theorems 2 and 3 hold, as claimed.

To see that (11) follows from (10), consider the following. From (10), we have

\[
E[u_{M_z}^n] = -\frac{q \ln (1 - pu) + p \ln (1 - qu)}{h} = \gamma(\log_{1/p} n, u) + O(n^{-1}).
\]

Observe

\[
-\frac{q \ln (1 - pu) + p \ln (1 - qu)}{h} = \sum_{j=1}^{\infty} \left( \frac{p^j q + q^j p}{j h} \right) u^j.
\]

Also note that

\[
\gamma(\log_{1/p} n, u) = \sum_{k \neq 0} e^{2k\pi i \log_{1/p} n \Gamma(z_k)} \frac{(1 - pu)^{-z_k} + p(1 - qu)^{-z_k} - p^{-z_k+1} - q^{-z_k+1}}{p^{-z_k+1} \ln p + q^{-z_k+1} \ln q}
\]

\[
= \sum_{j=1}^{\infty} \sum_{k \neq 0} e^{2k\pi i \log_{1/p} n \Gamma(z_k)} (p^j q + q^j p)(z_k)^j u^j.
\]

Then we apply these observations to (4.4) to conclude that (11) holds. Finally, we note that (12) is an immediate corollary of (11).

5 Analysis of LZS’77 via Suffix Trees

We recall that \( X = X_1X_2X_3 \ldots \), where the \( X_i \)'s are i.i.d. random variables on the alphabet \( \mathcal{A} = \{0, 1\} \) with \( \Pr(X_1 = 0) = p \) and \( \Pr(X_1 = 1) = q \). As before, without loss of generality, \( q \leq p \). Let \( X^{(i)} \) denote the \( i \)th suffix of \( X \). Then \( M_n \) is defined as the number of \( X^{(i)} \)'s (with \( 1 \leq i \leq n \)) that also have \( w \) as a prefix, that is,

\[
M_n = \# \{ 1 \leq i \leq n \mid X^{(i)} \text{ has } w \text{ as a prefix} \}.
\]

In Section 3 we redefined \( M_n \) as the multiplicity matching parameter in a suffix tree built over \( X \) (cf. Figure 7). In this section we analyze \( M_n \) and compare it to the distribution of \( M_n^L \) that we re-derive for easier comparisons. In short, we first obtain the bivariate generating functions for \( M_n \) and \( M_n^L \), denoted as \( M(z,u) \) and \( M^L(z,u) \), respectively. Then we prove a few useful lemmas concerning the autocorrelation polynomial. Next, we prove that \( M(z,u) \) can be analytically continued from the unit disk to a larger disk. Afterward, we determine the poles of \( M(z,u) \) and \( M^L(z,u) \). We write \( Q(z,u) = M(z,u) - M^L(z,u) \); we use Cauchy’s theorem to prove that
$$Q_n(u) := |z^n|Q(z, u) \to 0 \text{ uniformly for } u \leq p^{-1/2} \text{ as } n \to \infty.$$ Then we apply Cauchy's theorem again to prove that $Pr(M_n = k) - Pr(M'_n = k) = [u^k z^n]Q(z, u) = O(n^{-\epsilon} b^{-k})$ for some $\epsilon > 0$ and $b > 1$.

We conclude that the distribution of the multiplicity matching parameter $M_n$ is asymptotically the same in suffix trees as in tries built over independent strings, proving Theorem 4, i.e., $M_n$ and $M'_n$ have asymptotically the same distribution. Therefore, $M_n$ also follows the logarithmic series distribution plus some fluctuations, as claimed by Theorem 1.

5.1 BGF for the Multiplicity Matching Parameter of Independent Tries

First we re-derive the bivariate generating function for $M_n^I$ using a different approach (the so called “string-ruler” method) that is well suited for suffix trees. We deal here with a trie built over the independent strings $X(1), \ldots, X(n + 1)$, where $X(i) = X_1(i)X_2(i)X_3(i) \ldots$ and $\{X_j(i) \mid i, j \in \mathbb{N}\}$ is a collection of i.i.d. random variables with $Pr(X_j(i) = 0) = p$ and $Pr(X_j(i) = 1) = q = 1 - p$. We let $w$ denote the longest prefix of both $X(n + 1)$ and at least one other string $X(i)$ for some $1 \leq i \leq n$. We write $\beta$ to denote the $(|w| + 1)$st character of $X(n + 1)$. When $M_n^I = k$, we conclude that exactly $k$ strings $X(i)$ have $w\alpha$ as a prefix, and the other $n - k$ strings $X(i)$ do not have $w$ as a prefix at all. Thus the generating function for $M_n^I$ is exactly

$$M^I(z, u) := \sum_{n=1}^{\infty} \sum_{k=1}^{\infty} Pr(M_n^I = k) u^k z^n = \sum_{n=1}^{\infty} \sum_{k=1}^{\infty} \sum_{w \in A^*} \sum_{\alpha \in A} Pr(w\beta) \left( \binom{n}{k} (Pr(w\alpha))^k (1 - Pr(w))^{n-k} u^k z^n \right).$$

After simplifying, it follows immediately that

$$M^I(z, u) = \sum_{w \in A^*, \alpha \in A} \frac{u^{Pr(\beta)Pr(w)} z^{Pr(w)Pr(\alpha)}}{1 - z(1 - Pr(w))} \frac{zPr(w)Pr(\alpha)}{1 - z(1 + uPr(w)Pr(\alpha) - Pr(w))}.$$ (29)

The same line of reasoning about $M^I(z, u)$ can be applied in the next section to derive the generating function $M(z, u)$ for $M_n$, but the situation will be more complicated because the occurrences of $w$ can overlap.

