

Aspects of Insertion in Random Trees

A. Bagchi, Calcutta, and E. M. Reingold, Urbana, Ill.

Received May 29, 1981

Abstract — Zusammenfassung

Aspects of Insertion in Random Trees. A method formulated by Yao and used by Brown has yielded bounds on the fraction of nodes with specified properties in trees built by a sequence of random internal nodes in a random tree built by binary search and insertion, and show that in such a tree about bounds better than those now known. We then apply these methods to weight-balanced trees and to a type of “weakly balanced” trees. We determine the distribution of the weight-balance factors of the internal nodes in a random tree built by binary search and insertion and show that in such a tree about 72% of all internal nodes have weight balance factors lying between $1 - \sqrt{2}/2$ and $\sqrt{2}/2$.

AMS Subject Classification: 68C25, 68E99.

Key words: Binary search trees, height-balanced trees, AVL trees, weight-balanced trees.

Aspekte bei der Einfügung in Zufallsbäume. Eine Methode, die von Yao formuliert und von Brown angewendet wurde, gestattet es, Schranken für den Anteil von Knoten mit bestimmten Eigenschaften in Bäumen anzugeben, die durch eine Folge von zufälligen Einfügungen entstehen. Für den Fall von AVL-Bäumen (höhenbalanziert) zeigen wird, daß solche Methoden nicht erweitert werden können, um bessere Schranken als die bisher bekannten zu berechnen. Dann wenden wir diese Methode auf gewichtsbalanzierte Bäume und auf eine Art von „schwach balanzierten“ Bäumen an und bestimmen die Verteilung der gewichtsbalanzierten Faktoren der inneren Knoten in einem Zufallsbaum, der durch binäre Suche und Einfügen entsteht; ferner zeigen wir, daß in einem solchen Baum ungefähr 72% der inneren Knoten gewichtsbalanzierte Faktoren zwischen $1 - \sqrt{2}/2$ und $\sqrt{2}/2$ haben.

1. Introduction

Height-balanced (AVL) trees and weight-balanced trees are useful data structures because searching, insertion, and deletion can all be accommodated in time that is logarithmic in the number of items in the tree [4], [5], [6]. This excellent worst case performance has kindled interest in their average behaviour under random sequences of insertions and deletions [3], [7], [2].

Unfortunately, little progress has been made in the analysis of their average behaviour. Virtually no progress has been made in analyzing deletion. Insertion has fared somewhat better: Based on Yao's analysis of 2–3 trees [7], Brown [3] was

able to obtain, for random height-balanced trees, upper and lower bounds on the fraction of internal nodes that have a balance factor of zero. Those bounds are not very tight, however. While Yao's analysis of insertion into 2–3 trees can, in principle, be carried out to any desired degree of precision, Brown could not similarly extend his methods to get better bounds.

The approach of Yao and Brown is to use transitions between subtrees to model the insertion process. We therefore name their method *transition analysis*. The following questions naturally arise:

- i) Is it possible to do a more elaborate analysis of insertion in random height-balanced trees?
- ii) Can transition analysis be applied to weight-balanced trees?
- iii) If the balancing rules are relaxed (perhaps even to the point of sacrificing the logarithmic height of the trees), does the analysis of the insertion/deletion process become more tractable? In such a case, what happens to the bounds?
- iv) How badly out of balance is a tree built by random search and insertion ([4], page 424)? In other words, what is the distribution of balance factors in its internal nodes?

In the various section of this paper we try to answer the questions posed above. In section 2 we show why Brown's work on height-balanced trees cannot be extended, and in section 3 we look at the corresponding problem for weight-balanced trees. We give a transition analysis for trees under a weak rebalancing rule in section 4; surprisingly, the bounds we obtain on the fraction of internal nodes with balance factor zero are better than those obtained by Brown. In section 4, the distribution of weight-balance factors of the internal nodes in a tree built by random search and insertion is computed, and it is shown, for example, that about 72% of the nodes have weight-balance factors between $1 - \sqrt{2}/2$ and $\sqrt{2}/2$. Also, we show that at least 59% of the internal nodes are roots of height-balanced subtrees. The concluding section describes some open problems.

2. Height-Balanced Trees

To study the insertion process in height-balanced trees we introduce some terminology.

Definitions:

- a) A node p in a binary tree T is a *fringe node* if at least one of the sons of p is a leaf.
- b) A binary tree T is *labeled* if all edges of T are labeled with distinct labels. A labeled tree T is a subtree of a labeled tree T' if there is a node p in T' such that the subtree of T' rooted at p is identical (including labels) to T .
- c) A binary tree T is *unlabeled* if the edges of T are unlabeled.

d) Let T be a labeled binary tree and C a set of unlabeled binary trees. Then C covers T if there exists a set D of labeled binary trees such that

- i) each tree in D is isomorphic, viewed as a binary tree, to some tree in C ,
- ii) each tree in D is a subtree of T ,
- iii) each fringe node in T occurs in exactly one tree in D .

e) Let T be an unlabeled binary tree and C a set of unlabeled binary trees. Then C covers T if there exists a labeled tree T' such that

- i) T' is isomorphic, viewed as a binary tree, to T ,
- ii) C can cover T' .

