Annales Univ. Sci. Budapest., Sect. Comp. 42 (2014) 219–229

REMARKS ON THE APPROXIMATION FOR THE NUMBER OF ROOTED UNORDERED BINARY TREES

László Kovács (Miskolc, Hungary)

Dedicated to András Benczúr on the occasion of his 70th birthday

Communicated by Péter Racskó

(Received June 1, 2014; accepted July 1, 2014)

Abstract. The paper investigates the approximation formulas for the unlabeled rooted trees. In the case of ordered binary trees, the number of tree instances can be given with the Catalan numbers. It is an interesting fact, that for the case of unordered trees, only very few works can be found in the literature. The approximation formulas are usually built up with the application of appropriate generator functions. The paper presents an evaluation of a selected approximation formula from the literature.

1. Introduction

An efficient object lookup structure is the key component of information retrieval systems. The lookup operation is supported by index structures. The index structure is usually based on one dimensional object representation, like B-tree. On the other hand, there are many application areas where the objects cannot be modeled as vectors in Euclidean space and only the distances between the objects are known [3]. In these cases, a general metric space (GMS) approach [15] is used for object representation. The key element is a distance

Key words and phrases: trees, rooted unordered binary trees, weakly binary tree, Catalan numbers, approximation
2010 Mathematics Subject Classification: 05A16

matrix $H \in R^{N \times N}$, where N denotes the count of elements. The matrix element H_{ij} is equal to the distance between objects o_i and o_j. The most widely used indexing methods in general metric spaces use pivot elements. A pivot element p is a distinguished object from the object-set. The distance from an object x to p is used as the indexing key value to locate the bucket containing x. Usually more than one single pivot element are used in the algorithms. There are many variants of pivot-based index trees in general metric spaces. The Generalized Hyperplane Tree (GHT) is a widely used alternative. The corresponding structure is a binary tree where each node of the tree is assigned to a pair of pivot elements (p_1, p_2). If the distance of the object to p_1 is smaller than the distance to p_2, then the object is assigned to the left subtree, otherwise it is sent to the right subtree. According to authors, the GHT provides a better indexing structure than the usual vantage point trees [12].

An important characteristic of every partitioning structure is the balancing factor. The cost of a query operation depends on the actual balancing factor of the tree, the optimal cost is yielded in the case of perfect balancing. For one dimensional index structures, there are some efficient dynamic balancing methods, like AVL trees. In the case of GMS, no such general dynamic method exists. Thus the cost analysis of the generated index structure is an interesting question of GMS index structures. The goal of our investigation is to determine the number of different index tree structures.

In our investigation, the GHT index structure is modeled with a rooted unordered binary tree. The tree is one of the most frequently used data structure in computer science. The investigation is restricted to the rooted unordered binary trees, where each node may have maximum two children nodes. This tree structure is called weakly binary tree. The T binary tree structure can be defined on the following recursive way:

$$T = \{\Theta\} \cup \left\{ \left(\Theta^{'} \times T \times T \right) \right\}$$

where Θ denotes an external node and $\Theta^{'}$ denotes an internal root node. The size of a tree is defined as the number of its nodes. The usual tree representation form in programs is the label-based representation sequence. This form assigns an unique integer number $i \in (1..n)$ to every node where n is the size of the tree. The parent sequence shows the label of the parent for every node in the tree. For the tree given in Fig 1., the corresponding sequence is '01122334'. If each of the sibling nodes has a sequence number uniquely identifying the child, the tree is called ordered binary tree. If there is no ordering among the children the tree is called unordered tree.

Regarding the different tree manipulation algorithms, a key factor is the

computational cost. Considering the tree structure as a combinatorial object, the generation of the possible ordered or unordered trees is an intensively investigated problem domain. An important element in the cost analysis is the number of possible tree instances of a given size. The literature contains detailed analysis for each important subtype of the tree structure.

