A unified approach to linear probing hashing

Svante Janson[1]† and Alfredo Viola[2]

[1] *Department of Mathematics, Uppsala University, PO Box 480, SE-751 06 Uppsala, Sweden.*
[2] *Universidad de la República, Montevideo, Uruguay.*

We give a unified analysis of linear probing hashing with a general bucket size. We use both a combinatorial approach, giving exact formulas for generating functions, and a probabilistic approach, giving simple derivations of asymptotic results. Both approaches complement nicely, and give a good insight in the relation between linear probing and random walks. A key methodological contribution, at the core of Analytic Combinatorics, is the use of the symbolic method (based on q-calculus) to directly derive the generating functions to analyze.

Keywords: hashing; linear probing; buckets; generating functions; analytic combinatorics

1 Motivation

Linear probing hashing, defined below, is certainly the simplest "in place" hashing algorithm [10].

> *A table of length* m, $T[1\,..\,m]$, *with buckets of size* b *is set up, as well as a hash function* h *that maps keys from some domain to the interval* $[1\,..\,m]$ *of table addresses. A collection of* n *elements with* $n \leqslant bm$ *are entered sequentially into the table according to the following rule: Each element* x *is placed at the first bucket that is not full starting from* $h(x)$ *in cyclic order, namely the first of* $h(x), h(x)+1, \dots, m, 1, 2, \dots, h(x)-1$.

In [9] Knuth motivates his paper in the following way: "The purpose of this note is to exhibit a surprisingly simple solution to a problem that appears in a recent book by Sedgewick and Flajolet [12]:

Exercise 8.39 Use the symbolic method to derive the EGF of the number of probes required by linear probing in a successful search, for fixed M."

Moreover, at the end of the paper in his personal remarks he declares: "None of the methods available in 1962 were powerful enough to deduce the expected square displacement, much less the higher moments, so it is an even greater pleasure to be able to derive such results today from other work that has enriched the field of combinatorial mathematics during a period of 35 years." In this sense, he is talking about the powerful methods based on Analytic Combinatorics that has been developed for the last decades, and are presented in [6].

In this paper we present in a unified way the analysis of several random variables related with linear probing hashing with buckets, giving explicit and exact trivariate generating functions in the combinatorial

† Partly supported by the Knut and Alice Wallenberg Foundation

model, together with generating functions in the asymptotic Poisson model that provide limit results, and relations between the two types of results. Linear probing has been shown to have strong connections with several important problems (see [9; 5; 2] and the references therein). The derivations in the asymptotic Poisson model are probabilistic and use heavily the relation between random walks and the profile of the table. Moreover, the derivations in the combinatorial model are based in combinatorial specifications that directly translate into multivariate generating functions. As far as we know, this is the first unified presentation of the analysis of linear probing hashing with buckets based on Analytic Combinatorics ("if you can specify it, you can analyze it").

We will see that results can easily be translated between the exact combinatorial model and the asymptotic Poisson model. Nevertheless, we feel that it is important to present independently derivations for the two models, since the methodologies complement very nicely. Moreover, they heavily rely in the deep relations between linear probing and other combinatorial problems like random walks, and the power of Analytic Combinatorics.

The derivations based on Analytic Combinatorics heavily rely on a lecture presented by Flajolet whose notes can be accessed in [4]. Since these ideas have only been partially published in the context of the analysis of hashing in [6], we briefly present here some constructions that lead to q-analogs of their corresponding exponential generating functions. Proofs will be given in the full version [8] of this paper.

1.1 Some notation

We study tables with m buckets of size b and n elements, where $b \geqslant 1$ is a constant. We often consider limits as $m, n \to \infty$ with $n/bm \to \alpha$ with $\alpha \in (0,1)$. We consider also the Poisson model with $n \sim \mathrm{Po}(\alpha bm)$, and thus $\mathrm{Po}(b\alpha)$ elements hashed to each bucket; in this model we can also take $m = \infty$ which gives a natural limit object, see Section 4 and Lemma 5.1.

A *cluster* or *block* is a (maximal) sequence of full buckets ended by a non-full one. The *tree function* is $T(z) := \sum_{n=1}^{\infty} \frac{n^{n-1}}{n!} z^n$, which converges for $|z| \leqslant e^{-1}$. Let $\omega = \omega_b := e^{2\pi \mathrm{i}/b}$ be a primitive b:th unit root.

2 Combinatorial characterization of linear probing

As a combinatorial object, a non-full linear probing hash table is a sequence of almost full tables (or clusters) [9; 5; 13]. As a consequence, any random variable related with the table itself (like block lengths, or the overflow in the parking problem) or with a random element (like its search cost) can be studied in a cluster (that we may assume to be the last one in the sequence), and then use the sequence construction. Figure 1 presents an example of such a decomposition.

