The Fringe Analysis of Search Trees

Nivio Ziviani †

Department of Computer Science
University of Waterloo
Waterloo, Ontario
Canada

Research Report CS-82-15
May 1982

† Permanent address:

Departamento de Ciência da Computação
Universidade Federal de Minas Gerais
30000 Belo Horizonte MG
Brazil

The Fringe Analysis of Search Trees

Abstract

This thesis deals with the analysis of various search trees. These analyses are based on a method that considers only the bottom part of a tree structure and so it is known as fringe analysis. By considering only part of the nodes of a tree one is able to obtain bounds on most complexity measures considered and also some exact results.

We present a fringe analysis method based on a new way of describing the composition of a fringe in terms of tree collections. As a consequence, we obtain sharp bounds on the expected number of splits per insertion and on the expected depth of the deepest safe node in 2-3 trees and B-trees, and obtain improvements of the bounds on the expected number of nodes in 2-3 trees. (We also give bounds for 2-3 trees and B-trees using an overflow technique.)

We present a new way of handling tree collections that are not closed. An inherent difficulty posed by the transformations necessary to keep the AVL tree balanced makes its analysis difficult when using fringe analysis methods. We derive a technique to cope with this difficulty and obtain bounds on the expected number of rotations per insertion and on the expected number of balanced nodes.

We present an analysis of another balanced binary search tree, the symmetric binary B-tree, and obtain bounds on the expected number of split transformations per insertion and on the expected number of balanced nodes. Finally, an unexpected application of the fringe analysis method: we obtain a lower bound on the expected height of binary search trees constructed with no balance constraints.

Acknowledgements

I am deeply grateful to my supervisor Gaston H. Gonnet for his strong support and friendship. He suggested the thesis topic, spent a great deal of time with me, and read all my work. During our fruitful work together I was always able to proceed at my own pace. In dealing with a new research problem he stressed the importance of considering all possible alternatives with the same respect and intensity.

I am grateful to Frank Tompa for his advice and support during the first half of my studies, and to Ian Munro for introducing me to the areas of analysis of algorithms and computational complexity. Many people contributed to this thesis, especially Howard Johnson, Jack Kalbfleisch, Kurt Mehlhorn, Ian Munro, Thomas Ottmann, Frank Tompa, and Derick Wood. I wish to thank Patricio Poblete for his helpful suggestions and for reading the very first version of this thesis.

I am grateful to those persons who have supported me from a distance, especially Glauco Freitas, my sister Ivone, and my parents Célia and Nelson, who have encouraged all of their children to pursue excellence without putting pressure on them to succeed. I would like to thank the support that has been granted to me by the Brazilian Coordenação do Aperfeiçoamento de Pessoal de Nível Superior and by the Universidade Federal de Minas Gerais (Brazil).

I am grateful to the people with whom I have played tennis at the Waterloo Tennis Club, especially the tennis pro John Greenan, and my partners Bill, Clovis, Feofiloff, João, Kam, Maria, Ricardo, and Rolf. They did not contribute directly to this thesis, but they have surely contributed to improving my game.

I am most of all indebted to my wife Maria for her continuing support and encouragement, without which my graduate studies could not have been accomplished. Many thanks to Maria and our daughters Patricia and Paula who made our stay in Canada something special and pleasant.

Table of Contents

1. INTRODUCTION

1.1. Motivation

Balanced search trees (e.g. B-trees, 2-3 trees, 1-2 brother trees, symmetric binary B-trees, AVL trees, weight-balanced trees, etc) are efficient ways of storing information. They are particularly adequate when one needs to consider all or some combination of requirements, such as: (i) efficient access in random and sequential processing; (ii) ease of insertion and deletion of records; (iii) high storage utilisation; (iv) use in one-level and two-level storage.

These structures have been around for quite a long time (e.g. AVL trees appeared in 1962, B-trees in 1972), and their worst case behaviours are well-known (Knuth, 1973). However, no analytical results were known about the expected case behaviour of balanced search trees prior to the pioneering work of Yao (1978) on 2-3 trees and B-trees. Yao (1978) presented a technique of analysis now known as fringe analysis, which he used to find bounds on the expected number of nodes in a B-tree.

In this thesis we present a fringe analysis method based on a new way of describing the composition of a fringe in terms of tree collections, and new ways of handling tree collections. This enables us to answer a number of open problems and improve upon previous results on most of the balanced search trees mentioned above.

1.2. Related Work

The first valuable attempt to analyse a balanced search tree was performed by Yao (1978). In his work Yao presented a method which he used to obtain a partial analysis of the expected number of nodes in 2-3 trees and B-trees and the asymptotic storage utilisation in B-trees. The analysis is partial because the method considers only the bottom nodes of a tree (e.g., the first two lowest levels of a 2-3 tree in Yao's work), which allows one to obtain only bounds on most complexity measures considered. Yao's results on 2-3 trees were slightly extended by Brown (1979b). More recently and independently of our work on B-trees, Eisenbarth and Mehlhorn (1980) obtained the asymptotic storage utilisation of B-trees using an insertion technique called overflow technique.

The method introduced by Yao (1978) was later used by Brown (1979a) to obtain a partial analysis of AVL trees. In his analysis Brown considered the set of AVL subtrees with three or fewer leaves and called it the fringe of the AVL tree.

By analysing the fringe of large AVL trees Brown was able to derive bounds on the expected number of balanced nodes in the whole tree.