5.2 BGF for the Multiplicity Matching Parameter of Suffix Trees

Now we obtain the bivariate generating function for $M_n$, which is the multiplicity matching parameter for a suffix tree built over the first $n + 1$ suffixes $X^{(1)}, \ldots, X^{(n+1)}$ of a string $X$ (i.e., $X^{(i)} = X_i X_{i+1} X_{i+2} \ldots$). The bivariate generating function for the multiplicity matching parameter is much more difficult to derive in the dependent (suffix tree) case than in the independent (trie) case, because the suffixes of $X$ are dependent on each other. We let $w$ denote the longest prefix of both $X^{(n+1)}$ and at least one $X^{(i)}$ for some $1 \leq i \leq n$. We write $\beta$ to denote the $(|w| + 1)$st character of $X^{(n+1)}$; when $M_n = k$, we conclude that exactly $k$ suffixes $X^{(i)}$ have $w\alpha$ as a prefix, and the other $n - k$ strings $X^{(i)}$ do not have $w$ as a prefix at all. Thus, we are interested in finding strings with exactly $k$ occurrences of $w\alpha$, ended on the right by an occurrence of $w\beta$, with no other occurrences of $w$ at all. This set of words constitutes a language that can be also represented by two other simpler languages, namely $R_w = \{v \in A^* \mid v \text{ contains exactly one occurrence of } w, \text{ located at the right end}\}$ (30)

$$T_w = \{v \in A^* \mid wav \text{ contains exactly two occurrences of } w, \text{ located at the left and right ends}\}.$$
Thus the generating function for $M_n$ is

$$M(z, u) = \sum_{k=1}^{\infty} \sum_{w \in A^*} \sum_{s \in \mathcal{R}_w} \Pr(s\alpha)z^{|s|+1}u^k \left( \sum_{t \in \mathcal{T}_w^{(\alpha)}} \Pr(t\alpha)z^{|t|+1}u \right)^{k-1} \sum_{v \in \mathcal{T}_w^{(\alpha)}} \Pr(v\beta)z^{|v|+1-|w|-1}. \quad (31)$$

After simplifying the geometric sum, this yields

$$M(z, u) = \sum_{w \in A^*} \frac{uPr(\beta)R_w(z)}{z^{|w|}} \frac{Pr(\alpha)zT_w^{(\alpha)}(z)}{1 - Pr(\alpha)zT_w^{(\alpha)}(z)} \quad (32)$$

where

$$R_w(z) = \sum_{v \in \mathcal{R}_w} \Pr(v)z^{|v|}, \quad T_w^{(\alpha)}(z) = \sum_{v \in \mathcal{T}_w^{(\alpha)}} \Pr(v)z^{|v|}$$

are generating functions of languages $\mathcal{R}_w$ and $\mathcal{T}_w^{(\alpha)}$, respectively. From [17] it is known that $R_w(z)/z^{|w|} = Pr(w)/D_w(z)$, where

$$D_w(z) = (1 - z)S_w(z) + z^mPr(w)$$

and $S_w(z)$ denotes the autocorrelation polynomial for $w$, as defined in (13), which measures the degree to which a word $w$ overlaps with itself. It follows that

$$M(z, u) = \sum_{w \in A^*} \frac{uPr(\beta)Pr(w)}{D_w(z)} \frac{Pr(\alpha)zT_w^{(\alpha)}(z)}{1 - Pr(\alpha)zT_w^{(\alpha)}(z)}. \quad (33)$$

In order to derive an explicit form of $M(z, u)$, we still need to find $T_w^{(\alpha)}(z)$. If we define

$$\mathcal{M}_w = \{v \mid vw \text{ contains exactly two occurrences of } w, \text{ located at the left and right ends}\}, \quad (34)$$

then we observe that $\alpha T_w^{(\alpha)}$ is exactly the subset of words of $\mathcal{M}_w$ that begin with $\alpha$. We use $\mathcal{H}_w^{(\alpha)}$ to denote this subset (i.e., $\mathcal{H}_w^{(\alpha)} = \mathcal{M}_w \cap (\alpha A^*)$), and thus $\alpha T_w^{(\alpha)} = \mathcal{H}_w^{(\alpha)}$. So (33) simplifies to

$$M(z, u) = \sum_{w \in A^*} \frac{uPr(\beta)Pr(w)}{D_w(z)} \frac{H_w^{(\alpha)}(z)}{1 - uH_w^{(\alpha)}(z)}. \quad (35)$$

In order to compute the generating function $H_w^{(\alpha)}(z)$ of $\mathcal{H}_w^{(\alpha)}$, we write $\mathcal{M}_w = \mathcal{H}_w^{(\alpha)} + \mathcal{H}_w^{(\beta)}$, where $\mathcal{H}_w^{(\beta)}$ is the subset of words from $\mathcal{M}_w$ that start with $\beta$ (i.e., $\mathcal{H}_w^{(\beta)} = \mathcal{M}_w \cap (\beta A^*)$). (Note that every word of $\mathcal{M}_w$ begins with either $\alpha$ or $\beta$, because the empty word $\varepsilon \notin \mathcal{M}_w$.) The following useful lemma is the last necessary ingredient to obtain an explicit formula for $M(z, u)$ from (35).

**Lemma 1.** Let $\mathcal{H}_w^{(\alpha)}$ denote the subset of words from $\mathcal{M}_w$ that start with $\alpha$. Then

$$H_w^{(\alpha)}(z) = \frac{D_{w\alpha}(z) - (1 - z)}{D_w(z)}. \quad (36)$$
We note that the set of words with no occurrences of $w\beta$ and the similar (but slightly adapted for our proof) language
\[ U^{(\alpha)}_w = \{ v \mid v \text{ starts with } \alpha, \text{ and } uv \text{ has exactly } 1 \text{ occurrence of } w\alpha \text{ and no occurrences of } w\beta \}. \] (38)

We note that the set of words with no occurrences of $w\beta$ has generating function
\[ \frac{1}{1-z} - \frac{R_{w\beta}(z)U_{w\beta}(z)}{1-M_{w\beta}(z)}. \] (39)

Now we describe the set of words with no occurrences of $w\beta$ in a different way. The set of words with no occurrences of $w\beta$ and at least one occurrence of $w\alpha$ is exactly $R_w(\mathcal{H}^{(\alpha)}_w)^*U^{(\alpha)}_w$, which has generating function
\[ \frac{R_w(z)U^{(\alpha)}_w(z)}{1-H^{(\alpha)}_w(z)}. \]

The set of words with no occurrences of $w\beta$ and no occurrences of $w\alpha$ is exactly $R_w + (A^* \setminus R_w(M_w)^*U)$. (Note that the set of such words that end in $w$ is exactly $R_w$; on the other hand, the set of such words that do not end in $w$ is exactly $A^* \setminus R_w(M_w)^*U$.) So the set of words with no occurrences of $w\alpha$ and no occurrences of $w\beta$ has generating function
\[ \frac{R_w(z)U^{(\alpha)}_w(z)}{1-H^{(\alpha)}_w(z)} + R_w(z) + \frac{1}{1-z} - \frac{R_w(z)U_w(z)}{1-M_w(z)}. \] (40)

