How to Count Quickly and Accurately:

A Unified Analysis of Probabilistic Counting and Other Related Problems

Peter KIRSCHENHOFER* Helmut PRODINGER*
Technical University of Vienna

Wojciech SZPANKOWSKI†
Purdue University, West Lafayette

Abstract. *We consider a class of probabilistic counting algorithms parameterized by an integer $d \geq 0$ that estimate the number of elements N in a large set. Our algorithms generalize an idea of Flajolet and Martin who limited themselves to the case $d = 0$. As noted by Brassard and Bratley "it is far from obvious how to carry out a more precise analysis of the unbiased estimate of N ...". We present a novel and complete analysis of these new counting algorithms that – to the best of our knowledge – cannot be obtained by an extension of the analysis by Flajolet and Martin. We present results concerning the average value, the variance and the limiting generating function of an estimate of N. Moreover, our novel approach is not limited to probabilistic counting algorithms, and it can be applied in the investigation of several other "splitting algorithms" such as selecting the loser within a group of people, estimating the number of questions necessary to identify the number of distinct objects, searching algorithms based on digital tries, approximate counting, electing d finalists in a contest (cf. polling system), and so forth.*

1 Introduction

Flajolet and Martin in their paper [2] proposed an algorithm in order to estimate the cardinality of large (multi)sets of data. This algorithm is probabilistic in nature and works as follows. In order to estimate the cardinality N of a (multi)set $\mathcal{M}$ every element $x \in \mathcal{M}$ is hashed into a binary string of size m (the choice of m is easy, and $m = 5 + \log N$ suffices [1]). The bitwise OR-composition of modified hashed strings is used to build the so called *bitmap* and to obtain the estimate R_N of the

*Department of Algebra and Discrete Mathematics, Technical University of Vienna, Wiedner Hauptstrasse 8–10, A-1040 Vienna, Austria. This work was supported by Fonds zur Förderung der Wissenschaftlichen Forschung Project P7497-TEC

†Department of Computer Science, Purdue University, West Lafayette, IN 47907, U.S.A. This research was supported by the NSF Grant INT-8912631, and in part by AFOSR grant 90-0107, by the NSF grant CCR-8900305, and by grant R01 LM05118 from the National Library of Medicine.

cardinality N of $\mathcal{M}$. More precisely, the position of the leftmost zero in the *bitmap* string approximates $\log_2 N$ (for details see below and [2], [1]).

The original algorithm turns out to have the variance of the estimator R_N approximately to 1.257, which is rather large for practical implementations. Therefore a modification of the basic algorithm was also proposed in [2] but this new algorithm is more complex and costly.

In our paper, we propose for the instance of sets without replications a new estimate $R_{N,d}$ of N that is parameterized by an integer d. The value $d = 0$ corresponds to Flajolet-Martin's basic algorithm. Our algorithm has smaller variance (apart form fluctuations the value tending to $1/12 = 0.083$ as $d \to \infty$). The new idea behind our algorithm is to *count* the number of 1's up to $d+1$ in every position of the *bitmap*. As an estimate of N we propose to take the leftmost position that contains the value smaller than $d+1$. More formally, our algorithm works as follows. We assume that there exists a carefully implemented hashing function $hash: \mathcal{M} \to \{0,1\}^m$ where m is chosen as in [2]. Let y be the image of $x \in \mathcal{M}$ over the $hash(\cdot)$ function, that is, $y = hash(x)$. The procedure below constructs another string called *bitmap* that is used to estimate N. As in [2], we define the function $\rho(y)$ for a binary string y as the leftmost position that contains value 1.

procedure $BITMAP$
for $i = 0$ **to** $m-1$ **do** $bitmap(i) = 0$;
for $x \in \mathcal{M}$ **do**
  begin
  $y = hash(x)$;
  $index := \rho(hash(x))$
  $bitmap(index) = \min\{bitmap(index) + 1,\ d+1\}$;
  end;

Note that the resulting string *bitmap* is a $d+1$-ary string. For example, for $d = 3$ we may have $bitmap = 4444444444434100000$. As the estimate $R_{N,d}$ we take the position of the leftmost value smaller than $d+1$ (e.g., in the example above we have $R_{N,3} = 11$). More precisely we define:

$$R_{N,d} = \min\{k : \ bitmap(k) < d+1 \text{ and for all } 0 \le i < k \ \ bitmap(i) = d+1\} \ . \quad (1)$$

It is easy how our algorithm works. The *bitmap* string is a sequence of integers from the set $\{0, 1, \ldots, d+1\}$. If $bitmap(i) = k < d+1$, then exactly k strings y have at the ith position the leftmost one. Moreover, $bitmap(i) = d+1$ means that *at least* $d+1$ strings y contain the leftmost one at position i.

