Average-Case Analysis of Random Partial Match Queries in Random Relaxed K-d Trees ADS-MIRI

Amalia Duch

March 6, 2014

1 Generating Functions

Here you have a very brief introduction of those mathematical tools that are required for the analysis of the algorithms presented in next chapters. It is assumed that the reader is familiar with concepts such as analyticity, convergence of power series, singularity and residue complex analysis and, Taylor series developments. Excellent references on these issues are [10, 11].

The basic tool in the average case analysis of algorithms and data structures used in this work is the *generating function* [9].

Definition Given any sequence of complex numbers $\{a_k\}_{k\geq 0}$, its generating function is

$$A(z) = \sum_{k\geq 0} a_k z^k$$

where z is an auxiliary variable and k a non-negative integer.

The generating function of a sequence is often called *ordinary* generating function, in order to distinguish it from other kinds of generating functions such as *exponential* generating functions or *probability* generating functions.

Generating functions associate sequences of numbers to formal power series making possible to manipulate them with classical algebraic methods. In fact, elementary operations over sequences can be easily translated into operations over the corresponding generating functions. See Table 1, where operations (1), (3) and (4) correspond to *sum*, *backward shift* and *forward shift* of sequences. Operation (2) is known as *convolution* of sequences and operations (5) and (6) are *differentiation* and *integration* of sequences, respectively.

In many applications, it is often the case that the power series under study are convergent and in consequence they can be treated by analytical methods. In such cases the variable z of a generating function $f(z)$ is considered as a complex variable and the generating function as a complex function of z.

The n-th coefficient of a formal power series $f(z)$ will be denoted by $[z^n]f(z)$ (which also denotes the n-th coefficient of the Taylor expansion of an analytic function $f(z)$ around $z = 0$). Thus, if $f(z) = \sum_{n\geq 0} f_n z^n$ it follows that $[z^n]f(z) = f_n$.

Sequences	Generating Functions
1. $c_n = a_n \pm b_n$	$C(z) = A(z) \pm B(z)$
2. $c_n = \sum_{k=0}^{n} a_n b_{n-k}$	$C(z) = A(z) \cdot B(z)$
3. $c_n = a_{n-1}$	$C(z) = zA(z)$
4. $c_n = a_{n+1}$	$C(z) = \frac{A(z)-A(0)}{z}$
5. $c_n = na_n$	$C(z) = z\frac{d}{dz}A(z)$
6. $c_n = \frac{a_n}{n}$	$C(z) = \int_0^z (A(z) - A(0))\frac{dt}{t}$

Table 1: Translation of basic operations over sequences onto operations over generating functions.

In the average case analysis of algorithms it is necessary to recover the n-th coefficient of a generating function. There are two useful theorems that allow to extract this coefficient exactly, under particular circumstances. The first one is the so-called general expansion theorem for rational generating functions while the second is the Lagrange inversion formula, that can be applied when the generating functions satisfy a specific kind of implicit equation. The statements are as follows.

Theorem 1.1. (Expansion Theorem for Distinct Roots) *If $R(z) = P(z)/Q(z)$, for some polynomials $P(z)$ and $Q(z)$, such that the degree of $P(z)$ is smaller than the degree of $Q(z)$, and $Q(z)$ has r distinct roots $\rho_1, \rho_2, \ldots, \rho_r$ of multiplicities $d_1, d_2, \ldots, d_r$, then*

$$[z^n]R(z) = f_1(n)\rho_1^n + f_2(n)\rho_2^n + \cdots + f_r(n)\rho_r^n, \qquad n \geq 0,$$

where each $f_k(n)$ is a polynomial of degree $d_k - 1$ and its leading coefficient a_k is

$$a_k = \frac{d_k P(1/\rho_k)(-\rho)^{d_k}}{Q^{(d_k)}(1/\rho_k)}.$$

A proof of this theorem can be found in [9], in page 340.

Theorem 1.2. (Lagrange Inversion Formula) *Let $\phi(u) = \sum_{n \geq 0} \phi_n z^n$ be a formal power series such that $\phi(0) = \phi_0 \neq 0$. Then the equation*

$$y(z) = z\phi(y(z))$$

has a unique formal power series solution that satisfies

$$y(z) = \sum_{n>0} \frac{z^n}{n}[y^{n-1}]\phi^n(y).$$

However, to find an exact expression for the coefficients of a power series is not always possible and a good enough solution is to get asymptotic estimates. In order to get asymptotic estimates of the n-th coefficient of generating functions there are useful methods provided by the analysis of function's singularities that we describe in next section.