We briefly recall here some of the definitions presented in [13]. Let F_{bi+d} be the number of ways to construct an almost full table of length $i+1$ and size $bi+d$ (that is, there are $b-d$ empty slots in the last bucket). Define also

$$F_d(u) := \sum_{i \geq 0} F_{bi+d} \frac{u^{bi+d}}{(bi+d)!}, \qquad N_d(z,w) := \sum_{s=0}^{b-1-d} w^{b-s} F_s(zw), \quad 0 \leq d \leq b-1. \tag{2.1}$$

In this setting $N_d(z,w)$ is the generating function for the number of almost full tables with more than d empty locations in the last bucket. More specifically $N_0(z,w)$ is the generating function for all the almost

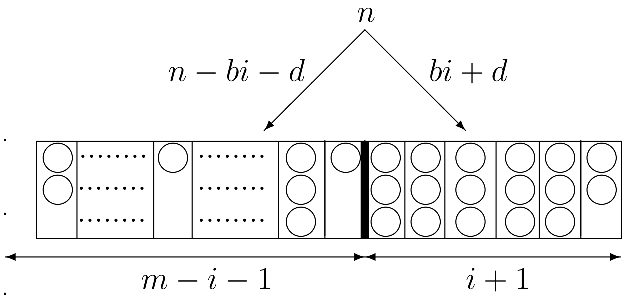

Fig. 1: A decomposition for $b=3$ and $d=2$.

full tables. We borrow from [13] the following identities:

$$\sum_{d=0}^{b-1} F_d(bz)x^d = x^b - \prod_{j=0}^{b-1}\left(x - \frac{T(\omega^j z)}{z}\right), \tag{2.2}$$

$$N_0(bz,w) = 1 - \prod_{j=0}^{b-1}\left(1 - \frac{T(\omega^j zw)}{z}\right), \tag{2.3}$$

$$\sum_{d=0}^{b-1} N_d(bz,w)x^d = \frac{\prod_{j=0}^{b-1}\left(1 - x\frac{T(\omega^j zw)}{z}\right) - \prod_{j=0}^{b-1}\left(1 - \frac{T(\omega^j zw)}{z}\right)}{1-x}. \tag{2.4}$$

Let also $Q_{m,n,d}$ be the number of ways of inserting n elements into a table with m buckets of size b, so that a given (say the last) bucket of the table contains more than d empty slots. In this setting, by a direct application of the sequence construction as presented in [6] we derive a result presented in [1]:

$$\Lambda_0(z,w) := \sum_{m\geq 0}\sum_{n\geq 0} Q_{m,n,0}\frac{z^n}{n!}w^{bm} = \frac{1}{1-N_0(z,w)}. \tag{2.5}$$

Then, $\Lambda_0(z,w)$ is the generating function for the number of ways to construct hash tables such that their last bucket is not full.

Consider a hash table of length m and n keys, where collisions are resolved by linear probing. Let P be a property (e.g. cost of a successful search or block length), related with the last cluster of the sequence, or with a random element inside it. Let $p_{bi+d}(q)$ be the probability generating function of P calculated in the cluster of length $i+1$ and with $bi+d$ elements. We may express $p_{m,n}(q)$, the generating function of P for a table of length m and n elements with at least one empty spot in the last bucket, as the sum of the conditional probabilities:

$$p_{m,n}(q) = \sum_{d=0}^{b-1}\sum_{i\geqslant 0} \#\{\text{tables where last cluster has size } i+1 \text{ and } bi+d \text{ elements}\}\ p_{bi+d}(q). \tag{2.6}$$

There are $Q_{m-i-1,n-bi-d,0}$ ways to insert $n-bi-d$ elements in the leftmost hash table of length $m-i-1$, leaving their rightmost bucket not full. Moreover, there are F_{bi+d} ways to insert $bi+d$ elements in the almost full table of length $i+1$. Furthermore, there are $\binom{n}{bi+d}$ ways to choose which $bi+d$ elements go to the last cluster. Therefore,

$$p_{m,n}(q) = \sum_{d=0}^{b-1} \sum_{i \geq 0} \binom{n}{bi+d} Q_{m-i-1,n-bi-d,0}\, F_{bi+d} p_{bi+d}(q). \tag{2.7}$$

Then, the trivariate generating function for $p_{m,n}(q)$ is

$$P(z,w,q) := \sum_{m,n \geq 0} p_{m,n}(q)\, w^{bm} \frac{z^n}{n!} = \frac{\hat{N}_0(z,w,q)}{1-N_0(z,w)}, \qquad \text{with} \tag{2.8}$$

$$\hat{N}_0(z,w,q) := \sum_{d=0}^{b-1} w^{b-d} \sum_{i \geq 0} F_{bi+d} \frac{(zw)^{bi+d}}{(bi+d)!}\, p_{bi+d}(q), \tag{2.9}$$

which could be directly derived with the sequence construction [6]. Notice that, as expected, $\hat{N}_0(z,w,1) = N_0(z,w)$ and $P(z,w,1) = \Lambda_0(z,w) - 1$, since we consider only $m \geq 1$ (we have a last, non-filled bucket).