We now summarize our main results concerning the probabilistic behavior of $R_{N,d}$. In particular, we prove that $ER_{N,d} = \log_2 N - 1 - C_d + P(\log_2 N) + O(\sqrt{N})$ where C_d is a constant such that $C_d = \log_2 d - 0.5 + O(d^{-1/2+\varepsilon})$ as $d \to \infty$, and $P(x)$ is a periodic fluctuation of period 1. More importantly, we show that the variance of $R_{N,d}$ fulfills, neglecting periodic fluctuations, $\mathrm{var} R_{N,d} \to 1/12 = 0.08333\ldots$ as $d \to \infty$. Interestingly enough the quantity $1/12$ is not uncommon in such a situation and known as Sheppard's correction (cf. [9]) which says that the discretization by itself introduces some fluctuations with the variance equal to $1/12$. Finally, in the next section we make some comments concerning the distribution and the limiting generating function for $R_{N,d}$.

The reader might have noticed that for $d \geq 1$ our algorithm does not necessarily neglect multiplicities. To be more precise: Only multiplicities greater than d will be ignored by the algorithm. Therefore for data base applications a priori knowledge about the expected multiplicities of the different elements is necessary. In this paper we confine our analysis on the quantity $R_{N,d}$, which is a good estimator provided there are no multiplicities.

Literature on the probabilistic counting is rather scanty. It was initiated by the paper of Flajolet and Martin [2] who analyzed the case $d = 0$ of our algorithm. Their analysis is based on an application of the inclusion-exclusion rule, and – to the best of our knowledge – cannot be extended to $d > 0$. This is also evident from the analysis of Greenberg, Flajolet and Ladner [6] who analyzed a conflict resolution algorithm which resembles the case $d = 1$ of our analysis. A simplification of Flajolet-Martin's analysis is presented in Kirschenhofer and Prodinger [11]. Finally, a different estimator for the probabilistic counting algorithm was analyzed in Szpankowski and Rego [20] who investigated the maximum over N geometrically distributed random variables. The authors of [20] considered the *last* bit that is equal to one in the *bitmap*. Unfortunately, in this case the variance is even bigger and equal to 3.507.

It must be stressed that the novelity of our paper also lies in the area of the mathematical analysis of algorithms. In fact, our technique of analysis can be applied to several other algorithms. As shown in the section below, we reduce our probabilistic counting algorithm to a problem on *(digital) trees*. Evaluating a parameter of such a tree, we obtain the following functional equation

$$g(z) = \beta a(z/2) g(z/2) + b(z) \ . \tag{2}$$

In general, this equation does not have an explicit finite solution unless $a(z)$ is a very particular function. For example, $a(z) = e^z$ was discussed in Knuth [12] (cf. also [19]), and Szpankowski [18], [19] (cf. also [12]) investigated the case $a(z) = e^z + 1$. For these two cases explicit final solutions of (2) were obtained, but it is rather hopeless to expect the existence of a useful closed form solution for general $a(z)$ and $b(z)$. An asymptotic solution to (2) was first suggested in Jacquet and Szpankowski [8] who analyzed the case $a(z) = 1 + (1+z)e^{-z}$. Recently, Flajolet and Richmond [4] investigated asymptotically the above equation with $a(z) = 2z^b/(1+z)^b$ for some integer $b > 0$. Here, analyzing probabilistic counting algorithms, we must investigate (2) with $a(z) = 1 - e_d(z)e^{-z}$ where $e_d(z) = \sum_{i=0}^{d} z^i/i!$.

2 Main Results

In this section we present our main results. We start with another description of the probabilistic counting algorithm which will slightly differ from the one presented in Section 1, but it helps us to establish recurrence equations necessary for the analysis.