Tree enumeration was possibly first found useful by chemists in the study of structurally isomeric, aliphatic hydrocarbons [6]. The algorithm of Beyer and Hedetniemi [1] provides a constant time cost using a level sequence generation method. Later, Wright [14] extended this algorithm to generate unlabeled free trees. Pallo [8] introduced a coding method for efficient generation of binary unordered trees. He shows that the proposed method uses constant amortized time per tree. The proposal of Iwata, Ishiwata and Nakano [5] provides an efficient encoding of unordered binary trees. In contrast with the standard encoding where a tree with n nodes requires $2n$ bits, the algorithm uses only $1.4n$ bits per node in average. Liand Ruskey [10] presented an algorithm for exhaustive generation of rooted and free trees where the algorithm uses linear space and the running time is proportional to the number of trees produced. Effantin [4] focuses on the generation of unordered binary trees and the proposed algorithm needs a sub-linear $O(log(n))$ average time per tree. Li [6] also developed a similar algorithm which uses canonic rooted trees where for any tree T of size m, its parent is generated by removing the last node m, and its children are obtained by adding node $m+1$ as the rightmost child of some node on the rightmost path of T.

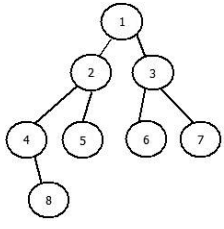

Figure 1. Sample tree

The first investigations on this area relate to Cayley [2] who created a formula for counting the number of rooted trees where the degree of the nodes are not limited. The number of unlabeled rooted trees (c_n) can be given as

$$c_n = \sum_{\sum_{i=1}^{n-1} i*j_i = n-1} \prod_{i=1}^{n-1} \binom{c_i + j_i - 1}{j_i}.$$

For efficient calculation of the number of possible tree structures, approximation formulas are used instead of the sequence definition [11]. In the case of ordered binary trees, the number of tree instances [11] is equal to

$$c_n = \frac{1}{n+1}\binom{2n}{n} = \frac{(2n)!}{n!(n+1)!} = \frac{1}{n}\binom{2n}{n-1}$$

The c_n numbers in this sequence are called the Catalan numbers. For unlabeled unrooted tree structure, the approximation function is

$$c_n = c_1 \frac{\alpha^n}{n^{5/2}},$$

where $c_1 \approx 0.5350$ and $\alpha \approx 2.9558$. The formula for unlabeled unordered rooted tree is

$$c_n = c_2 \frac{\alpha^n}{n^{3/2}},$$

where $c_2 \approx 0.4399$ and $\alpha \approx 2.9558$. For unlabeled ordered trees, the approximation formula is

$$c_n = c_3 \frac{4^n}{n^{3/2}},$$

where $c_3 \approx 0.1410$. For unlabeled rooted ordered binary tree, the corresponding approximation of the Catalan numbers is

$$c_n = 4 \cdot c_3 \frac{4^n}{n^{3/2}}.$$

In the case of labeled unrooted unordered tree structure, the approximation function is

$$c_n = n^{n-2}.$$

For labeled rooted unordered trees, the approximation formula is

$$c_n = n^{n-1}.$$

It is an interesting fact, that for the case of unordered trees, only very few works can be found in the literature [9]. The number of unordered trees can be given with the following recursive definition:

$$C_0 = C_1 = 1, \tag{1.1}$$

$$C_{2k} = C_0 C_{2k-1} + C_1 C_{2k-2} + .. + C_{k-1} C_k, \tag{1.2}$$

$$C_{2k+1} = C_0 C_{2k} + C_1 C_{2k-1} + .. + C_{k-1} C_{k+1} + C_k (C_k + 1)/2. \tag{1.3}$$

Table 1 shows the first 12 elements of B_n and C_n sequences.

n	1	2	3	4	5	6	7	8	9	10	11	12
B_n	1	2	5	14	42	132	429	1430	4832	16796	58786	208012
C_n	1	1	2	3	6	11	23	46	98	207	451	983

Table 1. The first elements of B_n and C_n sequences

2. Approximation formula for enumeration of unlabeled rooted binary trees

For the corresponding approximation formula on enumeration of unlabeled rooted binary trees, usually the work of Otter [7] is referenced. In the work of Otter, the number of trees of n vertices is given as

$$c_n = A_n^m - \frac{1}{2} \sum_{i+j=n, i>0, j>0} A_i A_j + \frac{1}{2} A_{n/2},$$

where m denotes the ramification number.