Moreover the Poisson Transform of $p_{m,n}(q)/m^n$ is, with $Q_{m,d}(u) := \sum_{n \geqslant 0} Q_{m,n,d} u^n/n!$,

$$\begin{aligned}\mathbf{P}_m[p_{m,n}(q)/m^n; b\alpha] &:= e^{-mb\alpha} \sum_{n \geq 0} p_{m,n}(q)\, \frac{(mb\alpha)^n}{m^n n!} \\ &= \sum_{d=0}^{b-1} e^{-(b-d)\alpha} \sum_{i \geq 0} F_{bi+d}\, \frac{(b\alpha e^{-\alpha})^{bi+d}}{(bi+d)!}\, p_{bi+d}(q)\; e^{-(m-i-1)b\alpha}\, Q_{m-i-1,0}(b\alpha). \end{aligned} \tag{2.10}$$

Furthermore, $Q_{m-i-1,0}(b\alpha) = [T_0(b\alpha)\, e^{(m-i-1)b\alpha}]_{b(m-i-1)-1}$ where $T_0(b\alpha)$ is, in the asymptotic Poisson model, the probability that a given bucket is not full [13]. It is proven in [1; 13] that

$$\lim_{m\to\infty} \mathbf{P}_m[Q_{m,n,0}/m^n; b\alpha] = \lim_{m\to\infty} e^{-mb\alpha} Q_{m,0}(b\alpha) = T_0(b\alpha) = \frac{b(1-\alpha)}{\prod_{j=1}^{b-1}\left(1 - \frac{T(\omega^j \alpha e^{-\alpha})}{\alpha}\right)}. \tag{2.11}$$

As a consequence, (2.10) and (2.9) yield

$$\lim_{m\to\infty} \mathbf{P}_m[p_{m,n}(q)/m^n; b\alpha] \quad = \quad T_0(b\alpha)\hat{N}_0(b\alpha, e^{-\alpha}, q). \tag{2.12}$$

Note that if $0 < \alpha < 1$ is a fixed constant, then $w = e^{-\alpha}$ is the dominant singularity of $P(b\alpha, w, q)$ (a root of $1 - N_0(b\alpha, w)$, for $j = 0$ in (2.3), cf. (2.8)), so the relation (2.12) can also be derived by standard asymptotic methods as in [6]. As a consequence, all the results found for exact m, n can easily been translated in the Poisson model.

3 A q-calculus to specify hashing random variables

All the generating functions in this paper are exponential in n and ordinary in m. As a consequence all the labelled constructions in [6] and their respective translation into EGF can be used. However, to specify the combinatorial properties related with the analysis of linear probing hashing, new constructions have to be added. These ideas have been presented by Flajolet in [4], but they do not seem to have been published in the context of hashing. As a consequence, we briefly summarize them in this section.

Adding an element $\mapsto \int$ $\mathcal{C} = \mathrm{Add}(\mathcal{A})$	$C_n = A_{n-1}$ $C(z) = \int_0^z A(w)dw$
Choosing a position $\mapsto \partial$ $\mathcal{C} = \mathrm{Pos}(\mathcal{A})$	$C_n = (n+1)A_n$ $C(z) = \frac{\partial}{\partial z}(zA(z))$
Averaging $\mapsto \frac{1}{Z}\int$ $\mathcal{C} = \mathrm{Ave}(\mathcal{A})$	$C_n = \frac{A_n}{n+1}$ $C(z) = \frac{1}{z}\int_0^z A(w)dw$
Adding a bucket $\mapsto \exp$ $\mathcal{C} = \mathrm{Bucket}(\mathcal{Z})$	$C_n = 1$ $C(z) = \exp(z)$

We present a list of combinatorial constructions used in hashing and their corresponding translation into EGF, where $\mathcal{Z}$ is an atomic class comprising a single element of size 1. Moreover, to keep track of the distribution of random variables (e.g. the displacement of a new inserted element), we need translations that belong to the area of q-calculus. Equations (3.1), (3.2) and (3.3) present some of these translations.

$$n \quad \mapsto \quad [n] = 1 + q + q^2 + \ldots + q^{n-1} = \frac{1-q^n}{1-q} \tag{3.1}$$

$$\sum (n+1) f_n z^n \quad \mapsto \quad \sum [n+1] f_n z^n \tag{3.2}$$

$$\frac{\partial}{\partial z}(zA(z)) \quad \mapsto \quad H[f(z)] = \frac{F(z) - qF(qz)}{1-q} \tag{3.3}$$

Moments result from using the operators ∂_q (differentiation w.r.t. q) and U (setting $q = 1$).

4 Probabilistic method: finite and infinite hash tables

In general, consider a hash table, with locations ("buckets") each having capacity b; we suppose that the buckets are labelled by $i \in \mathfrak{T}$, for a suitable index set $\mathfrak{T}$. Let for each bucket $i \in \mathfrak{T}$, X_i be the number of elements that have hash address i, and thus first try bucket i. Moreover, let H_i be the total number of elements that try bucket i and let Q_i be the *overflow* from bucket i, i.e., the number of elements that try bucket i but fail to find room and thus are transferred to the next bucket. We thus have the equations

$$H_i = X_i + Q_{i-1}, \qquad Q_i = (H_i - b)_+. \tag{4.1}$$

The final number of elements stored in bucket i is $Y_i := H_i \wedge b := \min(H_i, b)$; in particular, the bucket is full if and only if $H_i \geqslant b$.

Standard hashing is when the index set $\mathfrak{T}$ is the cyclic group $\mathbb{Z}_m$. Another standard case, called the *parking problem*, is when $\mathfrak{T}$ is an interval $\{1, \ldots, m\}$ for some integer m; in this case the Q_m elements that try the last bucket but fail to find room there are lost (overflow), and (4.1) uses the initial value $Q_0 := 0$.