Our new description goes as follows. Let us consider an empty *bitmap* string, that is, with all positions filled by zeros. Assume that N objects (e.g., data, persons, etc.) can randomly insert (hit) a 1 at any position of the *bitmap*, however, probability of hitting the $j \geq 0$ position is equal to 2^{-j-1}. In terms of the probabilistic counting, this means that the probability of the occurrence of the pattern like $0^j 1 \cdots$ is equal to 2^{-j-1} since 0 and 1 are equally likely (0^j denotes j consecutive zeros). Every object can hit only one time. In addition, we count the number of hits in any position of the *bitmap*, but we count the number of hits only up to some value $d+1$, where d is a given parameter. In other words, the *bitmap* is a $d+1$-ary string. Clearly, the *bitmap* will contain a lot of $d+1$'s at the begining of the string. This is due to the fact that the probability of a hit decreases exponentially fast with the increase of the position in the *bitmap*. The parameter of interest is the length $R_{N,d}$ of the longest run of $d+1$'s in the front of the *bitmap*. A precise definition of $R_{N,d}$ is given in (1).

Let $F_N(z) = Ez^{R_{N,d}}$ be the probability generating function of the estimator $R_{N,d}$. That is, $[z^j]F_N(z)$ (the jth coefficient at $F_N(z)$) is the probability that the *bitmap* is $(d+1)^j c \ldots$ where $c < d$, and $(d+1)^j$ denotes j consecutive values "$d+1$". To derive the recurrence for $F_N(z)$, one may represent the action of our algorithm as a splitting process in a *digital tree*. Imagine N persons flipping a fair coin, and those who get 1 (a hit) move to the right while the rest to the left. The process continues until less than $d+1$ persons flip a 1. Imagine that at the first step k persons flip 1. Then, the contribution to the generating function $F_N(z)$ is as follows: if $k \leq d$ then we need to add $z^0 \sum_{k \leq d} \binom{N}{k} 2^{-N}$, otherwise the contribution becomes $z \sum_{d+1 \leq k \leq N} \binom{N}{k} 2^{-N} F_{N-k}(z)$. In short, we obtain the following recurrence

$$F_N(z) = \sum_{k=0}^{d} \binom{N}{k} 2^{-N} + z \sum_{k=0}^{N-d-1} \binom{N}{k} 2^{-N} F_k(z) \ , \tag{3}$$

which is valid for all $N \geq 0$.

Using (3) we can derive recurrences for the first and second moments of $R_{N,d}$, namely: $ER_{N,d} = F'_N(1)$ and var $R_{N,d} = F''_N(1) + F'_N(1) - (F'_N(1))^2$. In particular, let $L(z)$ and $W(z)$ be the exponential generating function for $F'_N(1)$ and $F''_N(1)$ respectively. That is, $L(z) = \sum_{N \geq 0} F'_N(1) z^N/N!$ and $W(z) = \sum_{N \geq 0} F''_N(1) z^N/N!$. Define also $\widetilde{L}(z) = e^{-z} L(z)$ and $\widetilde{W}(z) = e^{-z} W(z)$. Then, (3) implies

$$\widetilde{L}(z) = f_d(z/2)\widetilde{L}(z/2) + f_d(z/2) \tag{4}$$

$$\widetilde{W}(z) = f_d(z/2)\widetilde{W}(z/2) + 2\widetilde{L}(z) f_d(z/2) \tag{5}$$

where $f_d(z) = 1 - e_d(z)e^{-z}$ and $e_d(z) = 1 + z^1/1! + \cdots + z^d/d!$.

Both functional equations fall into the general equation (2) which we repeat below

$$g(z) = \beta a(z/2)g(z/2) + b(z) \ . \tag{6}$$

where $\beta \leq 1$. Iterating (6), we obtain a general solution in the form of (cf. [8])

$$g(z) = \sum_{n=0}^{\infty} \beta^n b(z2^{-n}) \prod_{k=1}^{n} a(z2^{-k}) \tag{7}$$

if $g(0) = 0$. Define

$$\varphi(z) = \prod_{j=0}^{\infty} a(z2^j) \ , \tag{8}$$

provided the infinite product in (8) converges. Then, the general solution (7) can be rewritten as

$$g(z)\varphi(z) = \sum_{n=0}^{\infty} \beta^n b(z2^{-n})\varphi(z2^{-n}) \ . \tag{9}$$

The last form is the clue to our asymptotic solution of the functional equation (7), and in particular to solve (4) and (5). In Section 3 we present our detailed derivations.