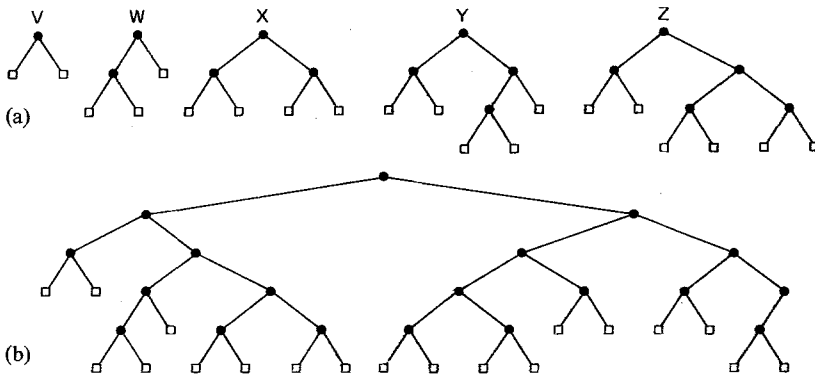


Fig. 1. (a) Five trees used as a cover, (b) the tree covered in example 1 by covers 1 and 2

Example 1: Let C consist of the five subtrees shown in Fig. 1 (a) and their symmetric variants, and let T be as in Fig. 1 (b). (In the figures, squares indicate leaves.) Then C can cover T in various ways. One way is:

Element of C	No. of copies
V	1
W	1
X	1
Y	1
Z	1

An alternative way could be:

Element of C	No. of copies
V	3
W	2
X	2
Y	0
Z	0

Definitions:

a) Let T and T' be two height-balanced trees. Then $T \rightarrow T'$ means that it is possible to get T' from T as the result of the insertion of a single node in T . (When we talk about the insertion of a node in a height-balanced tree, we mean insertion by the standard insertion algorithm, e.g., the one given in [4] page 455.)

b) A finite set C of height-balanced trees is a *closed set* if

- i) for every height-balanced tree T , C covers T , or T is a subtree of an element of C ,
- ii) for any $T \in C$, whenever $T \rightarrow T'$ then C covers T' .

Example 2: For each $i \geq 1$, let A_i consist of all height-balanced trees T such that either T has height i , or T has height $i + 1$ with one subtree of the root of height $i - 1$ and the other of height i . Then A_1 consists of the trees V and W of Fig. 1 (a) together with symmetric variants. It is easily verified, that if $T \in A_i$ and $T \rightarrow T'$, then either $T' \in A_i$ or the left and right subtrees of the root of T' are each individually in A_i . Thus A_i is a closed set. Moreover, A_i has the interesting property that given any height-balanced tree T of height at least i , A_i covers T in exactly one way.

A transition analysis of insertion in random height-balanced trees has the following features. There is a closed set C of k height-balanced subtrees U^1, U^2, \dots, U^k . For large enough n , an n -node height-balanced tree T can be covered by C . Suppose that in this cover u^i copies of U^i are used, for $1 \leq i \leq k$. Now, suppose that $T \rightarrow T'$. Then by the definition of a closed set, the $(n + 1)$ node height-balanced tree T' can also be covered by C . Let

$$\vec{u}_n = \begin{bmatrix} u^1 \\ u^2 \\ \vdots \\ u^k \end{bmatrix}$$

and let \vec{u}_{n+1} be the corresponding vector for T' . Under the assumption that T and T' are random height-balanced trees, it is possible to derive a matrix recurrence relation involving \vec{u}_n and \vec{u}_{n+1} . We find the steady state solution of the recurrence relation (i.e., a solution that holds for large n), and this solution is used to determine bounds on the fraction of internal nodes in a random height-balanced tree having a specified property.

Brown's analysis is exactly of this type. In his case, C consists of the subtrees V and W of Fig. 1 (a). Let v_n and w_n be the number of copies of V and W needed to cover an n -node random height-balanced tree. He gets the recurrence relation

$$\begin{bmatrix} v_{n+1} \\ w_{n+1} \end{bmatrix} = \left[I + \frac{1}{n+1} M \right] \begin{bmatrix} v_n \\ w_n \end{bmatrix}$$

where I is the 2×2 unit matrix and

$$M = \begin{bmatrix} -2 & 6 \\ 2 & -3 \end{bmatrix}.$$

In the limit as $n \rightarrow \infty$ the solution is

$$v_n = \frac{2(n+1)}{7}$$

$$w_n = \frac{n+1}{7}.$$

If B_n is the expected number of internal nodes with balance factor zero in a random n -node height-balanced tree, then using the above results he shows that, as $n \rightarrow \infty$,

$$\frac{10}{21}(n+1) \leq B_n \leq \frac{6}{7}(n+1) - 1. \tag{1}$$

It is clear that to get better bounds on B_n a larger closed set must be used. One possibility is to use an A_i of Example 2 for some $i > 1$. But, we show below that that cannot be done.

Lemma 1: *If C is a closed set then C contains a complete binary tree.*

Proof: Since C is finite there is a complete binary tree T that is not in C , and that is not a subtree of any element of C . T can only be covered by smaller complete binary trees, and since C must cover T , C must thus contain a complete binary tree. QED

In Fig. 1 (a), both V and X are complete binary trees. Of course, the only complete binary tree in a closed set C may be V .

