International Journal of Foundations of Computer Science Vol. 4 No. 4 (1993) 309–324

A TOP-DOWN UPDATING ALGORITHM FOR WEIGHT-BALANCED TREES

TONY W. LAI
Department of Computer Science, University of Toronto
Toronto, Ontario M5S 1A1, Canada

and

DERICK WOOD
Department of Computer Science, University of Western Ontario
London, Ontario N6A 5B7, Canada
e-mail: dwood@csd.uwo.ca

Received 1 May 1991
Revised 9 December 1993
Communicated by R. C. T. Lee

ABSTRACT

We prove that during any update in a weight-balanced tree, or BB[α] tree, a top-down restructuring pass is sufficient to rebalance the tree if $2/11 < \alpha \leq 1/4$. We also prove that if updates are known to be nonredundant, then a top-down pass is sufficient to rebalance the tree during an update if $2/11 < \alpha \leq 1 - \sqrt{2}/2$.

Keywords: Search trees, weight-balanced trees, top-down updating, top-down restructuring, redundant updates.

1. Introduction

The balanced binary search tree is a well-known data structure for representing a dictionary. Balanced trees such as AVL trees,[1] weight-balanced trees,[2,3] and red–black trees[4,5] can support the dictionary operations of **member**, **insert**, and **delete** in $O(\log n)$ worst-case time. Although AVL and red–black trees are well known and much used (particularly in courses on data structures), weight-balanced trees are less familiar. They have been discussed in a number of texts[6,7,8,9,10,11]; they have been used to solve computational geometry problems[12,13,14]; and they have been used as the basis of D-trees, dynamic trees for a nonuniform probability distribution.[15]

We consider the problem of rebalancing a weight-balanced tree during an update by use of a top-down restructuring pass. Nievergelt and Reingold[3] introduced the class of weight-balanced trees, or BB[α] trees, where $0 \leq \alpha \leq 1/2$. They proved that BB[α] trees have heights less than $1 + \log_{1/(1-\alpha)}(n+1)$ and gave insertion and deletion algorithms to perform updates in $O(\log_{1/(1-\alpha)} n)$ time if $\alpha \leq 1 - \sqrt{2}/2$. A particularly attractive feature of the insertion and deletion algorithms is that they use only one top-down restructuring pass. Unfortunately, the algorithms contain serious errors; they may fail to restore weight balance if $0 \leq \alpha \leq 2/11$. Blum and Mehlhorn,[2] however, proved that BB[α] trees can be rebalanced with an additional bottom-up pass if $2/11 < \alpha \leq 1 - \sqrt{2}/2$.

We prove that for $2/11 < \alpha \leq 1/4$, there exist top-down insertion and deletion algorithms for BB[α] trees in the sense of Nievergelt and Reingold: we perform a top-down restructuring pass while adjusting weight-balance information; if the update is redundant, we require a second top-down pass to correct the weight-balance information contained in the nodes; however, no further restructuring is necessary. The restructuring pass is not purely top-down during a deletion; to delete an internal node x with two children, we delete x's in order successor y and move y's contents into x; thus, we require a finger that points to x.

We also prove that if we can guarantee that updates are nonredundant or we can consult an oracle that determines whether an update is redundant, then, for $2/11 < \alpha \leq 1 - \sqrt{2}/2$, there exist top-down insertion and deletion algorithms for BB[α] trees.

2. Definitions and notation

Let T be a binary tree. We define $|T|$ to be the number of external nodes in T, and we define T_l and T_r to be the left and right subtrees of T, respectively. If T consists of a single external node, then the **root balance** of T is $\beta(T) = 1/2$; otherwise, we define $\beta(T) = |T_l|/|T|$. A binary tree T is of **bounded balance** α, or in the set BB[α], for $0 \leq \alpha \leq 1/2$, if

(i) $\alpha \leq \beta(T) \leq 1 - \alpha$, and
(ii) either T consists of a single external node or T_l and T_r are in BB[α].

A tree in BB[α] is said to be **weight-balanced.**

We define an insertion of a key x into a binary tree T to be **insertion redundant** if x is in T prior to the insertion; a deletion of a key x from T is **deletion redundant** if x is not in T prior to the deletion.

3. A general algorithm

3.1. The algorithm

To rebalance a BB[α]-tree, we use rotations and double rotations, as shown in Figure 1. (Symmetrical variants of the operations are not shown.) In the updating algorithm of Blum and Mehlhorn, a bottom-up restructuring pass is performed after the tree is updated; hence, restructuring is required only after a nonredundant update. Note that in the case of a deletion, we assume that only leaves are deleted, since the deletion of an internal node can always be transformed into the deletion of a leaf. Blum and Mehlhorn's algorithm works as follows. Let T be the currently inspected subtree, and let β_1 be the root balance of T. There are three cases to

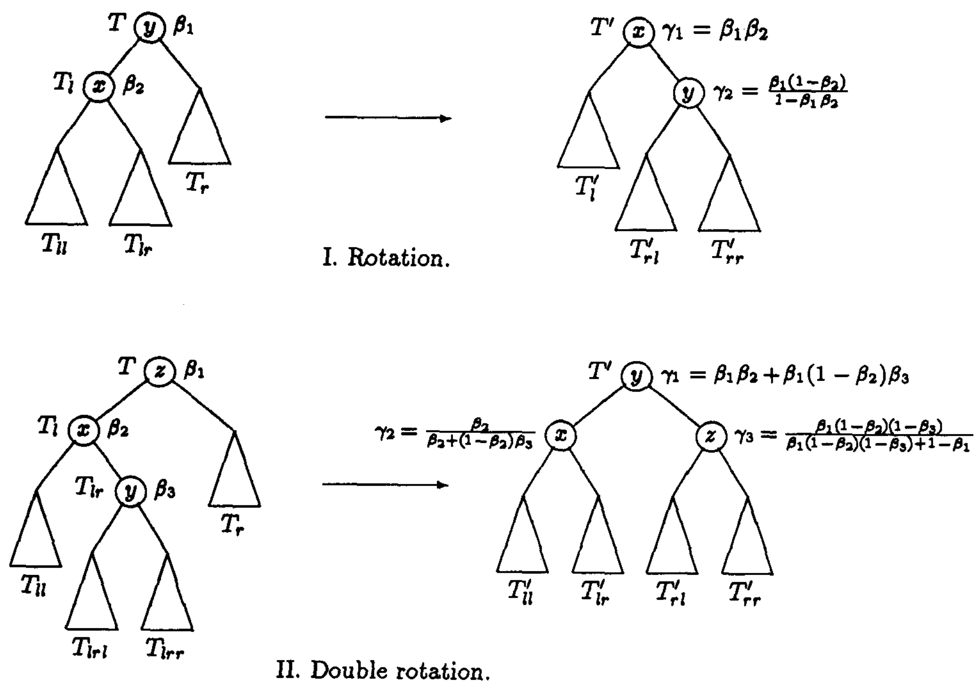

Fig. 1. The two rotations and the balance of nodes.

consider.

