JOURNAL OF COMPUTER AND SYSTEM SCIENCES 16, 226–274 (1978)

The Analysis of Double Hashing

LEO J. GUIBAS

Xerox Palo Alto Research Center, Palo Alto, California 94304

AND

ENDRE SZEMEREDI

Mathematical Institute of the Hungarian Academy of Sciences, Budapest, Hungary

Received October 1, 1976; revised November 11, 1977

In this paper we analyze the performance of *double hashing*, a well-known hashing algorithm in which we probe the hash table along arithmetic progressions where the initial element and the increment of the progression are chosen randomly and independently depending only on the key K of the search. We prove that double hashing is asymptotically equivalent to uniform probing for load factors α not exceeding a certain constant $\alpha_0 = 0.31\ldots$. Uniform hashing refers to a technique which exhibits no clustering and is known to be optimal in a certain sense. Our proof method has a different flavor from those previously used in algorithmic analysis. We begin by showing that the tail of the hypergeometric distribution a fixed percentage away from the mean is exponentially small. We use this result to prove that random subsets of the finite ring of integers modulo m of cardinality αm have always nearly the expected number of arithmetic progressions of length k, except with exponentially small probability. We then use this theorem to start up a process (called the *extension process*) of looking at snapshorts of the table as it fills up with double hashing. Between steps of the extension process we can show that the effect of clustering is negligible, and that we therefore never depart too far from the truly random situation.

1. INTRODUCTION

In this section we introduce the basic notions of hashing and of algorithmic analysis. We define terminology and notation to be used throughout this paper. Finally we present a summary of the results to be proved.

1.1. *Hashing Algorithms*

Hashing algorithms are a certain type of search procedure. We assume that we are given a set of *records*, where each record R is uniquely identified by its *key* K. Besides K the record R contains some unspecified useful information in the field INFO, as depicted in Fig. 1.1.1.

0022-0000/78/0162–0226$02.00/0

We wish to organize our records in such a way that (1) we can quickly find the record having a given key K (if such a record exists), and (2) we can easily add additional records to our collection. Since all retrieval and update requests are specified exclusively in terms of the key of a record, we will ignore the INFO field in most of the discussion that follows. A straightforward way to implement this organization is to maintain our records in a table. A table entry is either *empty*, or it contains one of our records, in which case it is *full*. We can look for a record with a given key by exhaustively examining all entries of the table. Similarly, a new record can be inserted into the table by searching for an empty position. It is clear that, unless we are careful, the searches in question can become quite protected for a large collection of records.

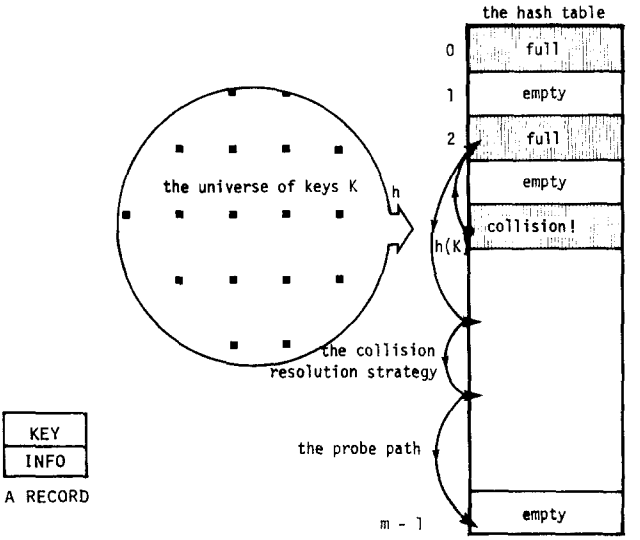


FIG. 1.1.1. The hash function h as a mapping.

The idea of hashing is that of using a transformation h on the key K which gives us a "good guess" as to where in the table the record containing our key K is located. Suppose our table has m entries or positions, numbered 0, 1,..., $m - 1$. Then h maps the universe of keys, which we assume very large, into the set $\{0, 1,..., m - 1\}$. We call h a *hash function*, and depict it as a mapping, as in Fig. 1.1.1.

If $h(K) = s$, then we will say that key K *hashes* to position s. Naturally, several keys may hash to the same position. Thus if we are trying to insert a new key K into the table, it may happen that entry $h(K)$ of the table is already occupied by another key. In that event we need a mechanism for probing the rest of the table until an empty entry is found. We will speak of a probe that encounters a full entry as a *collision*, and we will call our mechanism a *collision resolution strategy*. (It may, of course, happen that we are trying to insert a new key into an already full table, in which case we have an *overflow*.) Upon a retrieval request for the same key, we follow the same probe path until the record containing the key is found.

We will assume that our collision resolution strategy is such that every table position is examined exactly once before we return to the original location. The particular probe path we follow during a search may depend on the key K and the state of the table at that moment, as the examples of the next section will make clear. We will also assume that our hash function selects each of the table entries with equal probability. It is intuitively clear that we want our function h to "randomly scatter" the keys over the entire table as much as possible. We will elaborate on these probabilistic concepts in Section 1.3. For the moment the point we wish to make is that, once the "uniformity" of h has been assumed, the collision resolution strategy alone fully determines the behavior of the algorithm. Thus every hashing algorithm we consider naturally breaks up into two parts: (1) the construction of the hash function h mapping the universe of possible keys into the set $\{0, 1,..., m-1\}$ so that each set member is chosen with approximately equal probability, and (2) the formulation of an efficient collision resolution strategy. Since in this paper we are only concerned with the analysis of the performance of hashing algorithms, we will completely ignore the problem of constructing good hash functions. Similarly, if we use any additional randomizing transformations (hash functions) in the collision resolution strategy, we will only need to know the probability distribution of the values of such transformations. We will not concern ourselves with how such mappings can be explicitly constructed, given a specific universe of keys.

1.2. *Open Address Hash Techniques*

A hashing algorithm is an *open addressing* method if the probe path we follow for a given key K depends only on this key. Thus each key determines a permutation of $\{0, 1,..., m-1\}$ which indicates the sequence in which the table positions are to be examined. Let n denote the number of records currently in the table. Perhaps the two best known open addressing hash algorithms are *linear probing* and *double hashing*. We use the descriptions of these algorithms given in [11].

ALGORITHM L (Linear probing). This algorithm searches an m-node table, looking for a given key K. If K is not in the table and the table is not full, K is inserted.

The nodes of the table are denoted by TABLE[i], for $0 \leqslant i < m$, and they are of two distinguishable types, *empty* and *occupied*. An occupied node contains a key, called KEY[i], and possibly other fields. An auxiliary variable n is used to keep track of how many nodes are occupied; this variable is considered to be part of the table, and it is increased by 1 whenever a new key is inserted.

This algorithm makes use of a hash function $h(K)$, and it uses a linear probing sequence to address the table.

L1 [Hash]. Set $i \leftarrow h(K)$. (Now $0 \leqslant i < m$.)

L2 [Compare]. If KEY[i] $= K$, the algorithm terminates successfully. Otherwise if TABLE[i] is empty, go to L4.

L3 [Advance to next]. Set $i \leftarrow i - 1$; if now $i < 0$, set $i \leftarrow i + m$. Go back to step L2.

L4 [insert]. (The search was unsuccessful.) If $n = m - 1$, the algorithm terminates with overflow. (This algorithm considers the table to be full when $n = m - 1$, not when $n = m$.) Otherwise set $n \leftarrow n + 1$, mark TABLE[i] occupied, and setKEY[i] $\leftarrow K$. ▮

ALGORITHM D (Open addressing with double hashing). This algorithm is almost identical to Algorithm L, but it probes the table in a slightly different fashion by making use of two hash functions $h_1(K)$ and $h_2(H)$. As usual $h_1(K)$ produces a value between 0 and $m - 1$, inclusive; but $h_2(K)$ must produce a value between 1 and $m - 1$ that is *relatively prime* to m. (For example, if m is prime, $h_2(K)$ can be *any* value between 1 and $m - 1$ inclusive; or if $m = 2^p$, $h_2(K)$ can be any *odd* value between 1 and $2^p - 1$.) The probe sequences in this case are arithmetic progressions.

D1 [First hash]. Set $i \leftarrow h_1(K)$.

D2 [First probe]. If TABLE[i] is empty, go to D6. Otherwise if KEY[i] $= K$, the algorithm terminates successfully.

D3 [Second hash]. Set $c \leftarrow h_2(K)$.

D4 [Advance to next]. Set $i \leftarrow i - c$; if now $i < 0$, set $i \leftarrow i + m$.

D5 [Compare]. If TABLE[i] is empty, go to D6. Otherwise if KEY[i] $= K$, the algorithm terminates successfully. Otherwise go back to D4.

D6 [Insert]. If $n = m - 1$, the algorithm terminates with overflow. Otherwise set $n \leftarrow n + 1$, mark TABLE[i] occupied, and set KEY[i] $\leftarrow K$. ▮

We note that the main difference between these two algorithms is that in double hashing the decrement distance c can itself depend on the key K. As we will see later, this additional degree of freedom can have profound effects on the performance.

1.3. *Algorithmic Analysis*

We are concerned with analysing the performance of double hashing. A discussion of how the analysis of specific algorithms relates to computational complexity is given in [12]. We first have to define the cost measure by which we will evaluate the performance. The two usual cost measures are the space and time consumed by the algorithm. In order to make our time costs implementation independent we will use the number of probes made during a lookup as our basic cost function. This accounts, however, for only part of where the running time of a hashing algorithm is spent. The computation of the hash function(s) is another significant component. In comparing algorithms we cannot always factor this component out, as double hashing, for example, uses two hash function computations per search, vs only one for linear probing. Having made this caveat we now strictly confine our attention to the number of probes made.

With any hash function it can happen that all the keys we insert will select the same probe sequence. In this unfortunate situation all the algorithms of the previous section reduce to a linear search of the table. Thus the worst case of hashing methods is not very interesting. We will be concerned with performance on the average. Before we can make

precise the notion of the *average number of probes*, we need to specify the probability distribution of the inputs to our algorithms. We assume that every one of the hash functions we use will select each of its allowed values with equal probability, independently of all the others. Thus for Algorithm L we will assume that $h(K) = s$ $(0 \leqslant s \leqslant m - 1)$ with probability $1/m$. For double hashing we will take m to be prime and then assume that $(h_1(K), h_2(K)) = (i, j)$ with probability $1/m(m-1)$, for all (i, j) with $0 \leqslant i \leqslant m - 1$, $1 \leqslant j \leqslant m - 1$, $i \neq j$.

We now specify what we mean by the number of probes a bit more carefully. Consider the insertion of a new record. We will include in our count the very last probe in which an empty position was discovered. The other probes correspond to comparisons between keys. To avoid monotony of language we will use the terms probe and comparison interchangeably, even though this is misleading when it comes to the last probe. We clearly need to distinguish a successful from an unsuccessful search. We will measure the performance of a hashing algorithm by the following two quantities:

DEFINITION 1.3.1. Given any hashing algorithm we define C_n' to be the average number of probes made into the table when the $(n + 1)$ record is inserted (unsuccessful search). We include in this count the very last probe that discovered the empty position in an open addressing technique. We assume all hash functions involved to choose each of their allowed values independently with equal probability.

Similarly, C_n will denote the average number of comparisons (or probes) made in a successful search using the algorithm, when the table contains n records. For C_n we assume that we are equally likely to look up any record present in the table.

In an open addressing technique it is clear that the number of comparisons required to look up a specific record is the same as the number of probes made when that record was inserted. This observation implies that

$$C_n = (1/n) \sum_{i=0}^{n-1} C_i'.$$

Thus in open addressing C_n is just an average C_n'. For this reason C_n' will be the principal quantity we investigate for such algorithms.

The quantities C_n, C_n' naturally also depend on m, the table size. We will find that a convenient way to express the answers we seek is in terms of the *load* (or *occupancy*) *factor* α of the table, where $\alpha = n/m$. In several cases we will be unable to obtain C_n, C_n' as closed form expressions of n, m. But in these cases we will still be able to obtain formulas for C_n' and C_n as functions of α (and possibly m) that are asymptotically valid. That is, as the table size m gets large, if the load factor α, $0 < \alpha < 1$, stays fixed, these functions of α will differ from the true values by errors of the order of $O(1/m)$, and which therefore rapidly decrease as m increases. In terms of the load factor we may write C_α, C_α' rather than C_n, C_n'. In this "continuous" approximation the above relation between successful and unsuccessful searches for open addressing becomes

$$C_\alpha = (1/\alpha) \int_0^\alpha C_\alpha' \, d\alpha.$$

We will have occasion to appreciate the power of this notation throughout this paper.

1.4. *Clustering*

Since we are interested in the performance of hashing algorithms, we might ask the following question: what is the probability that two keys will follow exactly the same probe path? We can expect that the higher this probability, the more will different keys interfere with each other, and therefore the worse the performance of our algorithm will be. This interference phenomenon we will generally refer to as *clustering*. For example, in linear probing the probability that two keys will follow the same probe path is identical to the probability that they will hash to the same location, which is $1/m$. In double hashing this probabity is easily seen to be $1/m(m-1)$. Thus we expect double hashing to have smaller C_n' (and C_n) than linear probing, as is indeed borne out by the analyses.

Another way to appreciate the effect of clustering is by observing that (loosely speaking) configurations of occupied positions that have a relatively high C_n' grow with a higher probability than configurations with a low C_n'. For example, in linear probing a long block of contiguous occupied positions gives us a large contribution to the total C_n'. During the next insertion the probability that such a block will grow by one is proportional to the length of the block. Thus long blocks grow into even longer ones with higher probability than short ones. This "pile-up" effect accounts for the rapid increase in C_α' for linear probing as $\alpha \to 1$. Similarly, in double hashing the configurations that contribute greatly to the mean C_n' are those that contain a large number of arithmetic progressions among the occupied positions. In general the probability that a given empty position will be filled during the current insertion is proportional to the number of arithmetic progressions coming from the occupied positions to that empty position. Here we have made the convention that we have $m-1$ arithmetic progressions of length 0, so as to properly account for the probability of hitting our position on the first probe. Thus in double hashing, sets of occupied entries with an excessive number of arithmetic progressions will tend to grow into sets with even more progressions.

The connection between clustering and C_n' leads us to introduce a new family of classes of hashing techniques, those that exhibit secondary, tertiary, and in general k-ary clustering [11]. A hashing technique is said to exhibit *secondary* clustering, if the search into the table begins with *one* random probe, and then follows a fixed permutation which depends only on the location of this first probe. A hashing technique is said to exhibit *tertiary* clustering if it begins with *two* independently random probes into the table, and then probes the remaining table positions in a fixed permutation that can depend only on the locations of those first two probes. And in general a k-ary clustering technique begins the search in the table with k independent random probes and then continues along a permutation that depends on the locations of these first k probes only. (It is unfortunate that our terminology is somewhat inconsistent: secondary clustering is 1-ary clustering, tertiary is 2-ary; we have maintained the terms secondary and tertiary for historical reasons.) Thus linear probing exhibits secondary clustering, whereas double hashing exhibits tertiary clustering. More formally, we can think of a secondary

clustering technique as being specified by an $m \times (m-1)$ matrix, where we think of the rows of the matrix as indexed by $\{0, 1, \ldots, (m-1)\}$, and the row corresponding to i is a permutation of $\{0, 1, \ldots, (m-1)\} - \{i\}$ which specifies the order in which the remaining table positions are to be probed. Thus for linear probing we have the matrix depicted. by Fig. 1.4.1.

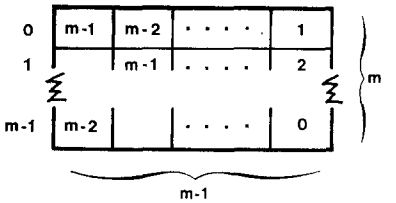


FIG. 1.4.1. The matrix for linear probing.

Similarly, a tertiary clustering technique is defined by an $(m(m-1)) \times (m-2)$ matrix, where we think of the rows as indexed by (i, j), $0 \leqslant i \neq j \leqslant m-1$ and row (i, j) specifies in which order to probe the remaining $m-2$ table positions when we make our first probe at i and our second probe at j. Thus the matrix corresponding to double hashing (assuming that m is prime) is as shown in Fig. 1.4.2, where the rows specify the arithmetic progressions to be followed in the search.

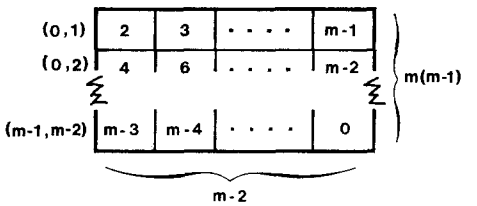


FIG. 1.4.2. The matrix for double hashing.

It is convenient to introduce at this point an open addressing technique that exhibits "no clustering," namely, *uniform hashing* (or *probing*). Uniform hashing has the property that after n keys have been inserted, all $C(m, n)$ possible subsets of occupied positions are equally likely. To achieve this we first prove the table at $h_1(K)$, then at $h_2(K)$ where $h_2(K) \neq h_1(K)$, then at $h_3(K) \neq h_1(K)$, $h_2(K)$, and so on. Here each h_i is assumed to select each of its allowed values with equal probability, independently of all the others. This method is certainly of no practical interest, since we have to compute arbitrarily many independent hash functions. On the other hand it is of theoretical importance, since Ullman has proved that no other open addressing technique can have a smaller C_n' for all n [14]. Thus the performance of uniform hashing can be used as a benchmark against which to measure the success of other open addressing techniques.