Theorem 1: *If a closed set C contains a complete binary tree of height at least 2, then C cannot be used in a transition analysis of insertion in random height-balanced trees.*

Proof: Suppose C contains the subtree X of Fig. 1 (a), which is a complete binary tree of height 2. The insertion of a node into X produces Y , unless the situation depicted in Fig. 2 occurs. (In this figure, leaves are not shown.) Here a node is inserted into the X subtree rooted at r , and a double rotation at p is needed to restore balance; the resulting tree has no Y subtree. The transitions from X are thus not well-defined. Similar problems arise with complete binary trees of height greater than 2. QED

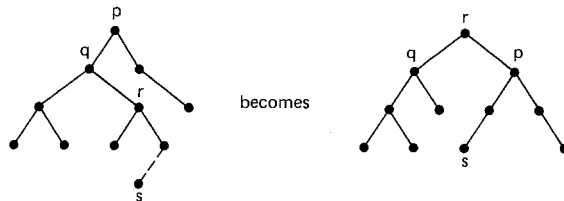


Fig. 2

Thus if a transition analysis is to be possible with a closed set C , then V must be the only complete binary tree in C . But since $V \rightarrow W$, W must also be in C , because W cannot be covered with any other subtrees. By Theorem 1, X is not in C , and since

$W \rightarrow X$, an insertion in a W tree will yield two copies of V . Using the notation defined above, we must have (see [7])

$$\vec{u}_{n+1} = \left[I + \frac{1}{n+1} M \right] \vec{u}_n$$

where I is the $k \times k$ unit matrix and M is a $k \times k$ matrix obtained from the transitions. For convenience we specify that $U^1 = V$ and $U^2 = W$. Normalizing the \vec{u}_n column vectors

$$\vec{t}_n = \frac{1}{n+1} \vec{u}_n$$

we get

$$\vec{t}_{n+1} = \left[I + \frac{1}{n+2} N \right] \vec{t}_n$$

where $N = M - I$ and N has the form

$$\begin{bmatrix} -3 & 6 & x & \dots & x \\ 2 & -4 & x & \dots & x \\ 0 & 0 & x & \dots & x \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & x & \dots & x \end{bmatrix}$$

i.e., $N_{ij} = 0$ for $3 \leq i \leq k$ and $1 \leq j \leq 2$; the x 's indicate irrelevant values. Now $C - \{V, W\}$ does not contain a complete binary tree, so it cannot be closed. Thus it must be the case that at least one of the entries in columns 3 through k in the first two rows of N is positive and nonzero. We also have

$$2t_n^1 + 3t_n^2 + \sum_{i=3}^k L_i t_n^i = 1 \quad (2)$$

where L_i is the number of leaves in U^i and t_n^i is the i -th component of \vec{t}_n .

To get the steady state solution, we have to solve (see [7])

$$N \vec{t}_n = 0 \quad (3)$$

together with (2). (This assumes that no eigenvalue of N has a positive real part.) The first two equations in (3) can be written in the form

$$\begin{aligned} -3t_n^1 + 6t_n^2 + \varepsilon_1 &= 0 \\ 2t_n^1 - 4t_n^2 + \varepsilon_2 &= 0 \end{aligned} \quad (4)$$

where ε_1 and ε_2 represent the contributions of t_n^3, \dots, t_n^k to the sums. Clearly $\varepsilon_1 \geq 0$ and $\varepsilon_2 \geq 0$, since the only negative entries in N are along the main diagonal, so from (4),

$$\varepsilon_1 = \varepsilon_2 = 0.$$

This shows that for $3 \leq i \leq k$, whenever either $N_{1,i} > 0$ or $N_{2,i} > 0$, we must have

$$t_n^i = 0. \quad (5)$$

But for any $U^i \in C$, $3 \leq i \leq k$, there is a sequence of insertions that will convert U^i to a complete binary tree. By (5), this says

$$t_n^i = 0, \quad 3 \leq i \leq k.$$

So, the steady state solution must be identical to the one found by Brown working with the closed set A_1 . Our conclusion can be restated as

Theorem 2: *Let C be a closed set with which a transition analysis of random insertions in height-balanced trees is possible. Then the bounds on B_n that the analysis yields can be no better than those given in (1).*

We clarify the proof of this theorem by constructing a directed graph G with the elements of C as its k nodes. G is similar to the state transition diagram of a Markov chain: There is a directed arc from node i to node j in G if $U^i \rightarrow U^j$, or if there is a height-balanced tree T such that $U^i \rightarrow T$ and U^j is an element of the smallest subset of C that covers T . G has the form shown in Fig. 3. Clearly, in the steady state (i.e., for large n), if there is an arc from node i to node j and $t_n^j = 0$, then $t_n^i = 0$. We showed above that $t_n^i = 0$ if there is an arc from node i to node 1 or node 2. Moreover, there is a directed path from any node i , for $i \geq 2$, to node 1. It follows that $t_n^i = 0$ for $3 \leq i \leq n$.

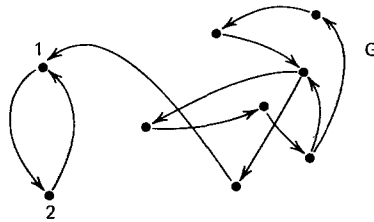


Fig. 3

It does not appear possible to do a transition analysis for random deletions in height-balanced trees: Suppose we modify the definition of a closed set appropriately, by considering transitions that result from deletions rather than from insertions. Let C be such a “closed set” for deletion, and let T be an element of C with the smallest number of internal nodes. Then the deletion of a node from T would give a tree that is not in C and that cannot be covered by C . So, a transition analysis would not be possible with C .

3. Weight-Balanced Trees

We now take a look at the insertion process in weight-balanced trees. Basic definitions can be found in [6], page 244. We are primarily interested in the class $WB[\alpha]$ where $\alpha = 1 - \sqrt{2}/2$. The notation and concepts introduced in section 2 are all meaningful in the context of weight-balanced trees, so we use them without redefining them. As far as we are concerned, the only real point of dissimilarity between height-balanced trees and weight-balanced trees is that the balancing rules

(i.e., the single and double rotations) are applied under different conditions. For weight-balanced trees, the conditions are given in [6], pages 247 – 250.