(i) $\beta_1 \in [\alpha, 1-\alpha]$. Do nothing.

(ii) $\beta_1 > 1-\alpha$. Let β_2 be the root balance of the left subtree T_l of T. If $\beta_2 \geq \frac{1-\alpha}{2-\alpha}$, perform a rotation, as shown in Figure 1; otherwise, perform a double rotation, as shown in Figure 1.

(iii) $\beta_1 < \alpha$. Let β_2 be the root balance of the right subtree T_r of T. If $\beta_2 \leq \frac{1}{2-\alpha}$, perform the mirror image of the rotation in Figure 1; otherwise, perform the mirror image of the double rotation in Figure 1.

The above algorithm is, unfortunately, insufficient if we want to perform a top-down restructuring pass while searching for the element. First, we cannot determine whether the update is redundant or not. Second, if the update is not redundant, then we may not know whether the subtrees of the current subtree T are weight-balanced, as before. To handle these problems, we modify the algorithm as follows. Let T be the currently inspected subtree during the top-down pass, and let β_1 or β_1' be the root balance of T if the current update is redundant or nonredundant, respectively. More formally, we define $\beta_1 = \beta(T)$ and

$$\beta_1' = \begin{cases} \frac{|T_l|+1}{|T|+1} & \text{if we perform an insertion into } T_l; \\ \frac{|T_l|-1}{|T|-1} & \text{if we perform a deletion from } T_l; \\ \frac{|T_l|}{|T|+1} & \text{if we perform an insertion into } T_r; \\ \frac{|T_l|}{|T|-1} & \text{if we perform a deletion from } T_r. \end{cases}$$

Again, there are three cases to consider.

(i) $\beta_1' \in [\alpha, 1-\alpha]$. Do nothing.

(ii) $\beta_1' > 1-\alpha$. Let β_2 or β_2' be the root balance of the left subtree T_l of T if the current update is redundant or nonredundant, respectively. If $\max(\beta_2, \beta_2') \geq \frac{1-\alpha}{2-\alpha}$, perform a rotation, as in Figure 1; otherwise, perform a double rotation, as in Figure 1.

(iii) $\beta_1' < \alpha$. Let β_2 or β_2' be the root balance of the right subtree T_r of T if the current update is redundant or nonredundant, respectively. If $\min(\beta_2, \beta_2') \leq \frac{1}{2-\alpha}$, perform the mirror image of the rotation in Figure 1; otherwise, perform the mirror image of the double rotation in Figure 1.

3.2. The analysis of the top-down updating algorithm

We now prove the following theorem. Note that the intermediate results that we need are all proved in a similar fashion. We have deliberately, however, retained the detailed proofs in all cases, except for the omission of some simple steps, because the proofs are nontrivial to discover. In addition, we believe that it is imperative that we demonstrate convincingly the correctness of our claims, rather than leaving you to demonstrate their correctness.

Theorem 1 *If* $2/11 < \alpha \leq 1/4$, *then rotations and double rotations along the search path in a BB[α] search tree suffice to rebalance the tree during the insertion or deletion of a leaf.*

We establish the theorem by way of a sequence of lemmas.

Lemma 1 (Blum and Mehlhorn[2]) *Let T be a tree in* BB$[\alpha]$. *If we insert a node into the left subtree of $T$to obtain T', then $\beta(T') \le 1/(1+\alpha)$. Furthermore, if $|T_r| \ge 2$, then $\beta(T') \le (2-\alpha)/(2+\alpha)$.*

Proof. Since T was balanced before the insertion, we know that

$$\frac{|T_r|}{|T|} = \frac{|T'_r|}{|T'|-1} \ge \alpha,$$

which implies that

$$|T'| \le \frac{|T'_r|}{\alpha} + 1.$$

Therefore,

$$\frac{|T'_r|}{|T'|} \ge \frac{|T'_r|}{\frac{|T'_r|}{\alpha}+1} = \frac{\alpha|T'_r|}{|T'_r|+\alpha}.$$

Since $|T'_r| \ge 1$, we know that $|T'_r|/|T'| \ge \alpha/(1+\alpha)$, which implies that $\beta(T') = 1 - |T'_r|/|T'| \le 1/(1+\alpha)$. If $|T_r| \ge 2$, then $|T'_r|/|T'| \ge 2\alpha/(2+\alpha)$, which implies that $\beta(T') = 1 - |T'_r|/|T'| \le (2-\alpha)/(2+\alpha)$. □

Lemma 2 *Let T be a tree in* BB$[\alpha]$. *If an insertion into the left subtree of T yields a tree T' such that $\beta(T')$ is greater than $1-\alpha$, then currently $\beta(T) > (1-2\alpha)/(1-\alpha)$. Furthermore, if $|T_r| \ge 3$, then $\beta(T) > (3-4\alpha)/(3-\alpha)$.*

Proof. If an insertion causes T' to become unbalanced, then

$$\frac{|T'_r|}{|T'|} = \frac{|T_r|}{|T|+1} < \alpha,$$

which implies that

$$|T| > \frac{|T_r|}{\alpha} - 1.$$

Therefore,

$$\frac{|T_r|}{|T|} \le \frac{|T_r|}{\frac{|T_r|}{\alpha}-1} = \frac{\alpha|T_r|}{|T_r|-\alpha}.$$

Since $|T_r| \ge 1$, we know that $|T_r|/|T| \le \alpha/(1-\alpha)$, which implies that $\beta(T) = 1 - |T_r|/|T| \ge (1-2\alpha)/(1-\alpha)$. If $|T_r| \ge 3$, then $|T_r|/|T| \ge 3\alpha/(3-\alpha)$, which implies that $\beta(T) = 1 - |T_r|/|T| \ge (3-4\alpha)/(3-\alpha)$. □

