Two Not-That-Dull Functional Equations Arising in the Analysis of Algorithms

Wojciech Szpankowski
Purdue University

June 15, 1998

[summary by Philippe Jacquet]

1. Introduction

The talk addresses two functional equations arising in the analysis of algorithms, namely, in the performance evaluation of the generalized digital search trees and the asymmetric leader election algorithm. These functional equations deal with Poisson transforms of the general recurrence

$$x_{n+b} = a_n + cp^n x_n + u \sum_{k=0}^{n} \binom{n}{k} p^k (1-p)^{n-k} (x_k + x_{n-k}),$$

where u and c are constants, a_n is a given sequence, and $b \geq 1$ is a parameter. Together with suitable initial condition, this recurrence describes both algorithms. It was extensively investigated either for $b = 1$ or $c = 0$. The speaker presents asymptotic expansions of x_n up to $O(1)$ term. Interestingly enough, for both algorithms there appears a constant that must be evaluated numerically from the original recurrence. Analytic techniques of (precise) analysis of algorithms are used to establish these conclusions. In particular, the author uses analytic poissonization/depoissonization, Mellin transform and singularity analysis.

The results presented in this talk were obtained jointly with S. Janson, G. Louchard and J. Tang.

2. The Generalized Digital Search Tree Algorithm

The basic Digital Search Tree (DST) is a tree-like data structure. Each node in the tree contains one data. We assume that all data are encoded over a common finite alphabet of size, say V. Each one of the edges pending from a node is in correspondence with one symbol of the alphabet. Consequently the branching degree of each node cannot exceed V.

The insertion of a new data X in the DST proceeds as follows:

1. Scan the first characters of data X in order to create a path in the DST with the symbol-edge correspondence;
2. use the character after the last scanned character of X to create a new edge in the DST pending from the last node visited, and create a new node to store X.

The basic DST can be used to implement Lempel-Ziv compression algorithms. The successive data inserted in the DST are phrases scanned on the text to be compressed. Therefore the original text is divided into phrases, and since each phrase points to another phrase via a symbol-edge in the DST structure, the compressed code replaces each phrase by a pair (pointer, symbol).

In the following we consider data generated from a Bernoulli binary source over a probability vector (p, q).

The average depth of insertion ED_n in the binary DST satisfies the following recursion valid for $n > 0$:

$$(n+1)ED_{n+1} = n + 1 \sum_{k=0}^{n} \binom{n}{k} p^k (1-p)^{n-k} (kED_k + (n-k)ED_{n-k}).$$

The probability generating function $D_n(u)$ of the depth of insertion satisfies:

$$(n+1)D_{n+1}(u) = 1 + u \sum_{k=0}^{n} \binom{n}{k} p^k (1-p)^{n-k} (kD_k(u) + (n-k)D_{n-k}(u)).$$

The general case for data generated from Bernoulli source over V-ary alphabet (with probability vector $(p_1, \ldots, p_V)$):

$$(n+1)D_{n+1}(u) = 1 + u \sum_{k_1,\ldots,k_V} \binom{n}{k_1 \cdots k_V} p_1^{k_1} \cdots p_V^{k_V} (k_1 D_{k_1}(u) + \cdots + k_V D_{k_V}(u)).$$

The generalized DST assume a capacity b for each node of the DST, *i.e.*, each can store up to b data. The insertion algorithm is the same excepted that if the last visited node contains less than b data, then data X is stored in this node and no new node is created.

The functional recursion of p.g.f. $D_n(u)$ is now the following:

$$(n+b)D_{n+b}(u) = b + u \sum_{k} \binom{n}{k} p^k (1-p)^{n-k} (kD_k(u) + (n-k)D_{n-k}(u)).$$

The generating function of $D(z) = \sum_n ED_n \frac{z^n}{n!} e^{-z}$ satisfies the functional equation:

$$\sum_{i=0}^{b} \binom{b}{i} \frac{\partial^i}{\partial z^i} D(z) = z + D(pz) + D(qz).$$

The aim is to find an accurate asymptotic expansion of ED_n. Via depoissonization argument it is equivalent to find an asymptotic expansion of $D(n) = ED_n + O(\log n)$. To this end one makes use of the Mellin transform $d(s)\Gamma(s) = \int_0^\infty D(x) x^{s-1} dx$ satisfies:

$$\sum_{i=0}^{b} \binom{b}{i} (-1)^i d(s-i) = (p^{-s} + q^{-s}) d(s).$$

In other words $(1 - p^{-s} - q^{-s})d(s)$ is a linear combination of $d(s-i)$, i varying from 1 to b:

$$d(s) = \frac{1}{1 - p^{-s} - q^{-s}} \sum_{i=0}^{b} \binom{b}{i} (-1)^{i+1} d(s-i).$$