In the analysis, we will mainly study infinite hash tables, either one-sided with $\mathfrak{T} = \mathbb{N} := \{1, 2, 3, \ldots\}$, or two-sided with $\mathfrak{T} = \mathbb{Z}$; as we shall see, these occur naturally as limits of finite hash tables. In the one-sided case, we again define $Q_0 := 0$, and then, given $(X_i)_1^\infty$, H_i and Q_i are uniquely determined recursively for all $i \geqslant 1$ by (4.1). In the doubly-infinite case, it is not obvious that the equations (4.1) really have a solution; we return to this question in Lemma 4.1 below.

In the case $\mathfrak{T} = \mathbb{Z}_m$, we allow (with a minor abuse of notation) also the index i in these quantities to be an arbitrary integer with the obvious interpretation; then X_i, H_i and so on are periodic sequences defined for $i \in \mathbb{Z}$.

We can express H_i and Q_i in X_i by the following lemma, which generalizes (and extends to infinite hashing) the case $b = 1$ treated in [10, Exercise 6.4-32], [3, Proposition 5.3], [7, Lemma 2.1].

Lemma 4.1 *Let X_i, $i \in \mathfrak{T}$, be given non-negative integers.*

(i) *If $\mathfrak{T} = \{1, \dots, m\}$ or $\mathbb{N}$, then the equations* (4.1)*, for all $i \in \mathfrak{T}$, have a unique solution given by, considering $j \geqslant 0$,*

$$H_i = \max_{j<i} \sum_{k=j+1}^{i} (X_k - b) + b, \qquad\qquad Q_i = \max_{j\leqslant i} \sum_{k=j+1}^{i} (X_k - b) \tag{4.2}$$

(ii) *If $\mathfrak{T} = \mathbb{Z}_m$, and moreover $n = \sum_1^m X_i < bm$, then the equations* (4.1)*, for all $i \in \mathfrak{T}$, have a unique solution given by* (4.2)*, now with $j \in \mathbb{Z}$. Furthermore, there exists $i_0 \in \mathfrak{T}$ such that $H_{i_0} < b$ and thus $Q_{i_0} = 0$.*

(iii) *If $\mathfrak{T} = \mathbb{Z}$, assume that*

$$\sum_{i=0}^{N-1} (b - X_{-i}) \to \infty \qquad \textit{as } N \to \infty. \tag{4.3}$$

Then the equations (4.1)*, for all $i \in \mathfrak{T}$, have a solution given by* (4.2)*, with $j \in \mathbb{Z}$, and this is the minimal solution. Furthermore, for each $i \in \mathfrak{T}$ there exists $i_0 < i$ such that $H_{i_0} < b$ and thus $Q_{i_0} = 0$. Conversely, this is the only solution such that for every i there exists $i_0 < i$ with $Q_{i_0} = 0$.*

In the sequel, we will always use this solution of (4.1) for hashing on $\mathbb{Z}$ (assuming that (4.3) holds); we can regard this as a definition of hashing on $\mathbb{Z}$.

5 Convergence to an infinite hash table

We are interested in hashing on Z_m with n elements having independent uniformly random hash addresses, thus $X_1, \dots, X_m$ have a multinomial distribution with parameters n and $(1/m, \dots, 1/m)$. (We denote these X_i by $X_{m,n;i}$.) We denote the profile of this hash table by $H_{m,n;i}$, where as above $i \in Z_m$ but we also can allow $i \in \mathbb{Z}$ in the obvious way.

We consider a limit with $m, n \to \infty$ and $n/bm \to \alpha \in (0, 1)$. The appropriate limit object turns out to be an infinite hash table on $\mathbb{Z}$ with $X_i = X_{\alpha;i}$ that are independent and identically distributed (i.i.d.) with the Poisson distribution $X_i \sim \operatorname{Po}(\alpha b)$; this is the asymptotic Poisson model mentioned earlier. Note that $\mathbb{E}\, X_i = \alpha b < b$, so $\mathbb{E}(b - X_i) > 0$ and (4.3) holds almost surely by the law of large numbers; hence this infinite hash table is well-defined. We denote the profile of this hash table by $H_{\alpha;i}$.

We claim that the profile $(H_{m,n;i})_{i=-\infty}^{\infty}$, regarded as a random element of the product space $\mathbb{Z}^{\mathbb{Z}}$, converges in distribution to the $(H_{\alpha;i})_{i=-\infty}^{\infty}$. (By the definition of the product topology, this is equivalent to convergence in distribution of any finite vector $(H_{m,n;i})_{-M}^{N}$ to $(H_{\alpha;i})_{-M}^{N}$.)

Lemma 5.1 *Let $m, n \to \infty$ with $n/bm \to \alpha$ for some α with $0 < \alpha < 1$. Then $(H_{m,n;i})_{i=-\infty}^{\infty} \xrightarrow{\mathrm{d}} (H_{\alpha;i})_{i=-\infty}^{\infty}$.*

Remark 5.2 Note that the convergence of the profile implies convergence of all other quantities that we study here. Thus the theorems in the sections below for hashing on $\mathbb{Z}$ contain (and are equivalent to) limit theorems for finite hashing as $m, n \to \infty$ with $n/bm \to \alpha$.