Lemma 3 *Let T be a tree in* BB$[\alpha]$. *If a deletion from the right subtree of T yields a tree T' such that $\beta(T')$ is greater than $1-\alpha$, then currently $\beta(T) > (1-\alpha)/(1+\alpha)$.*

Proof. If a deletion causes T' to become unbalanced, then

$$\frac{|T'_r|}{|T'|} = \frac{|T_r|-1}{|T|-1} < \alpha,$$

and, in a similar manner to the proof of Lemma 2, we obtain

$$\frac{|T_r|}{|T|} \le\le \frac{\alpha|T_r|}{|T_r|-1+\alpha}.$$

Since $|T_r| \geq 2$, we know that $|T_r|/|T| \leq 2\alpha/(1+\alpha)$, which implies that $\beta(T) = 1 - |T_r|/|T| \geq (1-\alpha)/(1+\alpha)$. □

We claim that if we update a tree $T \in \mathrm{BB}[\alpha]$, for $2/11 < \alpha \leq 1/4$, then the restructuring algorithm yields a tree $T' \in \mathrm{BB}[\alpha]$ such that $\beta(T') \in [\alpha, 1-\alpha]$ regardless of whether the update is redundant. By induction, this claim implies that the tree obtained after the update is in $\mathrm{BB}[\alpha]$, which implies that the updating algorithm is correct. This also implies that the cost of the algorithm is $O(\log_{1/(1-\alpha)} n)$ in the worst case, since the cost is proportional to the longest path in the tree.

If $\beta_1' \in [\alpha, 1-\alpha]$, where β_1' is the root balance of T for a nonredundant update, then the restructuring algorithm does nothing; that is, we obtain the tree $T' = T \in \mathrm{BB}[\alpha]$, which implies that the claim holds. Hence, it is sufficient to consider only the case $\beta_1' > 1-\alpha$, since the case $\beta_1' < \alpha$ is symmetric.

There are four cases to consider. During an update, we perform either an insertion or a deletion, and we perform either a rotation or a double rotation. In the following, let β_1, β_2, and β_3 be the current root balances of T, T_l, and T_{lr}, respectively, and let β_1', β_2', and β_3' be the root balances of T, T_l, and T_{lr} after the current update, respectively, if the current update is nonredundant.

Lemma 4 *Let T be a tree in* $\mathrm{BB}[\alpha]$. *Suppose we perform an insertion into T_l such that $\beta_1' > 1-\alpha$ and $\max(\beta_2, \beta_2') \geq \frac{1-\alpha}{2-\alpha}$. Then, the rotation shown in Figure 1 yields a tree T' such that*

$$\gamma_1, \gamma_2, \gamma_1' \in [\alpha, 1-\alpha],$$

where γ_1 and γ_2 are the root balances of T' and T_r' if the current update is redundant, and γ_1' is the root balance of T' if the update is nonredundant.

Proof. Observe that $\gamma_1 = \beta_1\beta_2$ is an increasing function of β_1 and β_2; $\gamma_2 = \frac{\beta_1(1-\beta_2)}{1-\beta_1\beta_2}$ is an increasing function of β_1 and a decreasing function of β_2; and $\gamma_1' = \beta_1'\beta_2'$ is an increasing function of β_1' and β_2'. Also, note that $\beta_1' > \beta_1$. We prove that $\gamma_2 \in [\alpha, 1-\alpha]$, $\min(\gamma_1, \gamma_1') \geq \alpha$, and $\max(\gamma_1, \gamma_1') \leq 1-\alpha$.

Lemma 2 implies that $\beta_1 > \frac{1-2\alpha}{1-\alpha}$. Moreover, $\beta_2 \leq 1-\alpha$. Since $\alpha \leq 1/4$, we know that

$$\begin{aligned}\gamma_2 &= \frac{\beta_1(1-\beta_2)}{1-\beta_1\beta_2} \\ &> \frac{(\frac{1-2\alpha}{1-\alpha})\alpha}{1-(\frac{1-2\alpha}{1-\alpha})(1-\alpha)} \\ &= \frac{1-2\alpha}{2(1-\alpha)} \\ &\geq \frac{1}{3};\end{aligned}$$

hence, $\gamma_2 > \alpha$.

By inspection, $\gamma_2 < \beta_1 \leq 1-\alpha$, which implies that $\gamma_2 \in [\alpha, 1-\alpha]$.

To prove that $\gamma_1, \gamma_1' \in [\alpha, 1-\alpha]$, we consider two cases: we perform either an insertion into T_{ll} or an insertion into T_{lr}.

(i) We perform an insertion into T_{ll}.

Observe that $\beta_2' > \beta_2$ and $\gamma_1' > \gamma_1$. It is sufficient to prove that $\gamma_1 \geq \alpha$ and $\gamma_1' \leq 1 - \alpha$.

To show that $\gamma_1 = |T_{ll}|/|T| \geq \alpha$, it is sufficient to show that $|T_{ll}| \geq |T_r|$, since $|T_r|/|T| \geq \alpha$. Observe that

$$\frac{|T_r|}{|T|+1} < \alpha,$$

which implies that

$$|T_r| < \alpha(|T_{ll}| + |T_{lr}| + |T_r| + 1),$$

and

$$|T_r| < \frac{\alpha}{1-\alpha}(|T_{ll}| + |T_{lr}| + 1).$$

Note that $\max(\beta_2, \beta_2') = \beta_2' = \frac{|T_{ll}|+1}{|T_l|+1} \geq \frac{1-\alpha}{2-\alpha}$. Hence,

$$\frac{|T_l|+1}{|T_{ll}|+1} = 1 + \frac{|T_{lr}|}{|T_{ll}|+1} \leq \frac{2-\alpha}{1-\alpha}$$

and

$$|T_{lr}| \leq \frac{|T_{ll}|+1}{1-\alpha}.$$

Thus,

$$|T_r| < \frac{\alpha}{1-\alpha}\left[|T_{ll}| + \frac{|T_{ll}|+1}{1-\alpha} + 1\right] = \frac{\alpha(2-\alpha)}{(1-\alpha)^2}(|T_{ll}| + 1).$$

Since $2/11 < \alpha \leq 1/4$, we know that $\frac{\alpha(2-\alpha)}{(1-\alpha)^2} < 1$, which implies that $|T_r| < |T_{ll}| + 1$, or $|T_r| \leq |T_{ll}|$. Thus $\gamma_1 \geq \alpha$.