Combining (39) and (40), it follows that
\[ \frac{1}{1-z} - \frac{R_{w\beta}(z)U_{w\beta}(z)}{1-M_{w\beta}(z)} = \frac{R_w(z)U^{(\alpha)}_w(z)}{1-H^{(\alpha)}_w(z)} + R_w(z) + \frac{1}{1-z} - \frac{R_w(z)U_w(z)}{1-M_w(z)}. \] (41)

Now we find the generating function for $U^{(\alpha)}_w$. For each word $v \in U^{(\alpha)}_w$, either $uv$ has exactly one or two occurrences of $w$. The subset of $U^{(\alpha)}_w$ of the first type is exactly $V^{(\alpha)}_w := U_w \cap (\alpha A^*)$, i.e., the subset of words from $U_w$ that start with $\alpha$. The subset of $U^{(\alpha)}_w$ of the second type is exactly $\mathcal{H}^{(\alpha)}_w$. We observe that
\[ V^{(\alpha)}_w \cdot A = (\mathcal{H}^{(\alpha)}_w + V^{(\alpha)}_w) \setminus \{ \alpha \} \] (see [25]). Hence, its generating function becomes
\[ V^{(\alpha)}_w(z) = \frac{H^{(\alpha)}_w(z) - Pr(\alpha)z}{z-1}. \]

Since $U^{(\alpha)}_w = V^{(\alpha)}_w + \mathcal{H}^{(\alpha)}_w$, it follows that
\[ U^{(\alpha)}_w(z) = \frac{H^{(\alpha)}_w(z) - Pr(\alpha)z}{z-1} + H^{(\alpha)}_w(z) = \frac{zH^{(\alpha)}_w(z) - Pr(\alpha)z}{z-1}. \] (43)
Recalling equation (41), we see that
\[
\frac{1}{1-z} R_w(z) U_w(z) = \frac{R_w(z)(zH_w^{(\alpha)}(z) - \Pr(\alpha)z)}{(1 - H_w^{(\alpha)}(z))(z - 1)} + R_w(z) + \frac{1}{1-z} R_w(z) U_w(z) - M_w(z),
\] (44)

Simplifying, and using \(U_w(z) = (1 - M_w(z))/(1 - z)\) and \(U_w(z) = (1 - M_w(z))/(1 - z)\) (see [17]), it follows that
\[
\frac{R_w(z)}{R_w(z)} = \frac{z\Pr(\beta)}{1 - H_w^{(\alpha)}(z)}.
\] (45)

Solving for \(H_w^{(\alpha)}(z)\) and then using \(R_w(z) = z^{m} \Pr(w)/D_w(z)\) and \(R_w(z) = z^{m+1} \Pr(w) \Pr(\beta)/D_w(z)\) (see [17]), it follows that
\[
H_w^{(\alpha)}(z) = \frac{D_w(z) - D_w(z)}{D_w(z)}.
\] (46)

Note
\[
D_w(z) - D_w(z) = (1 - z) S_w(z) + z^{m} \Pr(w) - (1 - z) S_w(z) - z^{m+1} \Pr(w) \Pr(\beta)
\]
\[
= (1 - z) (S_{wa}(z) - 1) + z^{m+1} \Pr(w) \Pr(\alpha)
\]
\[
= D_{wa}(z) - (1 - z).
\]

Thus, (46) completes the proof of the lemma.

Using the lemma above, we finally observe a form of \(M(z, u)\) that we summarize below.

Theorem 6. Let \(M(z, u) := \sum_{n=1}^{\infty} \sum_{k=1}^{\infty} \Pr(M_n = k) u^k z^n\) denote the bivariate generating function for \(M_n\), the multiplicity matching parameter of a suffix tree built over the first \(n + 1\) suffixes \(X^{(1)}, \ldots, X^{(n+1)}\) of a string \(X\). Then
\[
M(z, u) = \sum_{w \in A^*} u \Pr(\beta) \Pr(w) \frac{D_{wa}(z) - (1 - z)}{D_w(z) - u(D_{wa}(z) - (1 - z))}
\] (47)

for \(|u| < 1\) and \(|z| < 1\). Here \(D_w(z) = (1 - z) S_w(z) + z^{m} \Pr(w)\), and \(S_w(z)\) denotes the autocorrelation polynomial for \(w\), defined in (13).

5.3 On the Autocorrelation Polynomial

Note that \(p \leq \sqrt{\rho} < 1\), so there exists \(\rho > 1\) such that \(\rho \sqrt{\rho} < 1\) (and thus \(\rho \sqrt{\rho} < 1\)) too. Finally, define \(\delta = \sqrt{\rho}\). We establish a few lemmas about the autocorrelation polynomial that will be important for our analysis.

The autocorrelation polynomial \(S_w(z)\) contains the term \(\Pr(w^m_{k+1}) z^{m-k}\) if and only if \(w\) has an overlap with itself of length \(k\). Since each word \(w\) overlaps with itself trivially, then every autocorrelation polynomial has a constant term (i.e., \(z^{m-m} = z^0 = 1\) term). With high probability, however, \(w\) has very few large nontrivial overlaps with itself. Therefore, with high probability, all nontrivial overlaps of \(w\) with itself are small; such overlaps correspond to high-degree terms of \(S_w(z)\). Therefore, when \(w\) is a randomly chosen long word, then \(S_w(z)\) is very close to 1 with very high probability. The first lemma makes this notion mathematically precise.
Lemma 2. If $\theta = (1 - pp)^{-1} > 1$, then for $\delta = \sqrt{p}$ and $\rho \delta < 1$

\[
\sum_{w \in \mathcal{A}^k} |S_w(\rho) - 1| \leq (\rho \delta)^k \theta \Pr(w) \geq 1 - \delta^k \theta \tag{48}
\]

where $[A] = 1$ if $A$ holds, and $[A] = 0$ otherwise.