Now we are ready to present our main results.

Theorem 1 *Consider the probabilistic counting algorithms described above, and in particular the quantity $R_{N,d}$.*

(i) *The average value of $R_{N,d}$ becomes asymptotically as $N \to \infty$*

$$ER_{N,d} \sim \log_2 N - 1 - C_d + P_1(\log_2 N) + O(N^{-1/2+\varepsilon}) \tag{10}$$

where $P_1(x)$ is a periodic function with period 1 and mean zero, and with $L = \log 2$ we have

$$\begin{aligned} C_d &= \frac{1}{L^2}\int_0^\infty e^{-x} e_d(x)\varphi(2x)\frac{\log x}{x}dx \\ &\sim \log_2 d - 0.5 + O(d^{-1/2+\varepsilon}) \qquad as \quad d \to \infty \ . \end{aligned} \tag{11}$$

(ii) *The variance of $R_{N,d}$ is for $N \to \infty$*

$$\mathrm{var} R_{N,d} \sim D_d - C_d^2 + P_2(\log_2 N) + O(N^{-1/2+\varepsilon}),$$

where

$$D_d - C_d^2 \sim \frac{1}{12} + O(d^{-1/2+\varepsilon}) \qquad as \quad d \to \infty \ . \tag{12}$$

In particular

$$D_d = \frac{1}{L^3}\int_0^\infty e_d(x)e^{-x}\varphi(2x)\frac{\log^2 x}{x}dx$$
$$\sim -C_d - 1/6 + \log_2^2 d + O(d^{-1/2+\varepsilon}) \quad as \quad d\to\infty\ , \tag{13}$$

and $P_2(x)$ is a periodic function with period 1. ∎

Using numerical integration, we computed for the first few values of d the constants $ER_{N,d}-\log_2 N - P_1(\log_2 N)$ and var $R_{N,d}-P_2(\log_2 N)$. They are displayed in the table below. We note that the variance decreases substantially for initial values of d ($d\le 2$) and then the decrease slows down. From the table, one may conclude that $d\ge 5$ is a good practical choice.

Table 1: Performance of the algorithm for small values of d (neglecting fluctuations).

d	$ER_{N,d}-\log_2 N$	var $R_{N,d}$
0	-0.37	1.26
1	-1.40	0.78
2	-2.00	0.59
3	-2.42	0.49
4	-2.74	0.43
5	-3.02	0.38
6	-3.24	0.35
7	-3.43	0.32

Our technique also allows to deal with the limiting distribution for $R_{N,d}$ and the limiting generating function. For this, we introduce the Poisson transform of $F_N(z)$ as

$$G(z,x) = e^{-x}\sum_{N=0}^{\infty} F_N(z)\frac{x^N}{N!}\ . \tag{14}$$

Then, after some algebra, one proves that $H(z,x)=G(z,x)-1$ satisfies the following recurrence

$$H(z,x) = 2f_d(x/2)H(z,x/2) + (z-1)f_d(x/2)\ , \tag{15}$$

which falls into our general functional equation (6). In particular, as above we can write a solution for $H(z,x)$ as

$$\varphi(x)H(z,x) = (z-1)\sum_{n=0}^{\infty} z^n\varphi(x2^{-n-1})\ , \tag{16}$$

where $\varphi(x)$ is defined in (8).

With the help of (16), we are able to derive the limiting generating function for $F_N(z)$ as N becomes large.

Theorem 2 *Consider the probabilistic counting algorithm as in Theorem 1. For large N the limiting generating function becomes*

$$F_N(z) \sim z^{\log_2 N} \varphi^*(-\log_2 z)/L + P_3(\log_2 N) + O(N^{-1/2+\varepsilon}) \ . \tag{17}$$

where $\varphi^*(s)$ is the Mellin transform of $\varphi(x)$ defined in (8). ∎

In addition, in Section 3, we will present a conjecture (28) concerning the limiting distribution for $R_{N,d}$. Details and more precise statements are left to a journal version of the paper.