Because T is weight-balanced, $\beta_1 = \frac{|T_l|}{|T|} \leq 1 - \alpha$, which implies that

$$\gamma_1' = \frac{|T_{ll}|+1}{|T|+1} \leq \frac{|T_l|}{|T|+1} < 1 - \alpha.$$

(ii) We perform an insertion into T_{lr}.

Observe that $\beta_2' < \beta_2$ and $\gamma_1' < \gamma_1$. We prove that $\gamma_1' \geq \alpha$ and $\gamma_1 \leq 1 - \alpha$.

To show that $\gamma_1' \geq \alpha$, it is sufficient to show that $|T_{ll}| > |T_r|$, since this inequality implies that

$$\gamma_1' = \frac{|T_{ll}|}{|T|+1} \geq \frac{|T_r|+1}{|T|+1} > \frac{|T_r|}{|T|} \geq \alpha.$$

Since $\beta_1' > 1 - \alpha$, we know that

$$\frac{|T_l|+1}{|T|+1} > 1 - \alpha,$$

which implies that

$$|T_l| + 1 > (1-\alpha)(|T_l| + |T_r| + 1)$$

and

$$|T_l| + 1 > \frac{1-\alpha}{\alpha}|T_r|.$$

Since $\alpha \leq 1/4$, it follows that $|T_l| + 1 > 3|T_r|$, which implies that $|T_l| \geq 3|T_r|$. Recall that $\max(\beta_2, \beta_2') = \beta_2 \geq \frac{1-\alpha}{2-\alpha}$. Hence,

$$\frac{|T_{ll}|}{|T_l|} \geq \frac{1-\alpha}{2-\alpha}$$

and

$$|T_{ll}| \geq \left(\frac{1-\alpha}{2-\alpha}\right)|T_l| \geq \frac{3(1-\alpha)}{2-\alpha}|T_r|.$$

Since $\alpha \leq 1/4$, we know that $|T_{ll}| \geq \frac{9}{7}|T_r| > |T_r|$; thus, $\gamma_1' \geq \alpha$.

Finally, we claim that $\gamma_1 \leq 1-\alpha$. Since $\beta_1, \beta_2 \leq 1-\alpha$, by definition, $\gamma_1 = \beta_1\beta_2 < 1-\alpha$.

□

Lemma 5 *Let T be a tree in* BB$[\alpha]$. *Suppose we perform a deletion from T_r such that $\beta_1' > 1-\alpha$ and $\beta_2 \geq \frac{1-\alpha}{2-\alpha}$. (Note that $\beta_2 = \beta_2'$.) Then, the rotation shown in Figure 1 yields a tree T' such that*

$$\gamma_1, \gamma_2, \gamma_1' \in [\alpha, 1-\alpha],$$

where γ_1 and γ_2 are the root balances of T' and T_r' if the current update is redundant, and γ_1' is the root balance of T' if the update is nonredundant.

Proof. Since T_l is not affected by the deletion, we know that $\beta_2' = \beta_2 \in [\frac{1-\alpha}{2-\alpha}, 1-\alpha]$. Because $2/11 < \alpha \leq 1-\sqrt{2}/2$, the proof of correctness of Blum and Mehlhorn's rebalancing algorithm[2] implies that $\gamma_1', \gamma_2' \in [\alpha, 1-\alpha]$, where γ_2' is the root balance of T_r' if the deletion is nonredundant. Observe that $\gamma_1' > \gamma_1$ and $\gamma_2' > \gamma_2$. Hence it is sufficient to prove that $\gamma_1 \geq \alpha$ and $\gamma_2 \geq \alpha$.

Lemma 3 implies that $\beta_1 > \frac{1-\alpha}{1+\alpha}$. Note that $\beta_2 \geq \frac{1-\alpha}{2-\alpha}$. Since $\alpha \leq 1/4$, we have $\beta_1 \geq 3/5$ and $\beta_2 \geq 3/7$, so $\gamma_1 \geq 9/35 > \alpha$.

Finally, since $\beta_1 > \frac{1-\alpha}{1+\alpha}$ and $\beta_2 \leq 1-\alpha$,

$$\begin{aligned}
\gamma_2 &= \frac{\beta_1(1-\beta_2)}{1-\beta_1\beta_2} \\
&> \frac{(\frac{1-\alpha}{1+\alpha})\alpha}{1-(\frac{1-\alpha}{1+\alpha})(1-\alpha)} \\
&= \frac{1-\alpha}{3-\alpha} \\
&\geq \frac{3/4}{11/4} \\
&> \alpha.
\end{aligned}$$

□

Lemma 6 *Let T be a tree in* BB[α]. *Suppose we perform an insertion into T_l such that $\beta_1' > 1-\alpha$ and $\max(\beta_2, \beta_2') \le \frac{1-\alpha}{2-\alpha}$. Then, the double rotation shown in Figure 1 yields a tree T' such that*

$$\gamma_1, \gamma_2, \gamma_3, \gamma_1' \in [\alpha, 1-\alpha],$$

where γ_1, γ_2, γ_3 are the root balances of T', T_l', and T_r' if the current update is redundant, and γ_1' is the root balance of T' if the update is nonredundant.

Proof. Observe that $\gamma_1 = \beta_1\beta_2 + \beta_1(1-\beta_2)\beta_3$ is an increasing function of β_1, β_2, and β_3; $\gamma_2 = \frac{\beta_2}{\beta_2+(1-\beta_2)\beta_3}$ is an increasing function of β_2 and a decreasing function of β_3; $\gamma_3 = \frac{\beta_1(1-\beta_2)(1-\beta_3)}{\beta_1(1-\beta_2)(1-\beta_3)+1-\beta_1}$ is an increasing function of β_1 and a decreasing function of β_2 and β_3; and $\gamma_1' = \beta_1'\beta_2' + \beta_1'(1-\beta_2')\beta_3'$ is an increasing function of β_1', β_2', and β_3'. Also note that $\beta_1' > \beta_1$.