An improvement on Brown's results for AVL trees was obtained by Mehlhorn (1979a), through the study of 1-2 brother trees (Ottmann and Six, 1976). The main technical contribution of Mehlhorn's paper is a method for analysing the behaviour of 1-2 brother tree schemes where the rebalancing operations require knowledge about the brother of a node. Using the close relationship between 1-2 brother trees and AVL trees (Ottmann and Wood, 1979), Mehlhorn was able to improve the bounds on the expected number of balanced nodes in AVL trees.

Ottmann and Stucky (1980) presented a higher order analysis of the insertion algorithm for 1-2 brother trees by modifying it in such a way that Yao's technique becomes applicable. Mehlhorn (1979b) presented a fringe analysis of AVL trees under random insertions and deletions.

1.3. Contributions and Outline of the Thesis

The main contributions are:

(i) the way of describing the composition of a fringe in terms of tree collections and the resulting derivation of a fringe analysis method (Chapter 2);

(ii) the bounds on the expected number of splits per insertion and on the expected depth of the deepest safe node on an insertion path in 2-3 trees; the improvement of the best known bounds on the expected number of nodes in 2-3 trees; the bounds on the same complexity measures just mentioned for 2-3 trees, but using a particular insertion overflow technique (Chapter 3);

(iii) the bounds on the expected number of splits per insertion and on the expected depth of the deepest safe node on a particular insertion path in B-trees; the bounds on the expected number of internal nodes, on the expected number of splits per insertion, and on the expected depth of the deepest safe node on an insertion path in B-trees using an insertion overflow technique (Chapter 4);

(iv) the presentation of a closed AVL tree collection containing three types (Chapter 5);

(v) the new way of handling weakly-closed collections of AVL trees (Chapter 5);

(vi) the bounds on the expected number of rotations per insertion on AVL trees; the improvement of the bounds on the expected number of balanced nodes in AVL trees (Chapter 5);

(vii) the bounds on the expected number of splits per insertion and on the expected number of balanced nodes in symmetric binary B-trees (Chapter 6);

(viii) the lower bound on the expected height of binary search trees (Chapter 7).

Part of the results presented in this thesis appeared in Gonnet, Ziviani and Wood (1981).

Besides the seven central chapters, Chapter 8 contain the conclusions, and Appendices A, B, C, and D contains the basic definitions of AVL trees, 2-3 trees, B-trees, and symmetric binary B-trees, respectively.

In this thesis we do not consider the effect of deletions. The reason for this is because deletions do not preserve randomness, and it is not clear how to incorporate them in the fringe analysis problem. It is true that Mehlhorn (1979b) presented an analysis of AVL trees under random insertions and deletions, but there is no explicit reference in his paper to the problem mentioned above.

1.4. Notation and Basic Definitions

Let us start with the concept of a binary tree. We adopt the definition presented by Knuth (1968, p. 315, exercise 20). A *binary tree* consists of a single node, called its root, plus 0 or 2 disjoint binary trees. In other words, a binary tree is a set of nodes such that each node has exactly zero or two sons; when a node has two sons, they are called the left and right sons of the node.

The number of subtrees of a node is called the *degree* of that node. A node of degree zero is called an *external node* or *leaf*. (In this thesis we make no distinction between the two terms.) The other nodes are referred to as *internal nodes*. The *level* of a node is defined by saying that the root is at level 1; if a node is at level l then the roots of its subtrees are at level $l + 1$.

A *binary search tree* T_N of $N \geq 0$ internal nodes over an ordered set S of keys $k_1 < k_2 < \cdots < k_N$ is a binary tree such that each node is labelled with a unique key in S, and for each node v the following property holds: all the keys in the left subtree of v lexicographically precede the key that labels v, and all the keys in the right subtree of v lexicographically follow the key that labels v.

In a binary search tree the internal nodes are identified with their associated keys. Each external node or leaf can be identified with the keys of an interval of S such that for a tree containing keys $k_1,k_2,\ldots,k_N$, we define k_0 and k_{N+1} to be $k_0 < k < k_{N+1}$, for all $k \in S$. Thus the leaves can be named (k_j, k_{j+1}).

$0 \leq j \leq N$. Figure 1.3.1 shows one possible binary search tree of 3 nodes. In this thesis the internal nodes are circle-shaped and the external nodes are square-shaped.

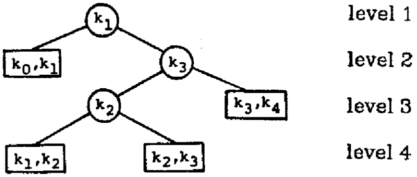


Fig. 1.3.1 Binary search tree with 3 nodes

Searching for a key k in a binary search tree T is described as follows. If T is a leaf, the search is unsuccessful. If k is equal to the key that labels the root node of T then the search has located k. If k is less than the key that labels the root node of T then search for k in the left subtree. If k is greater than the key that labels the root node of T then search for k in the right subtree.

These concepts have generalisations to ternary, quaternary, etc. trees. We define a ***t*-ary tree** as a set of nodes that is either empty or consists of a root and t ordered, disjoint t-ary trees. Like binary search trees, t-ary trees can be associated with an ordered set of keys to form ***t*-ary search trees**. Figure 1.3.2 shows a complete ternary search tree with four internal nodes and eight keys associated with the nodes.