2 Singularity Analysis

As we already said, generating functions can be considered as functions of complex variable in the complex plane, analytic in a disk around the origin. A *singularity* is a point at which the function ceases to be analytic. Let $f(z)$ be the generating function of a certain sequence. Since the power series of the function is analytic in the largest disk centered at the origin containing no singularities, the first step in the analysis will be to look for the singularities that are nearest to the origin. The nearest singularity is called *dominant singularity* and the distance from the origin to the dominant singularity is called *radius of convergence* of the power series. The radius of convergence provides useful information about the behavior of the coefficients of the power series, $f_n = [z^n]f(z)$, as stated by next theorem, which relates the location of singularities of a function to the exponential growth of its coefficients.

Theorem 2.1. (Exponential Growth Formula) *Let ρ be the radius of convergence of the power series $f(z) = \sum_{n\geq 0} f_n z^n$. Then, for all $\epsilon > 0$,*

$$(1-\epsilon)^n \rho^{-n} <_{i.o.} f_n <_{a.e.} (1+\epsilon)^n \rho^{-n},$$

where $<_{i.o.}$ means that the inequality holds for an infinite number of values of n, whereas $<_{a.e.}$ means that the inequality holds for all values of n, except a finite number of them.

Although these lower and upper bounds are useful information about the exponential growth of the coefficients f_n, it is usually insufficient and it is required to look for information on their sub-exponential growth, or preferable, to look for an asymptotic equivalent.

It is known from analysis that a non-entire function with positive coefficients has always a dominant positive real singularity. In most cases, it is possible to obtain information about the asymptotic behavior of the coefficients f_n by extracting information about the nature of the dominant singularity of the function $f(z)$ and the behavior of the function around it.

Singularity analysis methods are based on the assumption that a function $f(z)$ has, around its dominant singularity 1, an asymptotic expansion of the form: $f(z) = \sigma(z) + R(z)$ with $R(z) << \sigma(z)$ as $z \to 1$ and where $\sigma(z)$ is a standard set of functions that include $(1-z)^a \log^b(1-z)$ for constants a and b. Then, under general conditions,

$$[z^n]f(z) = [z^n]\sigma(z) + [z^n]R(z),$$

with $[z^n]R(z) << [z^n]\sigma(z)$, as $n \to \infty$.

Applications of this principle are based on varying the conditions imposed to functions $f(z)$ and $R(z)$ resulting in three principal methods:

1. *Transfer methods* where the approximation is established for $z \to 1$ and there are usually suppositions on the growth of the remainder term $R(z)$.

2. *Tauberian theorems* which impose conditions of positivity and monotonicity to be satisfied by the coefficients f_n and hold when z is real and less than 1. These theorems also require conditions on the growth of $R(z)$ but less restrictive than those of transfer methods.

3. *Darboux's method* imposes as condition the differentiability of $R(z)$.

We will use Transfer-Lemma 2.2 and Corollaries 2.3 and 2.4.

Lemma 2.2. (Transfer Lemma [7]) *Assume that $f(z)$ is analytic in $|z| < 1$. Assume further that as $z \to 1$ in this domain,*

$$f(z) = \mathcal{O}\left(|1-z|^{\alpha}\right),$$

for some real number $\alpha < -1$. Then the n-th Taylor coefficient of $f(z)$ satisfies,

$$f_n = [z^n]f(z) = \mathcal{O}\left(n^{-\alpha-1}\right).$$

Corollary 2.3. *Assume that $f(z)$ is analytic in $|z| < 1$. Assume further that as $z \to 1$ in this domain,*

$$f(z) = o\left(|1-z|^{\alpha}\right),$$

for some real number $\alpha < -1$. Then the n-th Taylor coefficient of $f(z)$ satisfies,

$$f_n = [z^n]f(z) = o\left(n^{-\alpha-1}\right).$$

Before giving the statement of next corollary we require the definition of asymptotic equivalence. We say that the coefficients a_n and b_n are *asymptotically equivalents* and we denote it $a_n \sim b_n$, if and only if

$$\lim_{n\to\infty} \frac{a_n}{b_n} = 1.$$

Equivalently, we say that the functions $f(n)$ and $g(n)$ are *asymptotically equivalents* and we denote it $f(n) \sim g(n)$, if and only if

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = 1.$$

This is slightly more informative than Θ as it gives the coefficient of the leading term as well as the precise order.