The notion of clustering can also help us understand why we wish to make our hash function h uniform, i.e., to make it equally likely to hash to any table entry. Suppose we are dealing with a technique with secondary clustering and let p_i denote the probability

of hashing to entry i, $0 \leqslant i \leqslant m-1$. Then the probability that two keys will follow the same probe path is

$$\sum_{i=0}^{m-1} p_i^2$$

which, since $\sum_{0 \leqslant i < m} p_i = 1$, is clearly minimized by setting $p_0 = p_1 = \cdots = p_{m-1} = 1/m$.

1.5. *Background and Summary of the Results*

The traditional tools of algorithmic analysis are the tools of classical combinatorial enumeration: special numbers (i.e., binomial coefficients, Fibonacci numbers, etc.), recurrence relations, and generating functions. Using these tools a large number of hashing algorithms have been analyzed in the past, and the results are summarized in [11].

Uniform hashing was one of the earliest algorithms analyzed. Its performance is given by

$$C_n' = 1 + n/(m - n + 1),$$

or, in asymptotic terms,

$$C_\alpha' = 1/(1 - \alpha) + O(1/m).$$

The considerably more difficult analysis of linear probing was first carried out correctly by Knuth, who showed that

$$C_\alpha' = (1 + 1/(1 - \alpha)^2)/2 + O(1/m).$$

Thus for load factors near 1 linear probing is quadratically worse than uniform probing.

In [6] we discuss hashing techniques that exhibit k-ary clustering. Among other results we show that *on the whole* (i.e., *averaged* over *all* techniques) k-ary clustering techniques for $k > 1$ are quite good. We prove that if the permutations described in the definition of k-ary clustering for $k > 1$ are randomly chosen, then C_α' is asymptotically $1/(1 - \alpha)$, the same as for uniform probing, which exhibits no clustering. We also analyze "random" secondary clustering ($k = 1$), in which case we find that C_α' is asymptotically $1/(1 - \alpha) - \alpha - \log(1 - \alpha)$. Thus secondary clustering techniques on the average are worse than tertiary (since $\alpha + \log(1 - \alpha) < 0$), although better than linear probing.

In this paper we exclusively concern ourselves with the analysis of double hashing. It has long been known from simulations that double hashing behaves essentially identically with uniform probing:

$$C_\alpha' \sim 1/(1 - \alpha)$$

with agreement to 1 or 2 tenths of a percent even for $m \sim 1000$ (see [1, 2]).

In the following two sections we prove that for $0 < \alpha \leqslant \alpha_0$, where α_0 is an absolute constant, $\alpha_0 \sim 0.319$, this is indeed the case: C_α' for double hashing is $1/(1 - \alpha) + o(1)$. This proof of this result uses techniques that have a different flavor from those previously employed in algorithmic analysis. We cannot appeal to recurrence relations of generating

functions. Instead we use a probabilistic argument to prove that the configurations of αm occupied positions that double hashing gives rise to have almost always nearly the expected number of arithmetic progressions, and thus nearly the expected C_α'. In the proof we will assume that m is prime, although it will be clear (as also pointed out below) that this is not essential.

This equivalence is somewhat surprising, since we would expect double hashing to do substantially worse than uniform hashing. The reason for this is that all probes in the case of uniform hashing are independent, while this is not so for double hashing. In other words, double hashing exhibits clustering; the probability that two keys will follow the same path is $O(1/m^2)$ not "zero" ($O(1/m!)$) as for uniform hashing. The bad configurations for double hashing are sets of occupied positions containing an excessive number of arithmetic progressions. Such sets will tend to grow into sets with even more arithmetic progressions, as a bit of thought will show. So it is by no means true that all sets of n occupied entries are equally likely under double hashing. The sets with an abnormally high number of arithmetic progressions are those that will make C_n' large and are also exactly those most likely to be obtained by double hashing. The effect of our results is to show that the clustering effect is negligible in the limit.

The most outstanding open problem left open by this research is whether one can extend the argument to work for all α, $0 < \alpha < 1$. The proof also can be applied to a modified double hashing algorithm, in which $h_2(K)$ is restricted to a linear segment of the table of size λm, for any fixed λ, $0 < \lambda \leqslant 1$. The number of probes in the modified algorithm can be proven to be asymptotically equal to that of double hashing. This modified algorithm allows us to handle tables of nonprime size. Perhaps we can prove this for $h_2(K)$ restricted in any subset of size λm. A number of purely number theoretic questions about arithmetic in the finite field Z_m of m elements also arise (m is prime). We make the following two conjectures: (1) Let I be fixed, $0 < I < \frac{1}{2}$, $S = \{1,..., m^I\}$, T any m^I-element subset of Z_m, $ST = \{st \mid s \in S, t \in T\}$. Then as $m \to \infty$, there exists a small constant ϵ such that $|ST| \geqslant m^{2I-\epsilon}$; (2) if $0 < x < m^{1/2}$, then no set of x elements of Z_m can have more than $O(x^2/k)$ arithmetic progressions of length k among its members, for any $k = 1, 2,..., x$. Establishing either of these conjectures would prove double hashing equivalent to uniform hashing for all α.

In spirit our techniques are mostly akin to the *probabilistic method* of Erdös [4]. It is to be hoped that this powerful method will be used as successfully in algorithmic analysis as it has been in pure combinatorics.

2. The Abundance of Near-Random Sets

We will use the terms *entry*, *cell*, *slot*, and *point* interchangeably to denote a position of the table. The word *element* or the adjective *occupied* will be used to distinguish the occupied positions. If we consider an arithmetic progression x, $x + d$, $x + 2d,..., x + kd$ (where we interpret all algebraic operations mod m), then d will be called its *distance* and k its *length*. We will speak of it as an arithmetic progression *coming* to x. If $x + d$, $x + 2d,...$, $x + kd$ all lie in some set S, then we will speak of it as an arithmetic progression *from* S.

Given a point x and a set $S \subseteq \{0, 1,..., m-1\}$ of cardinality αm, the expected number of arithmetic progressions of length k coming to x from S is approximately $\alpha^k m$, when we consider all such sets equally likely. This is so because there are $(m-1)$ choices for the distance d and for each such progression we have a probability of

$$C(m-k, \alpha m-k)/C(m, \alpha m) \sim \alpha^k$$

of belonging to S. We begin this section with a study of the tail of the hypergeometric distribution and the Farey subdivision of the circle. Using these results we then prove that except for a fraction of selections of S which is exponentially small, the number of arithmetic progressions of length k coming from S to x will be in the range $\alpha^k m(1 \pm \delta)$, for any small positive δ. This result gives us hope to prove what we want, as it shows that sets with an abnormally high number of arithmetic progressions are exceedingly rare.

2.1. *The Lattice of Arithmetic Progressions Coming From a Set to a Point*

Let Z_m denote the additive group of integers modulo m. We can think of these integers arranged in a circle, with 0 following $m-1$, as depicted in Fig. 2.1.1.

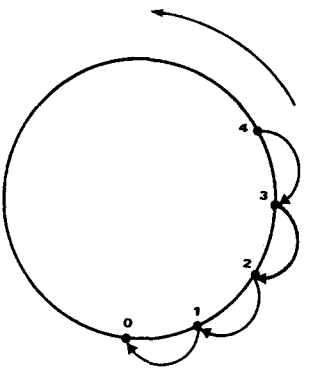


FIG. 2.1.1. The additive group Z_m.

In the entire context of this chapter m is a (sufficiently large) prime number. For any subset $H \subseteq Z_m$ we can count the number of arithmetic progressions that begin at 0 and whose next k elements lie in H. By an arithmetic progression we mean a sequence $x_0, x_1,..., x_k$, such that $x_0 = 0$, $x_1, x_2,..., x_k$ are elements of H, and $x_{i+1} - x_i \pmod m$ is the same for all $i = 1, 2,..., k-1$. (The point 0 need not itself belong to H.) The primality of m guarantees that all the x_i will be distinct. We will speak of such an arithmetic progression as a progression of length k *coming from* H to 0. We can generalize this concept if we allow that for each i, $1 \leqslant i \leqslant k$, we specify whether the corresponding element of the progression is to be in H, or in the complement of H. Thus we arrive at the concept of a type of a progression. A *type* τ of length k can be thought of as a Boolean vector of k bits. An arithmetic progression of length k coming to 0 is of type τ if the ith element of the progression is in H or in the complement of H, according to whether the ith bit of τ is a 1 or a 0. A 1 of the type will also be called a *hit*, whereas a 0 will be called

a *miss* (for obvious reasons). We will display a type by writing down the corresponding bit vector, e.g., $\tau = (10110001)$. Any type τ has a *length* that will be usually denoted by k, and a *number of hits*, that will be usually denoted by l, $0 \leqslant l \leqslant k$. Thus the above type has $k = 8$, $l = 4$. We will reserve the expression "a progression of length k coming from H to 0" to mean a progression of the type $(11 \cdots 1)$ with k hits. For any type τ and set H we can consider all the progressions of that type coming to 0. We will speak of these progressions as *belonging* to τ. For a fixed length k, the set of all types of that length forms a Boolean lattice (or algebra), in the usual way. Figure 2.1.2 illustrates some of the ordering relationships.

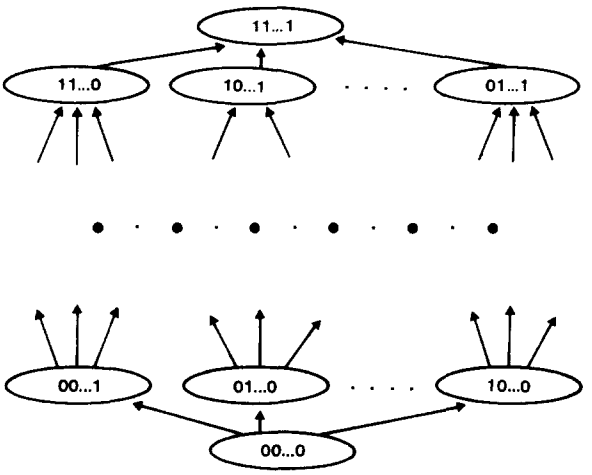


FIG. 2.1.2. The lattice structure of the types of arithmetic progressions.

The above lattice structure will not be important immediately, but will play a significant role in the latter half of this paper.

To fix the ideas let us now confine our attention to arithmetic progressions of length k belonging to the type of all hits. Clearly the number of progressions belonging to this type depends heavily on the set H. We can expect at most to make a probabilistic statement about the distribution of the number of these arithmetic progressions. We will be interested in such an estimation for large m with H of specified cardinality $|H| = \eta m$, where $0 < \eta < 1$. All subsets of this cardinality will be considered equally likely. As we let m get large, we will allow both k and η to vary with m. However, in order to make our argument work, we will see that we have to restrict the growth of k and/or the speed with which η can approach 0 or 1. In this and all subsequent sections, unless otherwise stated our O and o notations will always refer to $m \to \infty$. The implied constants, unless otherwise stated, will be absolute.

What is the expected number of arithmetic progressions of length k coming from H to 0? There are $m - 1$ arithmetic progressions in Z_m coming to 0, one for each possible distance $1, 2, \ldots, m - 1$. Each such progression occurs in H with probability

$$\frac{\eta m(\eta m - 1) \cdots (\eta m - k + 1)}{m(m - 1) \cdots (m - k + 1)} = (1 + o(1))\, \eta^k,$$

if k is suitably small ($\log k < (\frac{1}{2} - \epsilon) \log m$ will do, though in our applications we will always use a k that is $O(\log m)$) and η is bounded away from 0. Thus the expected number of arithmetic progressions of length k coming from H to 0 is $(1 + o(1))\, \eta^k m$. Let δ denote any small positive constant. In the following three sections we will prove that the fraction of choices of H for which the number of these arithmetic progressions is outside the range $\eta^k m(1 \pm \delta)$ is exponentially small in m. By exponentially small we mean that there exist positive constants C, s such that this fraction is

$$\exp[-C\delta^4 \eta^k m/(k^s \log^3(\eta^{-k}\delta^{-1}))]$$

provided $\eta^k m > m^\mu$, where μ denotes any positive constant.

Our method is briefly the following. In Section 2.2 we consider the hypergeometric distribution, which arises when we compute the probability that two subsets of size αm, βm of a set of m elements have an intersection of the expected size $\alpha\beta m$. We show that the probability of the intersection having a size outside the range $\alpha\beta m(1 \pm \epsilon)$ is exponentially small in m. In Section 2.3 we use the Farey series to subdivide the circle formed by the reals (mod 1) into arcs, such that all arcs except those containing certain fixpoints have the property that any two among such an arc's first k multiples are disjoint. By the jth multiple of an arc (interval) $[x, y)$ we mean the arc $[jx, jy)$ (mod 1). In Section 2.4 we use this subdivision, together with our estimation of the tail of the hypergeometric distribution, to prove the desired result.

The idea of Section 2.4 can be illustrated by an example. Suppose $k = 2$ and consider an arc $[x, y)$ which is disjoint from the arc $[2x, 2y)$, as shown in Fig. 2.1.3.

Now suppose we pick H, a random set of ηm points of the circle. What is the expected number of arithmetic progressions of length 2 coming from H to 0 whose first point lies

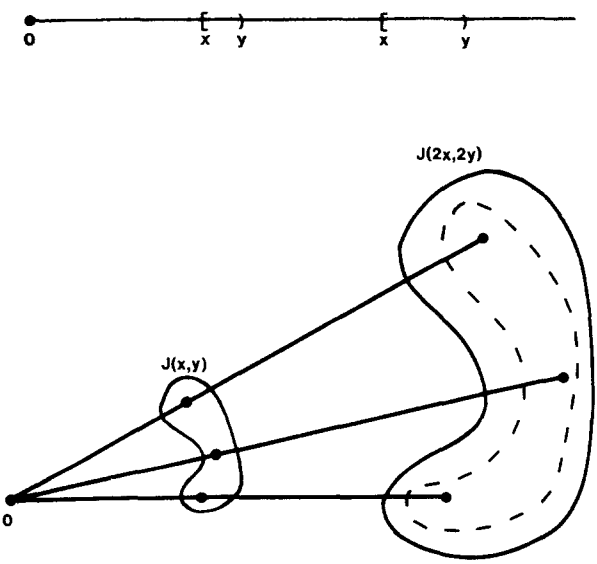


FIG. 2.1.3. An illustration of the "pull-back" argument.

in the set $J(x, y) = \{z \in Z_m \mid x \leqslant (z/m) < y\}$? Instead of repeating the argument given earlier, we can proceed as follows. The interval $J(x, y)$ has size $(y - x)m$. Consider the set $2J(x, y) = \{2z \mid z \in J(x, y)\} \subseteq J(2x, 2y)$. This set also has cardinality $(y - x)m$, and is the locus of the second points of progressions whose first point is in $J(x, y)$. We expect $(y - x)\eta m$ of the points of $2J(x, y)$ to be hit by H. The set of hit points can now be "pulled-back" to $J(x, y)$ to give us the candidate first points of these progressions. Since $J(x, y)$ and $J(2x, 2y)$ are disjoint, the expected number of points in this subset of $J(x, y)$ that will be hit is $(y - x)\,\eta^2 m$. So $(y - x)\,\eta^2 m$ is the desired average. This argument illustrates how we can translate our knowledge of the probabilities for the size of set intersections to probabilities for the occurrence of arithmetic progressions in H.

All of the above remarks apply verbatim to types other than the type with k hits. If our type has l hits then we only need to replace η^k by $\eta^l(1 - \eta)^{k-l}$ everywhere in the above discussion, and restrict η away from both 0 and 1. Circular symmetry implies that our results also hold for any point of Z_m, not just 0. Our method solves the corresponding problem when we do not allow wrap-around or when we specify an upper bound on the number of times we can wrap around. We do not need this result, so we will not dwell on it any longer here. We will, however, need a slight generalization of the case $k = 2$ shown above. Given two points $x, y \in Z_m$, we will say that these points are in the ratio $a : b$ if $xb = ya \pmod m$. Given a fixed ratio $a : b$, $1 \leqslant a, b \leqslant k$, we will want to estimate the number of pairs of points (x, y) of H that are in the ratio $a : b$.

2.2. *The Tail of the Hypergeometric Distribution*

In this section we estimate the tail of the hypergeometric distribution a specified percentage away from the mean. Properties of the hypergeometric distribution are discussed, for example, in [5]. Since we are interested in large deviations, the normal approximation will not be useful to us. Instead we will need an approximation more like the one done for the tail of the binomial distribution in [13].

Suppose we have a sample space of size n, and we select from this space two subsets, one of size αn, the other of size $\beta n (0 < \alpha, \beta < 1)$. The probability that the cardinality of the intersection of the two subsets is k is

$$a_k = C(\alpha n, k)\, C((1 - \alpha)n, \beta n - k)/C(n, \beta n).$$

↗	↗	↖
choose k of βn's from αn	choose others from rest of n	total number of ways to choose

The expected value of k is easily computed to be $\alpha\beta n(1 + o(1))$. We will estimate the probability that k lies outside the range $\alpha\beta n\,(1 \pm \epsilon)$. For this section only, our O notation will refer to $n \to \infty$.

THEOREM 2.2.1. *Let $Y(n, \alpha, \beta, \epsilon)$ denote the probability that if we randomly select two sets of sizes αn and βn, respectively, out of size n, their intersection will have cardinality outside the range $\alpha\beta n(1 \pm \epsilon)$. Then as $n \to \infty$, provided*

$$0 < \alpha, \beta \quad \text{and} \quad \alpha(1 + \epsilon), \beta(1 + \epsilon) < 1,$$

where α, β, ϵ *can vary with* n, *we have*

$$Y(n, \alpha, \beta, \epsilon) \leqslant K(1 + 1/\epsilon)\, e^{-\varphi(\epsilon)\alpha\beta n},$$

where $\varphi(\epsilon) \geqslant (1 + \epsilon)\log(1 + \epsilon) - \epsilon + \frac{1}{2}\epsilon^2[\alpha\beta/(1 - \alpha)(1 - \beta) + \alpha/(1 - \alpha) + \beta/(1 - \beta)]$ *and* K *is an absolute constant.*

The same conclusion holds if one of the two sets in question stays fixed.