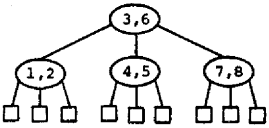


Fig. 1.3.2

In this thesis we assume that all trees are ***random trees***. Consider a t-ary search tree T with N keys and consequently $N + 1$ external nodes. As we have shown these N keys divide all possible key values into $N + 1$ intervals. An insertion into T is said to be a ***random insertion*** if it has an equal probability of being in any of the $N + 1$ intervals defined above. A ***random t-ary search tree*** with N keys is a t-ary search tree constructed by making N successive random insertions into an initially empty tree.

We analyse many different insertion algorithms for search trees. The formal definitions of each tree structure and their related insertion algorithms are presented in Appendices A, B, C and D. The complexity measures considered in each analysis will be defined as we need them.

The notation used throughout this thesis is standard. For well-known constants and functions we adopt the notation used by Abramowitz and Stegun (1972). The probability of a given event is denoted by Pr{event}.

The asymptotic notation is used as follows:

$g(n) = O(f(n))$ => there exists c and n_0 such that

$$|g(n)| < c\,|f(n)| \text{ for } n > n_0$$

$$g(n) = o(f(n)) \Rightarrow \lim_{n \to \infty} \frac{g(n)}{f(n)} = 0$$

$g(n) = \Theta(f(n))$ => there exists c_1, c_2 $(c_1 \times c_2 > 0)$ and n_0 such that

$$c_1 f(n) < g(n) < c_2 f(n) \text{ for } n > n_0$$

2. FRINGE ANALYSIS TECHNIQUE

In the first part of this chapter we introduce the concepts and the definitions necessary to describe the Markov chain used to model the insertion process in search trees. In the second part we study the matrix recurrence relation involved in the Markov process. In the third part we introduce some definitions necessary to describe the fringe of different classes of search trees.

2.1. The Markov Process

Let us define a *tree collection* C as a finite collection of trees. Consider the class of 2-3 trees of bounded height as an example.† The collection of 2-3 trees of height k ($k>0$) forms a different tree collection for each value of k. Figure 2.1.1 displays the two possible types of trees in a 2-3 tree collection of height 1. The dots represent the number of keys in each node.

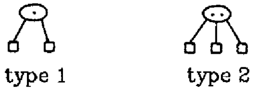


Fig. 2.1.1 Tree collection of 2-3 trees of height 1

The *fringe* of a tree consists of one or more subtrees that are isomorphic to members of a tree collection C. Typically, the fringe will contain all subtrees that meet this definition; for example the fringe of a 2-3 tree is obtained by deleting all nodes at a distance greater than k ($k>0$) from the leaves. In Chapter 7 we will restrict the fringe to contain one subtree only. Figure 2.1.2 shows an instance of a 2-3 tree with eleven keys in which the fringe that corresponds to the tree collection of 2-3 trees of height 1 is encircled.

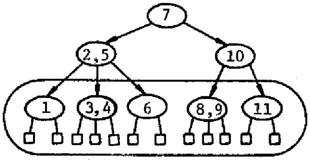


Fig. 2.1.2 A 2-3 tree and its fringe of height 1 subtrees

† See Appendix B for the definition of 2-3 trees.

The composition of the fringe can be described in several ways. One possible way is to consider the probability that a randomly chosen leaf of the tree belongs to each of the members of the corresponding tree collection. In other words, the probability p is

$$p_i(N) = \frac{\textit{Expected number of leaves of type } i \textit{ in an } N\textit{-key tree}}{N+1} \qquad (1)$$

Yao (1978) describes the fringe in a different way. His description of the composition of the fringe considers the expected number of trees of type i, while we describe it in terms of leaves as in Eq.(1). As we shall see our description of the composition of the fringe simplifies the notation necessary to present the fringe analysis technique, and also makes easier the task of finding which complexity measures can be obtained from the analysis of each search tree.

The transitions between trees of a tree collection can be used to model the insertion process. In an insertion of a key into the type 1 tree shown in Figure 2.1.1 two leaves of type 1 are lost and three leaves of type 2 are obtained. In an insertion of a key into the type 2 tree three leaves of the type 2 are lost and four leaves of the type 1 tree are obtained as a result of node splitting.

Clearly the probability that an insertion in one type of a tree collection C leads to another type of C depends only on the two types involved, and so the process is a Markov process (cf. Cox and Miller, 1965; Feller, 1968). A sequence $\{X_N\} = \{X_0, X_1, \cdots\}$ of random variables taking values on a state space S is a Markov chain if

$$Pr\{X_N = i \mid X_{N-1} = j,\, X_{N-2}=j_1, \cdots, X_0=j_{N-1}\} = Pr\{X_N = i \mid X_{N-1} = j\}$$

for all $i, j, j_1, \cdots, j_{N-1} \in S$. The current value of X_N depends on the history of the process only through the most recent value X_{N-1}.

To illustrate this fact consider the tree collection of 2-3 trees of height 1 shown in Figure 2.1.1. In this context, let X_N and Y_N be respectively the numbers of type 1 and type 2 leaves after the N^{th} insertion. Since the tree collection is closed, the value of X_N depends only on the value of X_{N-1} and as a consequence $\{X_N\}$ (or equivalently $\{Y_N\}$) is a Markov chain.