**Proof.** Our proof is the one given in [6]. Note that $S_w(z) - 1$ has a term of degree $i \leq j$ if and only if $m - i \in \mathcal{P}(w)$ with $1 \leq i \leq j$. Therefore, for each such $i$ and each $w_1 \ldots w_i$, there is exactly one word $w_{i+1} \ldots w_k$ such that $S_w(z) - 1$ has a term of degree $i$. Therefore, for fixed $j$ and $k$,

\[
\sum_{w \in \mathcal{A}^k} [S_w(z) - 1 \text{ has a term of degree } i] \Pr(w) \leq \sum_{1 \leq i \leq j} \Pr(w_1 \ldots w_i) \sum_{w_{i+1} \ldots w_k \in \mathcal{A}^{k-i}} [S_w(z) - 1 \text{ has a term of degree } i] \Pr(w_{i+1} \ldots w_k) \leq \sum_{1 \leq i \leq j} \Pr(w_1 \ldots w_i) p^{k-i} = \sum_{1 \leq i \leq j} p^{k-i} \frac{p^{k-j}}{1 - p}. \tag{49}
\]

We use $j = [k/2]$. Thus $\sum_{w \in \mathcal{A}^k} [\text{all terms of } S_w(z) - 1 \text{ have degree } [k/2]] \Pr(w) \geq 1 - \delta^k \theta$.

Note that, if all terms of $S_w(z) - 1$ have degree $> [k/2]$, then

\[
|S_w(\rho) - 1| \leq \sum_{i > [k/2]} (pp)^i = \frac{(pp)^{[k/2]+1}}{1 - pp} \leq \frac{(pp)^{k/2}}{1 - pp} \leq \frac{p^k p^{k/2}}{1 - pp} = (\rho \delta)^k \theta. \tag{50}
\]

This completes the proof of the lemma.

Using this lemma, we can quickly obtain another result that is similar but slightly stronger. First consider words $w$ such that $|S_w(\rho) - 1| \leq (\rho \delta)^k \theta$. Write $S_w(z) = \sum_{i=0}^{k-1} a_i z^i$ and $S_{wa}(z) = \sum_{i=0}^{k} b_i z^i$.

Observe that either $b_i = 0$ or $b_i = a_i$. The next lemma follows immediately.

Lemma 3. If $\theta = (1 - pp)^{-1} + 1$ and $\alpha \in \mathcal{A}$, then

\[
\sum_{w \in \mathcal{A}^k} \max\{|S_w(\rho) - 1|, |S_{wa}(\rho) - 1|\} \leq (\rho \delta)^k \theta \Pr(w) \geq 1 - \delta^k \theta, \tag{51}
\]

with the same notations as in Lemma 2

Also, the autocorrelation polynomial is never too small. In fact

Lemma 4. Define $c = 1 - \rho \sqrt{p} > 0$. Then there exists an integer $K \geq 1$ such that, for $|w| \geq K$ and $|z| \leq \rho$ and $|u| \leq \delta^{-1}$,

\[
|S_w(z) - uS_{wa}(z) + u| \geq c. \tag{52}
\]

**Proof.** The proof consists of considering several cases. The only condition for $K$ is $(1 + \delta^{-1}) \frac{(pp)^{K/2}}{1 - pp} \leq c/2$. The analysis is not difficult; all details are presented in [25].
5.4 Analytic Continuation

In order to establish (15) of Theorem 4 we need to prove that $M(z,u)$ can be analytically continued.

**Theorem 7.** The generating function $M(z,u)$ can be analytically continued for $|u| \leq \delta^{-1}$ and $|z| < 1$.

The proof requires several lemmas and observations. We always assume $|u| \leq \delta^{-1}$.

**Lemma 5.** If $0 < r < 1$, then there exists $C > 0$ and an integer $K_1$ (both depending on $r$) such that

$$|D_w(z) - u(D_{wa}(z) - (1-z))| \geq C$$

for $|w| \geq K_1$ and $|z| \leq r$ (and, as before, $|u| \leq \delta^{-1}$).

**Proof.** Consider the $K$ and $c$ defined in Lemma 4, which tells us that, for all $|w| \geq K$, we have

$$|S_w(z) - uS_{wa}(z) + u| \geq c$$

for $|z| \leq \rho$. So, for $|w| \geq K$, we have

$$|D_w(z) - u(D_{wa}(z) - (1-z))| \geq (1-r)c - r^m p^m (1-\delta^{-1}rp).$$

Note that $r^m p^m (1-\delta^{-1}rp) \to 0$ as $m \to \infty$. Therefore, replacing $K$ by a larger $K_1$ if necessary, we can assume without loss of generality that $r^m p^m (1-\delta^{-1}rp) \leq (1-r)c/2$. So we define $C = (1-r)c/2$, and the result follows immediately.

Now we can strengthen the previous lemma by dropping the condition $|w| \geq K_1$, i.e., by not requiring $w$ to be a long word.

**Lemma 6.** If $0 < r < 1$, then there exists $C > 0$ (depending on $r$) such that

$$|D_w(z) - u(D_{wa}(z) - (1-z))| \geq C$$

for $|z| \leq r$ (and, as before, $|u| \leq \delta^{-1}$).

**Proof.** Consider the $K_1$ defined in Lemma 5. Let $C_0$ denote the “$C$” from Lemma 5. There are only finitely many $w$’s with $|w| < K_1$, say $w_1, \ldots, w_i$. For each such $w_j$ (with $1 \leq j \leq i$), we note that $D_{w_j}(z) - u(D_{w_ja}(z) - (1-z)) \neq 0$ for $|z| \leq r$ and $|u| \leq \delta^{-1}$, so there exists $C_j > 0$ such that $|D_{w_j}(z) - u(D_{w_ja}(z) - (1-z))| \geq C_j$ for all $|z| \leq r$ and $|u| \leq \delta^{-1}$. Finally, we define $C = \min\{C_0, C_1, \ldots, C_i\}$.

**Proof of Theorem 7.** Consider $|z| \leq r < 1$. We proved in Lemma 6 there exists $C > 0$ depending on $r$ such that, for all $|u| \leq \delta^{-1}$, we have

$$\frac{1}{|D_w(z) - u(D_{wa}(z) - (1-z))|} \leq \frac{1}{C}.$$

Setting $u = 0$, we also have $|D_w(z)| \geq C$. Thus

$$|M(z,u)| \leq \frac{Pr(\beta)\delta^{-1}}{C^2} \sum_{\alpha \in A} \sum_{w \in A^*} Pr(w)|D_{wa}(z) - (1-z)|.$$  

(56)
Now we use Lemma 3. Consider \( w \) and \( \alpha \) with \( \max\{|S_w(\rho)|, |S_w(\rho) - 1|\} \leq (\rho \delta)^m \theta \). It follows immediately that

\[
|D_{wa}(z) - (1 - z)| = |(1 - z)(S_{wa}(z) - 1) + z^{m+1}\operatorname{Pr}(w)\operatorname{Pr}(\alpha)| \leq (1 + r)(\rho \delta)^m \theta + r^{m+1}p^m p = O(s^m),
\]

where \( s = \max\{\rho \delta, rp\} \). Now consider the other \( w \)'s and \( \alpha \)'s. We have

\[
|D_{wa}(z) - (1 - z)| = |(1 - z)(S_{wa}(z) - 1) + z^{m+1}\operatorname{Pr}(w)\operatorname{Pr}(\alpha)| \leq \frac{(1 + r)pp}{1 - pp} + r^{m+1}p^m p \leq \frac{(1 + r)pp}{1 - pp} + 1,
\]

so we define \( C_1 = \frac{(1 + r)pp}{1 - pp} + 1 \) to be a value which depends only on \( r \) (recall that \( r \) is fixed here).