For the reader's convenience we want to point out that the definition and basic properties of the *Mellin integral transform* are included in the paper [17] by U. Schmid *in this volume.*

Remarks

(i) *Special Cases.* As mentioned before, the case $d = 0$ was analyzed by Flajolet and Martin [2]. In addition, Greenberg *et al.* [6] investigated the function $\varphi(x)$ for $d = 1$ in the analysis of a conflict resolution algorithm for multiple channels. It must be stressed that the analysis of Flajolet and Martin [2] is based on the inclusion-exclusion rule, and seems to be very unlikely extendable to $d \geq 1$. This is partially evident from the analysis in [6]. We note that our analysis is based on a quite different approach, namely, functional equations, and can be proved very successful for some other algorithms (cf. Remark (iii)).

(ii) *Convergence in Probability.* Theorem 1 implies that $R_{N,d}$ converges *in probability* (pr.) to $ER_{N,d}$. More precisely, by Theorem 1 and Chebyshev's inequality we have

$$\Pr\{|R_{N,d}/ER_{N,d} - 1| > \varepsilon\} \leq \frac{\mathrm{var} R_{N,d}}{\varepsilon^2 E^2 R_{N,d}} \sim \frac{const}{\log_2^2 N} \to 0$$

as $N \to \infty$. This proves the announced convergence. It is not clear, however, whether one can extend this convergence to the *almost sure* convergence.

(iii) *Other Problems.* As mentioned in the abstract, our technique might be applied to analyze several other algorithms that are based on the "splitting process" described at the beginning of this section. More precisely, a splitting algorithm divides randomly N objects according to some rule. A good representation of such a splitting process is a digital tree (i.e., a trie [12]). For example, consider a multiaccess system with a large number of users. The following "conflict resolution" algorithm found to be very successful for some multiaccess systems (cf. [6], [8]). Let N be the size of a conflict, that is, the number of users that simultaneously sent packets. To split it, every user flips a fair coin, and those who get 1 "move right" in a digital tree represented this process, while the others "move left". This splitting process continues until all users sent successfully their packets (i.e., subtrees containing conflicting users are of size one). This algorithm can be analyzed in a unified manner by our approach through the functional equation of type (6) and its solution (9). Here is another example that generalizes PATRICIA tries (cf. [12], [19]). Consider N infinite strings that are built over a V-ary alphabet. Using a trie construction as in Knuth [12] (see also [19]) we can split all strings in such a manner that no two of them will share the common prefix. A trie built in such a manner may contain some internal nodes that are unary. To avoid such a waste of storage, one may compress this trie to obtain a PATRICIA trie. In the PATRICIA all internal nodes have degree greater or equal to two. But a generalization of a PATRICIA trie might be also explored by imposing that all internal nodes have degree of size greater or equal to $d \leq V$. What is the search time (i.e., the length of a path to an external node) in such a trie? What about the average height, etc.? The analysis of such a data structure seems to be a nontrivial task, and our approach can be applied to obtain some asymptotics. Finally, several other problems fall into the "splitting process" scenario. We mention here: selecting the loser [15], estimating the number of questions necessary to identify the number of distinct objects [14], selecting d finalists in a contest where questions are answered randomly, and so forth.

3 Analysis

In this section we present a sketch of proof for Theorem 1. We also offer some remarks regarding Theorem 2.

We start with the average value $ER_{N,d}$. Its modified exponential generating function $\widetilde{L}(z)$ satisfied the functional equation (4) which we repeat below

$$\widetilde{L}(z) = f_d(z/2)\widetilde{L}(z/2) + f_d(z/2) \tag{18}$$

where $f_d(z) = 1 - e_d(z)e^{-z}$ and $e_d(z) = 1 + z^1/1! + \cdots + z^d/d!$. Iterating this equation, as in (6) – (9), we finally obtain

$$\widetilde{L}(z)\varphi(z) = \sum_{n=0}^{\infty} \varphi(z2^{-n-1}) \ , \tag{19}$$

where

$$\varphi(z) = \prod_{j=0}^{\infty} f_d(z2^j) \ , \tag{20}$$

or, equivalently,

$$\widetilde{L}(z) = \sum_{n=0}^{\infty} \prod_{j=1}^{n+1} f_d\left(z2^{-j}\right) .$$

Observe that, by d'Alembert's criterion, the final expression converges absolutely for all complex numbers z.