6 The profile and overflow (parking problem)

In the combinatorial approach, let $\Omega(z,w,q)$ be the generating function for the number of elements that overflow from a hash table (i.e., the number of cars that cannot find a place in the parking problem)

$$\Omega(z,w,q) := \sum_{m\geqslant 0}\sum_{n\geqslant 0}\sum_{k\geqslant 0} N_{m,n,k} w^{bm} \frac{z^n}{n!} q^k, \tag{6.1}$$

where $N_{m,n,k}$ is the number of hash tables of length m with n elements and overflow k. (We include an empty hash table with $m = n = k = 0$ in the sum (6.1).) Thus w marks the number of places in the table, z the number of elements and q the number of elements that overflow. The following result has been independently presented by Panholzer in [11].

Theorem 6.1

$$\Omega(bz,w,q) = \frac{1}{q^b - w^b e^{qbz}} \cdot \frac{\prod_{j=0}^{b-1}\left(q - \frac{T(\omega^j zw)}{z}\right)}{\prod_{j=0}^{b-1}\left(1 - \frac{T(\omega^j zw)}{z}\right)}. \tag{6.2}$$

Proof: [Sketch] The number of elements that overflow from the table with $m \geqslant 1$ are the ones that overflow from a table of size $m-1$ plus the number of elements that hash into position m minus b (giving the factor $\frac{w^b e^{zq}}{q^b}$, corresponding to adding a last bucket, marking the elements that hash into this last bucket, and leaving b elements in it). However, we have to include a correction factor in case that the total number of elements that probe position m is less than b. As a consequence

$$\Omega(z,w,q) = 1 + \Omega(z,w,q)\frac{w^b e^{zq}}{q^b} + \sum_{s=0}^{b-1}(1 - q^{s-b}) O_s(z,w),$$

where $O_s(z,w)$ is the generating function for the number of hash tables that have s elements in bucket m.

From [13] we know that

$$O_s(z,w) = \frac{F_s(zw)w^{b-s}}{1 - N_0(z,w)},$$

and the result follows. □

For the probabilistic version, we use Lemma 5.1 and study in the sequel infinite hashing on $\mathbb{Z}$, with $X_i = X_{\alpha;i}$ i.i.d. random Poisson variables with $X_i \sim \text{Po}(\alpha b)$, where $0 < \alpha < 1$. Thus X_i has the probability generating function

$$\psi_X(z) := \mathbb{E}\, z^{X_i} = e^{\alpha b(z-1)}. \tag{6.3}$$

We begin by finding the distributions of H_i and Q_i. Let $\psi_H(z) := \mathbb{E}\, z^{H_i}$ and $\psi_Q(z) := \mathbb{E}\, z^{Q_i}$ denote the probability generating functions of H_i and Q_i (which obviously do not depend on $i \in \mathbb{Z}$), defined at least for $|z| \leqslant 1$.

Theorem 6.2 *Let $0 < \alpha < 1$. The probability generating functions $\psi_H(z)$ and $\psi_Q(z)$ extend to meromorphic functions given by*

$$\psi_H(z) = \frac{b(1-\alpha)(z-1)}{z^b e^{\alpha b(1-z)} - 1} \frac{\prod_{\ell=1}^{b-1}\bigl(z - T\bigl(\omega^\ell \alpha e^{-\alpha}\bigr)/\alpha\bigr)}{\prod_{\ell=1}^{b-1}\bigl(1 - T\bigl(\omega^\ell \alpha e^{-\alpha}\bigr)/\alpha\bigr)}, \tag{6.4}$$

$$\psi_Q(z) = \frac{b(1-\alpha)(z-1)}{z^b - e^{\alpha b(z-1)}} \frac{\prod_{\ell=1}^{b-1}\bigl(z - T\bigl(\omega^\ell \alpha e^{-\alpha}\bigr)/\alpha\bigr)}{\prod_{\ell=1}^{b-1}\bigl(1 - T\bigl(\omega^\ell \alpha e^{-\alpha}\bigr)/\alpha\bigr)}. \tag{6.5}$$

The formula (6.5), which easily implies (6.4), was shown by the combinatorial method in [13, Theorem 9]. It can also be obtained from Theorem 6.1; we omit the details.

Corollary 6.3 *For $k = 0, \dots, b-1$,*

$$\Pr(Y_i = k) = \Pr(H_i = k) = -b(1-\alpha)\frac{[z^k]\prod_{\ell=0}^{b-1}\bigl(z - T\bigl(\omega^\ell \alpha e^{-\alpha}\bigr)/\alpha\bigr)}{\prod_{\ell=1}^{b-1}\bigl(1 - T\bigl(\omega^\ell \alpha e^{-\alpha}\bigr)/\alpha\bigr)}. \tag{6.6}$$

Furthermore, the probability that a bucket is not full is given by

$$\Pr(Y_i < b) = \Pr(H_i < b) = T_0(b\alpha) = \frac{b(1-\alpha)}{\prod_{\ell=1}^{b-1}\bigl(1 - T\bigl(\omega^\ell \alpha e^{-\alpha}\bigr)/\alpha\bigr)} \tag{6.7}$$

and thus

$$\Pr(Y_i = b) = \Pr(H_i \geqslant b) = 1 - T_0(b\alpha). \tag{6.8}$$

The generating functions $T_d(u)$ defined in [13] for $0 \leqslant d \leqslant b-1$ have the property [13, p. 318] that $T_d(b\alpha)$ is the limit of the probability that a given bucket contains more than d empty slots, when $m \to \infty$ and $n \sim \mathrm{Po}(\alpha b m)$. By Lemma 5.1, this limit equals the probability that a given bucket in the infinite hashing has more than d empty slots. This gives the following relation.