Proof. We will first estimate the tail of the distribution above the mean. The tail below can be estimated in an essentially identical fashion.

We wish to estimate the sum

$$\sum_{k \geqslant \alpha\beta n(1+\epsilon)} a_k = \sum_{k \geqslant \alpha\beta n(1+\epsilon)} C(\alpha n, k)\, C((1 - \alpha)\, n, \beta n - k)/C(n, \beta n).$$

Note that the ratio of two successive terms in this sum is

$$a_{k+1}/a_k \leqslant (\alpha n - k)(\beta n - k)/k((1 - \alpha - \beta)n + k),$$

which is a decreasing function of k in the range of interest, i.e., $\alpha\beta n < k < \alpha n, \beta n$.

For $k = \alpha\beta n(1 + \epsilon)$ this ratio is less than

$$\rho = [(\alpha - \alpha\beta - \alpha\beta\epsilon)(\beta - \alpha\beta - \alpha\beta\epsilon)]/[(\alpha\beta + \alpha\beta\epsilon)(1 - \alpha - \beta + \alpha\beta + \alpha\beta\epsilon)].$$

It easily follows that $\rho < 1$. Therefore our sum is majorized by a convergent geometric series of ratio ρ, and we get a bound of

$$[1/(1 - \rho)]\, C[\alpha n, \alpha\beta n(1 + \epsilon)]\, C[(1 - \alpha)n, (\beta - \alpha\beta(1 + \epsilon))n]/C(n, \beta n).$$

Since, as we can easily check,

$$1/(1 - \rho) = 1 + (1 - \alpha - \alpha\epsilon)(1 - \beta - \beta\epsilon)/\epsilon \leqslant 1 + 1/\epsilon.$$

we are only left with estimating the density of the hypergeometric distribution at $k = \alpha\beta n(1 + \epsilon)$, as given above.

We will use Stirling's approximation for the factorial:

$$\log n! = n \log n - n + \tfrac{1}{2} \log n + \tfrac{1}{2} \log 2\pi + O(1/n).$$

From this we can easily derive the following fact:

$$\log C((x + y)n, xn) = (n + \tfrac{1}{2})((x + y)\log(x + y) - x \log x - y \log y) - \tfrac{1}{2} \log n + O(1).$$

We can now apply this fact to the binomial coefficients we have and obtain after simplification

$$\begin{aligned}\log[C(\alpha n, \alpha\beta n(1+\epsilon))\, C((1-\alpha)n, (\beta-\alpha\beta(1+\epsilon))n)/C(n, \beta n)] \\ = -[\alpha\beta(1+\epsilon)\log(1+\epsilon) \\ + \alpha(1+\beta)\{1-[(\beta\epsilon)/(1-\beta)]\}\log\{1-[(\beta\epsilon)/(1-\beta)]\} \\ + \beta(1-\alpha)\{1-[(\alpha\epsilon)/(1-\alpha)]\}\log\{1-[(\alpha\epsilon)/(1-\alpha)]\} \\ + (1-\alpha)(1-\beta)\{1+[(\alpha\beta\epsilon)/(1-\alpha)(1-\beta)]\}\log\{1+[(\alpha\beta\epsilon)/(1-\alpha)(1-\beta)]\}]n \\ + O(1).\end{aligned}$$

The following two inequalities are elementary:

$$(1+x)\log(1+x) \geqslant x \qquad \text{for } x \geqslant 0,$$
$$(1-x)\log(1-x) \geqslant -x + (x^2/2) \qquad \text{for } 0 \leqslant x \leqslant 1.$$

Since $\alpha(1+\epsilon) < 1$ is equivalent to $\alpha\epsilon/(1-\alpha) < 1$, and similarly for β, we can apply these inequalities to the above expression to obtain the upper bound

$$\begin{aligned}&-[\alpha\beta\epsilon + \alpha\beta[(1+\epsilon)\log(1+\epsilon) - \epsilon] \\ &-\alpha\beta\epsilon + \alpha\beta^2\epsilon^2/2(1-\beta) \\ &-\alpha\beta\epsilon + \alpha^2\beta\epsilon^2/2(1-\alpha) \\ &+ \alpha\beta\epsilon]n,\end{aligned}$$

from which the conclusion of the theorem is immediate.

For the lower tail a similar argument gives an upper bound of

$$-[\alpha\beta[(1-\epsilon)\log(1-\epsilon) + \epsilon] + \tfrac{1}{2}\alpha^2\beta^2\epsilon^2/(1-\alpha)(1-\beta)]n + O(1).$$

Now $(1-\epsilon)\log(1-\epsilon) + \epsilon \geqslant (1+\epsilon)\log(1+\epsilon) - \epsilon$ and the theorem follows. ∎

Remark 1. Notice that the above argument does not require ϵ to be small.

Remark 2. If, however, ϵ is small, say $\epsilon \leqslant \epsilon_0$, then $\varphi(\epsilon) \geqslant (1+\epsilon)\log(1+\epsilon) - \epsilon \geqslant C\epsilon^2$ where C depends on ϵ_0. If $\alpha\beta n\epsilon^2/\log(1/\epsilon) > N$, where N is a sufficiently large constant, then we can take C so small that the factor $K(1+1/\epsilon)$ is absorbed in the reduced exponent. Then we can state our conclusion as

COROLLARY 2.2.1. *$Y(n, \alpha, \beta, \epsilon) \leqslant \exp(-C\epsilon^2\alpha\beta n)$ for $\epsilon \leqslant \epsilon_0$, $\alpha\beta n\epsilon^2/\log(1/\epsilon) > N$, N and C positive constants depending on ϵ_0.*

This is the form in which Theorem 2.2.1 will be used most often. In our applications, in fact, $\alpha\beta n\epsilon^2/\log(1/\epsilon)$ will tend to ∞ with n.

The key property of our estimate is that it is exponentially small in n. An estimate obtained by using the variance and Chebycheff's inequality can only give us a bound for this tail that vanishes no faster than an inverse power of n.

2.3. *The Farey Subdivision of the Circle*

The Farey series F_n of order n is the ascending series of irreducible fractions between 0 and 1 whose denominators do not exceed n. For example, F_5 is

$$0/1,\ 1/5,\ 1/4,\ 1/3,\ 2/5,\ 1/2,\ 3/5,\ 2/3,\ 3/4,\ 4/5,\ 1/1.$$

The Farey series possesses many fascinating properties [9].

Property 1. If h/k, h'/k' are two successive terms of F_n, then $kh' - hk' = 1$.

Property 2. If h/k, h''/k'', and k'/k' are three successive terms of F_n, then

$$h''/k'' = (h + h')/(k + k').$$

Property 3. If h/k, h'/k' are two successive terms of F_n, then $k + k' > n$.

Property 4. If $n > 1$, then no two successive terms of F_n have the same denominator.

Property 5. The number of terms in the Farey series of order n is asymptotically $3n^2/\pi^2 + O(n \log n)$.

We will be interested in the circle of the reals (mod 1), denoted by U. The set U forms a group under addition. Consider the mappings

$$T_i\colon x \to ix \pmod 1, \qquad x \in U$$

for each $i = 2, 3,\ldots, k$. (T_1 is the identity.) It should be clear that the *fixpoints* (i.e., points x for which $T_j x = x$ for some j) of these mappings are the fractions a/b with $0 \leqslant a < b < k$. These are exactly the elements of $F_{k-1} \subset U$.

We wish now to partition U into a collection of disjoint intervals (taken to be left closed, right open) J with the property that (1) if $V \in J$, then T_1V, T_2V, $T_3V,\ldots, T_kV$ are all disjoint if V does not contain one of the above fixpoints, and (2) the $V \in J$ that contain a fixpoint can be made arbitrarily small in length.

Let φ_n denote (the cardinality of F_n) $- 1$. We now consider the subdivision of the circle defined by the Farey series F_{2k-2}. Clearly this contains the fixpoints discussed above ($F_{k-1} \subset F_{2k-2}$) and subdivides the circle into φ_{2k-2} intervals. We have

LEMMA 1. *No two fixpoints (i.e., elements of F_{k-1}) are adjacent in F_{2k-2}.*

Proof. If fixpoints h_1/k_1, h_2/k_2 are adjacent in F_{k-1}, then $k_1 + k_2 \leqslant 2k - 2$. Hence by Property 3 they cannot be adjacent in F_{2k-2}. ∎

For $i = 1, 2,\ldots, \varphi_{k-1}$, let us name L_i and R_i, respectively, the intervals of the above subdivision that lie to the "left" and to the "right" of the ith fixpoint (in the standard order).

From Lemma 1 it follows that the other endpoint of each L_i and R_i is not a fixpoint. We name the remaining intervals N_i, $i = 1, 2, \ldots, (\varphi_{2k-2} - 2\varphi_{k-1})$.

LEMMA 2. *Let X stand for one of the L_i or R_i. Then any two of*

$$T_1X, T_2X, \ldots, T_kX$$

will overlap only if they have an endpoint in common.

Proof. Let the ith fixpoint be h_1/k_1 and to make things concrete suppose we are dealing with R_i (the case of L_i is entirely analogous). Let h_2/k_2 be the other endpoint of R_i. Then by Lemma 1, $h_2/k_2 \notin F_{k-1}$, so $k_2 \geqslant k$. Then the length of R_i is

$$(h_2/k_2) - (h_1/k_1) = (1/k_1k_2) \leqslant (1/kk_1).$$

Thus the length of any of the T_jR_i is $\leqslant (1/k_1)$. But all multiples of $(h_1/k_1, h_2/k_2)$ start at a multiple of h_1/k_1. These multiples are spaced at least $1/k_1$ apart, so not two of the multiples of R_i will overlap unless they share a common left endpoint. ∎

LEMMA 3. *Let Y denote one of the N_i. Then the intervals*

$$T_1Y, T_2Y, \ldots, T_kY$$

are all disjoint.

Proof. Suppose intervals A and B in the above sequence intersect. By construction A and B do not share any endpoints. Thus if they intersect, we can assume that the left endpoint of B lies within A. Let Y be $(h_1/k_1, h_2/k_2)$. The distance between the left endpoints of A and B cannot be less than $1/k_1$, since the left endpoints are multiples of $1/k_1$, since $k_2 \geqslant k$ as no endpoint is a fixpoint. But this contradicts our assumption that the left endpoint of B lies within A. ∎

We now describe how to subdivide the L_i and R_i further, so as to make the intervals with an endpoint at a fixpoint as small as we please, while still maintaining the property that all other intervals have disjoint multiples. We describe the construction for R_i, that for L_i being analogous. Let us define for each i a subdivision into intervals RS_{ij}, $j = 1, 2, \ldots, l$, and RM_i. If $R_i = [x, y)$, these subintervals are defined as follows:

$$RS_{ij} = [x + ((k-1)/k)^j(y - x), x + ((k-1)/k)^{j-1}(y - x)), \qquad j = 1, 2, \ldots, l;$$

$$RM_i = [x, x + ((k-1)/k)^l(y - x)).$$

The following facts are then obvious:

(1) $RM_i \cup \bigcup_{j=1}^{l} RS_{ij} = R_i$;

(2) any two of $RS_{i1}, RS_{i2}, \ldots, RS_{il}, RM_i$ are disjoint,

(3) RS_{ij} has length $((k-1)/k)^{j-1}(y - x)/k$, and RM_i has length $((k-1)/k)^l(y - x)$ $(y - x =$ length of $R_i)$.

LEMMA 4. *Let Y denote any of the RS_{ij}. Then the intervals*

$$T_1Y, T_2Y,..., T_kY$$

are disjoint.

Proof. We first prove the lemma for $i = 1$, i.e., for a subdivision of the interval R_1 whose left endpoint is 0 (= 1). As $1/k \in F_{2k-2}$, $R_1 \subset [0, 1/k)$ so we do not have to worry about wrap-around problems for any of the RS_{1j}. To complete our argument we only need show that the right endpoint of the $t - 1$ multiple does not exceed the left endpoint of tth multiple. This is tantamount to

$$(t - 1)((k - 1)/k)^{j-1} \leqslant t((k - 1)/k)^j$$

or

$$(t - 1)/t \leqslant (k - 1)/k,$$

which is certainly true, as t only takes the values $1, 2,..., k$. This subdivision is nicely illustrated by Fig. 2.3.1.

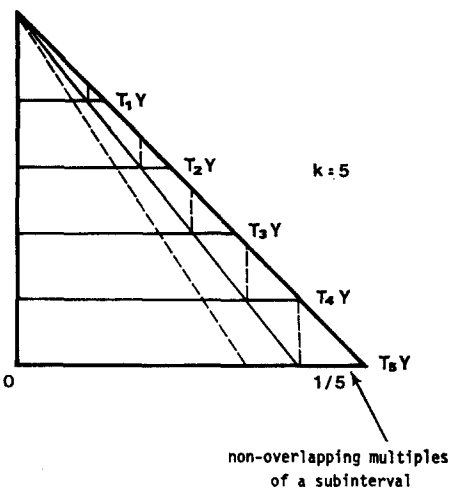


FIG. 2.3.1. The subdivision into intervals of nonoverlapping multiples near a fixpoint.

Now to handle the case of $i > 1$ we need only recall from Lemma 2 that two multiples of R_i overlap only if they have a common left endpoint. But then the situation at each such endpoint is a subcase of the situation described above for $i = 1$ around 0. So by the same argument the multiples of ES_{ij} are disjoint. ∎

We can of course repeat the whole construction and the proof of Lemma 4 for the L_i. Thus we obtain intervals LS_{ij}, $j = 1, 2,..., l$, and LM_i that also satisfy (1), (2), and (3) above.

Before we recapitulate what we have derived in this section we need to make a comment about the lengths of the intervals. Each R_i or L_i, being an interval $[h_1/k_1, h_2/k_2)$ between

two elements of F_{2k-2}, has length $1/k_1k_1 \geqslant 1/4k^2$. On the other hand, either k_1 or k_2 is $\geqslant k$, so the length is at most $1/k$.

Combining all our constructions we have the following theorem.

THEOREM 2.3.1. *We can construct a partition of the reals* mod 1 *into disjoint subintervals*

$$N_i, \quad i = 1, 2, \ldots, \varphi_{2k-2} - 2\varphi_{k-1},$$

$$LS_{ij}, LM_i, RS_{ij}, RM_i, \quad i = 1, 2, \ldots, \varphi_{k-1}, j = 1, 2, \ldots, l$$

so that

(1) *each of the* N_i, LS_{ij}, RS_{ij} *has* (a) *disjoint first* k *multiples and* (b) *length at least* $((k-1)/k)^{l-1}/4k^2$, *and*

(2) *each of the* LM_i, RM_i *has* (a) *an endpoint* (*the right of left one, respectively*) *which is a fixpoint and* (b) *length at most* $((k-1)/k)^l/k$.

2.4. *The estimation of the Arithmetic Progressions, and the Prevalence of Randomness*

We first map intervals on U to intervals on Z_m. Corresponding to an interval $[x, y) \subset U$ we have the set of all $i \in Z_m$. Corresponding to an interval $[x, y) \subset U$ we have the set of all $i \in Z_m$ with the property that $x \leqslant (i/m) < y$. (This should be interpreted cyclically; that is, if $x > y$, then we mean $x \leqslant i/m$ *or* $i/m < y$). We will now use the names of the intervals introduced in Section 2.3 to denote also the corresponding intervals in Z_m. For an interval $T = [x, y)$ we will denote by t its length in U ($t = (y - x)$) and by $|T|$ the number of integers it contains in Z_m. Clearly we have $|T|$ = (number of i such that $xm \leqslant i < ym$) $= [ym] - [xm]$. Thus

$$|T| = tm \pm 2.$$

We will write this as

$$|T| = tm(1 \pm \epsilon), \tag{i}$$

where ϵ will be a quantity used below. This is justified as long as ϵ is not too small. As we will see below, the smallest ϵ we will use will be $\Omega(1/(\log m)^r)$ for some positive r, while the smallest t will be such that $tm \geqslant m^\lambda$ for some positive λ. So as $m \to \infty$, Eq. (i) is justified; its form will simplify some of the computation below.

Recall that we are selecting a random subset H of Z_m of cardinality ηm. For any interval (in fact any subset) T of Z_m, we can ask for the number of elements of H that will fall in T. From Theorem 2.2.1 we know that the number of these elements will be $|T|\eta(1 \pm \epsilon)$, except with probability $Y(m, |T|/m, \eta, \epsilon)$, i.e., it will be $\eta tm(1 \pm \epsilon)^2$ except with probability $Y(m, t(1 \pm \epsilon), \eta, \epsilon)$.

Consider now the collection D of intervals composed of all the N_i, LS_{ij}, RS_{ij}, and the collection C composed of these same intervals *and* their first k multiples. In this latter case we are dealing with a total of $k(2\varphi_{k-1}l + \varphi_{2k-2} - 2\varphi_{k-1})$ intervals. For each interval T in the collection C we assume that H will intersect it in $\eta tm(1 \pm \epsilon)^2$ elements, as in the

discussion above. This will always be the case except for a fraction of choices of H that is bounded by

$$Q = \sum_{T \in C} Y(m, t(1 \pm \epsilon), \eta, \epsilon).$$

(This argument does not need any independence assumptions concerning the various choices of T.) Thus with probability $1 - Q$, our choice of H will intersect each interval in the collection about as often as we expect.