The transition probabilities of the chain $\{X_N\}$ are given by

$$Pr\{X_N = i \mid X_{N-1} = j\} = \begin{cases} \dfrac{j}{N} & i=j-2 \\[2ex] \dfrac{N-j}{N} & i=j+4 \end{cases}$$

while those of Y_N are

$$Pr\{Y_N = i \mid Y_{N-1} = j\} = \begin{cases} \dfrac{j}{N} & i=j-3 \\ \dfrac{N-j}{N} & i=j+3 \end{cases}$$

Let $j_N = E(X_N)$ and $k_N = E(Y_N)$. Then

$$\begin{aligned} j_N = E(X_N) &= E[E(X_N \mid X_{N-1}, Y_{N-1})] \\ &= E\left[\frac{X_{N-1}}{N}(X_{N-1}-2)+\frac{Y_{N-1}}{N}(X_{N-1}+4)\right] \\ &= j_{N-1}-\frac{2}{N}j_{N-1}+\frac{4}{N}k_{N-1} \end{aligned}$$

and similarly

$$k_N = k_{N-1}-\frac{3}{N}k_{N-1}+\frac{3}{N}j_{N-1} \quad .$$

But, by definition

$$j_{N-1} = Np_1(N-1); \quad j_N = (N+1)p_1(N);$$
$$k_{N-1} = Np_2(N-1). \quad k_N = (N+1)p_2(N).$$

Substituting these equations into the previous equations we get

$$p_1(N) = \frac{(N-2)p_1(N-1)+4p_2(N-1)}{N+1}$$

and

$$p_2(N) = \frac{3p_1(N-1)+(N-3)p_2(N-1)}{N+1}$$

In matrix notation

$$\begin{bmatrix} p_1(N) \\ p_2(N) \end{bmatrix} = \begin{bmatrix} \dfrac{N-2}{N+1} & \dfrac{4}{N+1} \\ \dfrac{3}{N+1} & \dfrac{N-3}{N+1} \end{bmatrix} \begin{bmatrix} p_1(N-1) \\ p_2(N-1) \end{bmatrix}$$

or

$$\begin{bmatrix} p_1(N) \\ p_2(N) \end{bmatrix} = \left[I + \frac{H}{N+1}\right] \begin{bmatrix} p_1(N-1) \\ p_2(N-1) \end{bmatrix}$$

where $H = \begin{bmatrix} -3 & 4 \\ 3 & -4 \end{bmatrix}$ and $I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$.

Thus the probability of an insertion occurring in each of the subtrees of the fringe can be obtained from the steady state solution of a matrix recurrence relation in a Markov chain. In general, let $p(N)$ be an m-component column vector containing $p_i(N)$. Then

$$p(N) = \left[I + \frac{H}{N+1} \right] p(N-1) \tag{2}$$

where I is the $m \times m$ identity matrix, and H is the transition matrix.

Extensions to other tree collections with more than two types requires consideration of a vector process $\{\vec{X}_N\}$ where X_{jN} is equal to the number of type j leaves at time N.

2.2. The Matrix Recurrence Relation

We start this section by presenting a formal definition of the components of the matrix H in Eq.2.1-2. In fringe analysis problems we always deal with a tree collection $C = \{T_1,...,T_m\}$ of trees. Let L_i be the number of leaves of T_i. An insertion into the k^{th} leaf, $k \in [1,...,L_j]$, of T_j will generate $l_{ij}(k)$ leaves of type T_i. As a consequence we must have

$$\frac{1}{L_j} \sum_{i=1}^{m} \sum_{k=1}^{L_j} l_{ij}(k) = L_j + 1 \text{ , for } 1 \le j \le m \tag{1}$$

This leads to the following definition:

Def. 2.2.1. A fringe analysis problem of size m consists of

(i) m integers $L_1, \ldots, L_m$

(ii) non-negative reals $l_{ij}(k)$, for $1 \le i,j \le m$, $1 \le k \le L_i$, such that

$$\frac{1}{L_j} \sum_{i=1}^{m} \sum_{k=1}^{L_j} l_{ij}(k) = L_j + 1 \text{ , for } 1 \le j \le m.$$

Let $p_i(N)$ be defined as in Eq.2.1-1. Then Eq.2.1-2 can be written as

$$p(N) = \left[I + \frac{H_2 - H_1 - I}{N+1} \right] p(N-1) \tag{2}$$

where

$$H_2 = \left[\frac{1}{L_j} \sum_{k=1}^{L_j} l_{ij}(k) \right]_{1 \le i,j \le m} , \qquad H_1 = \mathrm{diag}\,(L_1,\dots,L_m) ,$$

and I is the $m \times m$ identity matrix.

Def. 2.2.2. Consider a fringe analysis problem. Eq.(2) is the associated recursion equation, where $H = H_2 - H_1 - I = (h_{ij})$ is its transformation matrix. We have

$$h_{ij} = \frac{1}{L_j} \sum_{k=1}^{L_j} l_{ij}(k) - \delta_{ij}(L_j+1)$$

where δ_{ij} is the Kronecker symbol.

Intuitively, the elements in the diagonal of H represent the number of leaves lost due to an insertion minus one, and off diagonal elements represent the number of leaves obtained for each type times the probability that each type is reached in a transition.

Def. 2.2.3. A fringe analysis is *connected* if there is an $l \in [1...m]$ such that $\det(H_{ll}) \neq 0$, where H_{ll} is matrix H with the l^{th} column and l^{th} row deleted.