The goal of our investigation is to analyze this approximation formula for the C_n sequence. A simplified derivation for validation is presented yielding a similar result to the formula of Otter [7]. The construction of the formula is based on the following considerations.

Let us take the recursion formula of C_n. The formula can be transformed into the following expression

$$C_n = a_n + \frac{1}{2} \sum_{i+j=n-1} C_i \cdot C_j + b_n,$$

where

$$a_n = 1, \text{if } n = 0 \tag{2.1}$$

$$= 0 \text{ otherwise}, \tag{2.2}$$

$$b_n = \frac{C_{(n-1)/2}}{2}, \quad \text{if } n \text{ is odd} \tag{2.3}$$

$$= 0, \text{ otherwise}. \tag{2.4}$$

From the recursive formula, the equation for the corresponding generating function can be determined with summation for every n and generating the formal power series

$$\sum_n C_n \cdot x^n = \sum_n a_n x^n + \sum_n \frac{\sum_{i+j=n-1} C_i \cdot C_j}{2} x^n + \sum_n b_n x^n. \tag{2.5}$$

According the convolution rule of series, the equation can be given with

$$\sum_n C_n x^n = 1 + \frac{x}{2} \sum_n C_n x^n \sum_n C_n x^n + \frac{x}{2} \sum_n C_n x^{2n}$$

as only the a_0 tag is not zero. Let $f(x)$ denote the generating function of the series. The equation to be solved to get the generating function is

$$f(x) = 1 + \frac{x}{2} \left(f^2(x) + f(x^2) \right).$$

The given function-equation is very complex to solve, thus an approximation formula is used. Let us take the equation

$$f(x) = 1 + \frac{x}{2} \left(f^2(x) + 2c \right)$$

instead of the original one where c denotes a non-negative constant value. This equation for the generating function can be solved with the usual methods. The $g(x) = f(x) - 1$ substitution is introduced to apply the Lagrange-Bürmann inversion formula [13]

$$g(x) = \frac{x}{2} \left(2c + (g(x) + 1)^2 \right).$$

The Lagrange-Bürmann inversion formula can be used to solve $g(x)$ defined implicitly by

$$g(x) = x\Psi(g(x)).$$

The coefficients of an arbitrary $h(g(x))$ function meet the following formula

$$[a_n]h(g(a)) = \frac{[u_{n-1}]h(u) \cdot \Psi(u)^n}{n}.$$

In our case, the corresponding power series is

$$\Psi(u)^n = \frac{1}{2^n} \sum_{k=1}^{n} \binom{n}{k} (2c)^{n-k} (u+1)^{2k}.$$

With rearrangement of the summations, the coefficient for u_{n-1} is equal to

$$D_n = \frac{1}{2^n} \sum_{k=n*}^{n} \binom{n}{k} (2c)^{n-k} \binom{2k}{n-1},$$

where $n*$ denotes the $[(n-1)/2]$ integer value. In the next step, an approximation formula for D_n is generated. First, the asymptotic behavior of the D_n sequence was analyzed.

The goal of the investigation is to find the

$$\beta = \lim_{n\to\infty} \frac{D_{n+1}}{D_n}$$

value. After performing the substitutions, we get the key formula

$$\beta = \frac{2c}{-1+\sqrt{1+2c}}.$$

In the next steps, the value of $f(x^2)$ will be estimated for the C_n sequence. Let us denote the requested β value for the C_n sequence with β_c. The $f(x)$ function can be expressed explicitly by solving the equation

$$f(x) = \frac{1-\sqrt{1-2x(1+cx)}}{x}.$$

It can be seen that $f(x)$ is monotone increasing and

$$\lim_{x\to 0} f(x) = 1$$

and the R radius of convergence is equal to

$$R = \frac{1}{\beta}.$$

On the other hand,

$$\lim_{x\to R} f(x) = \beta$$

also holds.