Thus

\[
|M(z, u)| \leq \frac{\operatorname{Pr}(\beta)\delta^{-1}}{C^2} \sum_{\alpha \in A} \sum_{m \geq 0} \sum_{w \in A^m} |\operatorname{Pr}(w)(D_{wa}(z) - (1 - z))| \leq \frac{\operatorname{Pr}(\beta)\delta^{-1}}{C^2} \sum_{\alpha \in A} \sum_{m \geq 0} |1 - \delta^m \theta O(s^m) + \delta^m \theta C_1| \leq \frac{\operatorname{Pr}(\beta)\delta^{-1}}{C^2} \sum_{\alpha \in A} \sum_{m \geq 0} O(s^m) = O(1),
\]

and this completes the proof of the theorem.

\[\blacksquare\]

### 5.5 Singularity Analysis

We need some auxiliary results before we prove our main result of this section, namely Theorem 4. We first determine (for \( |u| \leq \delta^{-1} \)) the zeroes of \( D_w(z) - u(D_{wa}(z) - (1 - z)) \) and in particular the zeroes of \( D_w(z) \).

**Lemma 7.** There exists an integer \( K_2 \geq 1 \) such that, for \( u \) fixed (with \( |u| \leq \delta^{-1} \)) and \( |w| \geq K_2 \), there is exactly one root of \( D_w(z) - u(D_{wa}(z) - (1 - z)) \) in the closed disk \( \{z \mid |z| \leq \rho\} \).

**Proof.** Let \( K \) and \( c \) be defined as in Lemma 4. Without loss of generality (replacing \( K \) by a larger \( K_2 \), if necessary), we can also assume that \( 2(pp)^{K_2} < c(\rho - 1) \) and \( K_2 \geq K_1 \) (where \( K_1 \) is defined in Lemma 5). Also, we can choose \( K_2 \) large enough (for use later) such that \( \exists \epsilon_2 > 0 \) with

\[
\rho(1 - p^{K_2}(1 + \delta^{-1}p)) - 1 > c_2 \quad \text{and thus} \quad \rho(1 - p^{K_2}) - 1 > c_2.
\]

We recall \( 0 < pp\delta^{-1} < 1 \), and thus \( 0 < 1 - pp\delta^{-1} < 1 \). Since \( |u| < \delta^{-1} \) and \( |z| \leq \rho \), then for \( |w| \geq K_2 \) we have

\[
|\operatorname{Pr}(w)z^m(1 - uz\operatorname{Pr}(\alpha))| \leq (pp)^m(1 + \delta^{-1}pp) \leq 2(pp)^m < c(\rho - 1) \leq |(S_w(z) - uS_{wa}(z) + u)(\rho - 1)|.
\]

Therefore, for \( z \) on the circle \( \{z \mid |z| = \rho\} \), we have

\[
|\operatorname{Pr}(w)z^m(1 - uz\operatorname{Pr}(\alpha))| < |(S_w(z) - uS_{wa}(z) + u)(\rho - 1)|.
\]

Equivalently,

\[
|(D_w(z) - u(D_{wa}(z) - (1 - z))) - ((S_w(z) - uS_{wa}(z) + u)(\rho - 1))| < |(S_w(z) - uS_{wa}(z) + u)(\rho - 1)|.
\]
Therefore, by Rouché’s Theorem, \( D_w(z) - u(D_{wo}(z) - (1 - z)) \) and \( (S_w(z) - uS_{wo}(z) + u)(z - 1) \) have the same number of zeroes inside the disk \( \{z \mid |z| \leq \rho\} \). Since \( |S_w(z) - uS_{wo}(z) + u| \geq c \) inside this disk, we conclude that \( (S_w(z) - uS_{wo}(z) + u)(z - 1) \) has exactly one root in the disk. It follows that \( D_w(z) - u(D_{wo}(z) - (1 - z)) \) also has exactly one root in the disk.

When \( u = 0 \), this lemma implies (for \( |w| \geq K_2 \)) that \( D_w(z) \) has exactly one root in the disk \( \{z \mid |z| \leq \rho\} \). Let \( A_w \) denote this root, and let \( B_w = D_w(A_w) \). Also let \( C_w(u) \) denote the root of \( D_w(z) - u(D_{wo}(z) - (1 - z)) \) in the closed disk \( \{z \mid |z| \leq \rho\} \). Finally, we define

\[
E_w(u) := \left( \frac{d}{dz} (D_w(z) - u(D_{wo}(z) - (1 - z))) \right)_{z=C_w} = D'_w(C_w) - u(D'_{wo}(C_w) + 1). \tag{60}
\]

We have precisely determined the singularities of \( M(z, u) \). Next, we compare \( M(z, u) \) to \( M^I(z, u) \) to show that \( M_n \) and \( M_n^I \) have asymptotically similar behaviors.