The following theorem shows that the real part of the eigenvalues of the transition matrix H are non-positive.

Theorem 2.2.1. Consider a connected fringe analysis problem with a $m \times m$ transition matrix H as in Definition 2.2.2. Let $\lambda_1, \dots, \lambda_m$ be the eigenvalues of H. We can order them so that $\lambda_1 = 0$ and $0 > \mathrm{Re}\lambda_2 \ge \mathrm{Re}\lambda_3 \ge \cdots \ge \mathrm{Re}\lambda_m$.

Proof : Consider the sum of the elements in the j^{th} column of H:

$$\sum_{i=1}^{m} h_{ij} = \sum_{i=1}^{m} \left(\frac{1}{L_j} \sum_{k=1}^{L_j} l_{ij}(k) - \delta_{ij} L_j - \delta_{ij} \right)$$

$$= \frac{1}{L_j} \sum_{i=1}^{m} \sum_{k=1}^{L_j} l_{ij}(k) - (L_j+1) \qquad \text{by Eq.(1)}$$

$$= L_j + 1 - (L_j+1) = 0$$

From Gerschgorin's theorem (see Wilkinson, 1965, Chapter 2, § 13) it is known that all eigenvalues of H are contained in the union of the disks with center h_{ii} and radius $\sum_{j \neq i} |h_{ij}|$. Considering that the sum of the elements in any column of H is zero, then all eigenvalues of H have non-positive real part.

From $\sum_{l=1}^{m} h_{lj} = 0$, for $1 \leq j \leq m$, we infer that the vector $E^{(m)} = (1, \ldots, 1)$ is a left eigenvector of H with eigenvalue 0. To show that 0 is an eigenvalue of multiplicity 1, let us look at the characteristic polynomial of H:

$$\underline{\det}|(H-\lambda I) = (-\lambda)^m + S_1(-\lambda)^{m-1} + \ldots + S_{m-1}(-\lambda) + S_m = 0,$$

where S_q is the sum of the principal minors of order q of the matrix H, $q = 1, 2, \ldots, m$ (see Gantmacher, 1959, Chapter 3, § 7). We know that $S_m = 0$, and

$$S_{m-1} = \sum_{i=1}^{m} \det(H_{ii}),$$

where H_{ii} is the matrix H with the i^{th} row and the i^{th} column deleted. It is easy to see that $det\,(H_{ii}) \neq 0$ for some $i \in [1, \ldots, m]$. Thus the linear term of the characteristic polynomial of H is non-null, which implies that 0 is an eigenvalue of multiplicity 1. ▪

Def. 2.2.4. Let $T_j \to T_i$ if $\sum_{k=1}^{L_j} l_{ij}(k) > 0$, i.e. T_j can produce T_i. The symbol $\overset{*}{\to}$ is the reflexive transitive closure of $\to$.

The following theorem describes a test for connectedness.

Theorem 2.2.2. A fringe is connected if and only if there is a T_i such that $T_j \overset{*}{\to} T_i$ for all $j \in [1 \ldots m]$.

Proof : Consider H as in Definition 2.2.2.

Let i be such that $T_j \overset{*}{\to} T_i$ for all j. We will show that $det\,(H_{ii}) \neq 0$. Assume otherwise, i.e. $det\,(H_{ii}) = 0$. Let $\vec{u} = (u_1, \ldots, u_{i-1}, u_{i+1}, \ldots, u_m)$ be a left eigenvector of H_{ii} corresponding to eigenvalue 0. Let u_q be a component of maximal absolute value in $\vec{u}$ (without loss of generality $u_q > 0$) and let $J = \{j \,;\, u_j = u_q\} \subseteq [1 \ldots m] - \{i\}$. Since $T_j \overset{*}{\to} T_i$ for all $j \in J$ and $i \notin J$ there must be some $k \notin J$ and some $j \in J$ such that $T_j \to T_k$. Hence $h_{kj} > 0$. Since $\sum_{l=1}^{m} h_{lj} = 0$ (cf. proof of Theorem 2.2.1) we have

$$\sum_{\substack{l=1 \\ l \neq j}}^{m} u_l h_{lj} = \sum_{l \in J} u_l h_{lj} + \sum_{\substack{l \notin J \\ l \neq i}} u_l h_{lj}$$

$$\geq \sum_{l \in J} u_q h_{lj} - \sum_{\substack{l \notin J \\ l \neq i}} |u_l| h_{lj}$$

$$>\sum_{l \neq i} h_{lj} \geq 0 \,, \quad \text{a contradiction.}$$

The above inequality follows because $\vec{u}$ is a real vector and $h_{lj} \geq 0$ for $l \notin J$, $l \neq i$.

Assume $det(H_{ii}) \neq 0$. We will show $T_j \overset{*}{\rightarrow} T_i$ for all j.