Let us assume, that $g^*(x)$ is an approximation of $f(x)$. Based on the previous result, the $g_i(x)$ is an upper boundary for $f(x)$, where

$$\forall x : c_i > \frac{x^2 g^{*'}(x^2) + g^*(x^2)}{2}.$$

Let be

$$c_l = \inf\left\{\frac{x^2 g^{*'}(x^2) + g^*(x^2)}{2}\right\},$$

$$c_u = \sup\left\{\frac{x^2 g^{*'}(x^2) + g^*(x^2)}{2}\right\}.$$

It follows from the definition that both $g^*(x)$ and $g^{*'}(x)$ are monotone increasing, thus

$$\inf\left\{\frac{x^2 g^{*'}(x^2) + g^*(x^2)}{2}\right\} = \frac{g^*(0)}{2} = 0.5,$$

$$\sup\left\{\frac{x^2 g^{*'}(x^2)+g^*(x^2)}{2}\right\}=\frac{x_r^2 g^{*'}(x_r^2)+g^*(x_r^2)}{2},$$

where

$$x_r=\frac{-1+\sqrt{1+2c^*}}{2c^*}.$$

Thus every c^* can be assigned to a c_u and c_l values where $g_u(x)$ is an upper boundary and $g_l(x)$ is a lower boundary for $g^*(x)$. Let us denote these functions by $u(c)$ and $l(c)$. From he definition of $g()$ follows that

$$c_u \geq c^* \geq c_l$$

should be met. Taking the $u(x)$ and $l(x)$ functions, it can be seen that this condition holds only for

$$c_l = 0.5,$$

$$c_u = 0.631.$$

Based on these results, the following boundaries are given for the increase ratio

$$\beta_l = 2.41$$

and

$$\beta_u = 2.50.$$

In the next step, the investigated C_n value will be expressed with the help of the Catalan numbers. It is known for the B_n Catalan number that the increase ration meets

$$\beta_B=\lim_{n\to\infty}\frac{B_{n+1}}{B_n}=4.$$

With the symbol

$$\gamma=\frac{4}{\beta_c}$$

the value sequence

$$C_n=\frac{B_n}{\gamma^n}$$

meets the requested increase ratio. Using the Stirling formula, we get

$$C_n \approx \frac{(2n)^{2n}}{(n+1)^{n+1}(n-1)^{n-1}}\frac{\sqrt{\frac{n}{\pi(n-1)(n+1)}}}{n\gamma^n}.$$

For large n values, C_n can be approximated with

$$C_n \approx \frac{\beta_c^n}{\sqrt{\pi}n^{3/2}}.$$

In the next step, the β_c^n factor is evaluated with a numerical test.

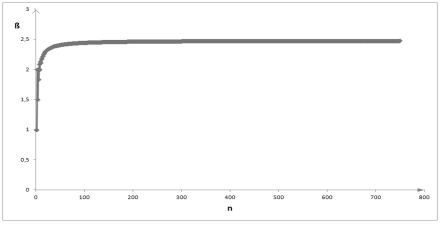

Figure 2. The β function

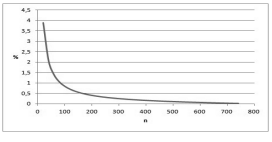

Figure 3. The error rate of the modified c_n approximation

3. Test experiments

For evaluation of the approximation formulas, some numerical test experiments were executed. The first test (Figure 3) shows the measured real β values as a function of n. As the resulted figure shows the β value is well approximated by the the calculated β_u, an upper bound value. In the next Figure 4, the relative approximation error of the modified formula is presented. As the result shows the modified formula provides a better approximation (near 0 %) of the real c_n values. As the test result shows the following modified formula

$$C_n \approx 0.7916 \frac{2.48325^n}{n^{3/2}}$$

can provide the required accuracy.