We first prove that $\gamma_1, \gamma_2, \gamma_3 \in [\alpha, 1-\alpha]$. By the hypothesis, we know that $\beta_2 \in [\alpha, \frac{1-\alpha}{2-\alpha})$ and $\beta_3 \in [\alpha, 1-\alpha]$. Also, Lemma 2 implies that $\beta_1 > \frac{1-2\alpha}{1-\alpha}$. We claim that $\gamma_1 \ge \alpha$. Since $\beta_1 \ge \frac{1-2\alpha}{1-\alpha}$, $\beta_2 \ge \alpha$, $\beta_3 \ge \alpha$, and $\alpha \le 1/4$, we have

$$\begin{aligned}\gamma_1 &= \beta_1\beta_2 + \beta_1(1-\beta_2)\beta_3 \\ &\ge \left(\frac{1-2\alpha}{1-\alpha}\right)\alpha + \left(\frac{1-2\alpha}{1-\alpha}\right)(1-\alpha)\,\alpha \\ &= \left(\frac{1-2\alpha}{1-\alpha}\right)(2-\alpha)\,\alpha \\ &> \alpha.\end{aligned}$$

By inspection, $\gamma_1 < \beta_1 \le 1-\alpha$. Also, by inspection, $\gamma_2 > \beta_2 \ge \alpha$. Furthermore, since $\beta_2 < \frac{1-\alpha}{2-\alpha}$ and $\beta_3 \ge \alpha$,

$$\begin{aligned}\gamma_2 &= \frac{\beta_2}{\beta_2 + (1-\beta_2)\beta_3} \\ &< \frac{\frac{1-\alpha}{2-\alpha}}{\frac{1-\alpha}{2-\alpha} + (\frac{1}{2-\alpha})\alpha} \\ &= 1-\alpha.\end{aligned}$$

Note that if $|T_r| \le 2$, then $\gamma_3 = \frac{|T_{lrr}|}{|T_{lrr}|+|T_r|} \ge 1/3 > \alpha$; otherwise, Lemma 2 implies that $\beta_1 \ge \frac{3-4\alpha}{3-\alpha}$. Because $\beta_2 < \frac{1-\alpha}{2-\alpha}$, $\beta_3 \le 1-\alpha$, and $\alpha \le 1/4$,

$$\begin{aligned}\gamma_3 &= \frac{\beta_1(1-\beta_2)(1-\beta_3)}{\beta_1(1-\beta_2)(1-\beta_3)+1-\beta_1} \\ &> \frac{(\frac{3-4\alpha}{3-\alpha})(\frac{1}{2-\alpha})\alpha}{(\frac{3-4\alpha}{3-\alpha})(\frac{1}{2-\alpha})\alpha + \frac{3\alpha}{3-\alpha}} \\ &= \frac{3-4\alpha}{9-7\alpha} \\ &> \alpha.\end{aligned}$$

Clearly, $\gamma_3 < \beta_1 \le 1-\alpha$.

We claim that $\gamma_1' \geq \alpha$. Lemma 1 implies that $\beta_2' \geq \frac{\alpha}{1+\alpha}$ and $\beta_3' \geq \frac{\alpha}{1+\alpha}$. Since $\beta_1' > 1-\alpha$ and $\alpha \leq 1/4$,

$$\begin{aligned}
\gamma_1' &= \beta_1'\beta_2' + \beta_1'(1-\beta_2')\beta_3' \\
&> (1-\alpha)\left(\frac{\alpha}{1+\alpha}\right) + (1-\alpha)\left(\frac{1}{1+\alpha}\right)\left(\frac{\alpha}{1+\alpha}\right) \\
&= \left(\frac{1-\alpha}{1+\alpha}\right)\left(\frac{2+\alpha}{1+\alpha}\right)\alpha \\
&> \alpha.
\end{aligned}$$

Finally, observe that $|T_l'| \leq |T_l| - 1$, which implies that

$$\gamma_1' \leq \frac{|T_l'|+1}{|T|+1} < \frac{|T_l|}{|T|} \leq 1-\alpha.$$

□

Lemma 7 *Let T be a tree in* BB$[\alpha]$. *Suppose that we perform a deletion from T_r such that $\beta_1' > 1-\alpha$ and $\beta_2 < \frac{1-\alpha}{2-\alpha}$. (Note that $\beta_2 = \beta_2'$.) Then, the double rotation shown in Figure 1 yields a tree T' such that*

$$\gamma_1, \gamma_2, \gamma_3, \gamma_1' \in [\alpha, 1-\alpha],$$

where γ_1, γ_2, γ_3 are the root balances of T', T_l', and T_r' if the current update is redundant, and γ_1' is the root balance of T' if the update is nonredundant.

Proof. Since neither T_{ll} nor T_{lr} are affected by the deletion, we know that $\beta_2' = \beta_2 \in [\alpha, \frac{1-\alpha}{2-\alpha})$ and $\beta_3' = \beta_3 \in [\alpha, 1-\alpha]$. Because $2/11 < \alpha \leq 1-\sqrt{2}/2$, the proof of correctness of Blum and Mehlhorn's rebalancing algorithm implies that $\gamma_1', \gamma_2', \gamma_3' \in [\alpha, 1-\alpha]$, where γ_2' and γ_3' are the root balances of T_l' and T_r' if the current deletion is nonredundant. Observe that $\gamma_1' > \gamma_1$, $\gamma_2' = \gamma_2$, and $\gamma_3' > \gamma_3$. Therefore, it is sufficient to show that $\gamma_1 \geq \alpha$ and $\gamma_3 \geq \alpha$.