It comes that $d(s)$ is defined for $-b - 1 < \Re(s) < -1$. The inverse Mellin transform expresses $D(z)$ via an integration formula:

$$D(z) = \frac{1}{2i\pi} \int_{c-i\infty}^{c+i\infty} d(s)\Gamma(s) z^{-s} ds$$

for $c \in]-b-1, -1[$. The asymptotic expansion of $D(z)$ when $z \to \infty$ comes from the poles of $d(s)$ and $\Gamma(s)$ via application of the residue theorem. The poles of $d(s)$ are the roots of $1 - p^{-s} - q^{-s}$, and the same roots repeat when translated on the right by integer values because of the identities between $d(s)$ and $d(s-i)$. The main singularity is at $s = -1$ which doubles the pole of $\Gamma(s)$ and will contribute in a $z \log z$ term in $D(z)$'s expansion.

Application of the residue theorem finally gives:

$$D(z) = \frac{1}{h} z \log z + \left(\frac{h_2 - h}{h^2} + \gamma - \frac{f'(-1)}{h} \right) z + z P_1(\log z) + O(\log z),$$

with $h = -p \log p - q \log q$ and $h_2 = p \log^2 p + q \log^2 q$. Quantity $P_1(z)$ is periodic with small amplitude when $\log p / \log q$ is rational (when the roots of $1 - p^{-s} - q^{-s}$ are regularly spaced on the vertical axis $\Re(s) = -1$), and $o(z)$ otherwise. Quantity $f'(-1)$ denotes the derivative of $\sum_{i=1}^{b} \binom{b}{i} (-1)^{i+1} d(s-i)$ at $s = -1$.

The question remains on how to compute a numerical evaluation of constant $f'(-1)$. To this end one computes a numerically tractable expression of $d(s)$ via the Mellin transform:

$$d(s) = \frac{1}{\Gamma(s)} \int_0^\infty z^{s-1} \sum_n E D_n \frac{z^n}{n!} e^{-z} dz = \sum_n \frac{E D_n}{n!} \frac{\Gamma(s+n)}{\Gamma(s)}.$$

Therefore

$$d(s) = \sum_n \frac{E D_n}{n!} s(s+1) \cdots (s+n-1).$$

Using this formula and truncating it up to certain rank N gives a numerical evaluation of $f'(-1)$. Unfortunately the error term can be proven to be of order $O(\log N / N)$. There are probably better estimates which converge geometrically.

3. Leader Election

We don't give details on the problem of leader election. Via successive Bernoulli splits over a group of n people, one randomly selects a leader. We denote X_n the average number of steps needed to achieve this election over a population of n people. When the Bernoulli splits are all done over the same probability vector (p, q), one obtains the recursion for $n \geq 2$:

$$X_n = 1 + q^n X_n + \sum_k \binom{n}{k} p^k q^{n-k} X_k$$

which translates into a functional equation for the generating function $L(z) = \sum_n X_n e^{-z} z^n / n!$:

$$L(z) = 1 - (1+z) e^{-z} + L(pz) + L(qz) e^{-pz}.$$

Using again the Mellin approach, $L^*(s)$ the Mellin transform of $L(z)$ satisfies the identity:

$$L^*(s) = \frac{f^*(s) - (1+s)\Gamma(s)}{1 - p^{-s}},$$

where $f^*(s)$ is the Mellin transform of $f(z) = L(qz) e^{-pz}$. Since $L(0) = 0$, $L^*(s)$ is defined for $-1 < \Re(s) < 0$, and thanks to the exponential decrease of $f(z)$, $f^*(s)$ is defined for all $\Re(s) > -1$. There is a sequence of poles on the axis $\Re(s) = 0$ regularly spaced: $s_k = 2i\pi k / \log p$, for k integer. The poles are simple for $k \neq 0$, the pole is double at $s = 0$ due to the contribution of the pole of $\Gamma(s)$. The double pole contributes in a $-\log z / \log p$ term in the residue theorem applied to reverse Mellin transform. The other poles contribute to a periodic function which depends on $f^{(s)}$ at $s = s_k$.

In summary for any $M > 0$:

$$L(z) = -\frac{\log z}{\log p} + \frac{1}{2} + \frac{1 - \gamma - f^*(0)}{\log p} + P_2(\log z) + O\left(\frac{1}{z^M} \right).$$

$P_2(x)$ is a periodic function of period $\log p$ and small amplitude. Using again depoissonization theorems one gets $X_n = L(n) + O(1/n)$.

As with the generalized DST, the numerical evaluation of constant $f^*(0)$ remains. In this case we are luckier than with DST since $f^*(0) = \sum_n X_n q^n/n$ which converges exponentially.