Analysis of a Block Arithmetic Coding: Discrete Divide and Conquer Recurrences

Michael Drmota
Inst. Discrete Mathematics and Geometry,
TU Wien,
A-1040 Wien, Austria,
Email: michael.drmota@tuwien.ac.at

Wojciech Szpankowski
Department of Computer Science,
Purdue University,
West Lafayette, IN 47907-2066 U.S.A.,
Email: spa@cs.purdue.edu

***Abstract*—In 1993 Boncelet introduced a block arithmetic scheme for entropy coding that combines advantages of stream arithmetic coding with algorithmic simplicity. It is a variable-to-fixed length encoding in which the source sequence is partitioned into variable length phrases that are encoded by a fixed length dictionary pointer. The parsing is accomplished through a complete parsing tree whose leaves represent phrases. This tree, in its suboptimal heuristic version, is constructed by a simple divide and conquer algorithm, whose analysis is the subject of this paper. For a memoryless source, we first derive the average redundancy and compare it to the (asymptotically) optimal Tunstall's algorithm. Then we prove a central limit theorem for the phrase length. To establish these results, we apply powerful techniques such as Dirichlet series, Mellin-Perron formula, and (extended) Tauberian theorems of Wiener-Ikehara.**

I. Introduction

We present a comprehensive analysis of a data compression algorithm due to Boncelet [3] known as *Block Arithmetic Coding* (BAC). Boncelet's algorithm is a variable-to-fixed data compression scheme. To recall, a variable-to-fixed length encoder partitions a source string over an m-ary alphabet $\mathcal{A}$ into a concatenation of variable-length phrases. Each phrase belongs to a given dictionary of source strings. A uniquely parsable dictionary is represented by a *complete parsing tree*, i.e., a tree in which every internal node has all m children nodes. The dictionary entries correspond to the *leaves* of the associated parsing tree. The encoder represents each parsed string by a fixed length binary code corresponding to its dictionary entry. There are several well known variable-to-fixed algorithms; e.g., Tunstall and Khodak schemes (cf. [10], [17], [25]). Boncelet's algorithm is based on a *divide and conquer* strategy, and therefore is fast and easy to implement.

Arithmetic entropy coders have been intensively studied in literature [9], [20], [21]. They are stream coders: an arbitrary long input sequence outputs a corresponding output stream. One disadvantage is that long input blocks are prone to the effect of transmission errors. Furthermore, in some applications the encoding and decoding are too complicated to be done in real time. On the other hand, Tunstall variable-to-fixed length scheme requires searching a codebook to find the most probable input sequence for the next splitting. To circumvent these difficulties, Boncelet designed a simple divide and conquer scheme that we briefly describe next.

In its simplest form – to which we restrict ourselves – Boncelet builds a parsing tree by splitting a fixed number n of leaves (codewords) into subtrees of predetermined number of leaves. The number of leaves in each subtree is proportional to the probability of the alphabet symbols. For example, for a binary alphabet with probabilities p_1 and $p_2 = 1 - p_1$ the expected phrase length $d(n)$ satisfies the following recurrence (other parameters such as variance, generating function of the phrase length fulfill similar recurrences)

$$d(n) = 1 + p_1 d\left(\lfloor p_1 n + \delta \rfloor\right) + p_2 d\left(\lceil p_2 n - \delta \rceil\right)$$

where is δ is a constant. This equation is an example of a general *discrete divide and conquer recurrence* that we studied extensively in [11]. We shall adopt it here in order to present a comprehensive analysis of the Boncelet's algorithm performance including its redundancy and limiting distribution of the phrase length.

A question arises how the Boncelet algorithm compares to the (asymptotically) optimal Tunstall algorithm. In Theorem 1 and Corollary 1 we provide an answer by first computing the redundancy of the Boncelet scheme (i.e., the excess of code length over the optimal code length) and compare it to the redundancy of the Tunstall code. Then in Theorem 2 we also prove that the phrase length of Boncelet's scheme obeys the central limit law, as the Tunstall algorithm [10].