Brown's analysis remains valid when viewed as an analysis of insertion in random weight-balanced trees, but better results are obtained here with the closed set A_2 consisting of the four trees W , X , Y and Z of Fig. 1 (a). Transitions with this closed set are not well-defined when insertions are made in height-balanced trees, but such problems do not arise with weight-balanced trees, provided a certain condition is satisfied.

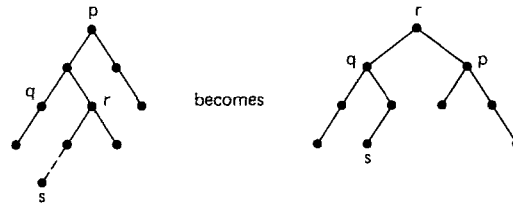


Fig. 4

Suppose a node is inserted into a weight-balanced tree; part of the tree is shown in Fig. 4, and the insertion is assumed to take place in the subtree T , which is one of W , X , Y , or Z . The weight-balanced tree may require more than one rotation, single or double, for rebalancing. As explained in [6], page 250, these rotations can be performed on the way down from the root of the tree to the leaf where the insertion will take place. However, our analysis requires that the rotations we made on the way up *after* the insertion is made, as is customary with height-balanced trees. Looking at Fig. 4 again, we find that the only situations of interest are

- i) double rotation at p ,
- ii) single rotation at q .

Other rotations do not affect the fringe of the tree. Let $L(p)$, $L(q)$ and $R(p)$, $R(q)$ indicate the numbers of leaves in the left and right subtrees of p and q prior to the insertion. For $\alpha = 1 - \sqrt{2}/2$, the following four cases arise:

Case 1, $T = W$: Here $R(q) = 3$ and $L(q) \geq 2$. In fact, we may assume $L(q) \geq 3$, for otherwise the subtree rooted at q is Y , and this case is treated separately below. Since $R(p) \geq 3$ as well, a double rotation at p or a single rotation at q can never be needed, and W just changes to X . Note that a single rotation at p , if it does take place, causes no problems as the fringe is not affected.

Case 2, $T = X$: Here again, $R(q) = 4$ and $R(p) \geq 3$. Also $L(q) \geq 3$, otherwise the subtree rooted at q is Z . A double rotation at p is required when $L(q) = 3$ and $R(p) = 3$, as illustrated in Fig. 5, where s is the newly inserted node, and leaves are not shown. In the figure the fringe still requires two W subtrees and one Y subtree, so there is no problem.

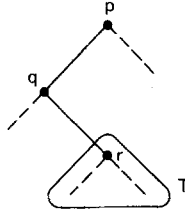


Fig. 5

Case 3, $T = Y$: In this case, $R(q) = 5$, $L(q) \geq 3$ and $R(p) \geq 4$. Double rotation is needed at p only when $L(q) = 4$ and $R(p) = 4$, as shown in Figs. 6(a) and 6(b). There are other subcases involving symmetrical variants that are not shown. The number of W , X , Y , Z subtrees in the fringe do not change as a result of the rotation.

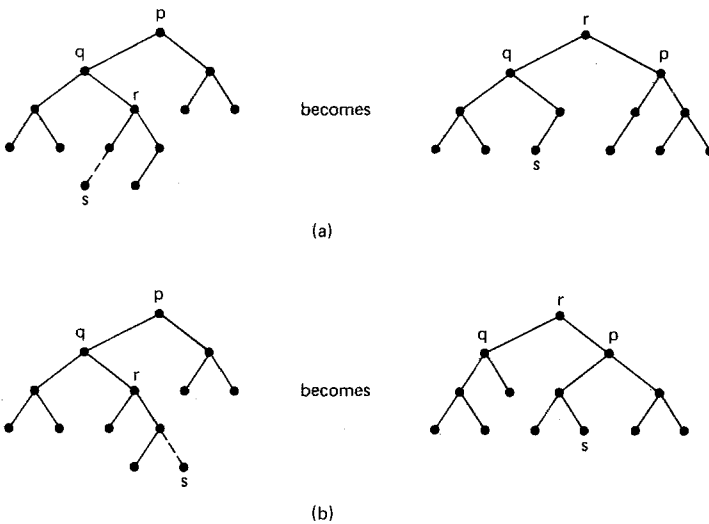


Fig. 6

Case 4, $T = Z$: Here $R(q) = 6$, $L(q) \geq 3$. Since a Z subtree is transformed into one W and one X subtree as a result of the insertion, double rotation, if needed at p can cause no difficulties, since the newly generated W and X subtrees are three levels below p .

If rotations are performed on the way down as described in [6], then Case 4 becomes a stumbling block in our analysis: When $L(q) = 3$ and $R(p) = 4$ the Z subtree can now get split by a double rotation at p , so transitions from Z are no longer well-defined.