Corollary 2.4. *Assume that $f(z)$ is analytic in $|z| < 1$. Assume further that as $z \to 1$ in this domain,*

$$f(z) \sim (|1 - z|^{\alpha}),$$

for some real number $\alpha < -1$. Then the n-th Taylor coefficient of $f(z)$ satisfies,

$$f_n = [z^n]f(z) \sim \left(n^{-\alpha-1}\right),$$

For further and insight information on singularity analysis we refer you to [7] and [14].

3 The Partial Match Algorithm

Let us start with the description of the partial match algorithm and the previous results given for its average-case performance in standard K-d trees, K-d-t trees and squarish K-d trees.

We briefly review that in a partial match search we are given a query $q = (q_0, q_1, \ldots, q_{K-1})$, with $q_i \in [0, 1] \cup \{*\}$ and the goal is to report those points in the file that match the query, that is, the points x such that $x_i = q_i$ if $q_i \neq *$, for all $0 \leq i < K$. For a query q, the bit-string $w = (w_0, w_1, \ldots, w_{K-1})$ such that $w_i = 0$ if $q_i = *$ and $w_i = 1$ otherwise, is called the *specification pattern* of the query. A query might then be thought as a pair consisting in a point $y \in [0, 1]^K$ and a bit-string w. Partial matches make sense if at least one coordinate of the query is specified and at least one coordinate is not.

The algorithm for partial match searches over relaxed K-d trees explores the tree in the following way. At each node of the K-d tree it examines the corresponding discriminant. If that discriminant is specified in the query then the algorithm recursively follows in the appropriate subtree, depending on the result of the comparison between the attribute of the query and the attribute of the key stored at the current node. Otherwise (that is, the discriminant at the current node is not specified in q), the algorithm recursively follows the two subtrees of the node.

The performance of partial match queries has been extensively studied for several multidimensional data structures (see for instance [2, 5, 8, 12, 13]). The average cost P_n of performing partial match searches in hierarchical K-dimensional data structures of size n, when s out of the K attributes of the query are specified is of the form $P_n = \beta \cdot n^{\alpha(s/K)}$. In the particular case of standard K-d trees, it has been shown by Flajolet and Puech [8] that partial match queries are efficiently supported by random standard K-d trees with an expected cost of $\beta \cdot n^{1-\frac{s}{K}+\theta(\frac{s}{K})}$, where n is the size of the tree and $\theta(x)$ is the unique real solution of

$$(\theta(x) + 3 - x)^x(\theta(x) + 2 - x)^{1-x} - 2 = 0.$$

whose value never exceeds 0.07. No closed expression for the β's is obtained in the paper, but their numerical values are given for $K \leq 4$. The complete characterization of the leading constant β is given by Chern and Hwang [2] using an asymptotic theory for Cauchy-Euler differential equations [1].

K-d-t trees are similar to standard K-d trees (when $t = 0$ they coincide) but subject to local re-balancing of subtrees of size $\geq 2t + 1$ [4]; for this variant $\theta(x) = \theta_t(x)$ is the unique solution of

$$[(\theta(x) + 3 + t - x)(\theta(x) \cdots (\theta(x) + 3 + 2t - x)]^x \cdot \\ [(\theta(x) + 2 + t - x)(\theta(x) \cdots (\theta(x) + 2 + 2t - x)]^{1-x} - \frac{2t + 2!}{t + 1!} = 0.$$

The authors provide the value of β for some specific patterns of the query, as well as the expected cost of standard partial matches for several values of t and n. Once again, characterizations for β are provided by Chern and Hwang [2].

Squarish K-d trees [5] have optimal performance since $\theta(x) = 0$; however, the values of the β's are not yet known and the only way for the moment to compute them would be through experimental measurement.