Literature on Boncelet's algorithm and *discrete* divide and conquer recurrences is very scarce. To the best of our knowledge, there is no precise redundancy analysis for the Boncelet's algorithm. In [3] some bounds on the average phrase length are derived. The Central Limit Law for the phrase length presented in Theorem 2 is new, too. Furthermore, we believe our contribution goes beyond analyzing precisely Boncelet's algorithm performance. We accomplish it by developing a methodology for solving general *discrete* divide and conquer recurrences (cf. [11]). The literature on *continuous* divide and conquer recurrence is very extensive [1], [6], [5], however, the discrete version of the recurrence has received much less attention. Flajolet and Golin [13] and Cheung et al. [4] use similar techniques to ours, however, their recurrences are much simpler and restricted to $p_1 = 0.5$ (see also [12], [16]). We apply a combination of methods such as Tauberian theorems and Mellin-Perron techniques.

II. Main Results

Let us start with a succinct description of the Boncelet algorithm in terms of its parsing tree. We consider a memoryless source over a general alphabet $\mathcal{A}$ of size m with probabilities of symbols denoted as p_i for $i = 1, \ldots, m$.

For fixed n (representing the number of leaves in the parsing tree and hence also the number of distinct phrases or codewords), the algorithm in each step creates m subtrees of predetermined number, n_i, of leaves (phrases). This continues recursively until less than m leaves are left. For example, for a binary alphabet, the root n is split into two subtrees with the number of leaves, respectively, equal to $n_1 = \lfloor p_1 n + \delta \rfloor$ and $n_2 = \lceil p_2 n - \delta \rceil$ for some $\delta \in (0, 1)$ that satisfies $2p_1 + \delta < 2$.

Let $\{v_1, \ldots v_n\}$ denote the phrases of the Boncelet code that correspond to the paths from the root to leaves of the parsing tree, and let $\ell(v_1), \ldots, \ell(v_n)$ be the corresponding phrase lengths. Furthermore, if $(i_1, i_2, \ldots, i_{\ell(v_k)})$ (with $i_j \in \{1, \ldots, m\}$) encodes the path from the root to phrase v_k we set $P(v_k) = p_{i_1} p_{i_2} \cdots p_{i_{\ell(v_k)}}$. Then $P(v_1), \ldots, P(v_n)$ sum up to 1 and represent a probability distribution on the phrases that corresponds to the distribution of phrases for a memoryless source. We denote by D_n the length of a phrase corresponding to the probability distribution P, that is, $\mathbb{P}[D_n = \ell(v_k)] = P(v_k)$. Its probability generating function is defined as $C(n, y) = \mathbb{E}\, y^{D_n} = \sum_{j=1}^{n} P(v_j) y^{\ell(v_j)}$. For a binary alphabet, the Boncelet splitting procedure leads to the following recurrence on $C(n, y)$ for $n \geq 2$

$$C(n, y) = p_1\, y\, C\left(\lfloor p_1 n + \delta \rfloor, y\right) + p_2\, y\, C\left(\lceil p_2 n - \delta \rceil, y\right) \quad (1)$$

with initial conditions $C(0, y) = 0$ and $C(1, y) = 1$ and some δ. Then the average phrase length, $d(n)$, defined as $\mathbb{E}\, D_n := d(n) = \sum_{j=1}^{n} P(v_j)\, \ell(v_j) = C'(n, 1)$ satisfies the following recurrence

$$d(n) = 1 + p_1 d\left(\lfloor p_1 n + \delta \rfloor\right) + p_2 d\left(\lceil p_2 n - \delta \rceil\right) \quad (2)$$

with $d(0) = d(1) = 0$. In general, for an m-ary alphabet recurrence (2) becomes

$$C(n, y) = y \sum_{i=1}^{m} p_i C([p_i n + \delta_i], y) \quad (3)$$

where $[x]$ is the quantized value of x; in our case it is replaced either by the floor function or the ceiling function.

These recurrences (1)–(3) are special cases of a general divide and conquer recurrence of the following form: For $m \geq 1$, let $p_1, \ldots, p_m$, $b_1, \ldots, b_m$ and $b'_1, \ldots, b'_m$ be positive real numbers such that $p_j < 1$ for $1 \leq j \leq m$. Then given $T(0) \leq T(1)$ for $n \geq 2$ we set

$$T(n) = a_n + \sum_{j=1}^{m} b_j T\left(\lfloor p_j n + \delta_j \rfloor\right) + \sum_{j=1}^{m} b'_j T\left(\lceil p_j n + \delta'_j \rceil\right) \quad (4)$$

where $(a_n)_{n \geq 2}$ is a known *non-negative* and *non-decreasing* sequence. We also assume that $2p_j + \delta_j < 2$ and $2p_j + \delta'_j \leq 1$ (for $1 \leq j \leq m$). In the next section we present in Theorem 3 a general solution of (4) as proved in [11]; we note that it's proof requires powerful tools of analytic combinatorics such as Dirichlet series [2], [24] and complex asymptotics [24].