Now, let w_n be the number of W subtrees needed to cover the fringe nodes of a random n -node weight-balanced tree, and define x_n, y_n, z_n similarly. Then

$$\begin{bmatrix} w_{n+1} \\ x_{n+1} \\ y_{n+1} \\ z_{n+1} \end{bmatrix} = \left[I + \frac{1}{n+1} M \right] \begin{bmatrix} w_n \\ x_n \\ y_n \\ z_n \end{bmatrix} \quad (6)$$

where I is the 4×4 unit matrix and

$$M = \begin{bmatrix} -3 & 0 & 4 & 6 \\ 3 & -4 & 0 & 6 \\ 0 & 4 & -5 & 0 \\ 0 & 0 & 3 & -6 \end{bmatrix}$$

Let $N = M - I$ as before. Then the eigenvalues of N are 0, -7 , and $-7.5 \pm 4.213i$. So solving (6) as in [7] for large values of n , we get

$$\begin{aligned} w_n &= \frac{23}{259}(n+1) & x_n &= \frac{3}{37}(n+1) \\ y_n &= \frac{2}{37}(n+1) & z_n &= \frac{6}{259}(n+1). \end{aligned}$$

Let $f(\alpha)$ be the fraction of internal nodes with weight-balance factor exactly α or $1-\alpha$ in an n -node random weight-balanced tree. Then for large n ,

$$\begin{aligned} 0.533 &\leq f(1/2) \leq 0.780 \\ 0.054 &\leq f(2/5) \leq 0.301 \\ 0.166 &\leq f(1/3) \leq 0.413. \end{aligned} \quad (7)$$

Moreover, when an insertion is made, a rotation, single or double, is certain to be required in

$$2w_n + 2y_n + 4z_n = 0.378(n+1)$$

cases; i.e., a rotation takes place at least once every 2.64 insertions.

Is it possible to do a similar analysis with a larger closed set? Let us first see what some typical closed sets for weight-balanced trees are like.

Example 3: Given α where $0 < \alpha \leq 1/2$ and given a positive integer k , let

$$j = \lfloor k/\alpha \rfloor - 1.$$

The set A'_k consists of all trees T in $WB[\alpha]$ such that

- i) T has m internal nodes where $k \leq m \leq j$,
- ii) at least one subtree of the root of T has less than k internal nodes.

Then A'_k is a closed set of weight-balanced trees. Note that $A'_1 = A_1$ and $A'_2 = A_2$. Brown's analysis uses A'_1 and our analysis above uses A'_2 .

An attempt at an analysis with a larger closed set encounters the same problems faced in section 2. It is clear that Lemma 1 holds for weight-balanced trees. Consider a closed set for weight-balanced trees that contains a complete binary tree of height at least 3. An example is shown in Fig. 7, where T_1 is a complete binary tree of 4 leaves, T_2 is a complete binary tree having 8 leaves, and T_3 is α -weight-balanced with 5 leaves, for $\alpha = 1 - \sqrt{2}/2$. The insertion of a node into T_2 forces a double rotation at p , and it is easy to see that transitions from T_2 are not well defined. This shows that Theorem 1 is also valid for weight-balanced trees, but only when the closed set contains a complete binary tree of height at least 3.

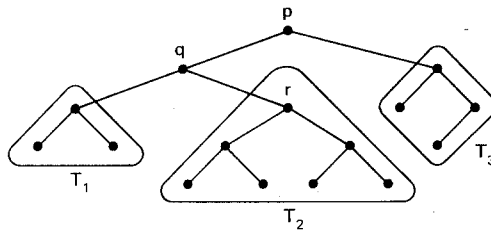


Fig. 7

An appropriate version of Theorem 2 also holds, but now there are two cases: If a closed set contains V but not X of Fig. 1(a); then Brown's results would be obtained, while if it contains X but not V then our results would be obtained. There is no advantage in putting both V and X in the closed set.

Thus, from the point of view of transition analysis, height-balanced trees and weight-balanced trees behave similarly. The only major difference is that while no analysis can be done with A_2 on height-balanced trees, A'_2 on weight-balanced trees does yield an analysis, provided we assume that balancing is done from the leaf upwards, rather than from the root downwards. It is curious and surprising that the two balancing procedures for weight-balanced trees give different distributions of subtrees at the fringe.

4. Weak Rebalancing Rules

We know from the previous sections that it is not possible to do transition analyses of insertion in height-balanced trees or weight-balanced trees using large closed sets, because transitions from subtrees are not always well-defined. Moreover, the bounds that we get from the analyses that we can do are not very tight. The balancing rules, particularly the double rotation rule, seem to be the source of the trouble, so it is of interest to investigate the consequences of weakening the balancing rules.

The most important property of balanced trees is that insertions and deletions can be done in time logarithmic in the number of nodes in the tree. If the balancing rules are weakened, this property would have to be sacrificed. In compensation, however, we may be able to get better bounds on, for example, the number of internal nodes with balance factor zero; if these bounds are superior to Brown's, we may get an indication of what better bounds for height-balanced trees would be like.

Definition:

A binary tree T is *weakly balanced* if for every fringe node p in T , the longest path from p to a leaf has length at most two.

Thus in a weakly balanced tree, the balance factor of a fringe node must be 0, -1 , or $+1$, but no restrictions are imposed on the difference in heights between the left and right subtrees of other internal nodes. Let T be a weakly balanced tree, and suppose we insert a node into T using the binary tree search and insertion procedure of [4], page 424. The resulting binary tree T' may not be weakly balanced. There may be a fringe node p in T' such that the longest path from p to a leaf has length 3, as shown in the lefthand side of Fig. 8. In this case the rotations shown in Fig. 8 would have to be performed to restore weak balance. These (and their symmetric variants) are the only permissible balancing operations.