The idea of our analysis is to obtain an asymptotic behavior of $\widetilde{L}(z)$ as $z \to \infty$, and then by *depoissonization* [8] to evaluate the coefficients $ER_{N,d}$ of $L(z)$. To obtain the asymptotics of $\widetilde{L}(z)$ we apply the Mellin transform technique (cf. [5] and [3]). We shall use the following depoissonization lemma proved in Rais, Jacquet and Szpankowski [16]. (An older and less versatile version of the depoissonization lemma may be found in Jacquet and Szpankowski [8].)

Depoissonization Lemma *Let S_θ be a cone $S_\theta = \{z : \ |\arg z| < \theta \ , \ 0 < \theta < \pi/2\}$. If for $z \in S_\theta$ and $z \to \infty$ the following holds for all u in a compact set $\mathcal{U}$*

$$|D(z,u)| < \beta_1 |z|^\varepsilon \tag{21}$$

for some $\beta_1, \varepsilon > 0$, and for $z \notin S_\theta$ and all $u \in \mathcal{U}$

$$|D(z,u)e^z| < \beta_2 |z|^\varepsilon e^{\alpha|z|} \tag{22}$$

for some $0 < \alpha < 1$ and a constant $\beta_2 > 0$, then for large n uniformly in $u \in \mathcal{U}$ the generating function $D_n(u) = [z^n/n!]D(z,u)e^z$ satisfies

$$D_n(u) = D(n,u) + O(n^{-1/2+\varepsilon}) \tag{23}$$

with $\varepsilon < 1/2$. ∎

Using the above idea, we first apply the Mellin transform to (19). Let $Q_1(z) = \widetilde{L}(z)\varphi(z)$ and $Q_1^*(s)$ be its Mellin transform. Since the sum in (19) is a harmonic sum, we have (cf. [5])

$$Q_1^*(s) = \frac{2^s}{1-2^s}\varphi^*(s) = \left(\frac{2^s}{1-2^s}\right)^2 \Phi^*(s) \tag{24}$$

where $\varphi^*(s)$ is the Mellin transform of $\varphi(z)$, and $\varphi^*(s) = \Phi^*(s)/(2^{-s}-1)$. Note that $\Phi^*(s)$ is an entire function (cf. [6]). Using the inverse of the Mellin transform (by computing the residues of $Q_1^*(s)$) and $\varphi(z) \sim 1 + O(z^{-M})$, any $M > 0$, we obtain for $z \to \infty$

$$\widetilde{L}(z) \sim \log_2 z \cdot \frac{\Phi^*(0)}{L} - \frac{\Phi^*(0)}{L} - \frac{\Phi^{*\prime}(0)}{L^2} + P(z) \tag{25}$$

where

$$\Phi^*(0) = \int_0^\infty e^{-x} e_d(x)\varphi(2x)\frac{dx}{x} \ ,$$

$$\Phi^{*\prime}(0) = \int_0^\infty e^{-x} e_d(x)\varphi(2x)\frac{\log x}{x}dx \ ,$$

and $P(z)$ is a fluctuating function with a small amplitude.

We need to evaluate $\Phi^*(0)$ and $\Phi^{*\prime}(0)$. The former is rather easy, and by properties of the Mellin transform we find out that $\Phi^*(0) = L = \log 2$. The latter quantity is harder to estimate. To do so, we first present another evaluation of $\Phi^*(0)$ which can be extended to obtain $\Phi^{*\prime}(0)$. Note that $\varphi(2x) - \varphi(x) = e^{-x}e_d(x)\varphi(2x)$. Define also a function $\mathbf{1}(x)$ as follows

$$\mathbf{1}(x) = \begin{cases} 1 & \text{if } x \geq 1 \\ 0 & \text{if } x < 1 \ . \end{cases}$$