Our first result concerns the average redundancy of Boncelet's algorithm. To present it succinctly, we need to introduce some properties of p_i.

Definition 1: We say that $\log(1/p_1), \ldots, \log(1/p_m)$ are *rationally related* if there exists a positive real number L such that $\log(1/p_1), \ldots, \log(1/p_m)$ are integer multiples of L, that is, $\log(1/p_j) = n_j L$, $n_j \in \mathbb{Z}$, $(1 \leq j \leq m)$ where $\gcd(n_1, \ldots, n_m) = 1$. Similarly, we say that $\log(1/p_1), \ldots, \log(1/p_m)$ are *irrationally related* if they are not rationally related.

Example. If $m = 1$, then we are always in the rationally related case. In the binary case $m = 2$, the numbers $\log(1/p_1)$, $\log(1/p_2)$ are rationally related if and only if the ratio $\log(1/p_1)/\log(1/p_2)$ is rational.

Theorem 1: Consider an m-ary memoryless source with positive probabilities $p_i > 0$ and the entropy rate $H = \sum_{i=1}^{m} p_i \log(1/p_i)$. Let $d(n) = \mathbb{E}\, D_n$ denote the expected phrase length of the binary Boncelet code.

(i) If $\log(1/p_1), \ldots \log(1/p_m)$ are irrationally related, then

$$d(n) = \frac{1}{H} \log n - \frac{\alpha}{H} + o(1), \quad (5)$$

where

$$\alpha = \overline{E}'(0) - \overline{G}'(0) - H - \frac{H_2}{2H}, \quad (6)$$

$H_2 = \sum_{i=1}^{m} p_i \log^2 p_i$, and $\overline{E}'(0)$ and $\overline{G}'(0)$ are the derivatives at $s = 0$ of the Dirichlet series defined in Section III.A (for a binary alphabet).

(ii) If $\log(1/p_1), \ldots \log(1/p_m)$ are rationally related, then

$$d(n) = \frac{1}{H} \log n - \frac{\alpha + \Psi(\log n)}{H} + O(n^{-\eta}) \quad (7)$$

for some $\eta > 0$, where $\Psi(t)$ is a periodic function of bounded variation that has usually an infinite number of discontinuities.

For practical data compression algorithms, it is important to achieve low redundancy defined as the excess of the code length over the optimal code length nH. For variable-to-fixed codes, the average redundancy is expressed as [10], [22]

$$R_n = \frac{\log n}{\mathbb{E}\, D_n} - H = \frac{\log n}{d(n)} - H.$$

Our previous results imply immediately the following.

Corollary 1: Let R_n denote the redundancy of the Boncelet code.

(i) If $\log(1/p_1), \ldots \log(1/p_m)$ are irrationally related, then

$$R_n = \frac{H\alpha}{\log n} + o\left(\frac{1}{\log n}\right) \quad (8)$$

with α defined in (6).

(ii) If $\log(1/p_1), \ldots \log(1/p_m)$ are rationally related, then

$$R_n = \frac{H(\alpha + \Psi(\log n))}{\log n} + o\left(\frac{1}{\log n}\right) \quad (9)$$

where $\Psi(t)$ is a periodic function of bounded variation.

We should compare the redundancy of Boncelet's algorithm to asymptotically optimal Tunstall algorithm. From [10], [22] we know that the redundancy of the Tunstall code is

$$R_n^T = \frac{H}{\log n}\left(-\log H - \frac{H_2}{2H}\right) + o\left(\frac{1}{\log n}\right)$$

for irrational case; in the rational case there is also a periodic term in the leading asymptotics.

Example. Consider $p = 1/3$ and $q = 2/3$. Then one computes

$$\alpha = \sum_{m\ge 1} \frac{d(m+2) - d(m+1)}{3}\left(\log\left\lceil 3m + \frac{5}{2}\right\rceil - \log(3m)\right)$$
$$+ 2\sum_{m\ge 1} \frac{d(m+2) - d(m+1)}{3}\left(\log\left\lfloor \frac{3}{2}m + \frac{5}{4}\right\rfloor - \log(\frac{3m}{2})\right)$$
$$+ \frac{\log 3}{3} - H - \frac{H_2}{2H} \approx 0.0518$$

while for the Tunstall code $-\log H - \frac{H_2}{2H} \approx 0.0496$.