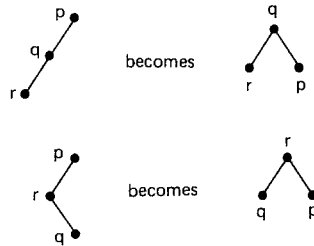


Fig. 8

Brown's work can be regarded as a transition analysis of insertion in random weakly balanced trees. Viewed in this light, his work can be extended, and an analysis is possible with the five subtrees V, W, X, Y and Z of Fig. 1 (a). With these five subtrees, covers are no longer unique, and we will make the assumption that we try to cover the fringe of a weakly balanced tree first with Z subtrees, then with Y subtrees, then with X subtrees, and so on, so that we use as many Z subtrees as possible, then as many Y subtrees as possible, etc. Let v_n be the number of V subtrees in the cover of a random n -node weakly balanced tree, and define w_n, x_n, y_n and z_n similarly. Then

$$\begin{bmatrix} v_{n+1} \\ w_{n+1} \\ x_{n+1} \\ y_{n+1} \\ z_{n+1} \end{bmatrix} = \left[I + \frac{1}{n+1} M \right] \begin{bmatrix} v_n \\ w_n \\ x_n \\ y_n \\ z_n \end{bmatrix}$$

where I is the 5×5 unit matrix and

$$M = \begin{bmatrix} -2 & 0 & 0 & 0 & 4 \\ 2 & -3 & 0 & 4 & 2 \\ 0 & 3 & -4 & 0 & 2 \\ 0 & 0 & 4 & -5 & 4 \\ 0 & 0 & 0 & 3 & -6 \end{bmatrix}$$

For large n the solution is

$$\begin{aligned} v_n &= \frac{8}{245}(n+1) & w_n &= \frac{3}{35}(n+1) \\ x_n &= \frac{3}{49}(n+1) & y_n &= \frac{2}{35}(n+1) \\ z_n &= \frac{6}{245}(n+1). \end{aligned}$$

If B'_n is the number of internal nodes with balance factor 0 in a random n -node weakly balanced tree, then

$$0.514(n+1) \leq B'_n \leq 0.776(n+1). \quad (8)$$

These bounds are better than Brown's. It would be nice if we could conclude from this that in a random height-balanced tree of n internal nodes at least $0.514(n+1)$ nodes have balance factor 0, but we have not been able to prove this. Actually, bounds even better than (8) can be derived by using larger closed sets, but there is little point in doing so until we can relate such bounds to height-balanced trees.

5. Random Binary Search Trees

A random binary tree of n internal nodes constructed by the usual search and insertion procedure ([4], page 424) can in the worst case have height n , but such cases rarely arise. In general, the tree would be fairly well balanced. We can determine how out of balance an n -node random binary search tree T is by computing the distribution of the weight balance factors of its internal nodes.

Let α be given, $0 < \alpha \leq 1/2$. Let a_n be the number of internal nodes in T whose weight balance factor is in the range $[\alpha, 1 - \alpha]$. Since T is random, all permutations of the integers 1 through n are equally likely as the order of the elements inserted in T , so that for $n \geq 1$

$$n a_n = \sum_{i=1}^n (a_{i-1} + a_{n-i}) + \lfloor (1 - \alpha)(n+1) \rfloor - \lceil \alpha(n+1) \rceil + 1. \quad (9)$$

We get this expression by observing that, if the root of T is regarded as an integer i lying between 1 and n , then the weight balance factor of i is in the required range if

$$\lfloor (1-\alpha)(n+1) \rfloor \geq i \geq \lceil \alpha(n+1) \rceil.$$

Note that $a_0 = 0$.

From (9), for $n \geq 1$

$$(n+1)a_{n+1} - (n+2)a_n = \lfloor (n+2)(1-\alpha) \rfloor - \lceil (n+2)\alpha \rceil - \lfloor (n+1)(1-\alpha) \rfloor + \lceil (n+1)\alpha \rceil. \quad (10)$$

Let RHS represent the right hand side of (10). Now suppose

$$\lceil (n+2)\alpha \rceil = \lceil (n+1)\alpha \rceil + 1.$$

Then we can write

$$(n+2)\alpha = k + 1 + \varepsilon_1$$

$$(n+1)\alpha = k + \varepsilon_2$$

for some nonnegative integer k , and $0 < \varepsilon_1, \varepsilon_2 \leq 1$. So

$$(n+2)(1-\alpha) = (n+2) - (k+1+\varepsilon_1) = (n-k+1) - \varepsilon_1$$

and

$$(n+1)(1-\alpha) = (n-k+1) - \varepsilon_2$$

i.e.,

$$\lfloor (n+2)(1-\alpha) \rfloor = \lfloor (n+1)(1-\alpha) \rfloor = n-k.$$

Thus $RHS = -1$. Similarly, when

$$\lceil (n+2)\alpha \rceil = \lceil (n+1)\alpha \rceil$$

then $RHS = +1$. Since $\alpha \leq 1/2$, for $n \geq 1$,

$$\lceil (n+2)\alpha \rceil > \lceil (n+1)\alpha \rceil \quad \text{implies} \quad \lceil (n+1)\alpha \rceil = \lceil n\alpha \rceil.$$

Moreover, when $\alpha = 1/2$,

$$\lceil (n+2)\alpha \rceil > \lceil (n+1)\alpha \rceil \quad \text{iff} \quad \lceil (n+1)\alpha \rceil = \lceil n\alpha \rceil.$$

It follows from above that for $\alpha = 1/2$, we have the recurrence relation

$$(n+1)a_{n+1} - 2a_n - (n+1)a_{n-1} = 0$$

for $n \geq 2$, with $a_0 = 0$, $a_1 = 1$. Using generating functions and putting

$$H(z) = \sum_{n=0}^{\infty} a_n z^n$$

we have

$$H' = \frac{2}{1-z} H + \frac{1}{1-z^2}$$

so that

$$H = \frac{2 \ln(1+z) - z}{(1-z)^2}.$$

This gives

$$a_n = 2 \sum_{k=0}^{n-1} \frac{n-k}{k+1} (-1)^k - n$$

which, for large n yields

$$a_n \approx (2 \ln 2 - 1)(n+1) \approx 0.386(n+1). \quad (11)$$

It is easy to show that in a random binary search tree a third of all the internal nodes have only leaves as sons. So (11) indicates that practically all the nodes with weight-balance factor exactly $1/2$ are at the bottommost levels of the tree.