We claim that $\gamma_1 \geq \alpha$. Lemma 3 implies that $\beta_1 > \frac{1-\alpha}{1+\alpha}$. We know that $\beta_2 \geq \alpha$, $\beta_3 \geq \alpha$, and $\alpha \leq 1/4$, which implies that

$$\begin{aligned}
\gamma_1 &= \beta_1\beta_2 + \beta_1(1-\beta_2)\beta_3 \\
&\geq \left(\frac{1-\alpha}{1+\alpha}\right)\alpha + \left(\frac{1-\alpha}{1+\alpha}\right)(1-\alpha)\,\alpha \\
&\geq \left(\frac{1-\alpha}{1+\alpha}\right)(2-\alpha)\,\alpha \\
&> \alpha.
\end{aligned}$$

We claim that $\gamma_3 = \frac{|T_{lrr}|}{|T_{lrr}|+|T_r|} \geq \alpha$. Since $\alpha \leq 1/4$, it is sufficient to prove $\frac{|T_{lrr}|}{|T_{lrr}|+|T_r|} \geq \frac{1}{4}$, or $|T_r| \leq 3|T_{lrr}|$. Observe that

$$\frac{|T_r|-1}{|T|-1} < \alpha,$$

which implies that

$$|T_r| - 1 < \alpha(|T_l| + |T_r| - 1),$$

and

$$|T_r| < \frac{\alpha}{1-\alpha}|T_l| + 1.$$

We know that

$$\beta_2 = \frac{|T_{ll}|}{|T_l|} \le \frac{1-\alpha}{2-\alpha}$$

which implies that

$$\frac{|T_{lr}|}{|T_l|} \ge \frac{1}{2-\alpha},$$

and

$$|T_l| \le (2-\alpha)|T_{lr}|.$$

Also,

$$\beta_3 = \frac{|T_{lrl}|}{|T_{lrl}| + |T_{lrr}|} \le 1-\alpha,$$

which implies that

$$\begin{aligned} |T_{lr}| &= |T_{lrl}| + |T_{lrr}| \\ &\le \frac{|T_{lrr}|}{\alpha}. \end{aligned}$$

Therefore,

$$\begin{aligned} |T_r| &< \frac{\alpha}{1-\alpha}|T_l| + 1 \\ &\le \frac{\alpha}{1-\alpha}[(2-\alpha)|T_{lr}|] + 1 \\ &\le \left(\frac{\alpha}{1-\alpha}\right)\left[(2-\alpha)\frac{|T_{lrr}|}{\alpha}\right] + 1 \\ &\le \left(\frac{2-\alpha}{1-\alpha}\right)|T_{lrr}| + 1. \end{aligned}$$

Because $\alpha \le 1/4$, we know that $|T_r| < \frac{7}{3}|T_{lrr}| + 1$, which implies that $|T_r| \le \lceil \frac{7}{3}|T_{lrr}| \rceil \le 3|T_{lrr}|$. Thus $\gamma_3 \ge \alpha$. □

4. An oracular update algorithm

4.1. The algorithm

If we guarantee that updates are nonredundant, or if we consult an oracle that decides whether an update is redundant, then we can, essentially, apply Blum and Mehlhorn's rebalancing algorithm in a top-down pass, except that we use future root balances instead of current root balances.

More precisely, we apply the following rebalancing algorithm. Assume that updates are nonredundant. Let T be the currently inspected subtree during our top-down pass, and let β_1' be the root balance of T after the update. There are three possibilities.

(i) $\beta_1' \in [\alpha, 1-\alpha]$. Do nothing.

(ii) $\beta_1' > 1-\alpha$. Let β_2' be the root balance of the left subtree T_l of T after updating the tree. If $\beta_2' \geq \frac{1-\alpha}{2-\alpha}$, perform a rotation, as shown in Figure 1; otherwise, perform a double rotation, as shown in Figure 1.

(iii) $\beta_1' < \alpha$. Let β_2' be the root balance of the right subtree T_r of T after updating the tree. If $\beta_2' \leq \frac{1}{2-\alpha}$, perform the mirror image of the rotation in Figure 1; otherwise, perform the mirror image of the double rotation in Figure 1.

4.2. Analysis

We now establish the following result.

Theorem 2 *If $2/11 < \alpha \leq 1-\sqrt{2}/2$, then rotations and double rotations along the search path in a* $\mathrm{BB}[\alpha]$ *search tree suffice to rebalance the tree during the insertion or deletion of a leaf, provided that the insertion or deletion is nonredundant.*

We claim that if we perform a nonredundant update in a tree $T \in \mathrm{BB}[\alpha]$ with the above rebalancing algorithm, for $2/11 < \alpha \leq 1-\sqrt{2}/2$, then any affected node will have a root balance within the interval $[\alpha, 1-\alpha]$ after the update. By induction, this claim implies that the tree obtained after the update is in $\mathrm{BB}[\alpha]$, which implies that the algorithm is correct. This claim also implies that the cost of the algorithm is $O(\log_{1/(1-\alpha)} n)$, since the cost is proportional to the longest path in the tree.

As in the analysis of the previous algorithm, it is sufficient to consider only the case when $\beta_1' > 1-\alpha$, where β_1' is the root balance of T after the update.

In the following, let β_1, β_2, and β_3 be the root balances of T, T_l, and T_{lr} before the update, respectively, and let β_1', β_2', and β_3' be the root balances of T, T_l, and T_{lr} after the update, respectively.

Note that the correctness of Blum and Mehlhorn's rebalancing algorithm[2] implies the correctness of the above algorithm if $\beta_2' \in [\alpha, 1-\alpha]$ and $\beta_3' \in [\alpha, 1-\alpha]$. Hence, we do not need to consider the case when a deletion is performed, since a deletion from T_r does not affect any weight balances in T_l. However, Blum and Mehlhorn's proof does not imply the correctness of the above algorithm if $\beta_2' \notin [\alpha, 1-\alpha]$ or $\beta_3' \notin [\alpha, 1-\alpha]$, which may occur after an insertion. Thus, we consider two cases. During an insertion, we either perform a rotation or a double rotation.

Lemma 8 *Let T be a tree in* $\mathrm{BB}[\alpha]$. *Suppose we perform a nonredundant insertion into T_l such that $\beta_1' > 1-\alpha$ and $\beta_2' \geq \frac{1-\alpha}{2-\alpha}$. Then, the rotation shown in Figure 1 yields a tree T' such that*

$$\gamma_1', \gamma_2' \in [\alpha, 1-\alpha],$$

where γ_1' and γ_2' are the root balances of T' and T_r' after the insertion is completed.

Proof. Observe that $\gamma_1' = \beta_1'\beta_2'$ is an increasing function of β_1' and β_2'; and $\gamma_2' = \frac{\beta_1'(1-\beta_2')}{1-\beta_1'\beta_2'}$ is an increasing function of β_1' and a decreasing function of β_2'. We prove that $\gamma_1', \gamma_2' \in [\alpha, 1-\alpha]$.