Finally, we deal with the limiting distribution of the phrase length D_n. The proof is presented in the next section.

Theorem 2: Consider a memoryless source generating a sequence of length n parsed by the Boncelet algorithm. If $(p_1, \dots, p_m)$ is not uniformly distributed, then the phrase length D_n satisfies the central limit law, that is,

$$\frac{D_n - \frac{1}{H}\log n}{\sqrt{\left(\frac{H_2}{H^3} - \frac{1}{H}\right)\log n}} \to N(0,1),$$

where $N(0,1)$ denotes the standard normal distribution, and

$$\mathbb{E}\, D_n = \frac{\log n}{H} + O(1), \qquad \mathbb{V}\mathrm{ar}\, D_n \sim \left(\frac{H_2}{H^3} - \frac{1}{H}\right)\log n$$

for $n \to \infty$.

III. Analysis and Asymptotics

We first present a general solution to our general discrete divide and conquer recurrence (4). We use analytic tools, in particular Dirichlet series. For our purpose, we define the following Dirichlet series

$$\widetilde{T}(s) = \sum_{n=1}^{\infty} \frac{T(n+2) - T(n+1)}{n^s}, \quad \widetilde{A}(s) = \sum_{n=1}^{\infty} \frac{a_{n+2} - a_{n+1}}{n^s}.$$

Recall that a_n of (4) is non-negative and non-decreasing and we also assume that $b_j \ge 0$ and $b'_j \ge 0$. If the sequence a_n is constant (for $n \ge n_0$) we set $\sigma_a = -\infty$. Otherwise we set $\sigma_a = \inf\{\sigma : a_n = O(n^\sigma)\}$. Then σ_a is the the abscissa of absolute convergence σ_a of $\widetilde{A}(s)$. Furthermore, let s_0 be the unique real solution of the equation

$$\sum_{j=1}^{m} (b_j + b'_j)\, p_j^s = 1. \tag{10}$$

By using the Arka-Bazzi theorem [1] it follows that $T(n) = O(n^{\max\{s_0,\sigma_a\}+\varepsilon})$ for every $\varepsilon > 0$. This means that the Dirichlet series $\widetilde{T}(s)$ converges for $\Re(s) > \max\{s_0, \sigma_a\}$. We will prove below that we actually have a representation of the form (15) for some entire function $\overline{G}(s)$ and some analytic function $\overline{E}(s)$ that is analytic for $\Re(s) > \max\{s_0, \sigma_a\} - 1$. For the precise asymptotic analysis, we appeal to the Tauberian theorem by Wiener-Ikehara [7], [19], and an analysis based on the Mellin-Perron formula [2], [24]. Both approaches rely on the singular behavior of $\widetilde{T}(s)$. From this representation it is clear that the asymptotic behavior of $T(n)$ will depend on the singular behavior of $\widetilde{A}(s)$ and the roots of (10) (that include s_0).

Actually, we have to deal with three different situations. If $\sigma_a < s_0$, then the asymptotics of $T(n)$ is driven by the recurrence; in the case $\sigma_a = s_0$ there is an interaction between the internal structure of the recurrence and the sequence a_n (resonance); and in the case $\sigma_a > s_0$ the asymptotic behavior of a_n dominates.

In [11] we proved a Master Theorem for our general discrete divide and conquer recurrence that we state below in slightly simplified form.

*Theorem 3 (*Discrete Master Theorem): Let $T(n)$ be the divide and conquer recurrence defined in (4), where b_j and b'_j are non-negative with $b_j + b'_j > 0$ and the sequence $(a_n)_{n\ge 2}$ is non-negative and non-decreasing. Let σ_a denote the abscissa of absolute convergence of the Dirichlet series $\widetilde{A}(s)$ and s_0 the real root of (10). If $\sigma_a \ge s_0 \ge 0$ assume further that a_n is nondecreasing sequence given by $a_n = Cn^\sigma(\log n)^\alpha$ with $C > 0$ (that is, $\sigma_a = \sigma$).
(i) If $\log(1/p_1), \dots, \log(1/p_m)$ are irrationally related, then $T(n)$ becomes as $n \to \infty$