To get the desired distribution of the weight-balance factors, it is enough to solve (10) for rational values of α . Let

$$\alpha = p/q, \quad 0 < \alpha \leq 1/2,$$

where p and q are positive integers. In (10), $RHS = -1$ if

$$n+1 = [k q p]$$

where k is any positive integer, and $RHS = +1$ otherwise. So

$$\sum_{n=1}^{\infty} (n+1) a_{n+1} z^{n+1} - \sum_{n=1}^{\infty} (n+2) a_n z^{n+1} = \sum_{n=1}^{\infty} z^{n+1} - 2 \sum_{k=1}^{\infty} z^{[kq/p]}.$$

Therefore

$$H' - \frac{2}{1-z} H = \frac{1}{(1-z)^2} - 2 \sum_{k=1}^p \frac{z^{jk-1}}{(1-z^q)(1-z)}$$

where

$$j_k = [k q/p] \quad \text{for } 1 \leq k \leq p.$$

For large n , we finally end up with

$$\frac{a_n}{n+1} \approx 1 - 2 \sum_{k=1}^p \sum_{i=0}^{\infty} \frac{1}{(i q + j_k)(i q + j_k + 1)}.$$

Table 1 gives the values of

$$A(\alpha) = \lim_{n \rightarrow \infty} \frac{a_n}{n+1}$$

and

$$\Delta(\alpha) = \begin{cases} A(1/2) = 2 \ln 2 - 1 & \alpha = 1/2 \\ \lim_{x \rightarrow \alpha^-} A(x) - \lim_{x \rightarrow \alpha^+} A(x) & \alpha < 1/2 \end{cases}$$

for different α . $A(\alpha)$ is the expected percentage of nodes with weight-balance factor in the range $[\alpha, 1-\alpha]$, and $\Delta(\alpha)$ is the expected percentage of nodes with weight-balance factor *exactly* α or $1-\alpha$. Using generating function techniques as above, we get

$$A(\alpha) = 4 \sum_{i=1}^{\infty} \frac{1}{(i q - 1) i q (i q + 1)}$$

where $\alpha = p/q \neq 1/2$, and p and q are relatively prime. The interesting point to observe is the very jerky manner in which $A(\alpha)$ changes with α . In fact, whenever α crosses a fraction with a small integer denominator like $1/3$ or $1/4$, $A(\alpha)$ changes by a relatively large amount. It turns out that $A(\alpha)$ is discontinuous for rational values of α and continuous for irrational values of α (see [2]). We also note that almost 72% of the internal nodes have weight-balance factors lying between $1 - \sqrt{2}/2$ and $\sqrt{2}/2$, which is a rather high percentage.

Table 1

α	$A(\alpha)$	$\Delta(\alpha)$
0 = 0.00000	1.00000	0.00000
1/60 = 0.01667	0.99910	0.00002
1/30 = 0.03333	0.99643	0.00018
1/20 = 0.05000	0.99206	0.00060
1/15 = 0.06667	0.98605	0.00143
1/12 = 0.08333	0.97845	0.00280
1/10 = 0.10000	0.96931	0.00485
7/60 = 0.11667	0.95490	0.00002
2/15 = 0.13333	0.94203	0.00143
3/20 = 0.15000	0.92389	0.00060
1/6 = 0.16667	0.91831	0.02281
11/60 = 0.18333	0.88865	0.00002
1/5 = 0.20000	0.88473	0.03984
13/60 = 0.21667	0.84026	0.00002
7/30 = 0.23333	0.82884	0.00018
1/4 = 0.25000	0.82512	0.07944
4/15 = 0.26667	0.74265	0.00143
17/60 = 0.28333	0.73435	0.00002
3/10 = 0.30000	0.71697	0.00485
19/60 = 0.31667	0.70589	0.00002
1/3 = 0.33333	0.70322	0.19722
7/20 = 0.35000	0.50353	0.00060
11/30 = 0.36667	0.49461	0.00018
23/60 = 0.38333	0.48170	0.00002
2/5 = 0.40000	0.47643	0.03984
5/12 = 0.41667	0.43315	0.00280
13/30 = 0.43333	0.41343	0.00018
9/20 = 0.45000	0.40302	0.00060
7/15 = 0.46667	0.39438	0.00143
29/60 = 0.48333	0.38814	0.00002
1/2 = 0.50000	0.38630	0.38630

The expected weight-balance factor of an internal node is around 0.366, if we consider all nodes as having weight-balance factors between 0 and $1/2$, while a lower bound on the corresponding parameter for $WB[\alpha]$ trees for $\alpha = 1 - \sqrt{2}/2$ is, from (7),

$$0.533 \times 0.5 + 0.054 \times 0.4 + 0.166 \times 0.333 + 0.247 \times 0.293 = 0.416.$$

This reaffirms the fact that random binary search trees are quite well balanced on the average.

The height-balance factor of a node in a binary tree, which is the difference in height between the right and left subtrees of the node, can take on any integral value, positive, negative or zero. It would be of great interest to determine the distribution of the height-balance factors of the internal nodes in a random binary search tree, but the problem seems very hard. It is possible, however, to get a reasonable lower bound on the fraction of nodes that are roots of height-balanced subtrees.