### 5.6 Comparing Suffix Trees to Tries

We shall finally prove here Theorem 4 by comparing the generating functions \( M(z, u) \) and \( M^I(z, u) \). Now we define

\[
Q(z, u) = M(z, u) - M^I(z, u). \tag{61}
\]

Using the notation from (29) and (47), if we write

\[
\begin{align*}
M^I_{w, \alpha}(z, u) &= \frac{u \Pr(\beta) \Pr(w)}{1 - z(1 - \Pr(\beta))} \frac{z \Pr(w) \Pr(\alpha)}{1 - z(1 + u \Pr(w) \Pr(\alpha) - \Pr(w))}, \\
M_{w, \alpha}(z, u) &= \frac{u \Pr(\beta) \Pr(w)}{D_w(z)} \frac{D_{wo}(z) - (1 - z)}{D_w(z) - u(D_{wo}(z) - (1 - z))}, \tag{62}
\end{align*}
\]

then we have proved that

\[
Q(z, u) = \sum_{w \in A^* \atop \alpha \in A} (M_{w, \alpha}(z, u) - M^I_{w, \alpha}(z, u)). \tag{63}
\]

We also define \( Q_n(u) = [z^n]Q(z, u) \). We denote the contribution to \( Q_n(u) \) from a specific \( w \) and \( \alpha \) as \( Q_{n(w, \alpha)}(u) = [z^n](M_{w, \alpha}(z, u) - M^I_{w, \alpha}(z, u)) \). Then we observe that

\[
Q_{n(w, \alpha)}(u) = \frac{1}{2\pi i} \int_{|z| = \rho} (M_{w, \alpha}(z, u) - M^I_{w, \alpha}(z, u)) \frac{dz}{z^{n+1}} \tag{64}
\]

where the path of integration is a circle about the origin with counterclockwise orientation.

We define

\[
I_{w, \alpha}(\rho, u) = \frac{1}{2\pi i} \int_{|z| = \rho} (M_{w, \alpha}(z, u) - M^I_{w, \alpha}(z, u)) \frac{dz}{z^{n+1}}. \tag{65}
\]

By Cauchy’s theorem, we observe that the contribution to \( Q_n(u) \) from a specific \( w \) and \( \alpha \) is exactly

\[
Q_{n(w, \alpha)}(u) = I_{w, \alpha}(\rho, u) - \text{Res}_{z=A_w} \frac{M_{w, \alpha}(z, u)}{z^{n+1}} - \text{Res}_{z=C_w(u)} \frac{M_{w, \alpha}(z, u)}{z^{n+1}} + \text{Res}_{z=1/(1-\Pr(w))} \frac{M^I_{w, \alpha}(z, u)}{z^{n+1}} + \text{Res}_{z=1/(1+u\Pr(w)\Pr(\alpha)-\Pr(w))} \frac{M^I_{w, \alpha}(z, u)}{z^{n+1}}. \tag{66}
\]

30
To simplify this expression, note that

\[
\text{Res}_{z=A_w} \frac{M_{w,\alpha}(z, u)}{z^{n+1}} = -\frac{\Pr(\beta)\Pr(w)}{B_w} \frac{1}{A_w^{n+1}},
\]

\[
\text{Res}_{z=C_w(u)} \frac{M_{w,\alpha}(z, u)}{z^{n+1}} = \frac{\Pr(\beta)\Pr(w)}{E_w(u)} \frac{1}{C_w(u)^{n+1}},
\]

\[
\text{Res}_{z=1/(1-\Pr(w))} \frac{M_{w,\alpha}(z, u)}{z^{n+1}} = \Pr(\beta)\Pr(w)(1-\Pr(w))^n,
\]

\[
\text{Res}_{z=1/(1+u\Pr(w)\Pr(\alpha)-\Pr(w))} \frac{M_{w,\alpha}(z, u)}{z^{n+1}} = -\Pr(\beta)\Pr(w)(1+u\Pr(w)\Pr(\alpha)-\Pr(w))^n. \quad (67)
\]

It follows from (66) that

\[
Q_n^{(w,\alpha)}(u) = I_{w,\alpha}(\rho, u) + \frac{\Pr(\beta)\Pr(w)}{B_w} \frac{1}{A_w^{n+1}} - \frac{\Pr(\beta)\Pr(w)}{E_w(u)} \frac{1}{C_w(u)^{n+1}} + \Pr(\beta)\Pr(w)(1-\Pr(w))^n - \Pr(\beta)\Pr(w)(1+u\Pr(w)\Pr(\alpha)-\Pr(w))^n. \quad (68)
\]

We next determine the contribution of the \(z = A_w\) terms of \(M(z, u)\) and the \(z = 1/(1-\Pr(w))\) terms of \(M^I(z, u)\) to the difference \(Q_n(u) = [z^n](M(z, u) - M^I(z, u))\).

**Lemma 8.** The “\(A_w\) terms” and the “\(1/(1-\Pr(w))\) terms” (for \(|w| \geq K_2\)) altogether have only \(O(n^{-\epsilon})\) contribution to \(Q_n(u)\), i.e.,

\[
\sum_{|w| \geq K_2, \alpha \in \mathcal{A}} \left( -\text{Res}_{z=A_w} \frac{M_{w,\alpha}(z, u)}{z^{n+1}} + \text{Res}_{z=1/(1-\Pr(w))} \frac{M_{w,\alpha}^I(z, u)}{z^{n+1}} \right) = O(n^{-\epsilon}), \quad (69)
\]

for some \(\epsilon > 0\).

**Proof.** We define

\[
f_w(x) = \frac{1}{A_w^{x+1}B_w} + (1-\Pr(w))^x \quad (70)
\]

for \(x\) real. So by (67) it suffices to prove that

\[
\sum_{|w| \geq K_2, \alpha \in \mathcal{A}} \Pr(\beta)\Pr(w)f_w(x) = O(x^{-\epsilon}). \quad (71)
\]

Note that \(\sum_{|w| \geq K_2, \alpha \in \mathcal{A}} \Pr(\beta)\Pr(w)f_w(x)\) is absolutely convergent for all \(x\). Also \(\tilde{f}_w(x) = f_w(x) - f_w(0)e^{-x}\) is exponentially decreasing when \(x \to +\infty\) and is \(O(x)\) when \(x \to 0\) (notice that we utilize the \(f_w(0)e^{-x}\) term in order to make sure that \(\tilde{f}_w(x) = O(x)\) when \(x \to 0\); this provides a fundamental strip for the Mellin transform in the next step). Therefore, its Mellin transform \(\tilde{f}_w^s(s) = \int_0^\infty \tilde{f}_w(x)x^{s-1}dx\) is well-defined for \(\Re(s) > -1\) (see [7] and [24]). We compute

\[
\tilde{f}_w^s(s) = \Gamma(s) \left( \frac{(\log A_w)^{-s} - 1}{A_wB_w} + (-\log(1-\Pr(w)))^{-s} - 1 \right), \quad (72)
\]

where \(\Gamma\) denotes the Euler gamma function, and we note that

\[
(\log A_w)^{-s} = \left( \frac{\Pr(w)}{S_w(1)} \right)^{-s} (1 + O(\Pr(w))),
\]

31


\[ (-\log(1 - \Pr(w)))^{-s} = \Pr(w)^{-s}(1 + O(\Pr(w))). \tag{73} \]