Then, we can write

$$\begin{aligned}\Phi^*(0) &= \int_0^\infty (\varphi(2x) - \varphi(x))\frac{dx}{x} \\ &= \int_0^\infty (\varphi(2x) - \mathbf{1}(2x))\frac{dx}{x} + \int_0^\infty (\mathbf{1}(2x) - \mathbf{1}(x))\frac{dx}{x} + \int_0^\infty (\mathbf{1}(x) - \varphi(x))\frac{dx}{x} \\ &= \int_0^\infty (\mathbf{1}(2x) - \mathbf{1}(x))\frac{dx}{x} = \int_{1/2}^\infty \frac{dx}{x} = L\ .\end{aligned}$$

In a similar manner we can deal with $\Phi^{*\prime}(0)$. This time, however, the calculation is more involved. Nevertheless, we obtain the following

$$\begin{aligned}\Phi^{*\prime}(0) &= -L^2/2 - L\int_0^\infty (\mathbf{1}(x) - \varphi(x))\frac{dx}{x} \\ &= \int_0^1 \varphi(x)\frac{dx}{x} + \int_1^\infty (\varphi(x) - 1)\frac{dx}{x} \\ &\sim O(d^{-1/2}) - \int_1^\infty e_d(x)e^{-x}\frac{dx}{x} \qquad d \to \infty\end{aligned}$$

where the last integral can be analyzed using the *incomplete gamma* function. In particular, we can prove that for $d \to \infty$ we have $C_d = \frac{1}{L^2}\Phi^{*\prime}(0) = \log_2 d - 1/2 + O(d^{-1/2+\varepsilon})$. Finally, using the Depoissonization Lemma we establish Theorem 1(i).

The proof of Theorem 1(ii) concerning the variance of $R_{N,d}$ is more intricate but it proceeds along the same lines as the above derivation. In particular, we deal now with the functional equation (5) which we repeat below

$$\widetilde{W}(z) = f_d(z/2)\widetilde{W}(z/2) + 2\widetilde{L}(z)f_d(z/2)\ . \tag{26}$$

Using the idea from Section 2 (cf. (6) – (9)) we find immediately an explicit solution to (26), namely

$$\widetilde{W}(z)\varphi(z) = 2\sum_{n=0}^\infty \widetilde{L}(z2^{-n-1})\varphi(z2^{-n-1})\ ,$$

which becomes, after inserting the solution (19) for $\widetilde{L}(z)$,

$$\widetilde{W}(z)\varphi(z) = 2\sum_{j=1}^\infty\sum_{n=1}^\infty \varphi(z2^{-n-j})\ . \tag{27}$$

To establish asymptotics of $\widetilde{W}(z)$, and also $F_N''(1)$, we proceed as before through the Mellin transform and depoissonization. Let $Q_2(z)$ and $Q_2^*(s)$ denote respectively the RHS of (27) and its Mellin transform. Then,

$$Q_2^*(s) = 2\left(\frac{2^s}{1-2^s}\right)^2\varphi^*(s) = 2\left(\frac{2^s}{1-2^s}\right)^3\Phi^*(s)$$

where $\Phi^*(s)$ is defined as before. Computing residues of the above, one immediately proves for $z \to \infty$

$$Q_2(z) \sim \log_2^2 z - (3 + 2C_d)\log_2 z + 2 + 3C_d + D_d + P(\log_2 z)$$

where C_d was defined before, and

$$D_d = \frac{1}{L^3}\Phi^{*\prime\prime}(0) = \frac{1}{L^3}\int_0^\infty e_d(x)e^{-x}\varphi(2x)\frac{\log^2 x}{x}dx \ .$$

We can analyze D_d in the same manner as before. After some algebra we arrive at

$$\begin{aligned} D_d &\sim C_d - 1/6 + 2/L^2\int_1^\infty e_d(t)e^{-t}\frac{\log t}{t}dt \\ &\sim C_d - 1/6 + \log_2^2 d + O(d^{-1/2+\varepsilon}) \end{aligned}$$

as $d \to \infty$. After applying the Depoissonization Lemma, we complete the proof of Theorem 1(ii).

The limiting generating function $\lim_{N\to\infty} F_N(z)$ can be analyzed in the same manner as above but this time the solution (16) should be used to derive Theorem 2.