We now restrict T to be one of the elements of D. In the sequel ϵ will denote a small quantity that will define all our relative errors. We allow $\epsilon \to 0$ as $m \to \infty$, and we also allow ϵ to depend on our choice for T. (We write ϵ_T when we need to make this dependency explicit.) Let us consider the first k multiples of T, and focus our attention on the last one T_kT. This interval has $tkm(1 \pm \epsilon)$ elements, and will almost certainly receive $tk\eta m(1 \pm \epsilon)^2$ elements of H. Within T_kT we have a subset S_k of cardinality $tm(1 \pm \epsilon)$, consisting of those elements that are k-fold multiples of elements of T. How many elements of this subset will H hit? (Note that these points are the endpoints of arithmetic progressions starting at 0, having their first element in T,..., and their kth in T_kT.) Now within T_kT itself we can invoke Theorem 2.2.1 to show that, except for a fraction of possibilities that does not exceed $Y(tkm(1 \pm \epsilon), (1/k)(1 \pm \epsilon)^2, \eta(1 \pm \epsilon)^3, \epsilon)$ the number of elements of S_k that H will hit will be $\eta tm(1 \pm \epsilon)^7$.

Consider now the $\eta tm(1 \pm \epsilon)^7$ progressions thus specified. We apply the "pull-back" process illustrated in Section 2.1 and in Fig. 2.4.1. What about the $k - 1$ points of these progressions—how many of these points will be hit by H? By construction, all these points from a subset S_{k-1} of $T_{k-1}T$, an interval *disjoint* from T_kT. By Theorem 2.2.1

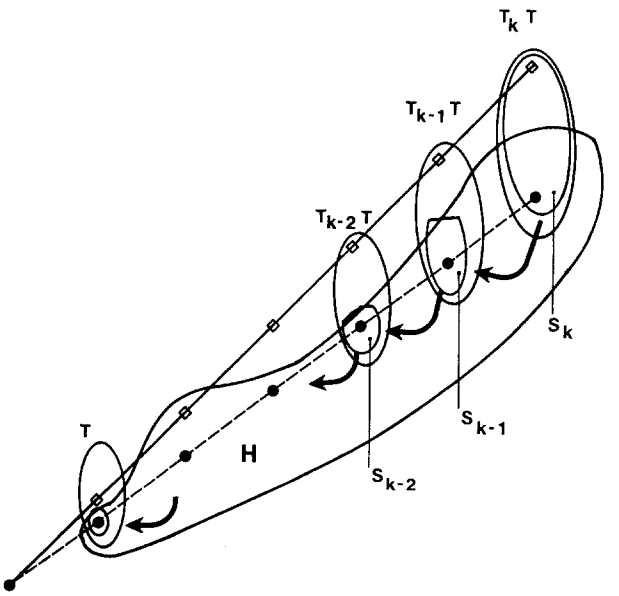


FIG. 2.4.1. The pull-back process.

confined now to the interval $T_{k-1}T$ we see that the intersection of S_{k-1} and H will be $\eta^2 t(1 \pm \epsilon)^{13}m$ points, except with probability $Y(t(k-1)m(1 \pm \epsilon), (\eta/(k-1))(1 \pm \epsilon)^3, \epsilon)$. (To amplify, we have here an interval of $t(k-1)m(1 \pm \epsilon)$ points; the set S_{k-1} is of size $(\eta/(k-1))(1 \pm \epsilon)^8$ times the size of $T_{k-1}T$; and $\eta(1 \pm \epsilon)^3$ is the probability that a point in $T_{k-1}T$ will be a hit by H. The basic rule we are using throughout is that if $x = X(1 \pm \epsilon)^i$, $y = Y(1 \pm \epsilon)^j$, then $xy = XY(1 \pm \epsilon)^{i+j}$, $x/y = (X/Y)(1 \pm \epsilon)^{i+j}$. To make these rules precise it is best to define $x = X(1 \pm \epsilon)$ to mean $x \in (X(1-\epsilon), X/(1-\epsilon))$. Note that this redefinition of $1 \pm \epsilon$ leaves Theorem 2.2.1 valid.)

We now have a set of progressions whose last two elements are guaranteed to be hit by H. At the next step we consider the $k-2$ points of these $\eta^2 tm(1 \pm \epsilon)^{13}$ progressions, which define a subset S_{k-2} of $T_{k-2}T$. By analogous computation we obtain that $\eta^3 tm(1 \pm \epsilon)^{16}$ of these points will belong to our random set H, except with probability $Y((k-2)tm(1 \pm \epsilon), (\eta^2/(k-2))(1 \pm \epsilon)^{14}, \eta(1 \pm \epsilon)^3, \epsilon)$. We now continue in this fashion with the $(k-3),\ldots, 1$ points of the arithmetic progressions. Figure 2.4.1 depicts this *pull-back* process in which successively commit H in the intervals T_iT, $i = k, k-1,\ldots, 1$. At the last step of this process we are considering T itself. After that step the number of candidate arithmetic progressions left will be $\eta^k tm(1 \pm \epsilon)^{6k+1}$. These are now confirmed to be entirely in H. The fraction of choices for H that we have eliminated in this process is bounded by the sum

$$\sum_{i=0}^{k-1} Y((k-i)\, tm(1 \pm \epsilon), (\eta^i/(k-1))(1 \pm \epsilon)^{6i+2}, \eta(1 \pm \epsilon)^3, \epsilon)$$

of all the excluding probabilities.

We now conceive of this process of selection of candidate arithmetic progressions as being carried out for T referring in turn to each of the intervals in D. The total fraction of choices for H thus excluded is bounded by

$$W = \sum_{T \in D} \sum_{i=0}^{k-1} Y((k-i)\, tm(1 \pm \epsilon_T), (\eta^i/(k-1))(1 \pm \epsilon_T)^{6i+2}, \eta(1 \pm \epsilon_T)^3, \epsilon_T).$$

What has all this accomplished? After excluding the choices for H accounted for in Q and W, we can be sure that the number of arithmetic progressions of length k coming from H to 0, and whose first point is T, is $\eta^k tm(1 \pm \epsilon_T)^{6k+1}$, where T is any of the above intervals. Thus the total number of arithmetic progressions coming to 0 from H of length k is

$$\sum_{T \in D} \eta^k tm(1 \pm \epsilon_T)^{6k+1} + E$$

where E is a correction coming from the fact that we cannot apply our argument to the fixpoint intervals LM_i and RM_i. But each of these special intervals cannot contribute more arithmetic progressions than its length. Thus the error committed is bounded by

$$0 \leqslant |E| \leqslant 2\varphi_{k-1}[(1/k)((k-1)/k)^t m + 2].$$

(See Theorem 3.3.1 and recall that there are $2\varphi_{k-1}$ such intervals.)

Now let δ be a given small positive constant. We choose l to be the smallest positive integer such that

$$2\varphi_{k-1}(1/k)((k-1)/k)^l m \leqslant \eta^k m\delta/4. \tag{ii}$$

Thus $l = \lceil (k \log(1/\eta) + \log \varphi_{k-1} - 3 \log 2 + \log(1/\delta))/(\log((k-1)/k)) \rceil$. By choosing δ small enough, and using the fact that $|\log(1-1/k)| \geqslant C/k$ for $k > 1$ and some positive constant C, we see that l will always satisfy

$$l \geqslant 2k. \tag{iii}$$

Since $\eta^k m > m^\mu$ it is also clear that for m sufficiently large

$$(1/k)((k-1)/k)^l m \geqslant 2,$$

and so

$$|E| \leqslant 2\varphi_{k-1}[(1/k)((k-1)/k)^l m + 2] \leqslant \eta^k m\delta/2. \tag{iv}$$

The total number of intervals in our partition is then

$$F = 2\varphi_{k-1}(l+1) + \varphi_{2k-2} - 2\varphi_{k-1} = \varphi_{2k-2} + 2l\varphi_{k-1}.$$

If t denotes the length of an interval in our collection D, then from Theorem 2.3.1 we have

$$\begin{aligned} t &\leqslant ((k-1)/k)^{l-1}/4k^2 \geqslant \eta^k\delta/(32\varphi_{k-1}k(k-1)) \\ &\geqslant S\eta^4\delta/k^4 \qquad \text{for some positive constant } S. \end{aligned}$$

We can therefore write

$$S\eta^4\delta/k^4 \leqslant t \leqslant 1/k. \tag{v}$$

(See Properties 3 and 5 of Section 2.3.)

For an interval $T \in D$ of length t, let ϵ_T be defined by

$$1/(1-\epsilon_T) = (1 + \delta/(2tF))^{1/(6k+1)},$$

so we assign larger relative errors to smaller intervals.

Now we are ready to total the number of arithmetic progressions we have of length k coming from H to 0. This number cannot exceed

$$\eta^k m(\delta/2) + \sum_{T\in D} \eta^k mt(1 + \delta/(2Ft)) \leqslant \eta^k m(\delta/2) + \eta^k m + \eta^k m(\delta/2) \leqslant \eta^k m(1+\delta).$$

in order to obtain a lower bound we use the elementary inequality

$$(1+x)^{-y} \geqslant (1-x)^y \qquad \text{for} \quad 1 \geqslant x, y \geqslant 0.$$

Then we have

$$1 - \epsilon_T = (1 + \delta/2tF)^{-1/(6k+1)} \geqslant (1 - \delta/2tF)^{1/(6k+1)}.$$

We must avoid, however, situations where $\delta/2tF$ comes too close to 1. We stipulate therefore that we will not attempt to maintain a lower bound during the pull-back process for an interval T, unless $tF \geqslant \delta$. Thus we will ignore lower bounds for intervals T much smaller than the average (the average length of an interval is $1/F$). As our intervals form sequences with lengths in geometric progressions, we expect that the total length of the uncontrolled intervals (we include in this the LM_i and RM_i) will be small. First of all it is easy to see that if δ is small enough then none of the intervals N_i will violate the condition $tF \geqslant \delta$. In each sequence the total length violating our condition is certainly less than

$$(\delta/F) \sum_{i=0}^{\infty} ((k-1)/k)^i = \delta k/F$$

for a total contribution not exceeding

$$\delta 2\varphi_{k-1} k/F.$$

But we have $F \geqslant 2l\varphi_{k-1} \geqslant 4k\varphi_{k-1}$ by (iii), and so the total length of the uncontrolled intervals does not exceed $\delta/2$. Therefore the total of progressions we are counting is at least

$$\sum_{\substack{T \in D \\ tF \geqslant \delta}} \eta^k mt(1 - (\delta/2tF)) \geqslant (1 - (\delta/2))\, \eta^k mt - \eta^k m(\delta/2) = \eta^k m(1 - \delta).$$

In summary, the total of our progressions is

$$\eta^k m(1 \pm \delta),$$

as we had hoped to prove. (Note again $1 + \delta < 1/(1 - \delta)$.)

This, of course, is a useful result only if we can show that the sum $Q + W$ of the excluding probabilities is small. To prove this we will need to restrict our η and k. We will assume that

$$\eta^k m \geqslant m^{\mu}, \qquad k = O(\log m)$$

where μ is any small positive constant. We now show that each term in Q or W is exponentially small in m. We will determine an upper bound for the largest term, which certainly occurs in W. A candidate term has the form

$$\exp(-\varphi(\epsilon_T)(1 \pm \epsilon_T)^{(6i+6)/(6k+1)} \eta^{i+1} tm).$$

We first treat the $1 - \epsilon_T$ case. For any interval T we are attempting to control we have

$$1 - \epsilon_T \geqslant (1 - (\delta/2tF))^{1/(6k+1)} \geqslant (\tfrac{1}{2})^{1/(6k+1)}.$$

Thus the absolute value of the above exponent is at least

$$\varphi(\epsilon_T\, \tfrac{1}{2}(\tfrac{1}{2})!^{/(6k+1)} \eta^{k+1} (\delta/F) m.$$

For the case $1 + \epsilon_T$ we have

$$\begin{aligned}(1 + (\delta/2tF))^{(6i+6)/(6k+1)}\eta^{i+1}tm &\geqslant (1 + (\delta/2tF))^{(6i+6)/(6k+6)}\eta^{i-1}tm \\ &\geqslant (\delta/2F)^{(6i+6)/(6k+6)}t^{6(k-i)/(6k+6)}\eta^{i+1}m \\ &\geqslant (\delta/2F)t^{6(k-i)/(6k+6)}\eta^{i+1}m,\end{aligned}$$

as certainly we can take $\delta < 1$. Let now t have its smallest value $S\eta^k\delta/k^4$ (from (v)) and obtain a lower bound of

$$S(\delta^2/(2Fk^4))\,\eta^{(6k(k-i)/(6k+6))+i+1}m \geqslant (S\delta^2/(2Fk^4))\,\eta^{k+1}m.$$

Thus the largest term does not exceed

$$\exp(-\varphi(\epsilon_T)(S\delta^2/2Fk^4)\,\eta^{k+1}m).$$

Finally for $\varphi(\epsilon_T)$ we have

$$\varphi(\epsilon_T) \geqslant (1 + \epsilon_T)\log(1 + \epsilon_T) - \epsilon_T$$

by Theorem 2.2.1; note also that $(1 + x)\log(1 + x) - x$ is an increasing function of x. Now

$$\begin{aligned}\epsilon_T &= 1 - (1 + (\delta/2tF))^{-1/(6k+1)} \\ &\geqslant 1 - (1 + (\delta k/2F))^{-1/(6k+1)}.\end{aligned}$$

Since $1 - (1 + x)^{-1/p} \geqslant c(x/p)$ if $0 < x < \rho < 1$, where p is an integer and c a constant depending on ρ, we can conclude

$$\epsilon_T \geqslant (c_1\delta/F),$$

for some positive constant c_1, if δ is small—say $\delta < \frac{1}{2}$. Therefore

$$\varphi(\epsilon_T) \geqslant \varphi(c_1\delta/F) \geqslant (c_2\delta^2/F^2)$$

for some other positive constant c_2. Combining all of the above we see that there exists a positive constant c_3 such that no term of Q or W exceeds

$$\exp(-c_3\delta^4\eta^{k+1}m/k^4F^3).$$

From the definition of l we see that

$$l = O(k^2\log(1/\eta) + k\log k + k\log(1/\delta)), \qquad F = O(k^2l)$$

where the implied constants are aboslute. We can finally find an absolute positive constant C that incorporates also the effect of these O constants and that of adding all the terms in Q and W. For that constant C we can then conclude that

$$Q + W \leqslant \exp(-C\delta^4\eta^{k+1}m/[k^{16}(k\log 1/\eta + \log k + \log 1/\delta)^3]).$$

We are now basically done. It only remains to check that the assumptions we used were justified. It is very easy to check that assumption (i) is satisfied for the values of ϵ_T we have chosen. For our repeated applications of the set intersection theorem we need to know that

$$t/(1-\epsilon_T) < 1, \qquad \eta/(1-\epsilon_T)^4 < 1.$$

Now

$$\begin{aligned} t/(1-\epsilon_T) &= t(1+(\delta/2tF))^{1/(6k+1)} \leqslant t\,(1+\delta/(2(6k+1)Ft)) \\ &\leqslant t+\delta/(2(6k+1)F) < 1 \qquad \text{since} \quad t < \tfrac{1}{2}, \delta < 1. \end{aligned}$$

Now note that when we choose ϵ_T we can assume that $\eta^k/(1-\epsilon_T)^{6k+1} \leqslant t$, for certainly we cannot have more progressions than the length of T. Thus to check $\eta/(1-\epsilon_T)^4 < 1$ we look at $\eta^{6k+1}/(1-\epsilon_T)^{4(6k+1)}$. Now

$$\eta^{6k+1}/(1-\epsilon_T)^{4(6k+1)} = (\eta^k/(1-\epsilon_T)^{6k+1})^4\eta^{2k+1} < \eta^{2k+1} < 1.$$

Thus $\eta/(1-\epsilon_T)^4 < 1$ is also proved. This completes the argument for the following result:

THEOREM 2.4.1. *If μ, δ_0 are positive constants while η, δ, and k can vary with m so that*

$$\begin{gathered} 0 < \eta < 1, \\ 1 \leqslant k = O(\log m), \\ \eta^k m \geqslant m^\mu, \\ 0 < \delta \leqslant \delta_0, \end{gathered}$$

then there exists a constant C depending at most on δ_0 such that as $m \to \infty$, except with probability not exceeding

$$\exp(-C\delta^4\eta^{k+1}m/[k^{16}(k\log 1/\eta + \log k + \log 1/\delta)^3]),$$

a selection H of ηm points in Zm will have

$$\eta^k m(1 \pm \delta)$$

arithmetic progressions of length k coming from H to 0.

THEOREM 2.4.2. *If τ is a type of length k and l hits, then Theorem 2.4.1 applies to the enumeration of progressions of type τ coming to 0, if throughout we replace η^k by $\eta^l(1-\eta)^{k-l}$. That is under the assumption that*

$$\eta^l(1-\eta)^{k-l}m \geqslant m^\mu,$$

we can conclude that the number of arithmetic progressions of type τ coming from H to 0 will be

$$\eta^l(1-\eta)^{k-l}m(1 \pm \delta),$$

except with probability not exceeding

$$\exp(-C\delta^4\eta^{k+1}(1-\eta)^{k-l+1}m/[k^{16}(k\log(\eta^{-1}(1-\eta)^{-1})\log k+\log 1/\delta)^3]).$$

Proof. In the argument above intersect with H or the complement of it according to whether the type specifies a hit or a miss. ∎

COROLLARY 2.4.1. *The conclusion of Theorem* 2.4.1 *or Theorem* 2.4.2 *can be made to apply* simultaneously *for all progressions coming to all points x in Z_m and all types not exceeding a certain length $k_0 = O(\log m)$.*

Proof. Simply look at the sum of all the excluding probabilities. The total number of conditions we are imposing is a polynomial in m (e.g., ⟨number of types⟩ × ⟨number of points⟩). Now use the fact that $P(m)\exp(-C_1m^\mu) < \exp(-C_2m^\mu)$ as $m \to \infty$ if P denotes a polynomial and $C_2 < C_1$. ∎

This last corollary illustrates the power of the exponentially small bounds.