Also
\[
A_w = 1 + \frac{1}{S_w(1)} \Pr(w) + O(\Pr(w)^2), \quad B_w = -S_w(1) + \left( \frac{2S_w(1)}{S_w(1)} + m \right) \Pr(w) + O(\Pr(w)^2). \tag{74}\]

Therefore
\[
\frac{1}{A_w B_w} = -\frac{1}{S_w(1)} + O(|w|\Pr(w)), \tag{75}\]

and
\[
\tilde{f}_w^*(s) = \Gamma(s) \left( \left( -\frac{1}{S_w(1)} + O(|w|\Pr(w)) \right) \left( \frac{\Pr(w)}{S_w(1)} \right)^{-s} (1 + O(\Pr(w))) - 1 \right) \\
+ \Pr(w)^{-s}(1 + O(\Pr(w))) - 1 \right) \\
= \Gamma(s) \left( \Pr(w)^{-s} (-S_w(1)^{s-1} + 1 + O(|w|\Pr(w))) + \frac{1}{S_w(1)} - 1 + O(|w|\Pr(w)) \right). \tag{76}\]

We define \( g^*(s) = \sum_{\alpha \in \mathcal{A}} \Pr(\beta)\Pr(w)\tilde{f}_w^*(s) \). Then we compute
\[
g^*(s) = \sum_{\alpha \in \mathcal{A}} \Pr(\beta) \sum_{|w| \geq K_2} \Pr(w) \tilde{f}_w^*(s) = \sum_{\alpha \in \mathcal{A}} \Pr(\beta) \Gamma(s) \sum_{m=K_2}^{\infty} \left( \sup \{ q^{-\Re(s)}, 1 \} \delta \right)^m O(1), \tag{76}\]

where the last equality is true because \( 1 \geq p^{-\Re(s)} \geq q^{-\Re(s)} \) when \( \Re(s) \) is negative, and also because \( q^{-\Re(s)} \geq p^{-\Re(s)} \geq 1 \) when \( \Re(s) \) is positive. We always have \( \delta < 1 \). Also, there exists \( c > 0 \) such that \( q^{-\epsilon} \delta < 1 \). Therefore, \( g^*(s) \) is analytic in \( \Re(s) \in (-1, c) \). Working in this strip, we choose \( \epsilon \) with \( 0 < \epsilon < c \). Then we have
\[
\sum_{|w| \geq K_2} \Pr(\beta)\Pr(w)f_w(x) = \frac{1}{2\pi i} \int_{\epsilon-i\infty}^{\epsilon+i\infty} g^*(s)x^{-s} ds + \sum_{|w| \geq K_2} \Pr(\beta)\Pr(w)f_w(0)e^{-x}. \tag{77}\]

Majorizing under the integral, we see that the first term is \( O(x^{-\epsilon}) \) since \( g^*(s) \) is analytic in the strip \( \Re(s) \in (-1, c) \) (and \(-1 < \epsilon < c \)). Also, the second term is \( O(e^{-x}) \). This completes the proof of the lemma. \( \blacksquare \)

Now we bound the contribution to \( Q_n(u) \) from the \( C_w(u) \) terms of \( M(z, u) \) and the \( z = 1/(1 + u\Pr(w)\Pr(\alpha) - \Pr(w)) \) terms of \( M^I(z, u) \).

**Lemma 9.** The “\( C_w(u) \) terms” and the “\( 1/(1 + u\Pr(w)\Pr(\alpha) - \Pr(w)) \) terms” (for \(|w| \geq K_2 \)) altogether have only \( O(n^{-\epsilon}) \) contribution to \( Q_n(u) \), for some \( \epsilon > 0 \). More precisely,
\[
\sum_{|w| \geq K_2} \Pr(\beta)\Pr(w)f_w(x) = \frac{1}{2\pi i} \int_{\epsilon-i\infty}^{\epsilon+i\infty} g^*(s)x^{-s} ds + \sum_{|w| \geq K_2} \Pr(\beta)\Pr(w)f_w(0)e^{-x}. \tag{77}\]

\[
\sum_{|w| \geq K_2} \Pr(\beta)\Pr(w)f_w(x) = \frac{1}{2\pi i} \int_{\epsilon-i\infty}^{\epsilon+i\infty} g^*(s)x^{-s} ds + \sum_{|w| \geq K_2} \Pr(\beta)\Pr(w)f_w(0)e^{-x}. \tag{77}\]

32
Proof. The proof technique is the same as the one for Lemma 8 above.

Next we prove that the $I_{w, \alpha}(\rho, u)$ terms in (68) have $O(n^{-\epsilon})$ contribution to $Q_n(u)$.

Lemma 10. The "$I_{w, \alpha}(\rho, u)$ terms" (for $|w| \geq K_2$) altogether have only $O(n^{-\epsilon})$ contribution to $Q_n(u)$, for some $\epsilon > 0$. More precisely,

$$\sum_{|w| \geq K_2, \alpha \in A} I_{w, \alpha}(\rho, u) = O(n^{-\epsilon}).$$

Proof. Here we only sketch the proof. A rigorous proof is given in [25]. Recall that

$$I_{w, \alpha}(\rho, u) = \frac{1}{2\pi i} \int_{|z| = \rho} u \Pr(\beta) \Pr(w) \left( \frac{1}{D_w(z)} \frac{D_{wa}(z) - (1 - z)}{D_w(z) - u(D_{wa}(z) - (1 - z))} \right) dz$$

By Lemma 7, $K_2$ was selected to be sufficiently large such that $(pp)^m(1 - \delta^{-1}pp) \leq (\rho - 1)/c/2$. Thus, by setting $C_1 = (\rho - 1)c/2$, we have $1/|D_w(z) - u(D_{wa}(z) - (1 - z))| \leq 1/C_1$ and thus $1/|D_w(z)| \leq 1/C_1$. Also $1/|1-z(1-Pr(w))| \leq 1/c_2$ and $1/|1-z(1+uPr(w)Pr(\alpha)-Pr(w))| \leq 1/c_2$ by (59). So we obtain

$$|I_{w, \alpha}(\rho, u)| = O((\rho^{-n})\Pr(w)(S_{wa}(\rho) - 1) + O(\rho^{-n})\Pr(w)O((pp)^m)).$$

Thus, by Lemma 3, $\sum_{\alpha \in A} \sum_{|w| = m} |I_{w, \alpha}(\rho, u)| = O(\rho^{-n})O((\rho \delta)^m)$. We conclude that $\sum_{\alpha \in A} \sum_{|w| \geq K_2} |I_{w, \alpha}(\rho, u)| = O(\rho^{-n})$, and the lemma follows.