Theorem 6.4 *For $d = 0, \dots, b-1$,*

$$T_d(b\alpha) = \Pr(Y_i < b-d) = \Pr(H_i < b-d) = \sum_{s=0}^{b-d-1} \Pr(Y_i = s), \tag{6.9}$$

It is easy to verify that the formula (6.6) is equivalent to [13, Theorem 8].

7 Robin Hood displacement

In Robin Hood, if ties are broken in a consistent way (e.g. by hash value) then the final table is the same, independently from the sequence of insertions. As a consequence, the last inserted element, has the same distribution as any other key. Let D^{RH} be the displacement of a given element x; we may assume that x hashes to bucket 0. We first study the number C^{RH} of elements that win over x in the competition for slots in the buckets; then $D^{\mathsf{RH}} = \lfloor C^{\mathsf{RH}}/b \rfloor$. As in [13], we note that C^{RH} is the sum of the number Q_{-1} of elements that overflow into 0 plus the number V of elements that hash to 0 that win over x; if there are k other elements hashing to 0, then V is by symmetry uniformly distributed in $\{0, \dots, k\}$, and has probability generating function $\frac{1}{k+1}\sum_{r=0}^{k} q^r$.

In the combinatorial model, the generating function $\mathcal{C}^{\mathsf{RH}}(z,w,q)$ of C^{RH} thus factors as $\Omega(z,w,q)$ times the generating function for V. The latter, as presented in Section 3, is given by the specification $\mathrm{Ave}(\mathrm{Pos}(\mathrm{Bucket}))$. We then arrive at

$$\begin{aligned}\mathcal{C}^{\mathsf{RH}}(bz,w,q) &= \Omega(bz,w,q)\,\mathrm{Ave}(\mathrm{Pos}(\mathrm{Bucket}(bz,w,q)))\\ &= \Omega(bz,w,q)w^b\frac{e^{bz}-e^{qbz}}{bz(1-q)} = \frac{(we^z)^b(1-e^{bz(q-1)})}{bz(1-q)(q^b-we^{qz})}\frac{\prod_{j=0}^{b-1}\left(q-\frac{T(\omega^j zw)}{z}\right)}{\prod_{j=0}^{b-1}\left(1-\frac{T(\omega^j zw)}{z}\right)}.\end{aligned}$$

The probabilistic argument for the infinite Poisson model is very similar. Again we have $C^{\mathsf{RH}} = Q_{-1}+V$, where Q_{-1} and V are independent, and a simple calculation shows that V has probability generating function $\psi_V(q) = \big(1-e^{b\alpha(q-1)}\big)/b\alpha(1-q)$. Using (6.5), this yields

$$\psi_C(q) = \psi_Q(q)\psi_V(q) = \frac{1-\alpha}{\alpha}\,\frac{1-e^{b\alpha(q-1)}}{e^{b\alpha(q-1)}-q^b}\,\frac{\prod_{\ell=1}^{b-1}\big(q-T\big(\omega^\ell\alpha e^{-\alpha}\big)/\alpha\big)}{\prod_{\ell=1}^{b-1}\big(1-T\big(\omega^\ell\alpha e^{-\alpha}\big)/\alpha\big)}. \tag{7.1}$$

The probability generating function for the displacement $D^{\mathsf{RH}} = \lfloor C^{\mathsf{RH}}/b\rfloor$ then equals, cf. [13],

$$\psi_{\mathsf{RH}}(q) = \frac{1}{b}\sum_{j=0}^{b-1}\psi_C\big(\omega^j q^{1/b}\big)\frac{1-q^{-1}}{1-\omega^{-j}q^{-1/b}}. \tag{7.2}$$

8 Block length

In an almost full table the length of the block is marked by w in $N_0(bz,w)$. Then, in the combinatorial model, the generating function $B(z,w,q)$ for the block length is

$$B(bz,w,q) = \Lambda_0(bz,w)N_0(bz,wq^{1/b}) = \frac{1-\prod_{j=0}^{b-1}\left(1-\frac{T(\omega^j zwq^{1/b})}{z}\right)}{\prod_{j=0}^{b-1}\left(1-\frac{T(\omega^j zw)}{z}\right)}.$$

For the probabilistic version, we consider one-sided infinite hashing on $\mathfrak{T} = \mathbb{N}$, with $X_i \sim \mathrm{Po}(\alpha b)$ i.i.d. as above. Let B be the length of the first block, i.e.,

$$B := \min\{i \geqslant 1 : Y_i < b\} = \min\{i \geqslant 1 : H_i < b\}. \tag{8.1}$$

Hence, B is the first positive index i such that the number of elements $S_i = X_1+\cdots+X_i$ hashed to the i first buckets is less than the capacity bi of these buckets, i.e.,

$$B = \min\{i \geqslant 1 : S_i < bi\}. \tag{8.2}$$

(This also follows from Lemma 4.1.) In other words, if we consider the random walk

$$S_n' := S_n - bn = \sum_{i=1}^{n}(X_i - b), \tag{8.3}$$

the block length B is the first time this random walk becomes negative. Since $\mathbb{E}(X_i - b) = \alpha b - b < 0$, it follows from the law of large numbers that almost surely $S'_n \to -\infty$ as $n \to \infty$, and thus $B < \infty$.