Assume otherwise. Then there is some j such that $\neg T_j \overset{*}{\rightarrow} T_i$. Let $J = \{l;\, T_j \overset{*}{\rightarrow} T_l\}$. Then $\phi \neq J \neq [1...m]$ and $h_{kl} = 0$ for all $k \notin J$ and $l \in J$. We may assume without loss of generality that $J = \{1,...,|J|\}$. Then H has the form

$$H = \begin{bmatrix} H' & 0 \\ 0 & H'' \end{bmatrix}$$

where H' is a $J \times J$ matrix. Note that $det(H_{ii}) = det(H') \cdot det(H''_{ii})$, where H''_{ii} is H'' with i^{th} column and i^{th} row deleted. But H' comes from the transition matrix of a fringe analysis problem (namely the restriction to J) and hence $det(H')=0$ by Theorem 2.2.1, a contradiction. ▪

It remains to solve Eq.(2) for connected fringe analysis problems. In a previous version of the proof of the convergence of the matrix recurrence relation (Gonnet, Ziviani, and Wood, 1981, Lemma 2.1, p.4) the eigenvalues of the transition matrix are assumed to be pairwise distinct. The following theorem, suggested by Mehlhorn (1981), extends the proof to the general case.

Theorem 2.2.3. Let H be the $m \times m$ transition matrix of a connected fringe analysis problem. Let $\lambda_1, \ldots, \lambda_m$ be the eigenvalues of H, where $\lambda_1 = 0 > \mathrm{Re}\lambda_2 \geq \mathrm{Re}\lambda_3 \geq \ldots \geq \mathrm{Re}\lambda_m$, and let x_1 be the right eigenvector of H corresponding to $\lambda_1 = 0$. Then for every vector $p(0)$ there is a c such that

$$|p(N) - cx_1| = O(N^{\mathrm{Re}\lambda_2})$$

where $p(N)$ is defined by Eq.(2).

Proof : For $N \in \mathbf{N}$ let $f_N : C \to \mathbf{C}$ be given by the polynomial $f_N(x) = \prod_{i=1}^{N}(1+\frac{x}{i})$.

Let $f(x) = \lim_{N \to \infty} f_N(x)$. Then $f(0) = 1$, $f(x) = 0$, for $Re(x) < 0$, and $|f(x) - f_N(x)| = O(N^{Re(x)})$ for $Re(x) < 0$, because

$$f_N(x) = \prod_{i=1}^{N}(1+\frac{x}{i})$$

$$
\begin{aligned}
&= \prod_{i=1}^{N} (\frac{x+i}{i}) \\
&= \frac{(x+1)(x+2)\cdots(x+N)}{N!} \\
&= \frac{\Gamma(N+x+1)}{\Gamma(x+1)\Gamma(N+1)} \qquad \text{(cf. Abramowitz, 1972, Eq. 6.1.21)} \\
&= O(N^x).
\end{aligned}
$$

Furthermore, $p(N) = \left[I + \frac{H}{N+1} \right] p(N-1) = f_N(H) p(0)$, and

$p(\infty) = \lim_{N\to\infty} p(N) = f(H) p(0)$. (cf. Gantmacher, 1959, Chapter 5).

Let

$$
J = THT^{-1} = \begin{bmatrix} J_1 & & & 0 \\ & J_2 & & \\ & & \ddots & \\ 0 & & & J_k \end{bmatrix}
$$

be the Jordan matrix corresponding to H, where $J_1, \ldots, J_k$ are the blocks of the Jordan matrix. We have $J_1 = (0)$, i.e. J_1 is a one by one matrix whose only entry is zero. Also

$$
J_l = \begin{bmatrix} \lambda_l & 1 & & \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ & & & \lambda_l \end{bmatrix} \qquad \text{with} \quad Re(\lambda_l) < 0 ,
$$

where λ_l is an eigenvalue of multiplicity l.

Considering that $f_N(x)$ is a polynomial in x then

$$
f(H) = f(T^{-1}JT) = T^{-1} f(J) T = T^{-1} \begin{bmatrix} f(J_1) & & & 0 \\ & \ddots & & \\ & & \ddots & \\ 0 & & & f(J_k) \end{bmatrix} T \cdot
$$

Next we have to compute $f(J_l)$. We have (cf.Gantmacher, 1959, Chapter 5, Example 2)

$$f(J_l) = \begin{bmatrix} f(\lambda_l) & \frac{f'(\lambda_l)}{1!} & \cdot & \cdot & \frac{f^{(r_l-1)}(\lambda_l)}{(r_l-1)!} \\ & \cdot & & & \\ & & \cdot & & \\ & & & \cdot & \\ 0 & & & & f(\lambda_l) \end{bmatrix}$$

where r_l is the multiplicity of λ_l, and $f^{(k)}$ is the k^{th} derivative of f.
Hence $f(J_1) = (1)$, the 1 by 1 matrix with entry 1, and $f(J_l) = (0)$, the r_l by r_l matrix with all entries 0.

$$\text{Thus } f(H) = T^{-1} Q\, T \text{ where } Q = \begin{bmatrix} 1 & 0 & \cdot & \cdot & 0 \\ 0 & 0 & & & \\ \cdot & & \cdot & & \\ \cdot & & & \cdot & \\ 0 & \cdot & \cdot & & 0 \end{bmatrix}$$

and

$$Hp(\infty) = H f(H) p(0) = T^{-1} T\, H\, T^{-1} Q\, T p(0) = T^{-1} J Q\, T p(0) = T^{-1} 0\, T p(0) = \begin{bmatrix} 0 \\ \cdot \\ \cdot \\ 0 \end{bmatrix}$$

since $JQ = 0$, the all zero matrix.

This shows that $p(\infty)$ is a multiple of x_1, say $p(\infty) = c x_1$, because $Hx_1 = \lambda_1 x_1$, or $Hx_1 = 0$ for $\lambda_1 = 0$, and $Hp(\infty) = 0$.