$$\begin{cases} C_1 + o(1) & \text{if } \sigma_a < 0 \text{ and } s_0 < 0, \\ C_2 \log n + C'_2 + o(1) & \text{if } \sigma_a < s_0 \text{ and } s_0 = 0, \\ C_3 (\log n)^{\alpha+1} \cdot (1 + o(1)) & \text{if } \sigma_a = s_0 = 0, \\ C_4\, n^{s_0} \cdot (1 + o(1)) & \text{if } \sigma_a < s_0 \text{ and } s_0 > 0, \\ C_5 n^{s_0} (\log n)^{\alpha+1} \cdot (1 + o(1)) & \text{if } \sigma_a = s_0 > 0 \text{ and } \alpha \ne -1, \\ C_5 n^{s_0} \log\log n \cdot (1 + o(1)) & \text{if } \sigma_a = s_0 > 0 \text{ and } \alpha = -1, \\ C_6\, (\log n)^\alpha (1 + o(1)) & \text{if } \sigma_a = 0 \text{ and } s_0 < 0, \\ C_7\, n^{\sigma_a} (\log n)^\alpha \cdot (1 + o(1)) & \text{if } \sigma_a > s_0 \text{ and } \sigma_a > 0, \end{cases} \tag{11}$$

where the explicitly computable constants $C_1, C_2, C_3, C_4, C_5, C_6, C_7$ are positive and C'_2 is real.
(ii) If $\log(1/p_1), \dots, \log(1/p_m)$ are rationally related, then $T(n)$ behaves as in the irrationally related case with the following two exceptions:

$$\begin{cases} C_2 \log n + \Psi_2(\log n) + o(1) & \text{if } \sigma_a < s_0 \text{ and } s_0 = 0, \\ \Psi_4(\log n)\, n^{s_0} \cdot (1 + o(1)) & \text{if } \sigma_a < s_0 \text{ and } s_0 > 0, \end{cases} \tag{12}$$

where C_2 is positive and $\Psi_2(t), \Psi_4(t)$ are periodic functions with period L (with usually countably many discontinuities).

We now briefly summarize the main steps to establish Theorem 3 and then provide a proof of Theorem 2.

A. Sketch of Proof of Theorem 3

We first apply the recurrence relation (4) to find the Dirichlet series $\widetilde{T}(s)$. To simplify our presentation, we assume that $b'_j = 0$, that it, we consider only the floor function on the right hand side of the recurrence (4). We thus obtain

$$\widetilde{T}(s) = \widetilde{A}(s) + \\ + \sum_{j=1}^{m} b_j \sum_{n=1}^{\infty} \frac{T\left(\lfloor p_j(n+2)+\delta_j \rfloor\right) - T\left(\lfloor p_j(n+1)+\delta_j \rfloor\right)}{n^s}.$$

Let

$$n = \left\lfloor \frac{k+2-\delta_j}{p_j} \right\rfloor - 2$$

for some integer k. For this k we have $\lfloor p_j(n+1)+\delta_j \rfloor = k+1$ and $\lfloor p_j(n+2)+\delta_j \rfloor = k+2$. For later use we split between $k \le 0$ and $k \ge 1$. Hence, setting

$$G_j(s) = \sum_{3p_j+\delta_j-2 \le k \le 0} \frac{T(k+2)-T(k+1)}{\left(\left\lfloor \frac{k+2-\delta_j}{p_j} \right\rfloor - 2\right)^s}$$

we obtain

$$\sum_{n=1}^{\infty} \frac{T\left(\lfloor p_j(n+2)+\delta_j \rfloor\right) - T\left(\lfloor p_j(n+1)+\delta_j \rfloor\right)}{n^s} \\ = G_j(s) + \sum_{k=1}^{\infty} \frac{T(k+2)-T(k+1)}{\left(\left\lfloor \frac{k+2-\delta_j}{p_j} \right\rfloor - 2\right)^s}.$$

We now compare the last sum to $p_j^s \widetilde{T}(s)$ and obtain

$$\sum_{k=1}^{\infty} \frac{T(k+2)-T(k+1)}{\left(\left\lfloor \frac{k+2-\delta_j}{p_j} \right\rfloor - 2\right)^s} = \sum_{k=1}^{\infty} \frac{T(k+2)-T(k+1)}{(k/p_j)^s} \\ = p_j^s \widetilde{T}(s) - E_j(s),$$

where

$$E_j(s) = \sum_{k=1}^{\infty} (T(k+2)-T(k+1)) \\ \times \left(\frac{1}{(k/p_j)^s} - \frac{1}{\left(\left\lfloor \frac{k+2-\delta_j}{p_j} \right\rfloor - 2\right)^s} \right). \quad (13)$$