For quad trees the analysis of partial match can be found in [6]. In this case $\theta(x)$ is the same as for standard K-d trees. But the β's depend only on K and s. For $K = 2$, $\beta = \Gamma(2\alpha + 2)/(2\alpha^3\Gamma^3(\alpha)) \approx 1.5950991$. For higher dimensions, the complete characterization is given by Chern and Hwang [3].

For relaxed K-d tries [13] $\alpha = \alpha(x) = \log_2(2 - x)$ and $\beta_w = \beta(\text{order}(w)/K)$ with

$$\beta(x) = \frac{1}{\log 2}\Big((\alpha(x) - 1)\Gamma(-\alpha(x)) + \delta(\log_2 n)\Big),$$

and $\delta(\cdot)$ a periodic function of period 1, mean 0 and small amplitude that also depends on α.

In what follows, we analyze the average cost of performing random partial match queries in random relaxed K-d trees.

4 The Cost of Partial Match Searches

The average-case analysis of the partial match algorithm for relaxed K-d trees is based on the assumption that the K-d tree is random as well as are the queries. By definition a relaxed K-d tree of size n is random if the $n!^K$ possible sequences to build it are equiprobable. For the queries, we say that a partial match query with s out of K attributes specified is random if each attribute has a probability s/K of being specified.

These definitions imply that the partial match algorithm in random relaxed K-d trees with random queries explores the tree in the following way. At each node the corresponding discriminant will be specified in the query with probability $\frac{s}{K}$, then the algorithm will recursively follow the appropriate subtree, depending on the result of the comparison between attributes. With complementary probability (that is $\frac{K-s}{K}$), the corresponding discriminant will be unspecified in the query, so the algorithm will follow the two subtrees recursively.

The following theorem gives the expected performance of a partial match query in a random relaxed K-d tree.

Theorem 4.1. *The expected cost P_n (measured as the number of comparisons) of a partial match query with s out of K attributes specified in a random relaxed K-d tree of size n is*

$$P_n = \beta n^\alpha + \mathcal{O}(1),$$

where

$$\begin{aligned}\alpha = \alpha(s/K) &= 1 - \frac{s}{K} + \phi(s/K) \\ &\sim_{(s/K)\to 0} 1 - \frac{2}{3}\left(\frac{s}{K}\right) + \mathcal{O}\left(\left(\frac{s}{K}\right)^2\right), \\ \beta &= \frac{\Gamma(2\alpha+1)}{(1-s/K)(\alpha+1)\Gamma^3(\alpha+1)}\end{aligned}$$

with $\phi(x) = \sqrt{9-8x}/2 + x - 3/2$ and $\Gamma(x)$ the Euler's Gamma function [9].

Proof. Let T be a random relaxed K-d tree of size n with left subtree L and right subtree R, and let $P(T)$ be the average search cost of a partial match search in T. Then, with probability $\frac{K-s}{K}$, the discriminant in the root of T is an unspecified attribute of the query. In this case, the search visits the root and then continues in both subtrees L and R. The cost will be the sum of the cost of visiting the root (one comparison) plus the cost of visiting subtrees L and R which corresponds to $P(L) + P(R)$ and it is reflected by the first term of the right hand side of the equation here-below.

With probability $\frac{s}{K}$ the discriminant in the root of T corresponds to some specified attribute then, after visiting the root, the partial match retrieval continues into the appropriate subtree. It will continue along L with probability $\frac{l+1}{n+1}$ (where l is the size of L) and along R with the complementary probability. In this case, the cost is reflected in the second term of the right hand side of the equation hereafter and it corresponds to the cost of visiting the root plus the cost of visiting either L or R with their corresponding probabilities. Thus, the search cost satisfies the relation

$$\begin{aligned}P(T \mid\; \mid L \mid = l) &= \frac{K-s}{K}(1 + P(L) + P(R)) \\ &\quad + \frac{s}{K}\left(1 + \frac{l+1}{n+1}P(L) + \frac{n-l}{n+1}P(R)\right).\end{aligned}$$