With regard to the limiting distribution the Depoissonization Lemma applied to the bivariate function $H(z,x)$ expressed as in (16) strongly suggests

$$\varphi\left(2^{-t-1-\{\log_2 N+t\}}\right) \tag{28}$$

as an approximation for $\Pr\{R_{N,d} - \log_2 N > t\}$ for t fixed and $N \to \infty$. (In the expression (28) $\{x\}$ denotes the fractional part of x.) The fluctuating term is due to the fact that a discrete parameter is approximated by a continuous function. Detailed derivations are left to a journal version of the paper.

Acknowledgment. The first two authors want to thank M. Drmota and P. Grabner for interesting discussions.

References

[1] G. Brassard and P. Bratley, *Algorithmics. Theory and Practice*, Prentice Hall, Englewood Cliffs, 1988.

[2] P. Flajolet and G.N. Martin, Probabilistic Counting Algorithms for Data Base Applications, *J. Comp. Syst. Sci.*, 31, 182-209, 1985.

[3] P. Flajolet and A. Odlyzko, Singularity Analysis of Generating Functions, *SIAM J. Disc. Methods*, 3, 216-240, 1990.

[4] P. Flajolet and B. Richmond, Generalized Digital Trees and Their Difference-Differential Equations, *Random Structures and Algorithms*, 3, 1992, in press.

[5] P. Flajolet, M. Régnier and R. Sedgewick, Some Uses of the Mellin Transform Techniques in the Analysis of Algorithms, in *Combinatorial Algorithms on Words*, Springer NATO ASI Ser. F12, 241-254, 1985.

[6] A. Greenberg, P. Flajolet and R. Ladner, Estimating the Multiplicities of Conflicts to Speed Their Resolution in Multiple Access Channels, *JACM*, 34, 289-325, 1987.

[7] P. Henrici, *Applied and Computational Complex Analysis*, vol. 2, John Wiley & Sons 1977.

[8] P. Jacquet and W. Szpankowski, Ultimate Characterizations of the Burst Response of an Interval Searching Algorithm: A Study of a Functional Equations, *SIAM J. Computing*, 18, 777-791, 1989.

[9] M. Kendall and A. Stuart, *The Advanced Theory of Statistics*, 4th Ed. vol. 1, Charles and Griffin and Co. Ltd., London (1977).

[10] P. Kirschenhofer and H. Prodinger, On the Analysis of Probabilistic Counting, *Lecture Notes in Mathematics*, 1452 (Eds. E. Hlawka and R. Tichy), 117-120, Springer Verlag 1990.

[11] P. Kirschenhofer and H. Prodinger, Approximate Counting: An Alternative Approach, *Informatique Théorique et Applications/Theoretical Informatics and Applications*, 25, 43-48, 1991.

[12] D.E. Knuth, *The Art of Computer Programming. Sorting and Searching*, vol. 3., Addison-Wesley, Reading, MA 1973.

[13] R. Morris, Counting Large Numbers of Events in Small Registers, *Comm. ACM*, 21, 840-842, 1978.

[14] B. Pittel and H. Rubin, How Many Random Questions Are Necessary to Identify n Distinct Objects?, *J. Combinatorial Theory, Ser. A*, 55, 292-312, 1990.

[15] H. Prodinger, How to Select a Loser, *Discrete Mathematics*, to appear.

[16] B. Rais, P. Jacquet, and W. Szpankowski, Limiting Distribution for the Depth in Patricia Tries, *SIAM Journal on Discrete Mathematics*, to appear.

[17] U. Schmid, The Average CRI-length of a Tree Collision Resolution Algorithm in Presence of Multiplicity-Dependent Capture Effects, *this Volume.*

[18] W. Szpankowski, Solution of a Linear Recurrence Equation Arising in the Analysis of Some Algorithms, *SIAM J. Alg. Disc. Methods*, 8, 233-250, 1987.

[19] W. Szpankowski, Patricia Tries Again Revisited, *JACM*, 37, 691-711, 1990.

[20] W. Szpankowski and V. Rego, Yet Another Application of a Binomial Recurrence. Order Statistics, *Computing*, 43, 401-410, 1990.