Let T be a random n -node binary search tree, and let b_n be the number of nodes in T such that the subtree of T rooted at the node is height-balanced. To get a lower bound on b_n , we only consider subtrees of seven or fewer nodes. In that case, for $n \geq 8$

$$b_n = \frac{1}{n} \sum_{i=1}^n (b_{i-1} + b_{n-i})$$

or

$$b_{n+1} = \frac{n+2}{n+1} b_n, \quad \text{for } n \geq 8,$$

so that

$$b_n = \frac{n+1}{9} b_8, \quad \text{for } n \geq 9,$$

and b_8 has to be computed directly.

All height-balanced trees with seven or fewer internal nodes are shown in Fig. 9 (leaves are omitted, and symmetrical variants are not shown). We have

$$b_0 = 0$$

$$b_1 = 1$$

$$b_2 = 2$$

$$b_3 = \frac{2}{3} \times 2 + \frac{1}{3} \times 3 = \frac{7}{3}$$

$$b_4 = \frac{1}{2} \times \frac{7}{3} + \frac{1}{2} \times 4 = \frac{19}{6}$$

$$b_5 = \frac{2}{5} \times \frac{19}{6} + \frac{2}{5} \times \left(\frac{2}{3} \times 3 + \frac{1}{3} \times 5 \right) + \frac{1}{5} \times 5 = \frac{56}{15}$$

$$b_6 = \frac{1}{3} \times \frac{56}{15} + \frac{1}{3} \left(1 + \frac{19}{6} \right) + \frac{1}{3} \left(\frac{2}{3} \times 4 + \frac{1}{3} \times 6 \right) = \frac{377}{90}$$

$$b_7 = \frac{2}{7} \times \frac{377}{90} + \frac{2}{7} \left(1 + \frac{56}{15} \right) + \frac{2}{7} \left(\frac{1}{2} \times 4 + \frac{1}{2} \times 7 \right) + \frac{1}{7} \left(\frac{1}{9} \times 7 + \frac{4}{9} \times 5 + \frac{4}{9} \times 4 \right) = \frac{1513}{315}$$

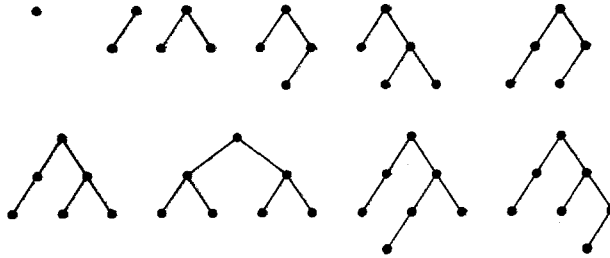


Fig. 9

Finally,

$$b_8 = \frac{2}{8} \left(1 + 2 + \frac{7}{3} + \frac{19}{6} + \frac{56}{15} + \frac{377}{90} + \frac{1513}{315} \right) = \frac{3343}{630}.$$

Thus

$$\frac{b_n}{n+1} \approx 0.590.$$

Hence at least 59% of all internal nodes are roots of height-balanced subtrees. Better lower bounds can be obtained by considering larger height-balanced subtrees, but the computation becomes laborious.

6. Conclusions

The major import of sections 2 and 3 of this paper is a negative one, namely that transition analysis as it is currently formulated is unlikely to lead to any further insights into the nature and properties of the insertion process in height-balanced and weight-balanced trees. Nor is transition analysis likely to be useful in the study of deletion. A new analytical technique is necessary.

In the course of our investigations two questions remained unanswered, and they deserve further investigation. First, weak balancing rules gave good lower bounds for B'_n in section 4, but what implication, if any, does this have on lower bounds for B_n ? Since it is easy to get even better lower bounds on B'_n by just doing a more elaborate transition analysis with a larger closed set, the discovery of a correlation between the two lower bounds would be important. The second problem concerns the distribution of height-balance factors of internal nodes in random binary search trees. Is there a "top-down" method for getting this distribution, similar to the method used for weight-balanced trees in section 5?

References

- [1] Bagchi, A., Reingold, E. M.: A naturally occurring function continuous only at irrationals. *Amer. Math. Monthly* 89 (1982).
- [2] Blum, N., Mehlhorn, K.: On the average number of rebalancing operations in weight-balanced trees. Technical Report A-78/06, Fachbereich Angewandte Mathematik und Informatik, Universität des Saarlandes, Saarbrücken, Federal Republic of Germany, June 1978.
- [3] Brown, M. R.: A partial analysis of random height-balanced trees. *SIAM J. Comp.* 8, 33—41 (1979).
- [4] Knuth, D. E.: The art of computer programming; Vol. 3: Sorting and searching. Reading, Mass.: Addison-Wesley 1972.
- [5] Nievergelt, J., Reingold, E. M.: Binary search trees of bounded balance. *SIAM J. Comp.* 2, 33—43 (1973).
- [6] Reingold, E. M., Nievergelt, J., Deo, N.: Combinatorial algorithms: theory and practice. Englewood Cliffs, N.J.: Prentice-Hall 1977.
- [7] Yao, A. C.-C.: On random 2—3 trees. *Acta Informatica* 9, 159—170 (1978).

A. Bagchi
Indian Institute of Management Calcutta
P.O. Box 16757
Calcutta-700027
India

E. M. Reingold
Department of Computer Science
University of Illinois at Urbana-Champaign
1304 W. Springfield Ave.
Urbana, IL 61801, U.S.A.