Taking the average over all the possible values of l, and since the probability that L has size l is $1/n$ for all l, with $0 \le l < n$, because T is assumed to be random, we find that, for $n > 0$, the expected cost P_n of a partial match query in a random relaxed K-d tree of size n is

$$P_n = 1 + \frac{K-s}{K}\left(\frac{1}{n}\sum_{l=0}^{n-1}[P_l + P_{n-1-l}]\right) + \frac{s}{K}\left(\frac{1}{n}\sum_{l=0}^{n-1}\left[\frac{l+1}{n+1}P_l + \frac{n-l}{n+1}P_{n-1-l}\right]\right),$$

which by symmetry is equivalent to

$$P_n = 1 + 2\left(\frac{K-s}{K}\right)\frac{1}{n}\sum_{l=0}^{n-1} P_l + 2\left(\frac{s}{K}\right)\frac{1}{n}\sum_{l=0}^{n-1}\frac{l+1}{n+1}P_l. \qquad (1)$$

To derive the expression for P_n we use generating functions and singularity analysis [7]. By Definition 1 the ordinary generating function of the sequence $\{P_n\}_{n\geq 0}$, is $P(z) = \sum_{n\geq 0} P_n z^n$, with $P(0) = 0$. Let us define the generating function $R(z) = \sum_{n\geq 0}(n+1)P_n z^n = zP'(z) + P(z)$, where $R(0) = 0$. Multiplying Equation (1) by $(n+1)$ gives

$$(n+1)P_n = (n+1) + 2\left(\frac{K-s}{K}\right)\frac{n+1}{n}\sum_{l=0}^{n-1} P_l + 2\left(\frac{s}{K}\right)\frac{1}{n}\sum_{l=0}^{n-1}(l+1)P_l.$$

This recurrence translates to the following integral equation

$$R(z) = \frac{1}{(1-z)^2} - 1 + 2\frac{K-s}{K}\left(\frac{P(z)}{1-z} - P(z) + \int_0^z \frac{P(t)}{1-t}dt\right) + 2\frac{s}{K}\int_0^z \frac{R(t)}{1-t}dt.$$

Taking derivatives and expressing $R(z)$ in terms of $P(z)$ gives the second order non-homogeneous differential equation

$$P''(z) - 2\frac{(2z-1)P'(z)}{z(1-z)} - 2\frac{(2-s/K-z)P(z)}{z(1-z)^2} - 2\frac{1}{z(1-z)^3} = 0. \qquad (2)$$

The homogeneous differential equation associated to Equation (2) has only z and $(1-z)^p$ as divisors for $p = 1, 2$. Thus, $P(z)$ has a single singularity at $z = 1$ and because p is integer, the function is meromorphic with a single pole at $z = 1$. Thus, the dominant contribution in the local expansion of $P(z)$, when $z \to 1$, is of the form $P(z) \sim \beta(1-z)^\xi$, with ξ the smallest root of the indicial equation: $\xi^2 + \xi - 2(1 - s/K) = 0$, which results in $\xi = \frac{-1-\sqrt{(9-8s/K)}}{2}$.

Because of Lemma 2.2 it follows that,

$$P_n = [z^n]P(z) \sim \beta n^\alpha \qquad (3)$$

for $\alpha = -\xi - 1$ and some constant β.

Expanding $\alpha(s/K)$ in Taylor series, we obtain that $\alpha(s/K) \sim 1 - 2/3(s/K) + \mathcal{O}((s/K)^2)$.

The approach that we have sketched above for the analysis of partial match can be further explored finding the result,

$$P_n = [z^n]P(z) = \beta n^\alpha + \mathcal{O}(1), \qquad (4)$$

which is stronger than (3). In fact in [13] it has been shown that the exact solution of Equation (2) is,

$$P(z) = \frac{1}{1-s/K}\left({}_2F_1\left(\begin{matrix}a, b\\ 2\end{matrix}\,\middle|\, z\right)(1-z)^\xi - \frac{1}{1-z}\right), \qquad (5)$$

where ${}_2F_1\left(\begin{smallmatrix}a,b\\ c\end{smallmatrix}\,\middle|\, x\right)$ is the hypergeometric function [9], with $a = 2 + \xi$ and $b = 1 + \xi$. Then, we can study the asymptotic behavior of $P(z)$ when $z \to 1$