Note also that $S'_{B-1} \geqslant 0$, and thus $0 > S'_B \geqslant -b$. In fact, the number of elements hash to the first B buckets is $S_B = S'_B + bB$, and since all buckets before B are full and thus take $(B-1)b$ elements, the number of elements in the final bucket of the block is

$$Y_B = H_B = S_B - (B-1)b = S'_B + b \in \{0, \dots, b-1\}. \tag{8.4}$$

Theorem 8.1 *The probability generating function $\psi_B(z) := \mathbb{E}\, z^B$ of B is given by*

$$\psi_B(z) = 1 - \prod_{\ell=0}^{b-1} \Bigl(1 - T\bigl(\omega^\ell \alpha e^{-\alpha} z^{1/b}\bigr)/\alpha\Bigr). \tag{8.5}$$

More generally,

$$\mathbb{E}\bigl(z^B t^{Y_B}\bigr) = \mathbb{E}\bigl(z^B t^{H_B}\bigr) = t^b - \prod_{\ell=0}^{b-1} \Bigl(t - T\bigl(\omega^\ell \alpha e^{-\alpha} z^{1/b}\bigr)/\alpha\Bigr). \tag{8.6}$$

9 Unsuccessful search

In a cluster with n keys, the number of visited buckets in a unsuccessful search, is the same as the one needed to insert the $(n+1)$st element. As a consequence, in the combinatorial model, the specification $\mathrm{Pos}(N_0)$ leads, from equation (3.3), to

$$U(bz, w, q) = \Lambda_0(bz, w) \frac{N_0(bz, w) - N_0(bz, wq^{1/b})}{1-q} = \frac{\prod_{j=0}^{b-1}\Bigl(1 - \frac{T(\omega^j zwq^{1/b})}{z}\Bigr) - \prod_{j=0}^{b-1}\Bigl(1 - \frac{T(\omega^j zw)}{z}\Bigr)}{(1-q)\prod_{j=0}^{b-1}\Bigl(1 - \frac{T(\omega^j zw)}{z}\Bigr)}.$$

This result is also derived in [1, Lemma 4.2].

In the probabilistic model, for an unsuccessful search for an element that does not exist in the hash table, let $U_i \geqslant 0$ denote the number of *full* buckets that we search, when we start with bucket i. Thus $U_i = k - i$ where k is the index of the bucket that ends the block containing i. In the probabilistic version, we consider again hashing on $\mathbb{Z}$, with $X_i \sim \mathrm{Po}(\alpha b)$ independent. Obviously, all U_i have the same distribution, so we may take $i = 0$.

Theorem 9.1 *The probability generating function $\psi_U(z) := \mathbb{E}\, z^{U_i}$ of U_i is given by*

$$\psi_U(z) = \frac{T_0(b\alpha)}{1-z} \prod_{\ell=0}^{b-1} \Bigl(1 - T\bigl(\omega^\ell \alpha e^{-\alpha} z^{1/b}\bigr)/\alpha\Bigr). \tag{9.1}$$

10 FCFS displacement

In the combinatorial model, from section 9, $U(bz, w, q) = \sum_{m\geq 1} w^{bm} \sum_{n\geq 0} \frac{(bmz)^n}{n!} P_{m,n}(q)$, where $P_{m,n}(q)$ is the probability generating function for the displacement of the $(n+1)$st inserted element. The generating function for the displacement of a random element when having $n+1$ elements in the table is $FC_{m,n}(q) := \frac{\sum_{i=0}^{n} P_{m,i}(q)}{n+1}$. We need then a transform $w^{bm} \frac{(bmz)^n}{n!} P_{m,n}(q) \mapsto w^{bm} z^n \frac{\sum_{i=0}^{n} P_{m,i}(q)}{n+1}$.

In this regard, the Laplace transform leads to the *ordinary* generating function

$$\int_0^\infty U(byt, we^{-t}, q)\,dt = \sum_{m\geq 1} \frac{w^{bm}}{bm} \sum_{n\geq 0} y^n P_{m,n}(q).$$

As a consequence we have the ordinary generating function

$$FCFS(bz, w, q) = \sum_{m\geq 1} w^{bm} \sum_{n\geq 0} FC_{m,n}(q) z^n = \frac{w\partial_w}{z} \int_0^z \left(\int_0^\infty U(byt, we^{-t}, q)\,dt\right) \frac{dy}{1-y}. \quad (10.1)$$

In the probabilistic model, when inserting a new element in the hash table with the FCFS rule, we do exactly as in an unsuccessful search, except that at the end we insert the new element. Hence the displacement of a new element has the same distribution as U_i in Section 9. However (unlike the RH rule), the elements are never moved once they are inserted, and when studying the displacement of an element already in the table, we have to consider U_i at the time the element was added.