Defining $E(s) = \sum_{j=1}^m b_j E_j(s)$ and $G(s) = \sum_{j=1}^m b_j G_j(s)$ we finally obtain the relation

$$\widetilde{T}(s) = \frac{\widetilde{A}(s) + G(s) - E(s)}{1 - \sum_{j=1}^m b_j\, p_j^s}. \quad (14)$$

The same procedure applies if some of the b'_j are positive leading to

$$\widetilde{T}(s) = \frac{\widetilde{A}(s) + \overline{G}(s) - \overline{E}(s)}{1 - \sum_{j=1}^m (b_j + b'_j)\, p_j^s}, \quad (15)$$

with a slightly modified functions $\overline{G}(s)$ and $\overline{E}(s)$.

By our previous assumptions, we know the analytic behaviors of $\widetilde{A}(s)$ and $\left(1 - \sum_{j=1}^m b_j\, p_j^s\right)^{-1}$: $\widetilde{A}(s)$ has a pole-like singularity at $s = \sigma_a$ (if $\sigma_a \ge s_0$) and a proper continuation to a complex domain that contains the (punctuated) line $\Re(s) = \sigma_a$, $s \ne \sigma_a$, as discussed in [11]. On the other hand, $(1 - \sum_{j=1}^m b_j\, p_j^s)^{-1}$ has a polar singularity at $s = s_0$ (and infinitely many other poles on the line $\Re(s) = s_0$ if the numbers $\log(1/p_j)$ are rationally related), and also a meromorphic continuation to a complex domain that contains the line $\Re(s) = s_0$. Heuristically, the asymptotic behavior (of the partial sums) of the coefficients of $\widetilde{T}(s)$ is reflected by the singular behavior of $\widetilde{T}(s)$. Recall that $T(n) = O(n^\sigma)$ implies that the series $\widetilde{T}(s)$ converges for $\Re(s) > \sigma$. Hence, if $s = \sigma$ is a singularity of $\widetilde{T}(s)$, then we expect that $T(n)$ behaves (more or less) like n^σ. Actually there is a very precise correspondence by Tauberian theorems (of Wiener-Ikehara and Delange, see [7], [11], [19]) if σ is the only singularity on the line $\Re(s) = \sigma$. Hence, Tauberian theorems can be applied if the $\log(1/p_j)$ are irrationally related. In the rationally related case the problem is more subtle but can be handled with the help of the Mellin-Perron formula stated next (Theorem 4).

In our formulation we use Iverson's notation $[\![P]\!]$ which is 1 if P is a true proposition and 0 else.

Theorem 4 (see [2]): For a sequence $c(n)$ define the Dirichlet series $C(s) = \sum_{n=1}^{\infty} c(n) n^{-s}$ and assume that abscissa of absolute convergence σ_a is finite or $-\infty$. Then for all $\sigma > \sigma_a$ and all $x > 0$

$$\sum_{n<x} c(n) + \frac{c(\lfloor x \rfloor)}{2} [\![x \in \mathbb{Z}]\!] = \lim_{T\to\infty} \frac{1}{2\pi i} \int_{\sigma-iT}^{\sigma+iT} C(s) \frac{x^s}{s}\, ds.$$

Note that the Mellin-Perron formula enables us to obtain precise information about the function $\overline{c}(v) = \sum_{n \ge v} c(n)$ if we know the behavior of $\frac{1}{s}C(s)$. In our context we have $c(n) = T(n+2) - T(n)$, that is,

$$T(n) = T(2) + \lim_{T\to\infty} \frac{1}{2\pi i} \int_{c-iT}^{c+iT} \widetilde{T}(s) \frac{(n-\frac{3}{2})^s}{s}\, ds \quad (16)$$

where $\widetilde{T}(s)$ is given by (15). Informally, one shifts the line of integration to the left and collects the contributions from the residues of the (polar) singularities at $s = \sigma_a$, $s = s_0$ and $s = 0$; if the $\log(1/p_j)$ are rationally related there are infinitely many polar singularities on the line $\Re(s) = s_0$ that contribute to the periodic term $\Psi(t)$. Details can be found in [11].