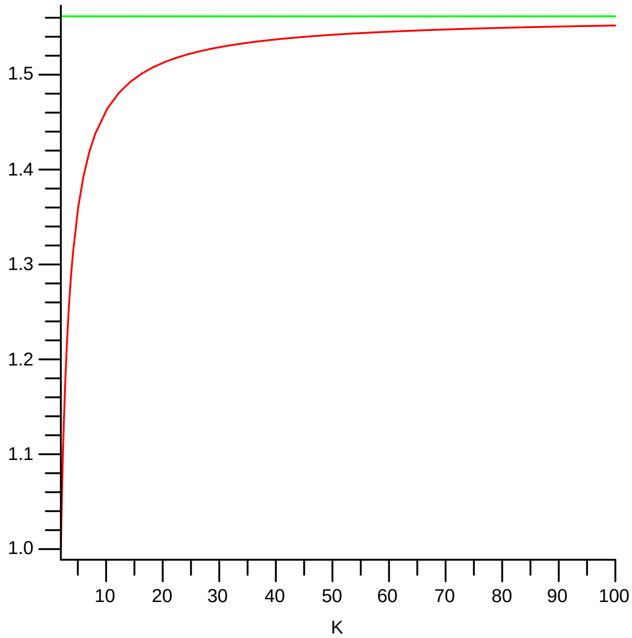

Figure 1: The value of the exponent in the average cost of partial match queries in relaxed random K-d trees.

to get not only the precise order of magnitude of P_n but the coefficient of the main order term and the magnitude of the lower order terms. The second term of $P(z)$ makes a contribution which is $\mathcal{O}(1)$ and the hypergeometric function is analytic at $z = 1$. Therefore,

$$\beta = \frac{{}_2F_1\left(\left.{a,b \atop 2}\right| 1\right)}{(1 - s/K)\Gamma(-\xi)} = \frac{\Gamma(2\alpha + 1)}{(1 - s/K)(\alpha + 1)\Gamma^3(\alpha + 1)}.$$

□

It is interesting to point out that, although Theorem 4.1 is valid only if $0 < \frac{s}{K} < 1$, it provides meaningful information for the limiting cases-at least, to some extent. If $\frac{s}{K} \to 0$, that is, no attribute is specified, then $P_n = n$. Indeed, $\alpha \to 1$ and $\beta \to 1$ as $\frac{s}{K} \to 0$. On the other hand, in an exact match all attributes are specified and $s = K$. In this case, we know that $P_n = \Theta(\log n)$. And we have that $\alpha \to 0$ and $\beta \to \infty$ if $\frac{s}{K} \to 1$, which is an approximate way to say with a formula like βn^α that P_n grows with n, but slower than any function of the type n^ϵ, for real positive ϵ.

In Figures 1 and 2 we plot respectively the value of the exponent α in the average cost of partial match queries in random K-d trees and random relaxed K-d trees, and the excess of the corresponding exponents with respect to $1 - \frac{s}{K}$, since $\Theta(n^{1-s/K})$ is the best known lower bound for partial match queries. In figure 3 we plot the value of β as a function of the ratio $\rho = s/K$.

The expected cost of partial match queries in random relaxed K-d trees is slightly higher than the one given by Flajolet and Puech [8] for random K-d

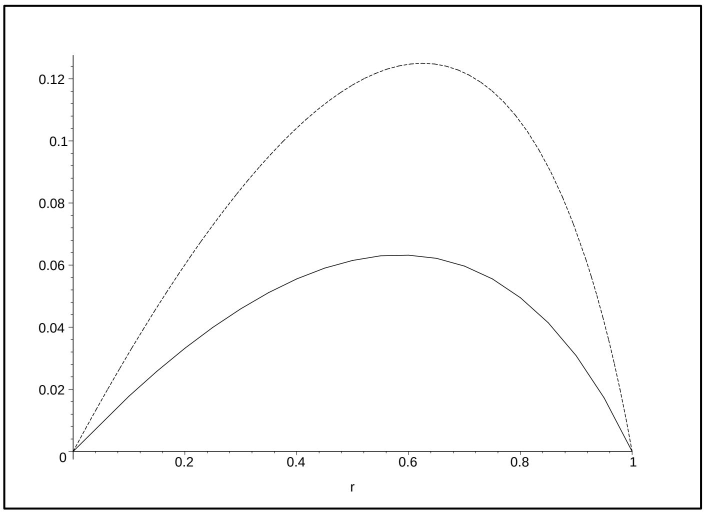

Figure 2: Excess (with respect to $1 - \rho$) of the exponent in the average cost of partial match queries in relaxed (dashed line) and standard (solid line) random K-d trees.