COROLLARY 2.4.2. *Under the conditions of Theorem* 2.4.1, *a selection H of ηm points in Z_m will have no more than*

$$2\eta^2 m$$

pairs (x, y), $x, y \in H$, of points in the specified ratio $a : b$, $1 \leqslant a, b \leqslant k$, except with probability not exceeding

$$\exp(-C\eta^{k+1}m/[k^{16}(k\log 1/\eta+\log k)^3]).$$

Proof. Assume the ratio $a : b$ is in lowest terms. Then apply the argument of this section while only considering T_aT and T_bT in the pull-back process for each interval T. ∎

The following generalization of Corollary 2.4.2 is needed in Section 3.3 The arbitrary notation used below is chosen to correspond to the context of that section.

THEOREM 2.4.4. *Let A denote a fixed subset of Z_m of cardinality at least $m^{1/4+\delta_2}$. Let $\eta = m^{-1/4-\delta_1}$, where δ_1, δ_2 are small positive constants satisfying $\delta_2 > \delta_1$. Then there exists a small positive constant δ such that: if a subset H of ηm elements is randomly chosen in Z_m, then the number of pairs of points (u, v) in H with $v \in A$, and u, v in the prespecified ratio $a : b$, $1 \leqslant a < b \leqslant k = O(\log m)$, is $O(\eta \mid A \mid m^{-\delta})$, except with probability that does not exceed* $\exp(-m^\delta)$.

Proof. Let δ_3 be such that $\delta_1 < \delta_3 < \delta_2$. Consider the pull-back process for a certain interval T of the partition. Let S_b denote as before the bth multiples of points of T. Then S_b is a subset of T_bT. Let x denote the number of points of A in S_b. We distinguish two cases.

Case 1. $x > m^{1/4+\delta_3}$. We will apply the pull-back argument to only T_bT and T_aT. We start by a weak bound on the intersection of A and H in S_b. What is the probability

that this intersection will exceed $xm^{-\delta}$ in size, where δ is positive but less than δ_1, $\delta_3 - \delta_1$, and $\delta_2 - \delta_3$? We use our Theorem 2.2.1. We have

$$\alpha = x/m, \qquad \beta = \eta,$$
$$1 + \epsilon = m^{-\delta}/\eta \geqslant m^{1/4}.$$

Then

$$\varphi(\epsilon)\,\alpha\beta m \geqslant [(1+\epsilon)\log(1+\epsilon) - \epsilon]\,\alpha\beta m \geqslant [\tfrac{1}{4}\log mm^{-\delta}/\eta - (m^{-\delta}/\eta - 1)]\,x\eta$$
$$\geqslant Cxm^{-\delta} = Cm^{1/4+\delta_3-\delta}$$

for some positive constant C. Thus the probability under consideration is exponentially small, and we may assume that our intersection does not exceed $xm^{-\delta}$. We now pull back this intersection to T_aT, thereby defining S_a. This is a disjoint set from S_b, and applying once more our intersection theorem with ϵ this time small, say $\epsilon = \frac{1}{2}$, we immediately conclude that, except with probability not exceeding

$$\exp(-C'\eta xm^{-\delta}) \leqslant \exp(-m^{\delta}),$$

the number of pairs (u, v) with $u \in H \cap T_aT$, $v \in H \cap A \cap T_bT$, u, v in the ratio $a : b$, is no greater than

$$O(\eta xm^{-\delta}).$$

Case 2. $x \leqslant m^{1/4+\delta_3}$. We now simply do not bother with the T_bT step of the above argument. Just pull back the entire $A \cap S_b$ to T_aT in order to obtain S_a. Thus S_a has size $\leqslant m^{1/4+\delta_3}$, and since we are interested in maximizing the number of pairs (u, v), we will in fact assume that S_a has size $m^{1/4+\delta_3}$. Applying Theorem 2.2.1. to S_a and H we obtain that, except with probability not exceeding

$$\exp(-C'x\eta) \leqslant \exp(-C'm^{\delta_3-\delta_1}) \leqslant \exp(-m^{\delta}),$$

the total of pairs (u, v) with $u \in H \cap T_aT$, $v \in A \cap T_bT$ (and *a fortiori* those for which $v \in A \cap H \cap T_aT$) and u, v in the ratio $a : b$, is no greater than

$$O(x\eta) = O(m^{\delta_3-\delta_1}).$$

Now we sum the contributions over all T. The contributions from Case 1 are $O(\eta(\Sigma_T x)m^{-\delta})$. Since the mapping $z \to bz$ is 1–1, we must have $\Sigma_T x \leqslant |A|$. Thus the total from Case 1 is $O(\eta \mid A \mid m^{-\delta})$, which is at least $Lm^{\delta_2-\delta_1-\delta}$, by our assumption about the size of A. By taking L sufficiently large (say $L = Ck \log m$, for some constant C) we can make the contribution of the fixpoint intervals negligible compared to this, say no more than $m^{\delta_3-\delta_1}$. Also the contributions of Case 2 are no more than $O(k^3(\log m)\, m^{\delta_3-\delta_1})$ (just let all T's have a Case 2 contribution), and that too is negligible.

By choosing the δ in the statement of the theorem slightly smaller than the δ we have used in the proof, the result follows. ∎

Clearly the results of the last Lemma and Theorem can also be generalized so that they apply to all points and all ratios and types up to some maximum length $k_0 = O(\log m)$ simultaneously.

In a certain light what we have shown is that the occurrence of one arithmetic progression of length k in H influences very little the occurrence of another such progression. These progressions are nearly independent in the sense that they give rise to a distribution analogous to that of independent Bernoulli trials. This is why the results of this section could not have been obtained by variance arguments alone. It is interesting that for $k = 3$ a similar result can be proved using the exponential sums technique of analytic number theory [7]. Unfortunately that proof does not appear to generalize to $k > 3$.

3. The Impotence of Clustering

In this section we use the results of Section 2 to start up a process (called the *extension process*) of looking at snapshots of the table as it fills up with double hashing. Between steps of the extension process we can show that the effect of clustering is negligible, and that we therefore never depart too far from the truly random situation. We begin by showing how the near-randomness maintained by the extension process can be used to derive the desired result.

3.1. *The Seed Set and the Final Argument*

In this section we prove that double hashing is asymptotically equivalent to uniform hashing for $0 < \alpha \leqslant \alpha_0$ by using the results of the following two sections. Here α_0 denotes an absolute constant, $\frac{1}{4} < \alpha_0 < \frac{1}{3}$. Let us prestate here Corollary 3.3.1, which is the result we will need:

Corollary 3.3.1. *Given any α, β such that $0 < \beta < \alpha \leqslant \alpha_0$, there exist a constant C_α and an initial configuration of βm occupied positions such that for any small positive constant θ, if we add $(\alpha - \beta)m$ points to the table using the double hashing process then, except with probability less than $\exp(-C_\theta m^{\delta_0})$, we will arrive at a configuration of αm occupied elements such that for each point x of the table and for each length k, $2 \leqslant k \leqslant k_\alpha' = C_\alpha \log m$ the number of arithmetic progressions of length k coming to x from the occupied points is $\alpha^k m(1 \pm \underline{\theta}_{\alpha,k})$. Here the $\underline{\theta}_{\alpha,k}$ denote relative errors satisfying*

$$\text{(a)} \quad \sum_{k=2}^{k_\alpha'} \underline{\theta}_{\alpha,k}\alpha^k \leqslant \theta,$$

and

$$\text{(b)} \quad \alpha^{k_\alpha'}(1 + \underline{\theta}_{\alpha,k_\alpha'})\, m \leqslant m^{1/2-\delta'},$$

where δ' and δ_0 are small positive constants, and C_θ is a constant depending on θ only.

What is the average number of comparisons we need in order to find an empty slot in the resulting configuration, using double hashing? (Recall that, as in Section 1, we count the final probe into an empty slot as a comparison.) As we make such a search, let p_l denote the conditional probability that we will make at least $(l+1)$ comparisons before we find an empty slot, $0 \leqslant l < m$, given that we hit at least one of the occupied positions. Thus we must on the first probe ($h(K)$) select one position among the set S of occupied positions, and then select a distance ($g(K)$) that leads to an arithmetic progression of length (at least) l among elements of S. The average number of comparisons for an unsuccessful search will then be

$$1 + \alpha \sum_{l=0}^{m-1} p_l . \tag{i}$$

We first dispose of the arithmetic progressions of length greater than k_α'. Let us consider an occupied point $x \in S$ (all such points are equivalent for the computation below). We claim that there are no arithmetic progressions coming from S to x of length exceeding $(m^{1/2-\delta'}+1)\,k_\alpha'$. This is so, since if we had such a progression, then we would have more than $m^{1/2-\delta'}$ progressions of length k_α' coming to x from S. (If d is the distance of the original progression, then $d, 2d, 3d, \ldots, (m^{1/2-\delta'}+1)d$ would all be distances of progressions of length k_α' that are subsets of the long progression). But this contradicts condition (b) of the above corollary. Now for lengths between k_α' and $k_\alpha'(m^{1/2-\delta'}+1)$, we can have at most as many arithmetic progressions as we have at k_α'. The total contribution of these to (i) is

$$\alpha m \sum_{k=k_\alpha'}^{k_\alpha'(m^{1/2-\delta'}+1)} m^{1/2-\delta'} \times 1/m(m-1)$$

choices for x | contribution of all distances in question | probability of choosing a specific x and a specific distance

$$= O(k_\alpha' m^{-2\delta'}) = o(1) \quad \text{as} \quad m \to \infty,$$

since $k_\alpha' = O(\log m)$.

Here we have ignored the contribution of the excluded configurations, but these can contribute at most a total of

$$m \exp(-C_\theta m^{\delta_0}) = o(1) \qquad \text{as} \quad m \to \infty$$

maximum number of probes for any search to the mean, and so from now on they will be ignored for good. For the shorter k we see that our corollary implies

$$p_k = \alpha^k(1 \pm \underline{\theta}_{\alpha,k}), \qquad 2 \leqslant k \leqslant k_\alpha',$$

and certainly $p_0 = 1$, $p_1 = \alpha$. So the contribution of short lengths to Σp_k is

$$1 + \alpha + \sum_{l=2}^{k_\alpha'} \alpha^l(1 \pm \theta_{\alpha,l}) = \sum_{l=0}^{k_\alpha'} \alpha^l \pm \theta = 1/(1-\alpha) \pm \theta + o(1),$$

where we have used conclusion (a) of the corollary and the fact that

$$\sum_{l=k_\alpha'+1}^{\infty} \alpha^l = o(1) \qquad \text{as} \quad m \to \infty$$

since $k_\alpha' \to \infty$ as $m \to \infty$. Combining all of our conclusions, we have proved that the average number of comparisons needed to find an empty position in a table filled up to load factor α as described in the corollary is

$$1 + \alpha/(1 - \alpha) \pm \alpha\theta + o(1) = 1/(1 - \alpha) \pm \alpha\theta + o(1). \tag{ii}$$

Unfortunately we are not done, *because we did not start from an empty table.* The double hashing algorithm was applied only after an initial seed of βm points was already strategically placed in the table.

In order to complete our argument, we need to investigate the effect of these initial βm points. We have added $(\alpha - \beta)m$ keys using the double hashing process. What if we had added these same $(\alpha - \beta)m$ keys to an initially empty table using double hashing? Let us select a specific hash sequence $(h(K_1), g(K_1))$, $(h(K_2)), g(K_2))$,... and so on. Let S denote the set obtained by adding points with this sequence to the initial βm set, and S' the corresponding set obtained by adding points using the same hash sequence to an initially empty table. Then we claim $S' \subseteq S$. Consider the first point K in our sequence, whose insertion would cause an alleged violation of this condition. Either our key K ends up in the same position in both S' and S, in which case there can be no violation, or our key continues on a longer search path in S than it did in S'. But then the location where K ends up in S' must have already been occupied in S, and so again no violation is possible. The above remark implies that the average number of probes to find an empty slot with configuration S is an upper bound for $C'_{(\alpha-\beta)m}$, i.e.,

$$C'_{(\alpha-\beta)m} \leqslant 1/(1 - \alpha) + \alpha\theta + o(1),$$

or

$$C'_{\alpha m} \leqslant 1/(1 - \alpha - \beta) + (\alpha + \beta)\,\theta + o(1),$$

by a simple change of variable. (Assume β is so small that $\alpha + \beta < 1$.)

Next we get a lower bound for $C'_{\alpha m}$. We just saw that if we start with βm points rather than an empty table, we can only do worse. But how much worse? Again let us fix our attention to the particular hash sequence on hand $(h(K_1), g(K_1))$, $(h(K_2), g(K_2))$..., etc. In the set difference $S - S'$ we have βm points. Now suppose we are at the final configuration S and let us look at an arithmetic progression of length k of occupied cells in S coming to x. If this progression contains at least one point in $S - S'$, we shall say that it is destroyed. (This means that it contributed to the computation of p_l for α but will not contribute to the one for $\alpha - \beta$.) No point in $S - S'$ can destroy more than k such progressions, so the total number of progressions of length k coming to x that is destroyed is bounded by $k\beta m$. Of course we can never destroy more arithmetic progressions than there are, which is

$\alpha^k m(1 + \underline{\theta}_{\alpha,k})$. Now let $k_0 = \log(1/\beta)/\log(1/\alpha)$. Then the number of progressions coming to x of length greater or equal to k_0 that can possibly be destroyed is

$$\sum_{k=k_0}^{m-1} \alpha^k m(1 + \underline{\theta}_{\alpha,k}) = O(\alpha^{k_0} m) = O(\beta m),$$

where we estimated the sum as we estimated sum (i) (using also the obvious fact that the errors $\underline{\theta}_{\alpha,k}$ are bounded for fixed k, as $m \to \infty$). From 1 to k_0 we can destroy at most

$$k_0^2 \beta m = O(\beta \log^2(1/\beta) m)$$

arithmetic progressions. Thus the total of destroyed progressions coming to x is

$$O(\beta \log^2(1/\beta) m),$$

and we have shown

$$C'_{\alpha m} \geqslant 1/(1 - \alpha - \beta) - (\alpha + \beta)\theta - O(\beta \log^2(1/\beta)) + o(1)$$

by arguing as in the previous paragraph.

To summarize, we have

$$\begin{aligned} &1/(1 - \alpha - \beta) - (\alpha + \beta)\theta - O_\beta(\beta \log^2(1/\beta)) + o(1) \\ &\quad \leqslant C'_{\alpha m} \leqslant 1/(1 - \alpha - \beta) + (\alpha + \beta)\theta + o(1). \end{aligned}$$

Since θ, β can be taken to be arbitrarily small, we have proved

THEOREM 3.1.1. *The average number of comparisons needed to find an empty slot with double hashing in a table of size m, filled up to load factor α, $\alpha \leqslant \alpha_0$, is*

$$C'_{\alpha m} = 1/(1 - \alpha) + o(1) \qquad \text{as} \quad m \to \infty.$$

3.2. *The Lattice Flows and the Extension Process*

Let x be a point of the table, and let τ be a type of length k with h hits and $k - h$ misses. If γm points of the table are occupied, $0 < \gamma < 1$, then the expected number of arithmetic progressions of type τ coming to x is $\gamma^h(1 - \gamma)^{k-h} m$. In this and the following section we show that if we start with a configuration of βm occupied points in which every point has nearly the expected number of arithmetic progressions of every type and grow this table to αm elements using the double hashing process, then if we only exclude an exponentially small fraction of possibilities, we can be sure that the resulting configuration of αm points will also have nearly the expected number of arithmetic progressions of every type coming to every point.

To illustrate the argument we first discuss how we can prove such a statement if the additional $(\alpha - \beta)m$ elements were randomly inserted. We add the new points in groups of ηm at a time, where η is very small compared to α or β. Suppose we currently have γm elements in the table and are about to add ηm new ones. Fix a point x of the table and

consider the arithmetic progressions of length k coming to x. The various types to which these progressions may belong form a Boolean lattice, as illustrated by Fig. 3.2.1.

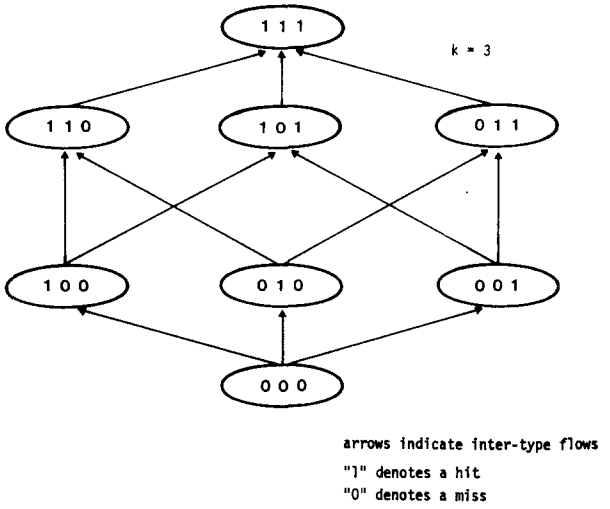


FIG. 3.2.1. The lattice of types of arithmetic progressions of a given length coming to a point.

As the new ηm points are added, there will be flows upwards in this lattice, that is, some arithmetic progressions will shift into types with more hits. For example, the first point of a progression of type (001) may become occupied by one of the ηm points, whereas the second may stay empty, thus causing the progression to shift into type (101). In order to estimate the magnitude of these intertype flows we need to introduce some notation.