Furthermore

$$f_N(H) = T^{-1} \begin{bmatrix} f_N(J_1) & & \\ & \cdot & \\ & & \cdot \\ & & & f_N(J_k) \end{bmatrix} T = T^{-1} \begin{bmatrix} 1+\varepsilon(N) & \varepsilon(N) & \cdot & \cdot & \varepsilon(N) \\ & \varepsilon(N) & \cdot & \cdot & \varepsilon(N) \\ & \cdot & & & \cdot \\ & & & \cdot & \cdot \\ & & & & \varepsilon(N) \end{bmatrix} T$$

where $\varepsilon(N) = O(N^{\mathrm{Re}\lambda_2})$.

$$\text{Thus } p(N) - p(\infty) = (f_N(H) - f(H)) p(0) = \begin{bmatrix} \delta(N) \\ \cdot \\ \cdot \\ \delta(N) \end{bmatrix} \text{ with } \delta(N) = O(N^{\mathrm{Re}\lambda_2}).$$

This finishes the proof of the theorem. ▪

It is important to note that:

(i) Consider an $m \times m$ transition matrix H of a connected fringe analysis problem. Theorem 2.2.3 says that $p(N)$, the m-component column vector solution of

Eq.(2), converges to the solution of

$$Hq = 0\,, \quad as\ N \to \infty \tag{3}$$

where q is also an m-component column vector that is independent of N, and

$$p(N) = \alpha_1 x_1 + O(N^{\mathrm{Re}\lambda_2}) \tag{4}$$

where x_1 is the right eigenvector of H corresponding to eigenvalue λ_1=0. Furthermore, the eigenvalues of H do not need to be pairwise distinct.

(ii) Let $A_i(N)$ be the expected number of trees of type i in a random search tree with N keys. Let L_i be the number of leaves of the type i tree. We observe that Eq.(1) can be written as

$$p_i(N) = \frac{A_i(N)L_i}{N+1} \tag{5}$$

This section was improved with suggestions from Mehlhorn (1981). In particular, the test for connectedness (cf. Theorem 2.2.2), and the generalisation to consider the case when the matrix H does not have pairwise distinct eigenvalues (cf. Theorem 2.2.3) are Mehlhorn's suggestions.

1.3. Application to Various Search Trees

In fringe analysis problems we always deal with a collection $C = \{T_1, \ldots, T_m\}$ of trees. To be able to apply the technique developed in the previous two sections to different classes of search trees we have to introduce some concepts about the fringes of these search trees.

Def. 2.3.1. A tree collection $C = \{T_1, \ldots, T_m\}$ is *weakly-closed* if for all $j \in [1,...,m]$ an insertion into T_j always leads to one or more T_i, $i \in [1,...,m]$.

Def. 2.3.2. A tree collection $C = \{T_1, \ldots, T_m\}$ is *closed* when (i) C is weakly-closed and (ii) the effect of an insertion on the composition of the fringe is determined only by the subtree of the fringe where the insertion is performed.

The tree collection of Figure 2.1.1 is an example of a closed tree collection because inserting into either type 1 or type 2 tree results in trees of type 1 or type 2. In fact the tree collection of 2-3 trees of height k is a closed tree collection for any k>0. On the other hand the collection of AVL trees † with more than

† See Appendix A for the definition of AVL trees

2 and fewer than 7 leaves (see Figure 2.3.1) is not closed. This is because an insertion into a type 2 tree of Figure 2.3.1, when the type 2 tree is part of the fringe of an AVL tree, may cause a rotation higher in the tree, and the composition of the fringe depends on this rotation at the higher level. Figure 2.3.2 shows an instance of an AVL tree where an insertion into a type 2 tree does not lead to a type 3 tree as expected.

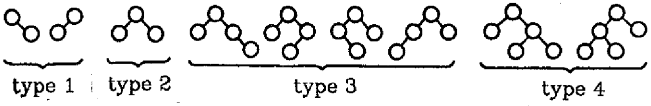


Fig. 2.3.1 Tree collection of AVL trees with more than 2 and less than 7 leaves (leaves not shown)

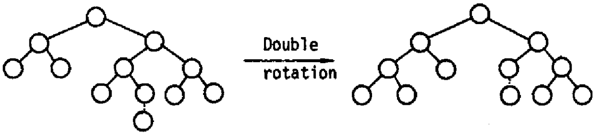


Fig. 2.3.2 Example of an insertion that unexpectedly changes the fringe of an AVL tree (dotted edge shows the point of insertion)

Def. 2.3.3. A tree collection $C = \{T_1, \ldots, T_m\}$ is *ambiguous* when a tree in C appears as a subtree of another tree in C. Figure 2.3.3 shows an AVL tree collection that is ambiguous, since a tree of type 1 is a subtree of trees of type 3.

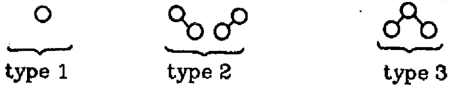


Fig. 2.3.3 Tree collection of AVL trees with more than 1 and less than 5 leaves (leaves not shown)

Def. 2.3.4. A tree collection $C = \{T_1, \ldots, T_m\}$ is *open* if it is not weakly-closed.