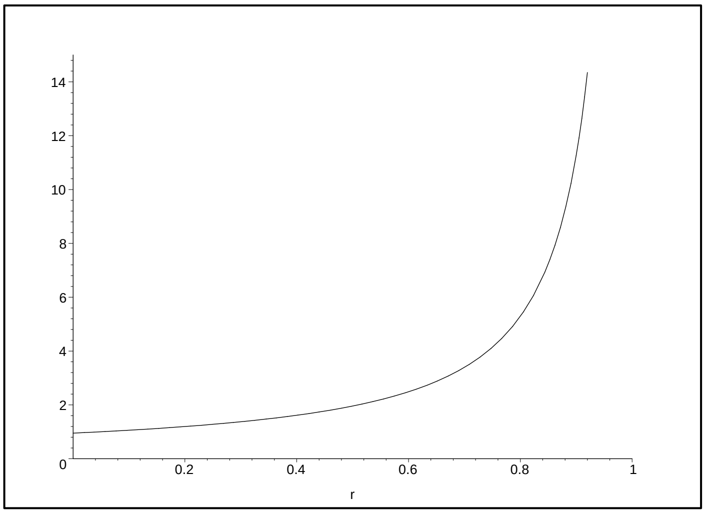

Figure 3: The value of the constant β in the average cost of partial match in random relaxed K-d trees.

trees. In fact, the values of $\phi(s/K)$ never exceed 0.12. It is possible to show that the difference in the exponent of n of these costs is at most 0.08, though-and for extreme values of s/K the difference is much smaller. Notice also that the constant β in the main order term of the expected cost of partial match queries in relaxed K-d trees is independent of the specification pattern of the query whereas for standard K-d trees such a constant is dependent on the particular pattern of the query [2, 8].

References

[1] H. H. Chern and H. K. Hwang. An asymptotic theory for Cauchy-Euler differential equations with applications to the analysis of algorithms. *Journal of Algorithms*, 44(1):177–225, 2002.

[2] H. H. Chern and H. K. Hwang. Partial match queries in random k-d trees, 2003. Submitted for publication.

[3] H. H. Chern and H. K. Hwang. Partial match queries in random quad trees. *SIAM Journal on Computing*, 32(4):904–915, 2003.

[4] W. Cunto, G. Lau, and Ph. Flajolet. Analysis of KDT-trees: KD-trees improved by local reorganisations. In F. Dehne, J. R. Sack, and N. Santoro, editors, *Workshop on Algorithms and Data Structures (WADS'89)*, volume 382 of *LNCS*, pages 24–38. Springer–Verlag, 1989.

[5] L. Devroye, J. Jabbour, and C. Zamora-Cura. Squarish k-d trees. *SIAM Journal on Computing*, 30:1678–1700, 2000.

[6] Ph. Flajolet, G. Gonnet, C. Puech, and J. M. Robson. Analytic variations on quad trees. *Algorithmica*, 10:473–500, 1993.

[7] Ph. Flajolet and A. Odlyzco. Singularity analysis of generating functions. *SIAM Journal of Discrete Mathematics*, 3(1):216–240, 1990.

[8] Ph. Flajolet and C. Puech. Partial match retrieval of multidimensional data. *Journal of the ACM*, 33(2):371–407, 1986.

[9] R. L. Graham, D. E. Knuth, and O. Patashnik. *Concrete Mathematics, 2nd edition.* Addison–Wesley, 1994.

[10] P. Henrici. *Applied and Computational Complex Analysis, volume 1.* Wiley-Interscience, 1974.

[11] P. Henrici. *Applied and Computational Complex Analysis, volume 2.* Wiley-Interscience, 1977.

[12] P. Kirschenhofer and H. Prodinger. Multidimensional digital searching in alternative data structures. *Random Structures and Algorithms*, 5(1):123–134, 1994.

[13] C. Martínez, A. Panholzer, and H. Prodinger. Partial match queries in relaxed multidimensional search trees. *Algorithmica*, 29(1–2):181–204, 2001.

[14] J. S. Vitter and Ph. Flajolet. *Average-case analysis of algorithms and data structures. In J. van Leewen, editor, Handbook of Theoretical Computer Science, volume A, chapter 9.* North–Holland, 1990.