DEFINITION 3.2.1. Let x be a point of the table, τ a type of length k, i an integer $1 \leqslant i \leqslant k$, and Γ a configuration of γm occupied positions; then by $S(i, \tau, x, \Gamma)$ we will denote the set of the ith points of the arithmetic progressions of type τ coming to x. We also introduce the density

$$\sigma(\tau, x, \Gamma) = | S(i, \tau, x, \Gamma)|/m,$$

which is clearly independent of i (since m is prime).

Throughout the arguments that follow we will be dealing with inequalities on the $\sigma(\tau, x, \Gamma)$. We introduce the symbol $\sigma(\tau, \gamma)$ to stand for any of $\sigma(\tau, x, \Gamma)$, where x ranges over all points and Γ over all nonexcluded configurations of γm occupied positions. Thus when we write

$$\sigma(\tau, \gamma) = \lambda(1 \pm \mu),$$

we mean $\lambda(1 - \mu) < \sigma(\tau, x, \Gamma) < \lambda/(1 - \mu)$ for all x and Γ as described above.

We now present a heuristic argument for the case of random insertions. Assume that our configuration Γ is such that $\sigma(\tau, \gamma) = \gamma^h(1 - \gamma)^{k-h}$ for all types τ, where h denotes

the number of hits and $k - h$ the number of misses of the type. Thus $S(i, \tau, x, \Gamma) = \gamma^h(1 - \gamma)^{k-h}m$. What happens to the arithmetic progressions of type τ coming to x as the new ηm points are added? Consider a type τ' which is τ except a hit of τ in the ith position is a miss in τ', e.g., $\tau = (101)$, $\tau' = (001)$ in the example above. Then for each point of $S(i, \tau, x, \Gamma)$ that is hit by the ηm, a progression may change from type τ' to type τ. This is illustrated in Fig. 3.2.2.

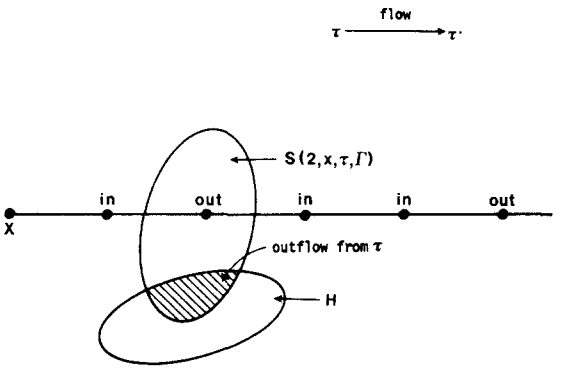


FIG. 3.2.2. The mechanism of intertype flows.

There are $(1 - \gamma)m$ unoccupied elements, of which we are choosing ηm, thus the probability of selecting a point is $\eta/(1 - \gamma)$. By hypothesis the size of $S(i, \tau, x, \Gamma)$ is $\gamma^{h-1}(1 - \gamma)^{k-h+1}m$, and therefore the expected size of its intersection with the ηm is $\eta\gamma^{h-1}(1 - \gamma)^{k-h}m$. Since τ has h hits, there are h possible positions at which such inflows into type τ can occur (i.e., there are h possible feeder types τ'), for a total of $h\eta\gamma^{h-1}(1 - \gamma)^{k-h}m$. Now some of the progressions of type τ can move out of this type. For each i that corresponds to a miss of τ, this will happen whenever the set $S(i, \tau, x, \Gamma)$ is intersected by the ηm. We easily compute the size of the outflow to be $(k - h)\,\eta\gamma^h(1 - \gamma)^{k-h-1}m$. In the above we have ignored the possibility that a transition between two types can occur with more than one of the points of a progression being hit by the ηm. Any flows arising out of such transitions, however, will have an expected magnitude of $O(k^2\eta^2 m)$ and since we take η to be very small, they can be ignored. To total up, when the new ηm points have been added, the expected number of arithmetic progressions of type τ coming to x will be

$$\gamma^h(1 - \gamma)^{k-h}m + h\eta\gamma^{h-1}(1 - \gamma)^{k-h}m - (k - h)\,\eta\gamma^h(1 - \gamma)^{k-h-1}m$$

which is $(\gamma + \eta)^h(1 - \gamma - \eta)^{k-h}m$ if again we ignore $O(k^2\eta^2 m)$ terms.

Thus $\sigma(\tau, \gamma) = (\gamma + \eta)^h(1 - \gamma - \eta)^{k-h}$ on the average, as we had hoped. By iterating this heuristic argument we see how we can grow from βm to αm elements while having the expected number of arithmetic progressions of any type at each point at each step.

A natural question at this point is: Why did we not use this method to carry out the proof of the results in Section 2? The reason is that the above argument needs a "good"

configuration already in the table to get started. The law of large numbers cannot be used to show that the various set intersections will be nearly what we expect, unless the sets involved are large enough. In the next section we will show how to carry out the above argument rigorously. A special trick that uses the results of Section 2 will allow the above process to get started.

For the remainder of the current section we confine ourselves to some definitions and general remarks. We shall use the term *the extension process* for this process of building up the table we are describing. This process consists of steps of adding ηm points at a time. During each step, given any two types of progressions coming to a point x, there may be transitions of actual progressions from one type to the other. These intertype transitions will be called *flows*. For each type we will have a certain *inflow* and *outflow* of progressions from it. Naturally we cannot assume that a type will have exactly the expected number of progressions, as we have done in the heuristic argument above. We introduce relative errors $\theta_{\gamma,\tau}$ on this expected value that describe the deviation we are willing to allow. In other words, when we are at load factor γ, we assume that for each point x and each type τ of length k (k not exceeding a certain maximum) and h hits, we will have

$$\gamma^h(1-\gamma)^{k-h}m(1 \pm \theta_{\gamma,k})$$

arithmetic progressions of type τ coming to x. Here we have already adopted the convention that we will follow in the actual argument and suppressed the dependency of $\theta_{\gamma,\tau}$ on anything but the length k of τ. We will find that the errors $\theta_{\gamma,k}$ grow faster for larger k, but if we compute the *total* number of arithmetic progressions coming to x of types consisting entirely of hits, then the relative error on this total we will be able to make as small as we please. Now double hashing chooses each of the empty points with probability proportional to the number of arithmetic progressions coming from the occupied points to that point. The above remark then implies that during the current step every empty position is nearly equally likely to be filled. So we are not too far from the random situation. But how can we be sure that we will maintain the same good situation during the next step? Here we invoke Theorem 2.1.1 to assure that all intersections between the ηm points and the various sets $S(i, \tau, x, \Gamma)$ of Definition 3.2.1 are nearly the same size. In doing this we exclude an exponentially small fraction of choices of the ηm points, while increasing the relative errors $\theta_{\gamma,k}$ to $\theta_{\gamma+\eta,k}$ for the next step. We will speak of using Theorem 2.2.1 for *controlling the intersections*, and therefore the flows. In order to keep the error propagation equations for $\theta_{\gamma,k}$ relatively clean, we will allow certain absolute errors as well (the "residual" progressions of the next section). During any step, if there is a number of progressions flowing between two types that is allowed by our control but cannot be accounted for in the relative errors we allow, this number we will speak of as an *excessive flow*. The gist of the argument then is that by excluding an exponentially small fraction of possibilities, we maintain at each step every empty position nearly equally likely to be filled. We never give clustering a chance to build up a really bad configuration.

We now make a number of remarks that the reader should keep in mind while reading the next section.

Remark 1. The types that ultimately play a role in double hashing are those consisting entirely of hits. Because, however, the population of types changes by intertype flows, we have to attempt to control all types at once.

Remark 2. Suppose we wish to maximize the number of progressions in a type τ consisting of k hits. During each step the significant inflows into τ are those from types with $k-1$ hits. Obviously we should maximize these inflows. Now these inflows are also outflows from the "feeder" types one step below in the lattice. In order to maximize those same inflows during the next step, we want to maximize the growth of the feeder types during the current step. But these types have their outflows already chosen, so the best we can do is to maximize the inflows into them. An inductive extension of this argument shows that all flows in the lattice should take their maximum allowed value during every step, if we are interested in maximizing the growth at the apex of the lattice. Similarly if we wish to minimize this growth, all flows should be minimized. The point made here is important and somewhat subtle, and the reader should dwell on it for a moment. Another way to see the point is this. Consider one of the sets S corresponding to one of the feeder types. At the current step a fraction ρ_1 of S will flow, where ρ_1 is allowed to vary within certain limits. At the next step a fraction ρ_2 of the part of S that is left will flow, and so on, say up to ρ_ν. Then it is simple to see that the total fraction of S that has flowed is

$$1 - \prod_{i=1}^{\nu} (1 - \rho_i)$$

and this expression is maximized when all of the ρ_i are maximized. The intuitive interpretation of this is that if we wish to maximize the total flow between two types, we should never trade the certainty of a specific transition in the current step for the probability of that same transition in some future step.

Remark 3. If we are interested in maximizing the flows, it will only hurt our upper bound to make any of the sets S of Definition 3.2.1 that partake in the controlled intersections larger than it really is.

Remark 4. Since we are dealing with nonnegative quantities, a relative error smaller than -1 clearly does not make sense. We do, however, allow such fictitiously large negative errors in the argument of the next section, since they can only make our lower bounds worse and they avoid consideration of special cases.

Remark 5. If $P(m)$ denotes any polynomial in m, C, δ_1, δ_2 constants with $C > 0$, $\delta_2 > \delta_1 > 0$, then m sufficiently large

$$P(m) \exp(-Cm^{\delta_2}) < \exp(-Cm^{\delta_1}).$$

Remark 6. Let θ denote an arbitrarily small positive number and let $\psi(m)$ be a quantity which is $o(1)$ as $m \to \infty$. Then we will say that ψ can be incorporated in θ to mean that, given any positive constant θ', for m sufficiently large we can assume that the sum $\theta + \psi(m)$ does not exceed θ'. We use this terminology on a number of occasions.

This is justified because it will be trivial to cheek that *the sum* of the $\psi(m)$ over all instances of the terminology that refer to the same θ is $o(1)$.

Remark 7. We will make some use of the O, o notations. They always refer to $m \to \infty$, and the implied constants are either absolute or depend at most on α, which is a constant of the entire problem. In Corollary 3.3.1 we also use the notations $\ll$, $\approx$ with their usual heuristic meaning. If the reader wishes to have an exact meaning, then he may take, in the context where these occur, $f \approx g$ to mean $gm^{-\delta} \leqslant f \leqslant gm^{\delta}$, and $f \ll g$ to mean $f \leqslant gm^{-\delta}$, for some small positive δ.

Remark 8. The reader should realize that the process of intertype flows we have described is only a model for what occurs in the real table. The model will be used to bound the number of progressions we can actually have in the table. It need not be the case that the flows we use in the estimations of the next section can be realized by some sequence of insertions into the actual table.

3.3. *The Propagation of Errors and the Impotence of Clustering*

We will now carry out a precise estimation of the error propagation in the extension process. We assume α, β are fixed constants, β small, $0 < \beta < \alpha < 1$. In the course of the computation we will find that we have to restrict α to be below some absolute constant α_0, $\alpha_0 < 1$. We take

$$\eta = m^{-1/4-\delta_1},$$

and define

$$\begin{aligned} k_\beta &= [(3/4 - \delta_2)/\log(1/\beta)] \log m \qquad (\text{so } \beta^{k_\beta} m = m^{1/4+\delta_2}), \\ k_\alpha &= [(1/2 + \delta_3)/\log(1/\alpha)] \log m \qquad (\text{so } \alpha^{k_\alpha} m = m^{1/2-\delta_3}) \end{aligned} \tag{i}$$

where δ_0, δ_1, δ_2, δ_3, δ_4 are small positive constants such that

$$\delta_2 > \delta_1, \ \delta_2 - \delta_1 > \delta_4 > \delta_0 > 0 \qquad (\delta_0, \delta_4 \text{ will be used later}). \tag{ii}$$

Our choice for η is a compromise between two conflicting requirements. On the one hand we want to make η as large as possible so as to get the maximum benefit from the law of large numbers and Theorem 2.2.1. On the other hand we want to take η sufficiently small so that we can ignore the iteractions of the ηm points among themselves. During the extension we need to maintain control over arithmetic progressions of length k_α since the argument of Section 3.1 depends heavily on our ability to push the number of arithmetic progressions of length k_α below some power less than $m^{1/2}$. Unfortunately in the early stages of the extension process we are then out of luck. For types τ of length k_α and many hits, the expected number of progressions of that type coming to a point will be too small to either assert anything initially, or to control the intertype flows by bounding the size of the intersections with the ηm points. To circumvent this shortcoming we introduce a technical device. For each point x and for each type τ of length between k_β and k_α we introduce an initial maximum positive "error" of size $E_0 = m^{1/4+\delta_2}$ in the

number of arithmetic progressions of type τ coming to x. This error is in addition to the regular relative errors discussed in Section 3.2. As we will see, it provides us with a way of masking out the fact that we cannot control the size of the relative errors during the early stages of the extension process. These additional progressions will of course flow among the types like the normal ones we have already considered. We will call them the *residual* progressions and will control their flows independently of the regular progressions.

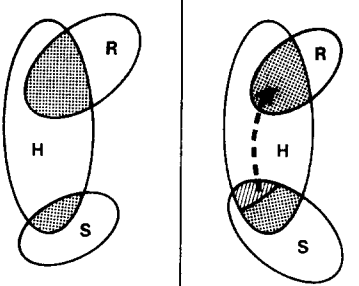
We will ignore the outflow of residual progressions from any given type. Thus their number can only grow and will never become less than E_0. By analogy with Definition 3.2.1 we introduce the notations $R(i, \tau, x, \Gamma)$, $\rho(\tau, x, \Gamma)$ to denote the corresponding quantities for the residual progressions. Thus $\rho(\tau, x, \Gamma) \geqslant m^{-3/4+\delta_1}$. If at any moment during the extension process we have

$$\sigma(\tau, x, \Gamma) < \rho(\tau, x, \Gamma)$$

then we will not attempt to control the intersection of any of $S(i, \tau, x, \Gamma)$ with the ηm. Instead we will control *only* $S(i, \tau, x, \Gamma) \cup R(i, \tau, x, \Gamma)$, which has cardinality $(\sigma(\tau, x, \Gamma) + \rho(\tau, x, \Gamma))m$. We also use the notation

$$r(\tau, x, \Gamma) = |R(i, \tau, x, \Gamma)|.$$

If the intersection of the ηm with $S(i, \tau, x, \Gamma)$ was excessively large, then any excess we will relabel as *residual* progressions for the receiving type. Thus we can guarantee that none of the regular flows (i.e., flows of regular progressions) will be excessive, by allowing sometimes the residual flows to be excessively large (by at most the same relative error). The quantitative argument will be given in the proof of Theorem 3.3.1. Figure 3.3.1 attempts to summarize this camouflaging with the residual progressions.



Case.1: $R > S$ $S \cap H$ within bounds | Case 2: $R > S$ $S \cap H$ out of bounds

FIG. 3.3.1. Camouflaging with the residual progressions.

DEFINITION 3.3.1. The generic variable ρ_γ will stand for any of $\rho(\tau, x, \Gamma)$, the densities of the residual progressions.

As we saw in the last section, it is our goal to perform the extension process so that at each step all empty points are nearly equally likely to be filled. Since at each step we introduce not one but ηm points all at once, we have to understand the interactions among the ηm points themselves. It is possible that an initial fragment of the ηm points will be placed so badly that it will greatly affect where the remaining of the ηm points will go. This, however, can only occur if during an insertion one of the ηm points interacts heavily with those previously inserted.

DEFINITION 3.3.2. Suppose we have a configuration Γ of γm occupied positions and are inserting ηm additional points. An insertion of one of these points will be called *bad* if its probe path (i.e., the sequence of examined points before insertion)

(1) contains an initial segment of length at least k_α consisting of positions of the γm and at most one position occupied by one of the ηm points, or

(2) contains (at least) two of the ηm points among its first k_α (or fewer) steps.

An insertion which is not bad will be called *good*. We let b_γ denote the total number of bad insertions that have occurred when we reach a load factor of γ.

Figure 3.3.2 illustrates the different cases of good and bad insertions.

What we will prove below is that the conditional probabilities that any two empty positions will be filled, given that they are filled with good insertions, are nearly equal.

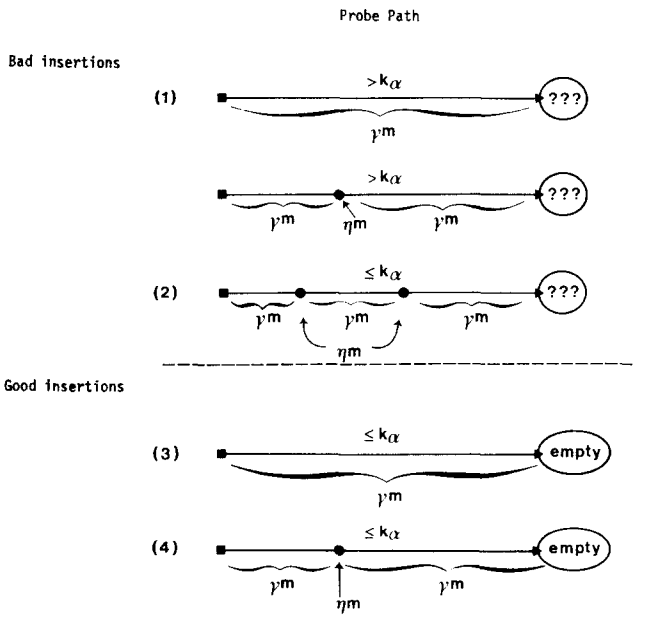


FIG. 3.3.2. The good and bad insertions.

We introduce the quantity χ_γ to capture the relative error in the probabilities (recall $\theta_{\gamma,1} = 0$).

DEFINITION 3.3.3. We let

$$\chi_\gamma = \sum_{k=2}^{k_\alpha} \gamma^k \theta_{\gamma,k} .$$

We now have all the concepts we need to begin the quantitative argument.