B. Proof of Theorem 2

Finally we indicate the proof of Theorem 2 for the non-symmetric binary case. For simplicity, we shall write p for p_1 and q for $p_2 = 1 - p \ne p_1$.

We recall that $C(n, y)$ satisfies the recurrence (1) with initial conditions $C(0, y) = 0$ and $C(1, y) = 1$. It is clear that for every fixed positive real number y we can apply Theorem 3. However, we have to be careful since we need an asymptotic representation for $C(n, y)$ uniformly for y in an interval that contains 1 in its interior. Note that $C(n, 1) = 1$.

For the proof of Theorem 2, one has to consider the Dirichlet series

$$C(s,y) = \sum_{n=1}^{\infty} \frac{C(n+2,y) - C(n+1,y)}{n^s}.$$

For simplicity we just consider here the case $y > 1$. Then $C(s,y)$ converges for $\Re(s) > s_0(y)$, where $s_0(y)$ denotes the real zero of the equation $y(p^{s+1} + q^{s+1}) = 1$. We find

$$C(s,y) = \frac{(y-1) - \tilde{E}(s,y)}{1 - y(p^{s+1} + q^{s+1})},$$

where $E(s,y)$ converges for $\Re(s) > s_0(y) - 1$ and satisfies $\tilde{E}(0,y) = 0$ and $\tilde{E}(s,1) = 0$.

Then by the Wiener-Ikehara theorem only the residue at $s_0(y)$ contributes to the main asymptotic leading term. (Recall that we just consider the case $y > 1$ and the irrationally related case). We thus have

$$C(n,y) \sim \operatorname{Res}\left(\frac{((y-1) - \tilde{E}(s,y))(n-3/2)^s}{s(1 - y(p^{s+1} + q^{s+1}))}; s = s_0(y)\right)$$

$$= \frac{((y-1) - \tilde{E}(s_0(y),y))(n-3/2)^{s_0(y)}}{-s_0(y)(\log(p)p^{s_0(y)+1} + \log(q)q^{s_0(y)+1}))}(1 + o(1)).$$

The essential but non-trivial observation is that this asymptotic relation holds uniform for y in an interval around 1. In order to make this precise we can use the Mellin-Perron formula from Theorem 4

$$C(n,y) = C(2,y) + \frac{1}{2\pi i}\int_{c-i\infty}^{c+i\infty} C(s,y)\frac{\left(n - \frac{3}{2}\right)^s}{s}\,ds$$

and apply the methods presented in [11] which can be made uniform in y; this works for the irrationally related case as well as for the rationally related case. Hence we find (in all cases)

$$C(n,y) = (1 + O(y-1))n^{s_0(y)}(1 + o(1))$$

uniformly for real y that are contained in an interval around 1; note that the case $y \le 1$ can be handled similarly and leads to the same result. Finally by using the local expansion

$$s_0(y) = \frac{y-1}{H} + \left(\frac{H_2}{2H^3} - \frac{1}{H}\right)(y-1)^2 + O((y-1)^3), \quad (17)$$

and by setting $y = e^{t/(\log n)^{1/2}}$ we obtain

$$n^{s_0(y)} = \exp\left(\frac{1}{H}t\sqrt{\log n} + \left(\frac{H_2}{H^3} - \frac{1}{H}\right)\frac{t^2}{2} + O(t^3/\sqrt{\log n})\right)$$

and consequently

$$\mathbb{E}\left[e^{D_n t/\sqrt{\log n}}\right] = C\left(n, e^{t/\sqrt{\log n}}\right) =$$

$$\exp\left(\frac{1}{H}t\sqrt{\log n} + \left(\frac{H_2}{H^3} - \frac{1}{H}\right)\frac{t^2}{2}\right)(1 + o(1)).$$

Hence, we arrive at

$$\mathbb{E}\left[e^{t(D_n - \frac{1}{H}\log n)/\sqrt{\log n}}\right] = e^{-(t/H)\sqrt{\log v}}\mathbb{E}\left[e^{D_n t/\sqrt{\log n}}\right] \quad (18)$$

$$= e^{\frac{t^2}{2}\left(\frac{H_2}{H^3} - \frac{1}{H}\right)} + o(1).$$

By the convergence theorem for the Laplace transform or Goncharov theorem (see [24]) this proves the normal limiting distribution as $n \to \infty$ and also convergence of (centralized) moments.