A further investigation can be focused on enumeration of tree instances having a height value k. In the literature, where the random generation of binary search trees are analyzed, the main result is that the H_n height of

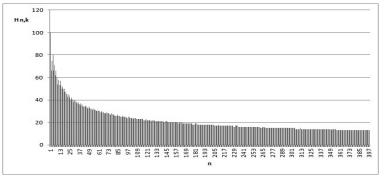

Figure 4. The $H_{n,k}/n$ ratio function

a random binary search tree on n nodes can be approximated with $\alpha lnn - \beta lnlnn + O(1)$, where $\alpha = 4.311 and \beta = 1.953$ [12]. A tree of maximum height of k, can be constructed as a union of left and right subtrees having a maximum height of $k-1$. Taking the different combinations on the size of the subtrees into account, the following formula can be derived

(3.1)
$$H_{0,k} = H_{1,k} = 1,$$

(3.2)
$$H_{2n,k} = H_{0,k-1}H_{2n-1,k-1} + H_{1,k-1}H_{2n-2,k-1} + .. + H_{n-1,k-1}H_{n,k-1},$$

(3.3)
$$H_{2n+1,k} = H_{0,k-1}H_{2n,k-1} + .. + H_{n-1,k-1}H_{n+1,k-1} + H_{n,k-1}(H_{n,k-1}+1)/2.$$

The $H_{n,k}$ symbol denotes the number of rooted unordered unlabeled binary trees having a height not greater than k and containing n vertices. If $k \to \infty$, the $H_{n,k}$ value is equal to the previously investigated C_n value. The detailed analysis of this enumeration function is an open interesting problem.

References

[1] **Beyer, T. and S.M. Hedetniemi,** Constant time generation of rooted trees, in *SIAM J. Computing*, **9** (1980), 706–712.

[2] **Cayley, A.,** *Collected Mathematical Papers,* Cambridge, 1889-1897.

[3] **Dohnal, W.,** *Indexing Structures for Searching in Metric Spaces,* PhD Thesis, Masaryk University, Brno, 2004.

[4] **Effantin, B.,** A compact encoding of unordered binary trees, *TAMC 2011,* eds. M. Ogihara and J. Tarui, LNCS **6648**, 2011, 106–113.

[5] **Iwata, K., S. Ishiwata and S. Nakano,** Generation of unordered binary trees, *ICCSA 2004,* ed. Lagana et al., LNCS **3045**, 2004, 648–655.

[6] **Li, G.,** *Generation of Rooted Trees and Free Trees,* Thesis, University of Victoria, 1996.

[7] **Otter, R.,** The number of trees, *Annals of Mathematics*, **49** (1948), 583–599.

[8] **Pallo, J.,** Lexicographic generation of binary unordered trees, *Pattern Recognition Letters*, **10** (1989), 217–221.

[9] **Rosen, K.H.,** *Handbook of Discrete and Combinatorial Mathematics,* CR Press, 2000.

[10] **Ruskey, F.,** Listing and counting subtrees of a tree, *SIAM J. Computing*, **10** (1981), 140–150.

[11] **Sedgewick, R. and P. Flajolet,** *An Introduction to the Analysis of Algorithms,* AddisonWesley, 2013.

[12] **Uhlmann, K.,** Satisfying general proximity similarity queries with metric trees, *Information Processing Letters*, **40** (1991), 175-179

[13] **Tokuda, N.,** A new application of Lagrange-Burmann expansions I. General principle, in *Zeitschrift für angewandte Mathematik und Physik ZAMP*, **34** (1983), 697–727.

[14] **Wright, R.A., B. Richmond, B., A. Odlyzko, and B.D. McKay,** Constant time generation of free trees, *SIAM J. Computing*, **15**, (1986), 540–548.

[15] **Zezula, P., G. Amato, V. Dohnal and M. Batko,** The metric space approach, *Advances in Database Systems*, Springer-Verlag, Heidelberg, 2006.

Laszlo Kovacs
University of Miskolc
Miskolc, Hungary
kovacs@iit.uni-miskolc.hu