Finally, we consider the contribution to $Q_n(u)$ from small words $|w|$. Basically, we prove that $|w|$ has a normal distribution with mean $\frac{1}{h} \log n$ and variance $\theta \log n$, where $h = -p \log p - q \log q$ denotes the entropy of the source, and $\theta$ is a constant. Therefore, $|w| \leq K_2$ is extremely unlikely, and as a result, the contribution to $Q_n(u)$ from words $w$ with $|w| \leq K_2$ is very small.

Lemma 11. The terms $\sum_{\alpha \in A} \sum_{|w| < K_2} (M_{w, \alpha}(z, u) - M'_{w, \alpha}(z, u))$ altogether have only $O(n^{-\epsilon})$ contribution to $Q_n(u)$.

Proof. Let $D_n$ denote the depth of the $(n+1)$st insertion in a suffix tree, i.e., $D_n < k$ if and only if

$$X_{n+1} \ldots X_{n+k} \neq X_{i+1} \ldots X_{i+k} \quad \text{for all } 0 \leq i < n$$

i.e., $D_n = |w|$ in the notation of Section 5.2. Similarly, let $D'_n$ denote the depth of the $(n+1)$st insertion in a trie built over $n+1$ independent strings, i.e., $D'_n < k$ if and only if

$$X_1(n+1) \ldots X_k(n+1) \neq X_i(i) \ldots X_i(i) \quad \text{for all } 1 \leq i \leq n$$

i.e., $D'_n = |w|$ in the notation of Section 5. Therefore,

$$[z^n] \sum_{\alpha \in A} \sum_{|w| < K_2} (M_{w, \alpha}(z, u) - M'_{w, \alpha}(z, u)) = \sum_{i < K_2} \sum_{k=1}^n \left( \Pr(M_n = k \& D_n = i) - \Pr(M'_n = k \& D'_n = i) \right) u^k.$$

(84)
Noting that \( \Pr(M_n = k & D_n = i) \leq \Pr(D_n = i) \) and \( \Pr(M_n^I = k & D_n^I = i) \leq \Pr(D_n^I = i) \), it follows that

\[
[z^n] \sum_{|u| < K_2} \sum_{\alpha \in A} |M_{w,\alpha}(z, u) - M_{w,\alpha}^I(z, u)| \leq \sum_{i < K_2} \sum_{k=1}^{n} (\Pr(D_n = i) + \Pr(D_n^I = i)) |u|^k. \tag{85}
\]

In [9], the typical depth \( D_{n+1}^T \) in a trie built over \( n+1 \) independent strings was shown to be asymptotically normal with mean \( \frac{1}{n} \log(n+1) \) and variance \( \theta \log(n+1) \). We observe that \( D_n^I \) (defined in (83)) and \( D_{n+1}^T \) have the same distribution; to see this, observe that

\[
\Pr(D_n^I < k) = \sum_{|w|=k} \Pr(w)(1 - \Pr(w))^n = \Pr(D_{n+1}^T < k).
\]

Therefore, \( D_n^I \) is also asymptotically normal with mean \( \frac{1}{n} \log n \) and variance \( \theta \log n \). In [25], we rigorously prove that \( D_n^I \) and \( D_n \) have asymptotically the same distribution, namely, a normal distribution with mean \( \frac{1}{n} \log(n+1) \) and variance \( \theta \log(n+1) \). Therefore, considering (85) (and noting that \( K_2 \) is a constant), it follows that

\[
[z^n] \sum_{|w| < K_2} \sum_{\alpha \in A} |M_{w,\alpha}(z, u) - M_{w,\alpha}^I(z, u)| = O(n^{-\epsilon}). \tag{86}
\]

This completes the proof of the lemma.

All contributions to (68) have now been analyzed. We are finally prepared to summarize our results.

### 5.7 Summary

Combining the last four lemmas, we see that \( Q_n(u) = O(n^{-\epsilon}) \) uniformly for \( |u| \leq \delta^{-1} \), where \( \delta^{-1} > 1 \). For ease of notation, we define \( b = \delta^{-1} \). Finally, we apply Cauchy’s theorem again. We compute

\[
\Pr(M_n = k) - \Pr(M_n^I = k) = [u^k z^n]Q(z, u) = [u^k]Q_n(u) = \frac{1}{2\pi i} \int_{|u|=b} \frac{Q_n(u)}{u^{k+1}} \, du. \tag{87}
\]

Since \( Q_n(u) = O(n^{-\epsilon}) \), it follows that

\[
|\Pr(M_n = k) - \Pr(M_n^I = k)| \leq \frac{1}{|2\pi i|} (2\pi b) O(n^{-\epsilon}) \frac{n^{-\epsilon}}{b^{k+1}} = O(n^{-\epsilon} b^{-k}). \tag{88}
\]

Thus Theorem 4 holds. It follows that \( M_n \) and \( M_n^I \) have asymptotically the same distribution, and therefore \( M_n \) and \( M_n^I \) asymptotically have the same factorial moments. The main result of [26] gives the asymptotic distribution and factorial moments of \( M_n^I \). As a result, Theorem 4 follows immediately. Therefore, \( M_n \) follows the logarithmic series distribution, i.e., \( \Pr(M_n = j) = \frac{p^j q + q^j p}{j^n} \) (plus some small fluctuations if \( \ln p / \ln q \) is rational). Theorem 1 is finally proved.
6 Concluding Remarks

From the algorithmic perspective, two immediate challenges remain. First, we would like to make LZRS’77 on-line. The implementation of LZRS’77 described here is off-line because the blocks need to be processed backwards, but it is not clear if this is absolutely necessary. Second, we would like to be able to protect the first block while maintaining the backward compatibility. Note that we cannot embed the parity bits of the first block in the pointers of the last, because otherwise we would introduce a circular dependency in the process. From an analytic perspective, it would be interesting to extend Theorem 1 to Markov sources. While it is well-known [28] that the expectation for Markov sources is $E[M_n] = O(1)$ (cf. [13]), not much is known about the distribution of $M_n$ under that probabilistic model. The recent work of Fayolle and Ward [6], in which they extend the analysis of [9] to Markov sources, is a step in that direction.

References