ACKNOWLEDGMENT

This work was supported in part by the Austrian Science Foundation FWF Grant No. S9604, by NSF Science and Technology Center for Science of Information Grant CCF-0939370, NSF Grants DMS-0800568, and CCF-0830140, NSA Grant H98230-11-1-0184, and the MNSW grant N206 369739.

REFERENCES

[1] M. Akra and L. Bazzi, On the Solution of Linear Recurrence Equations, *Computational Optimization and Applications*, 10,195-201, 1998.
[2] T. M. Apostol, Introduction to analytic number theory, Undergraduate Texts in Mathematics, Springer, New York, 1976.
[3] C. G. Boncelet, Block arithmetic coding for source compression, *IEEE Trans. Inform. Theory*, IT-39, no.5, pp.1546-1554, 1993.
[4] Y.K. Cheung, P. Flajolet, M. Golin, and C.Y. Lee, Multidimensional Divide-and-Conquer and Weighted Digital Sums, *ANALCO*, 2008.
[5] V. Choi and M. J. Golin, Lopsided trees. I. Analyses. *Algorithmica*, 31, 3, 240-290, 2001.
[6] T. Cormen, C. Leierson, and R. Rivest, *Introduction to Algorithms*, MIT Press, Cambridge, Mass., 1990.
[7] H. Delange, Généralisation du théorème de Ikehara, *Ann. Sci. Éc. Norm. Supér.*, 71, 213–242, 1954.
[8] H. Delange, Sur la fonction sommatoire de la fonction "Somme des Chiffres." *Enseignement Math.* 21, 31–47, 1975.
[9] M. Drmota, H-K. Hwang, and W. Szpankowski, Precise Average Redundancy of an Idealized Arithmetic Coding, *Data Compression Conference*, 222-231, Snowbirds, 2002.
[10] M. Drmota, Y. Reznik, and W. Szpankowski, Tunstall Code, Khodak Variations, and Random Walks *IEEE Trans. Information Theory*, 56, 2010.
[11] M. Drmota and W. Szpankowski, A Master Theorem for Discrete Divide and Conquer Recurrence, *Proc. SODA'11*, San Francisco, 2011.
[12] P. Erdos, A. Hildebrand, A. Odlyzko, P. Pudaite, and B. Reznick, The Asymptotic Behavior of a Family of Sequences, *Pacific Journal of Mathematics*, 126, 227-241, 1987.
[13] P. Flajolet, and M. Golin, Mellin Transforms and Asymptotics: The Mergesort Recurrence, *Acta Informatica*, 31, 673–696, 1994.
[14] P. Flajolet, X. Gourdon, and P. Dumas, Mellin Transforms and Asymptotics: Harmonic sums, *Theoretical Computer Science*, 144, 3–58, 1995.
[15] P. Flajolet and R. Sedgewick, *Analytic Combinatorics*, Cambridge University Press, Cambridge, 2008.
[16] H-K. Hwang, Distribution of the number of factors in random ordered factorizations of integers, *Journal of Number Theory* ,81, 61–92, 2000.
[17] G. L. Khodak, Connection Between Redundancy and Average Delay of Fixed-Length Coding, *All-Union Conference on Problems of Theoretical Cybernetics* (Novosibirsk, USSR, 1969) 12 (in Russian)
[18] D.E. Knuth, *The Art of Computer Programming, Vol. 2: Seminumerical Algorithms*, 3rd ed. Reading, MA: Addison-Wesley, 1998.
[19] J. Korevaar, A Century of complex Tauberian theory, *Bull. Amer. Math. Soc.* 39 (2002), 475–531.
[20] G. Langdon, An Introduction to Arithmetic Coding, *IBM. J. Res. Develop.*, 28, 135-149, 1984.
[21] J. Rissanen and G. Langdon, Arithmetic Coding, *IBM. J. Res. Develop.*, 23, 149-162, 1979.
[22] S. A. Savari and Robert G. Gallager; Generalized Tunstall codes for sources with memory, IEEE Trans. Info. Theory, vol. IT-43, pp. 658 - 668, March 1997.
[23] R. Sedgewick, and P. Flajolet, *An Introduction to the Analysis of Algorithms*, Addison-Wesley, Reading, MA, 1995.
[24] W. Szpankowski, Average Case Analysis of Algorithms on Sequences, Wiley, New York, 2001.
[25] B. P. Tunstall, Synthesis of Noiseless Compression Codes, Ph.D. dissertation, (Georgia Inst. Tech., Atlanta, GA, 1967)