Since $\beta_1' > 1-\alpha$ and $\beta_2' \geq \frac{1-\alpha}{2-\alpha}$, we know that

$$\gamma_1' = \beta_1'\beta_2' > \frac{(1-\alpha)^2}{2-\alpha}.$$

Because $\alpha \leq 1 - \sqrt{2}/2$, we have $\frac{(1-\alpha)^2}{2-\alpha} \geq \alpha$, which implies that $\gamma_1' > \alpha$.

By inspection, $\gamma_1' < \beta_1 \leq 1-\alpha$.

Lemma 1 implies that $\beta_2' \leq \frac{1}{1+\alpha}$. Moreover, $\beta_1' > 1-\alpha$. Hence,

$$\begin{aligned} \gamma_2' &= \frac{\beta_1'(1-\beta_2')}{1-\beta_1'\beta_2'} \\ &> \frac{(1-\alpha)\frac{\alpha}{1+\alpha}}{1-(1-\alpha)\frac{1}{1+\alpha}} \\ &= \frac{1-\alpha}{2} \\ &\geq \alpha. \end{aligned}$$

Finally, by inspection, $\gamma_2' \leq \beta_1 \leq 1-\alpha$. □

Lemma 9 *Let T be a tree in* BB$[\alpha]$. *Suppose we perform a nonredundant insertion into T_l such that $\beta_1' > 1-\alpha$ and $\beta_2' < \frac{1-\alpha}{2-\alpha}$. Then, the double rotation shown in Figure 1 yields a tree T' such that*

$$\gamma_1', \gamma_2', \gamma_3' \in [\alpha, 1-\alpha],$$

where γ_1', γ_2', γ_3' are the root balances of T', T_l', and T_r' after the insertion is completed.

Proof. Observe that $\gamma_1' = \beta_1'\beta_2' + \beta_1'(1-\beta_2')\beta_3'$ is an increasing function of β_1', β_2', and β_3'; $\gamma_2' = \frac{\beta_2'}{\beta_2'+(1-\beta_2')\beta_3'}$ is an increasing function of β_2' and a decreasing function of β_3'; and $\gamma_3' = \frac{\beta_1'(1-\beta_2')(1-\beta_3')}{\beta_1'(1-\beta_2')(1-\beta_3')+1-\beta_1'}$ is an increasing function of β_1' and a decreasing function of β_2' and β_3'. We prove that $\gamma_1', \gamma_2', \gamma_3' \in [\alpha, 1-\alpha]$.

To prove that $\gamma_1' \geq \alpha$, we consider two cases.

(i) $|T_{ll}| = 1$.

$\beta_2 = \frac{|T_{ll}|}{|T_l|} \geq \alpha$ implies that $|T_l| \leq \frac{1}{\alpha}$, and $\beta_1' = \frac{|T_l|+1}{|T|+1} > 1-\alpha$ implies that

$$\begin{aligned} |T|+1 &< \frac{|T_l|+1}{1-\alpha} \\ &\leq \frac{\frac{1+\alpha}{\alpha}}{1-\alpha}. \end{aligned}$$

Since $\alpha < 1/3$, we know that

$$\begin{aligned} \gamma_1' &= \frac{|T_l|+1}{|T|+1} \\ &\geq \frac{|T_{ll}|+|T_{lrl}|}{|T|+1} \end{aligned}$$

$$
\begin{aligned}
&> \frac{2}{\frac{1+\alpha}{\alpha(1-\alpha)}}\\
&= \frac{2(1-\alpha)}{1+\alpha}\cdot\alpha\\
&> \alpha.
\end{aligned}
$$

(ii) $|T_{ll}| > 1$.
If we perform an insertion into T_{ll}, then clearly $\beta_2' > \alpha$; if we perform an insertion into T_{lr}, then Lemma 1 implies that $\beta_2' \geq \frac{2\alpha}{2+\alpha}$. Also, Lemma 1 implies that $\beta_3' \geq \frac{\alpha}{1+\alpha}$, since $|T_{ll}| \geq 2$. Finally, we know that $\beta_1' > 1-\alpha$ and $\alpha \leq 1-\sqrt{2}/2$. Hence,

$$
\begin{aligned}
\gamma_1' &> (1-\alpha)\left(\frac{2\alpha}{2+\alpha}\right) + (1-\alpha)\left(\frac{2-\alpha}{2+\alpha}\right)\frac{\alpha}{1+\alpha}\\
&= \left(\frac{4-3\alpha-\alpha^2}{2+3\alpha+\alpha^2}\right)\alpha\\
&> \alpha.
\end{aligned}
$$

By inspection, $\gamma_1' < \beta_1 \leq 1-\alpha$. Also, by inspection, $\gamma_2' \geq \beta_2 \geq \alpha$. To prove that $\gamma_2' \leq 1-\alpha$, we consider two cases.

(i) $\beta_3' \geq \alpha$.
Since $\beta_2' \leq \frac{1-\alpha}{2-\alpha}$ and $\beta_3' \geq \alpha$,

$$
\begin{aligned}
\gamma_2' &= \frac{\beta_2'}{\beta_2' + (1-\beta_2')\beta_3'}\\
&\leq \frac{\frac{1-\alpha}{2-\alpha}}{\frac{1-\alpha}{2-\alpha} + \frac{1}{2-\alpha}\cdot\alpha}\\
&= 1-\alpha.
\end{aligned}
$$

(ii) $\beta_3' < \alpha$.
$\beta_3' < \alpha$ is possible only if an insertion is performed in T_{lrr}. To prove that $\gamma_2' = \frac{|T_{ll}|}{|T_{ll}|+|T_{lrl}|} \leq 1-\alpha$, it is sufficient to show $|T_{ll}| \leq |T_{lrr}|$, since $1-\beta_3 = \frac{|T_{lrr}|}{|T_{lrl}|+|T_{lrr}|} \leq 1-\alpha$. Note that $|T_{lr}|+1 < \frac{|T_{lrr}|+1}{1-\alpha}$, since $1-\beta_3' = \frac{|T_{lrr}|+1}{|T_{lr}|+1} > 1-\alpha$. Since

$$
\begin{aligned}
\beta_2' &= \frac{|T_{ll}|}{|T_{ll}|+|T_{lr}|+1}\\
&< \frac{1-\alpha}{2-\alpha},
\end{aligned}
$$

we know that

$$
\begin{aligned}
|T_{ll}| &< (1-\alpha)(|T_{lr}|+1)\\
&< (1-\alpha)\left(\frac{|T_{lrr}|+1}{1-\alpha}\right)\\
&= |T_{lrr}|+1.
\end{aligned}
$$

Thus, $|T_{ll}| \leq |T_{lrr}|$.