We consider again infinite hashing on $\mathbb{Z}$, and add a time dimension by letting the elements arrive to the buckets by independent Poisson process with intensity 1. At time $t \geqslant 0$, we thus have $X_i \sim \mathrm{Po}(t)$, so at time αb we have the same model as before, but with each element given an arrival time, with the arrival times being i.i.d. and uniformly distributed on $[0, b\alpha]$. (We cannot proceed beyond time $t = b$; at this time the table becomes full and an infinite number of elements overflow to $+\infty$; however, we consider only $t < b$.)

Consider the table at time αb, containing all element with arrival times in $[0, \alpha b]$. We are interested in the FCFS displacement of a "randomly chosen element". Since there is an infinite number of elements, this is not well-defined, but we can interpret it as follows (which gives the correct limit of finite hash tables): By a basic property of Poisson processes, if we condition on the existence of an element, x say, that arrives to a given bucket i at a given time t, then all other elements form a Poisson process with the same distribution as the original process. Hence the FCFS displacement of x has the same distribution as U_i, computed with the load factor α replaced by $\beta := t/b$. Furthermore, as said above, the arrival times of the elements are uniformly distributed in $[0, \alpha b]$, so β is uniformly distributed in $[0, \alpha]$. Hence, the FCFS displacement D^{FC} of a random element is (formally by definition) a random variable with the distribution

$$\Pr(D^{\mathsf{FC}} = k) = \frac{1}{\alpha} \int_0^\alpha \Pr\bigl(U_i(\beta) = k\bigr)\,\mathrm{d}\beta, \quad (10.2)$$

where $U_i(\beta)$ means U_i with the load α replaced by β. This leads to the following, where we now write α as an explicit parameter of all quantities that depend on it.

Theorem 10.1 *The probability generating function $\psi_{\mathsf{FC}}(z) := \mathbb{E}\, z^{D_i^{\mathsf{FC}}}$ of D_i^{FC} is given by*

$$\begin{aligned}\psi_{\mathsf{FC}}(z;\alpha) &= \frac{1}{\alpha}\int_0^\alpha \psi_U(z;\beta)\,\mathrm{d}\beta = \frac{1}{\alpha}\int_0^\alpha \frac{\tau(\beta)}{1-z}\prod_{\ell=0}^{b-1}\bigl(1-\zeta_\ell(z;\beta)\bigr)\,\mathrm{d}\beta \\ &= \frac{1}{\alpha}\int_0^\alpha \frac{b(1-\beta)\prod_{\ell=0}^{b-1}\bigl(1-\zeta_\ell(z;\beta)\bigr)}{(1-z)\prod_{\ell=1}^{b-1}(1-\zeta_\ell(1;\beta))}\,\mathrm{d}\beta.\end{aligned} \quad (10.3)$$

Acknowledgements

Philippe Flajolet has had a strong influence in our scientific careers. The core of the use of the symbolic method in hashing problems has been taken from [4]. Thank you Philippe for all the work you have left to inspire our research. We also thank Alois Panholzer for interesting discussions.

References

[1] Ian F. Blake and Alan G. Konheim, Big buckets are (are not) better! *J. Assoc. Comput. Mach.* **24** (1977), no. 4, 591–606.

[2] Philippe Chassaing and Philippe Flajolet, Hachage, arbres, chemins & graphes. *Gazette des Mathématiciens* **95** (2003), 29–49.

[3] Philippe Chassaing and Svante Janson, A Vervaat-like path transformation for the reflected Brownian bridge conditioned on its local time at 0. *Ann. Probab.* **29** (2001), no. 4, 1755–1779.

[4] Philippe Flajolet, Slides of the lecture "On the Analysis of Linear Probing Hashing", 1998. `http://algo.inria.fr/flajolet/Publications/lectures.html`

[5] Philippe Flajolet, Patricio Poblete and Alfredo Viola, On the Analysis of Linear Probing Hashing. *Algorithmica* **22** (1998), no. 4, 490–515.

[6] Philippe Flajolet and Robert Sedgewick, *Analytic Combinatorics*. Cambridge University Press, 2009.

[7] Svante Janson, Asymptotic distribution for the cost of linear probing hashing. *Random Struct. Alg.* **19** (2001), no. 3–4, 438–471.

[8] Svante Janson and Alfredo Viola, A unified approach to linear probing hashing with buckets. In preparation.

[9] Donald E. Knuth, Linear Probing and Graphs. *Algorithmica* **22** (1998), no. 4, 561–568.

[10] Donald E. Knuth, *The Art of Computer Programming. Vol. 3: Sorting and Searching*. 2nd ed., Addison-Wesley, Reading, Mass., 1998.

[11] Alois Panholzer, Slides of the lecture "Asymptotic results for the number of unsuccessful parkers in a one-way street", 2009. `http://info.tuwien.ac.at/panholzer/`

[12] Robert Sedgewick and Philippe Flajolet, *An Introduction to the Analysis of Algorithms*. Addison-Wesley, Reading, Mass., 1996.

[13] Alfredo Viola, Distributional analysis of the parking problem and Robin Hood linear probing hashing with buckets. *Discrete Math. Theor. Comput. Sci.* **12** (2010), no. 2, 307–332.