THEOREM 3.3.1. *Let α, β be positive constants such that $0 < \beta < \alpha < \alpha_0$. There exist absolute positive constants s, D such that, given an arbitrarily small positive constant θ, there exist positive constants i_θ, C_θ (tending to 0 as $\theta \to 0$) such that: if we begin the extension process with a configuration of βm elements placed so that for each point x and type τ of length less than or equal to k_α we have the expected number of arithmetic progressions of that type coming to x within a relative error of i_θ and (for those τ of length at least k_β) a residual error of at most $E_0 = m^{1/4+\delta_2}$ progressions, then, except with probability* $\exp(-C_\theta m^{\delta_0})$ *where δ_0 is a constant, $0 < \delta_0 < \delta_2 - \delta_1\,\delta_1$, δ_2 as defined by η, k_α in* (i), *when we reach a load factor γ we will have*

(a) $\theta_{\gamma,k} \leqslant (1 + i_\theta)\, e^{5\theta(\gamma_s/s)} - 1 \qquad$ *for* $\quad 2 \leqslant k \leqslant k_\alpha$,

(b) $\chi_\gamma \leqslant \theta\gamma^s$,

(c) $\rho_\gamma m \leqslant E_0 m^{(1/2+\delta_3)\log[1+2\log(1/(1-\gamma))]/\log(1/\alpha)}$,

(d) $b_\gamma \leqslant D_\gamma m^{1/2-\delta}, \qquad$ *with* δ, $0 < \delta < \delta_3$, $2\delta_1$, *a constant,*

where θ_γ, k, χ_γ, ρ_γ, b_γ are as given by Definitions 3.2.2, 3.3.3, 3.3.1, *and* 3.3.2, *respectively.*

Proof. We will prove assertions (a), (b), (c), and (d) by induction on the number of steps in the extension process. Thus we will assume that they hold for γ and prove them for $\gamma + \eta$. For $\gamma = \beta$ all assertions are true trivially, except for (b) which requires that we take $i_\theta \leqslant \theta_\beta \beta^s(1 - \beta)$, as we certainly can. We will see how to choose the constants D, s in the course of the proof.

The proof is in two parts. First we examine the effect of the bad insertions, and second we look at the propagation of the errors.

What is the probability of a bad insertion? Let us go back to Definition 3.3.2. An initial segment of length at least k_α will be entirely within the γm with probability $\sigma(\tau_0, \gamma)$, where τ_0 is the type of k_α hits. Similarly, the probability of encountering one of the ηm points in this segment is certainly bounded by $k_\alpha \sigma(\tau_1, \gamma)$, where τ_1 denotes any type of length k_α and $k_\alpha - 1$ hits. Thus the probability of condition (1) of the definition being satisfied does not exceed

$$\gamma^{k_\alpha}(1 + \theta_{\gamma,k_\alpha}) + k_\alpha \gamma^{k_\alpha - 1}(1 - \gamma)(1 + \theta_{\gamma,k_\alpha}). \tag{iii}$$

We estimate the probability that condition (2) will be satisfied somewhat differently. We ask how many pairs $(h(k), g(k))$ are there that lead to a prove path satisfying (2). The probe path is completely specified once we know the two ηm points involved, and the

positions of the two points on the path, say they are the bth and cth points, respectively. Since we can take $1 \leqslant b < c \leqslant k_\alpha$ we have at most $\frac{1}{2}k_\alpha{}^2\eta^2m^2$ distinct probe paths. Each candidate pair $(h(k), g(k))$ defines a distinct path. Since each pair occurs with probability $1/m(m-1)$, we have an overall probability (per insertion) of satisfying (2) that is bounded by $k_\alpha{}^2\eta^2$.

From assertion (a) we have

$$\theta_{\gamma,k_\alpha} \leqslant (1 + i_\theta)\, e^{5\theta(\gamma^s/s)k_\alpha} - 1$$

and since

$$k_\alpha = [(1/2 + \delta_3)\log m]/[\log 1/\alpha]$$

it follows that

$$\theta_{\gamma,k_\alpha} \leqslant (1 + i_\theta)\, m^{5\theta(\gamma^s/s\ [[(1/2)+\delta_3]/(\log 1/\alpha)]}$$

which can be made $\leqslant m^{\delta_5}$ by taking θ sufficiently small, where δ_5 is such that $0 < \delta_5 < \delta_3 - \delta,\ 2\delta_1 - \delta$. Thus the quantity specified in (iii) is less than or equal to $m^{-1/2-\delta_6}$ for some $\delta_6 > \delta$, and so is $k_\alpha{}^2\eta^2$ as the reader can easily check. The residual progressions of the types accounted for in (iii) have to be added in also, of course, but their number as given by (c) is less than or equal to

$$\frac{m^{1/4+\delta_2}m^{(1/2+\delta_3)\log[1+3\log(1/(1-\gamma))]/\log(1/\alpha)}}{m}$$

which is less than $m^{-1/2-\delta_6}$ for $\gamma \leqslant \alpha \leqslant \alpha_0$ as can be easily checked. We will encounter this α_0 later also, so we will not dwell on its value any longer here. Thus we have proved

Claim 1. The probability of a bad insertion is never greater than $m^{-1/2-\delta}$ for $\beta \leqslant \gamma \leqslant \alpha \leqslant \alpha_0$.

By Theorem 2.2.1 (or its equivalent for Bernoulli trials) the probability that we will have more than $Dm^{-1/2-\delta}\eta m = D\eta m^{1/2-\delta}$ bad insertions, for some constant D slightly larger than 1, is less than $\exp(-C(D-1)^2\eta m^{1/2-\delta}) < \exp(-m^{1/s-2\delta-\delta_1})$ and thus this event can be excluded. Therefore we can assert that at load factor $\gamma + \eta$ the total of bad insertions will not exceed

$$D(\gamma + \eta)m^{1/2-\delta},$$

as we desire in order to prove (d). Although we cannot say anything about where the badly inserted points will go, their number is so small that, as we shall see, they cannot destroy the final assertion of regularity of our configuration.

We next show that any two empty positions have nearly equal probabilities of being filled with good insertions. Under double hashing the probability that a given empty position will be filled is proportional to the number of arithmetic progressions coming to that position from the occupied positions. Recall also that in a good insertion, the probe path is at most k_α long and in this path at most one of the new ηm points can occur. Let x be any empty point. The number of regular (i.e., nonresidual) arithmetic progres-

sions of length k, $0 \leqslant k \leqslant k_0$, coming to x from the occupied points is by assumption $\gamma^k(1 \pm \theta_{\gamma,k})m$, for a total of

$$\sum_{k=0}^{k_\alpha} \gamma^k(1 \pm \theta_{\gamma,k})\, m,$$

or the discrepancy over the expected value is in absolute value at most

$$\left(\sum_{k=0}^{k_\alpha} \gamma^k \theta_{\gamma,k}\right) m = \chi_\gamma m.$$

Each of the new ηm points can occur in the path, and each such point can introduce at most k additional progressions of length k coming to x (by being the 1st, 2nd,..., kth point of the progression), for a total of

$$\eta m \sum_{k=0}^{k_\alpha} k \leqslant k_\alpha{}^2 \eta m. \tag{iv}$$

The number of residual progressions coming to x is at most

$$\sum_{k=k_\beta}^{k_\alpha} \rho_\gamma m \leqslant k_\alpha m^{1/2-\delta} \tag{v}$$

for $\alpha \leqslant \alpha_0$. Finally the previously badly inserted points can introduce each at most k progressions of length k, for a total as in (iv) of at most

$$b_\gamma \sum_{k=0}^{k_\alpha} k \leqslant k_\alpha{}^2 D\gamma m^{1/2-\delta} \tag{vi}$$

progressions. We only demand in (b) that χ_γ can be made as small as any prescribed constant, and so the combined effect of (iv), (v), and (vi) can be accounted for by asserting that the deviation of the number of the arithmetic progressions coming to x from the expected value does not exceed $2\chi_\gamma m$. Thus we have proved

Claim 2. The probability that at a certain moment any specified empty point will be filled with a good insertion during the γ to $\gamma + \eta$ step is $((1 \pm 2\chi_\gamma)/(1 - \gamma))m$, *independently* of where any previously inserted elements among the ηm were located.

(We have written $2\chi_\gamma$ instead of $2(1 - \gamma)\chi_\gamma$ so as to incorporate the error that the $(1 - \gamma)$ in the denominator can really vary between $(1 - \gamma)$ and $(1 - \gamma - \eta)$.) The unavoidable bad insertions and the above small deviation from randomness is the way that clustering manifests itself in this argument. When we insert the new ηm points, the probability that an empty position will be filled is

$$\eta^* = \eta(1 \pm 2\chi_\gamma)/(1 - \gamma). \tag{vii}$$

This ignores the effect of the bad insertions, but their contribution can easily be incorporated in the overgenerous factor of 2 introduced above, since their number is $O(m^{1/2-\delta})$ which is much less than $\eta m = m^{3/4-\delta_1}$.

We are now ready to begin excluding those choices of the ηm points that would cause any of the intertype lattice flows to be excessively different from the expected value. We do this simultaneously for the lattices corresponding to all k, $2 \leqslant k \leqslant k_\alpha$ ($k = 0, 1$ cannot vary from the average) and all points x. We control the flows by allowing a maximum relative error of θ_0 for the intersections of our ηm points with each of the sets $S(i, \tau, x, \Gamma)$ of Definition 3.2.1, where Γ denotes our current configuration of γm occupied positions. By Theorem 2.2.1 we can do this while excluding only an exponentially small fraction of the choices of the ηm points as long as the expected size of the intersection is not too small. At the beginning of this section we introduced the residual progressions as a device for handling the small $S(i, \tau, x, \Gamma)$. For each i, τ, and x we demand that the intersections of both $S(i, \tau, x, \Gamma)$ and $R(i, \tau, x, \Gamma)$ with the ηm are within $(1 \pm \theta_0)$ of what we expect *if* $|S(i, \tau, x, \Gamma)| \geqslant |R(i, \tau, x, \Gamma)|$, otherwise we only demand this of the (disjoint) union $S(i, \tau, x, \Gamma) \cup R(i, \tau, x, \Gamma)$. In the latter case the intersection will have up to $(\sigma(\tau, x, \Gamma) + \rho(\tau, x, \Gamma))\, \eta^*(1 + \theta_0)m$ points. By relabeling some regular progressions as residual we can then still claim that the flow corresponding to the intersection of $S(i, \tau, x, \Gamma)$ with the ηm is $\eta^*\sigma(\tau, x, \Gamma)(1 \pm \theta_0)m$, provided we allow the flow corresponding to $R(i, \tau, x, \Gamma)$ to get as large as $\rho(\tau, x, \Gamma)\, \eta^*(1 + \theta_0)m$. Furthermore now no set whose intersection with the m we desire to control has cardinality less than $E_0 = m^{1/4+\delta_2}$. Theorem 2.2.1 then implies

Claim 3. During the step from load factor γ to load factor $\gamma + \eta$, if we exclude a fraction of choices of the ηm points that does not exceed $\exp(-C_0 m^{\delta_4})$ (for δ_4 as in (ii)), then we can assume that the intersection of the ηm points with each of $S(i, \tau, x, \Gamma)$ (τ a type of length at most k_α) will have cardinality $\sigma(\tau, x, \Gamma)\, \eta^*(1 \pm \theta_0)m$, and the intersection of the ηm with each of $R(i, \tau, x, \Gamma)$ will not be larger than $\rho(\tau, x, \Gamma)\eta^*(1+\theta_0)m$.

We now compute the relative error $\theta_{\gamma+\eta,k}$ in terms of $\theta_{\gamma,k}$. Let τ, the type we are now considering, have length k and l hits. We saw in Section 3.2 that in order to maximize the relative error for the type of k hits, we may assume that all intertype flows are maximal. As we will see momentarily, we can ignore any flows caused by progressions hit by more than one of the ηm points. Thus the maximal number of progressions of type τ we can have at any point when we reach load factor $(\gamma + \eta)$ is

$$[\gamma^l(1-\gamma)^{k-l}(1+\theta_{\gamma,k}) + l\gamma^{l-1}(1-\gamma)^{k-l+1}(1+\theta_{\gamma,k})\,\eta^*(1+\theta_0) - (k-l)\,\gamma^l(1-\gamma)^{k-l}(1+\theta_{\gamma,k})\,\eta^*(1+\theta_0)]m$$

already there — inflow — outflow — "+" since *all* flows are maximal

Figure 3.3.3 illustrates the inflow and outflow of progressions from a type.

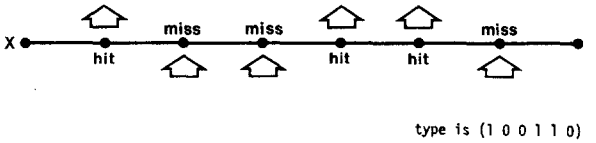


FIG. 3.3.3. The inflow and outflow of progressions from a type.

Ignoring the factor of m we can write the above expression as

$$\gamma^l(1-\gamma)^{k-l}(1+\theta_{\gamma,k}) + l\gamma^{l-1}(1-\gamma)^{k-l+1}(1+\theta_{\gamma,k})(1+\theta_0)\,\eta(1+2\chi_\gamma)/(1-\gamma)$$
$$-(k-l)\,\gamma^l(1-\gamma)^{k-l}(1+\theta_{\gamma,k})(1+\theta_0)\,\eta(1+2\chi_\gamma)/(1-\gamma).$$

This has to equal $(\gamma+\eta)^l(1-\gamma-\eta)^{k-l}(1+\theta_{\gamma+\eta,k})$, and so we get

$$1+\theta_{\gamma+\eta,k} = \frac{\gamma^l(1-\gamma)^{k-l}}{(\gamma+\eta)^l(1-\gamma-\eta)^{k-l}}[1+\theta_{\gamma,k}+[l\eta/\gamma](1+\theta_{\gamma,k})(1+2\chi_\gamma)(1+\theta_0)$$
$$-[(k-l)\,\eta/(1-\gamma)](1+\theta_{\gamma,k})(1+2\chi_\gamma)(1+\theta_0)].$$

Now

$$\frac{\gamma^l(1-\gamma)^{k-l}}{(\gamma+\eta)^l(1-\gamma-\eta)^{k-l}} = 1 - l\eta/\gamma + (k-l)\,\eta/(1-\gamma) + O(\eta^2).$$

If we ignore the η^2 terms, then we can rewrite the above as

$$\theta_{\gamma+\eta,k} = \theta_{\gamma,k} + (\eta l/\gamma)(\theta_0 + 2\chi_\gamma + 2\theta_0\chi_\gamma)\,\theta_{\gamma,k}$$
$$+ (\eta l/\gamma)(\theta_0 + 2\chi_\gamma + 2\theta_0\chi_\gamma)$$
$$- [\eta(k-l)/(1-\gamma)](\theta_0 + 2\chi_\gamma + 2\theta_0\chi_\gamma)\,\theta_{\gamma,k}$$
$$- [\eta(k-l)/(1-\gamma)](\theta_0 + 2\chi_\gamma + 2\theta_0\chi_\gamma).$$

The effect of the η^2 terms can be incorporated in the constants θ_0 or χ_γ, and so these terms can be justifiably ignored. We maximize the error $\theta_{\gamma+\eta,k}$ by taking $l = k$ above, so our final error propagation equation becomes

$$\theta_{\gamma+\eta,k} = \theta_{\gamma,k} + (\eta k/\gamma)(\theta_0 + 2\chi_\gamma + 2\theta_0\chi_\gamma)\,\theta_{\gamma,k} + (\eta k/\gamma)(\theta_0 + 2\chi_\gamma + 2\theta_0\chi_\gamma). \quad \text{(ix)}$$

Now we have $\chi_\gamma \leqslant \theta\gamma^s$ and we take $\theta_0 \leqslant \theta\gamma^s$, where s is a constant to be chosen below. For θ sufficiently small we will have $\theta_0 \leqslant 1$ and so we can make the errors $\theta_{\gamma+\eta,k}$ only larger by writing

$$\theta_{\gamma+\eta,k} = 5\theta\gamma^{s-1}k\eta(1+\theta_{\gamma,k}) + \theta_{\gamma,k}. \quad \text{(x)}$$

Going back through the above derivation and changing the signs of $\theta_{\gamma+\eta,k}$, $\theta_{\gamma,k}$, θ_0, and χ_γ gives us the error propagation equation for the negative errors. (Now we want all flows to be *minimal*.) We get the equivalent of (viii) for the absolute value of the error:

$$\theta_{\gamma+\eta,k} = \theta_{\gamma,k} - (\eta l/\gamma)(\theta_0 + 2\chi_\gamma - 2\theta_0\chi_\gamma)\,\theta_{\gamma,k}$$
$$+ (\eta l/\gamma)(\theta_0 + 2\chi_\gamma - 2\theta_0\chi_\gamma)$$
$$+ [\eta(k-l)/(1-\gamma)](\theta_0 + 2\chi_\gamma - 2\theta_0\chi_\gamma)\,\theta_{\gamma,k}$$
$$- [\eta(k-l)/(1-\gamma)](\theta_0 + 2\chi_\gamma - 2\theta_0\chi_\gamma).$$

Since $\gamma \leqslant \alpha_0$ which we can take less than $\frac{1}{2}$, we have $\gamma \leqslant 1 - \gamma$, and so we can conclude that (ix) is valid for the absolute value of the negative errors as well. Thus equation (x) is justified for the absolute value of both positive and negative errors.