To prove that $\gamma_3' \geq \alpha$, we consider two cases.

(i) $\beta_3' \leq 1-\alpha$.
Since $\beta_1' > 1-\alpha$, $\beta_2' \leq \frac{1-\alpha}{2-\alpha}$, $\beta_3' \leq 1-\alpha$, and $\alpha \leq 1-\sqrt{2}/2$,

$$\begin{aligned}\gamma_3' &= \frac{\beta_1'(1-\beta_2')(1-\beta_3')}{\beta_1'(1-\beta_2')(1-\beta_3')+1-\beta_1'}\\ &\geq \frac{(1-\alpha)\frac{1}{2-\alpha}\cdot\alpha}{(1-\alpha)\frac{1}{2-\alpha}\cdot\alpha+\alpha}\\ &= \frac{1-\alpha}{3-2\alpha}\\ &\geq \alpha.\end{aligned}$$

(ii) $\beta_3' > 1-\alpha$.
$\beta_3' > 1-\alpha$ is possible only if an insertion is performed in T_{lrl}. To prove that $\gamma_3' = \frac{|T_{lrr}|}{|T_{lrr}|+|T_r|} \geq \alpha$, it is sufficient to show that $|T_r| \leq |T_{lrl}|$, since $1-\beta_3 = \frac{|T_{lrr}|}{|T_{lrl}|+|T_{lrr}|} \geq \alpha$. Note that $|T_{lr}|+1 < \frac{|T_{lrl}|+1}{1-\alpha}$, since $\beta_3' = \frac{|T_{lrl}|+1}{|T_{lr}|+1} > 1-\alpha$. To show that $|T_r| \leq |T_{lrl}|$, we first show that $|T_{ll}| \leq |T_{lrl}|$. Since

$$\begin{aligned}\beta_2' &= \frac{|T_{ll}|}{|T_{ll}|+|T_{lr}|+1}\\ &< \frac{1-\alpha}{2-\alpha},\end{aligned}$$

we know that

$$\begin{aligned}|T_{ll}| &< (1-\alpha)(|T_{lr}|+1)\\ &< (1-\alpha)\left(\frac{|T_{lrl}|+1}{1-\alpha}\right)\\ &= |T_{lrl}|+1.\end{aligned}$$

Thus, $|T_{ll}| \leq |T_{lrl}|$. We now show that $|T_r| \leq |T_{lrl}|$. Since

$$\begin{aligned}1-\beta_1' &= \frac{|T_r|}{|T_l|+|T_r|+1}\\ &< \alpha,\end{aligned}$$

we know that

$$\begin{aligned}|T_r| &< \frac{\alpha}{1-\alpha}(|T_l|+1)\\ &= \frac{\alpha}{1-\alpha}(|T_{ll}|+|T_{lr}|+1)\\ &< \frac{\alpha}{1-\alpha}\left(|T_{lrl}|+\frac{|T_{lrl}|+1}{1-\alpha}\right)\\ &= \left(\frac{\alpha}{1-\alpha}\right)\left(\frac{2-\alpha}{1-\alpha}\right)|T_{lrl}|+\frac{\alpha}{(1-\alpha)^2}.\end{aligned}$$

Because $\alpha \leq 1-\frac{\sqrt{2}}{2}$, we have $\frac{\alpha(2-\alpha)}{(1-\alpha)^2} \leq 1$, which implies that $|T_r| < |T_{lrl}|+1$, or $|T_r| \leq |T_{lrl}|$.

Finally, by inspection, $\gamma_3' < \beta_1 \leq 1 - \alpha$. □

Acknowledgements

The work of the first author was supported by an NSERC Postgraduate Scholarship and that of the second by grants from the Natural Sciences and Engineering Research Council of Canada and the Information Technology Research Centre of Ontario. The work was started when both authors were at the University of Waterloo.

References

1. G. M. Adel'son-Vel'skii and E. M. Landis, "An algorithm for the organization of information," *Sov. Math. Dokl.* **3** (1962) 1259–1262.
2. N. Blum and K. Mehlhorn, "On the average number of rebalancing operations in weight-balanced trees," *Theoretical Computer Science* **11** (1980) 303–320.
3. J. Nievergelt and E. M. Reingold, "Binary search trees of bounded balance," *SIAM Journal on Computing* **2** (1973) 33–43.
4. R. Bayer, "Symmetric binary B-trees: Data structure and maintenance algorithms," *Acta Informatica* 1 (1972) 290–306.
5. L. J. Guibas and R. Sedgewick, "A dichromatic framework for balanced trees," in *Proceedings of the 19th Annual IEEE Symposium on Foundations of Computer Science*, 1978, pp. 8–21.
6. T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms* (McGraw-Hill, New York, NY, 1990).
7. D.E. Knuth, *Sorting and Searching*, volume 3 of *The Art of Computer Programming*, (Addison-Wesley, Reading, MA, 1973).
8. K. Mehlhorn, *Sorting and Searching*, volume 1 of *Data Structures and Algorithms*, (Springer-Verlag, New York, NY, 1984).
9. E. M. Reingold and W. J. Hansen, *Data Structures*, (Little, Brown & Company, Boston, MA, 1983).
10. E.M. Reingold, J. Nievergelt, and N. Deo, *Combinatorial Algorithms: Theory and Practice*, (Prentice-Hall, Inc., Englewood Cliffs, NJ, 1977).
11. H.F. Smith, *Data Structures: Form and Function*, (Harcourt Brace Jovanovich, Publishers, New York, NY, 1987).
12. H. Edelsbrunner, A new approach to rectangle intersections, Part I, *International Journal of Computer Mathematics* **13** (1983) 209–219.
13. H. Edelsbrunner, A new approach to rectangle intersections, Part II, *International Journal of Computer Mathematics* **13** (1983) 221–229.
14. K. Mehlhorn, Dynamic binary search, *SIAM Journal on Computing* **8** (1979) 175–198.
15. K. Mehlhorn, Arbitrary weight changes in dynamic trees, *RAIRO Informatique théorique* **15** (1981) 183–211.