We still have to estimate the size of the flows that involve more than one of the ηm points. Let us look at an arithmetic progression of length k coming to x that changes type by receiving two of the ηm points. Suppose these two points occupy positions i and j of the progression, respectively, $1 \leqslant i < j \leqslant k$. Let us fix the two types involved, which fixes i and j, and then ask how many progressions can flow between these two types. If the donating type is τ, then at most one such progression can flow for each pair (a, b) of the ηm points with the property that $a \in S(i, \tau, x, \Gamma)$ and a and b are in a distance ratio $i : j$ from x. If we allow the ηm points to range over all m points, not just the remaining $(1 - \gamma)m$ ones, we can only increase the flow in question. But now we are exactly in the situation covered by Theorem 2.4.4 and thus we can assert that our flow, except with exponentially small probability, will be $O(\eta \mid S(i, \tau, x, \Gamma)\mid m^{-\delta})$. Summing over all possible choices of i and j we still get a total possible inflow into the receiving type of $O(k_\alpha{}^2\gamma^{l-2}(1 - \gamma)^{k-l+2}\eta m^{1-\delta})$ only, where l denotes the number of hits of the receiving type τ'. A trivial extension of the above argument shows that any flows into τ' arising from types with 3 (or 4, etc.) fewer hits will not be any greater. Thus the total magnitude of these flows combined will be $o(\gamma^l(1 - \gamma)^{k-l}\eta m)$ and can therefore be incorporated into the relative errors permitted in Eq. (ix). Our derivation of Eq. (x) by ignoring type transitions which involve more than one of the ηm points has been justified.

We are finally at the point where we can push assertion (a) of our theorem through the induction step. We would like to prove

$$\begin{aligned}\theta_{\gamma+\eta,k} &= (1 + i_\theta)\, e^{5\theta[(\gamma+\eta)^s/s]k} - 1 \\ &= (1 + i_\theta)\, e^{5\theta[(\gamma^s/s)+\eta\gamma^{s-1}+O(\eta^2)]k} - 1,\end{aligned}$$

and since θ can be taken arbitrarily small, this is

$$\begin{aligned}&(1 + i_\theta)\, e^{5\theta(\gamma^s/s)k}(1 + 5\theta\gamma^{s-1}k\eta + O(k\eta^2)) - 1 \\ &\quad = (1 + \theta_{\gamma,k})(1 + 5\theta\gamma^{s-1}k\eta + O(k\eta^2)) - 1 \\ &\quad = \theta_{\gamma,k} + 5\theta\gamma^{s-1}k\eta(1 + \theta_{\gamma,k}) + (1 + \theta_{\gamma,k})\, O(k\eta^2);\end{aligned}$$

again the $O(k\eta^2)$ term is negligible compared to the $5\theta\gamma^{s-1}k\eta$ terms, and can be incorporated in the constant θ. Thus we need

$$\theta_{\gamma+\eta,k} = \theta_{\gamma,k} + 5\theta\gamma^{s-1}k\eta(1 + \theta_{\gamma,k}),$$

which is exactly what we have proved in (x).

Next we prove (b) and determine the constant s. It will be simpler to let the sum

$$\chi_\gamma = \sum_{k=2}^{k_\alpha} \theta_{\gamma,k}\gamma^k$$

go to infinity, which we are allowed to do, since this can only increase the bound for χ_γ. So

$$\chi_\gamma \leqslant \sum_{k=2}^{\infty} \theta_{\gamma,k}\gamma^k$$

then substituting $\theta_{\gamma,k}$ from (a) and letting

$$e^A = e^{5\theta(\gamma^s/s)},$$

we get

$$\begin{aligned}\chi_\gamma &\leqslant (1 + i_\theta)\,\gamma^2 e^{2A}/(1 - \gamma e^A) - \gamma^2/(1 - \gamma)\\ &= i_\theta \gamma^2 e^{2A}/(1 - \gamma e^A) + (e^A - 1)\,\gamma^2(1 + e^A - \gamma e^A)/(1 - \gamma)(1 - \gamma e^A).\end{aligned}$$

For small θ we have $e^A \sim 1$, $e^A - 1 \sim 5\theta(\gamma^s/s)$, and so if we take s so large that

$$(5/s)(2\alpha^2/(1 - \alpha)^2) < \tfrac{1}{2},$$

and i_θ so small that

$$i_\theta < ((1 - \alpha)/2\alpha^2)\,\beta^s\theta$$

then we will have

$$\chi_\gamma \leqslant \theta\gamma^s,$$

as desired. Note that s is independent of θ and γ.

The last object of interest is the residual sets. How fast can they grow? Let $r_\gamma^{(k,l)}$ denote $\rho(\tau, \gamma)m$ for τ a type of length k and l hits. The change in the r_γ's as we go from γ to $\gamma + \eta$ load factor can be computed in a manner analogous to the above. The maximum flow into τ during the current step will be

$$l r_\gamma^{(k,l-1)}\eta(1 + \theta)(1 + 2\chi_\gamma)/(1 - \gamma) \leqslant 2l r_\gamma^{(k,l-1)}\eta/(1 - \gamma).$$

We are not counting the outflows, so we obtain the recurrence relation

$$r_{\gamma+\eta}^{(k,l)} = r_\gamma^{(k,l)} + 2l r_\gamma^{(k,l-1)}\eta/(1 - \gamma), \qquad 0 \leqslant l \leqslant k \quad (r_\beta^{(k,l)} = E_0).$$

(Here we see that types with more hits will grow faster than types with fewer, since they have more types feeding into them. The same, of course, was true in our computation of the relative errors, but there we decided to ignore this improvement. Because the residual sets cause the argument to fail for large α, we want to do a better job of estimating their growth.) Since all initial values are identical, it is clear from (xi) that $r_\gamma^{(k_\alpha,k_\alpha)}$ is the maximally growing type. In what follows therefore we restrict ourselves to estimating its growth. Again, since η is infinitesimal compared to l or r_γ, we can easily check that the solution to the above difference equations (which we can think of as the system of differential equations

$$dr^{(k,l)}/d\gamma = 2l r^{(k,l-1)}/(1 - \gamma) \qquad (0 \leqslant l \leqslant k)$$

is of the form

$$r_\gamma^{(k,l)} = E_0(1 + 2\log(1/(1-\gamma)) - 2\log(1/(1-\beta)))^l$$
$$\leqslant E_0(1 + 2\log(1/(1-\gamma)))^l.$$

Thus

$$r_\gamma^{(k_\alpha,k_\alpha)} \leqslant E_0[1 + 2\log(1/(1-\gamma))]^{(1/2+\delta_3)\log m/\log(1/\alpha)}$$
$$= E_0 m^{(1/2+\delta_3)\log[1+2\log(1/(1-\gamma))]/\log(1/\alpha)},$$

and so

$$\rho_\gamma \leqslant (E_0 m^{(1/2+\delta_3)\log[1+2\log(1/(1-\gamma))]/\log(1/\alpha)})/m,$$

as (c) of Theorem 3.3.1 requires.

There are at most $1/\eta = m^{1/4+\delta_1}$ steps, at most m points x, at most $\sum_{2\leqslant k\leqslant k_\alpha} 2^{k_\alpha} \leqslant 2^{k_\alpha+1} = 2m^{(1/2+\delta_3)\log 2/\log(1/\alpha)}$ distinct types, and at most k_α values for i in the context $S(i, \tau, x, \Gamma)$. Thus the total number of excluded cases is a polynomial in m, and no case has probability higher than $\exp(-C_\theta m^{\delta_4})$. Thus as m gets large the total of the probabilities of the excluded cases is less than $\exp(-C_\theta m^{\delta_0})$, where δ_0 is as constrained in (ii).

This completes our proof of Theorem 3.3.1. ▌

COROLLARY 3.3.1. *Given $0 < \beta < \alpha < \alpha_0 < 1$, for any small positive constant θ, there exists an initial configuration of βm occupied points, such that if we add $(\alpha - \beta)m$ additional points using the double hashing process, then except with probability* $\exp(-C_\theta m^{\delta_0})$, *we will arrive at a configuration of αm occupied positions such that for each point x and for each length k, $2 \leqslant k \leqslant k_\alpha'$, the number of arithmetic progressions coming to x of length k from the occupied points will be $\alpha^k(1 \pm \underline{\theta}_{\alpha,k})$. Here the $\underline{\theta}_{\alpha,k}$ are relative errors satisfying*

$$\text{(a)} \quad \sum_{k=2}^{k_\alpha'} \underline{\theta}_{\alpha,k}\alpha^k \leqslant \theta$$

and

$$\text{(b)} \quad \alpha^{k_\alpha'}(1 \pm \underline{\theta}_{\alpha,k_\alpha'})\, m \leqslant m^{1/2-\delta'},$$

for some positive constants δ_0, δ'. (Notice that we have excluded any references to bad insertions or residual progressions.)

Proof. This is a direct consequence of Theorem 3.3.1, except for a few items that we need to check. First is the existence of a good initial configuration, the "seed" of the extension process. We have to choose a configuration of βm points so that for each point x and each type τ of length k and l hits we have

$$\beta^l(1-\beta)^{k-l}m(1 \pm i_\theta)$$

progressions of type τ coming to x, with i_θ, k, l restricted as in the theorem. Now if $\beta^l(1-\beta)^{k-l}m \geqslant m^{1/4}$, then by Theorem 2.4.2, all configurations except for a fraction not exceeding $\exp(-Ci_\theta^2 m^{1/5})$ of them will satisfy the above condition. If $\beta^l(1-\beta)^{k-l}m \ll$

$m^{1/4}$, then let τ_1 denote a shorter type which is an initial segment of τ, of length k_1 and l_1 hits, such that $\beta^{l_1}(1-\beta)^{k_1-l_1}m \approx m^{1/4}$. Then we can apply Corollary 2.4.2 to τ_1 and claim it has at most $O(m^{1/4})$ progressions. Clearly τ cannot have more progressions than τ_1, so therefore the excess of progressions τ can have is at most $O(m^{1/4})$. Any such excessive progressions we label as residual for out type τ. This is consistent with the assumptions of Theorem 3.3.1 that allow initial residual errors as large as $m^{1/4+\delta_2}$ per type. Therefore all configurations of the βm points except for an exponentially small fraction of them satisfy the initial conditions of Theorem 3.3.1. We start the extension process by choosing one of them. This is an interesting "nonconstructive" aspect of our proof. We do not know how to find a specific such good configuration, though we have just proved that almost all configurations are good.

We now perform the extension process till we reach the load factor α, as described in Theorem 3.3.1. The number of points inserted with bad insertions is $O(m^{1/2-\delta})$. For each point x and each length k, no bad point can introduce (influence) more than k progressions of length k coming to x. Thus the bad points can introduce at most $O(km^{1/2-\delta})$ progressions of length k coming to x, $k \leqslant k_\alpha = O(\log m)$. The number of residual progressions of length k coming to x is at most

$$E_0 m^{(1/2+\delta_3)\log[1+2\log(1/(1-\alpha))]/\log(1/\alpha)} = m^{1/4+\delta_2+(1/2+\delta_3)\log[1+2\log(1/(1-\alpha))]/\log(1/\alpha)}.$$

The absolute constant α_0 is chosen so that

$$1/4 + \delta_2 + (1/2 + \delta_3)\log[1 + 2\log(1/(1-\alpha))]/\log(1/\alpha) \leqslant 1/2 - \delta$$

for $\alpha \leqslant \alpha_0$. A rough numerical computation shows that

$$\alpha_0 \sim 0.319.$$

Thus the contribution of the residual progressions at any length is at most $O(m^{1/2-\delta})$. Let now δ', δ'' be such that $0 < \delta' < \delta'' < \delta, \delta_3$. Let $k_\alpha' < k_\alpha$ be such that

$$\alpha^{k_\alpha'} m = m^{1/2-\delta''};$$

such a k_α' clearly exists since

$$\alpha^{k_\alpha} m = m^{1/2-\delta_3}.$$

We have

$$\begin{aligned} 1 + \theta_{\alpha,k_\alpha'} &= (1 + i_\theta)\, e^{5\theta(\alpha^s/s)k_\alpha'} \\ &= (1 + i_\theta)\, m^{5\theta(\alpha^s/s)(1/2+\delta'')/\log(1/\alpha)}. \end{aligned}$$

Recall that $i_\theta \to 0$ as $\theta \to 0$, and so by choosing θ sufficiently small we can obtain

$$1 + \theta_{\alpha,k_\alpha'} \leqslant m^{\delta'''}, \qquad \text{for} \quad \delta''' < \delta'' - \delta'.$$

So we have

$$\alpha^{k_\alpha'} m(1 + \theta_{\alpha,k_\alpha'}) \leqslant m^{1/2-\delta''+\delta'''}.$$

We can add to this the contribution of the bad insertions and the residual progressions, and since they both are $O(m^{1/2-\delta''})$, the grand total of progressions of length k_α' coming to x is

$$\alpha^{k_\alpha'} m(1 + \underline{\theta}_{\alpha,k_\alpha'}) \leqslant m^{1/2-\delta'}.$$

This proves part (b) of the Corollary.

For part (a) we work analogously. We know that

$$\sum_{k=2}^{k_\alpha} \theta_{\alpha,k}\alpha^k \leqslant \theta\alpha^s \leqslant \theta. \tag{xii}$$

The contributions of the bad insertions and the residual progressions estimated as above are $O(m^{1/2-\delta''})$ even when summed over all allowed lengths k. Thus these contributions to (xii) can be incorporated in the constant θ. Since $k_\alpha' \leqslant k_\alpha$, we have shown that the *true* relative errors satisfy

$$\sum_{k=2}^{k_\alpha'} \underline{\theta}_{\alpha,k}\alpha^k \leqslant \theta$$

as desired.

By Theorem 3.3.1 the probability of the excluded events is $\exp(-C_\theta m^{\delta_0})$. This completes the argument. ∎

Remark. It is worth pointing out the reason why we have carried out the computation of the growth of the residual progressions separately from the regular ones. For the regular progressions, the initial number of progressions of a type τ of length k and l hits is approximately $\beta^l(1-\beta)^{k-l}m$. Thus for β small and a particular k we have most regular progressions in types with few hits. The initial number of residual progressions, however, is the same for all types, thus giving rise to a quantitatively different model.

Figure 3.3.4 is to be used for reference. It summarizes the various δ's we have introduced and the relations among them.

Definitions

$\eta = m^{-1/4-\delta_1}$
$\beta^k\beta m = m^{1/4+\delta_2}$
$\alpha^k\alpha m = m^{1/2-\delta_3}$
$E_0 = m^{1/4+\delta_2}$
Prb. of excluded events $= e^{-C_\theta m^{\delta_0}}$
Prb. of bad insertion $= m^{-1/2-\delta}$

Constraints

$\delta_2 > \delta_1 > 0$
$\delta_2 - \delta_1 > \delta_4 > \delta_0 > 0$
$2\delta_1, \delta_3 > \delta_6 > \delta > 0$
$2\delta_1 - \delta, \delta_3 - \delta > \delta_5 > 0$
$\delta_3, \delta > \delta'' > \delta' > 0$

FIG. 3.3.4. The proliferation of deltas.

ACKNOWLEDGMENTS

The results of this paper form part of the first author's Ph.D. Dissertation at Stanford University [8]. Both authors would like to acknowledge the encouragement of Professor D. E. Knuth throughout this research. The germ of the "pull-back process" idea of Section 2.4 is due to Vasek Chvatal. The first author would also like to thank the Fannie and John Hertz Foundation and the Xerox Palo Alto Research Center for financial support during the period in which this research was carried out. Mark R. Brown, Louis Trabb-Pardo, Janet R. Roberts, Edward M. McCreight, and the anonymous referee offered valuable suggestions for improving the exposition of this paper.

REFERENCES

1. J. R. BELL AND C. H. KAMAN, The linear quotient hash code, *Comm. ACM* **13** (1970), 675–677.
2. R. P. BRENT, Reducing the retrieval time of scatter storage techniques, *Comm. ACM* **16** (1972), 105–109.
3. H. CHERNOFF, A measure of asymptotic efficiency for tests of hypotheses based on a sum of observations, *Ann. Math. Statist.* **23** (1952), 493–509.
4. P. ERDÖS AND J. SPENCER, "Probabilistic Methods in Combinatorics," Academic Press, New York, 1974.
5. W. FELLER, "An Introduction to Probability Theory and Its Applications," 3rd ed., Vol. 1, Sect. II.6, Wiley, New York, 1968.
6. L. J. GUIBAS, The analysis of hashing algorithms that exhibit k-ary clustering, *in* "Proceedings of the 1976 FOCS Conference, Houston, Texas, October 1976."
7. L. J. GUIBAS, "On the Distribution of Arithmetic Progressions of Length 3 in Random Samples of the Integers 1, 2,..., N," Unpublished Manuscript, Stanford, Calif., February 1975.
8. L. J. GUIBAS, "The Analysis of Hashing Algorithms," Computer Science Ph.D. Thesis at Stanford Univ. June 1976 (also available as Xerox PARC CSL Report CSL-76-3).
9. G. H. HARDY AND E. M. WRIGHT, "An Introduction to the Theory of Numbers," 4th ed., Chapts. III, X, Oxford, London/New York, 1968.
10. D. E. KNUTH, "The Art of Computer Programming," Vol. 1, "Fundamental Algorithms," 2nd ed., Sect. 6.4, Addison–Wesley, Reading, Mass., 1975.
11. D. E. KNUTH, "The Art of Computer Programming," Vol. 3, "Sorting and Searching," Sect. 1.2, Addison–Wesley, Reading, Mass., 1973.
12. D. E. KNUTH, Mathematical analysis of algorithms, Inf. Process. Lett. (1972), 19–27.
13. A. RENYI, "Probability Theory," Chap. VII, North–Holland, Amsterdam, 1970.
14. J. D. ULLMAN, A note on the efficiency of hash functions, *J. Assoc. Comput. Mach.* **19** (1972), 569–575.

Printed in Belgium