3. AN ANALYSIS OF 2-3 TREES

3.1. Motivation

2-3 trees were introduced by John Hopcroft in 1970 (see Knuth, 1973, p.468). In a 2-3 tree every internal node contains either one or two keys, and all leaves appear at the same level. The definition of 2-3 trees, the description of the insertion algorithm, and the transformations (called splits) necessary to keep the tree balanced are presented in Appendix B.

The class of 2-3 trees is a special class of B-trees. Unlike B-trees, 2-3 trees are more appropriate for use in primary store than secondary. For this reason they have become equal contenders with AVL trees, often being the preferred data structure (Aho, Hopcroft and Ullman (1974), and Huddleston and Mehlhorn(1980)).

The first analytical results about 2-3 trees were obtained by Yao (1978). Although his results were slightly extended by Brown (1979b), many questions of interest were left open. Some of these questions are:

(i) The expected number of nodes in a 2-3 tree after N random insertions is certainly of interest, since this measure indicates storage utilisation. We extend and refine the results of Yao with regard to this measure;

(ii) When considering insertions, the costliest operation is surely that of splitting an overfull node, since this involves not only the creation of a new node but also an insertion into the next higher level of the tree. Knuth (Chvatal, Klarner, and Knuth, 1972, Problem 37) raised the following question related to 2-3 trees: " how many splittings will occur on the n^{th} random insertion, on the average, ...". We present the first partial analysis of this measure for 2-3 trees;

(iii) A different insertion algorithm for B-trees, which uses a technique called overflow, was presented by Bayer and McCreight (1972, p.183) and also by Knuth (1973, pp. 477-478, § 6.2.4). In the overflow technique, instead of splitting an overfull node, we look first at its sibling nodes and make a rearrangement of keys when possible. The effect of the overflow technique is to produce trees with fewer internal nodes on the average. This results in a better storage utilisation. We present an analysis of 2-3 trees using an overflow technique which is a particular case of the overflow technique presented by Bayer and McCreight;

(iv) Consider the concurrency of operations on 2-3 trees; see Kwong and Wood (1980) for a survey of the techniques used. One basic technique identified there was first used by Bayer and Schkolnick (1977), namely lock the deepest safe node (dsn) on the insertion path. A node is insertion-safe if it contains fewer than the maximum number of keys allowed. Then a safe node is the deepest one in a particular insertion path if there are no safe nodes below it. Since locking the deepest safe node effectively prevents access by other processes it is of interest to determine how deep the deepest safe node can be expected to be. Our results enable us to provide some insight into this question.

We now define certain complexity measures:

(i) Let $\bar{n}(N)$ be the expected number of nodes in a 2-3 tree after the random insertion of N keys into an initially empty tree;

(ii) Let $Pr\{j\ splits\}$ be the probability that j splits occur on the $(N+1)^{st}$ random insertion into a random 2-3 tree with N keys;

(iii) Let $Pr\{j\ or\ more\ splits\}$ be the probability that j or more splits occur on the $(N+1)^{st}$ random insertion into a random 2-3 tree with N keys;

(iv) Let $\bar{s}(N)$ be the expected number of splits that occur in a 2-3 tree during the random insertion of N keys into an initially empty tree;

(v) Let $E[s(N)]$ be the expected number of splits that will occur on the $(N+1)^{st}$ insertion into a random 2-3 tree with N keys;

(vi) Let $Pr\{dsn\ at\ j^{th}\ lowest\ level\}$ be the probability that the deepest safe node on a random search is located at the j^{th} $(j \geq 1)$ lowest level of a random 2-3 tree with N keys;

(vii) Let $Pr\{dsn\ above\ j^{th}\ lowest\ level\}$ be the probability that the deepest safe node on a random search is located above the j^{th} lowest level of a random 2-3 tree with N keys.

In Sections 3.2, 3.3, and 3.4 we shall derive exact values for $Pr\{0\ splits\}$, $Pr\{1\ split\}$, $Pr\{2\ splits\}$, $Pr\{3\ or\ more\ splits\}$, and bounds on $\bar{s}(N)$, $E[s(N)]$, and improve Yao's previous results on $\bar{n}(N)$. In Section 3.5 we shall derive exact values for $Pr\{0\ splits\}$, $Pr\{1\ split\}$, $Pr\{2\ or\ more\ splits\}$, and bounds on $\bar{n}(N)$, $\bar{s}(N)$, and $E[s(N)]$ for an insertion algorithm that uses an overflow technique. In Section 3.6 we shall derive exact values for $Pr\{dsn\ at\ 1^{st}\ lowest\ level\}$, $Pr\{dsn\ at\ 2^{nd}\ lowest\ level\}$, $Pr\{dsn\ at\ 3^{rd}\ lowest\ level\}$, and $Pr\{dsn\ above\ 3^{rd}\ lowest\ level\}$ for the normal insertion algorithm, and $Pr\{dsn\ at\ 1^{st}\ lowest\ level\}$, $Pr\{dsn\ at\ 2^{nd}\ lowest\ level\}$, and $Pr\{dsn\ above\ 2^{nd}\ lowest\ level\}$ for the insertion algorithm using an overflow technique. In Section 3.7 we discuss the possibilities of higher order analyses.

Table 3.1.1 shows the summary of the results related to 2-3 trees using the normal insertion algorithm. The lower order analyses are included to indicate the improvements achieved by the third order analysis. Table 3.1.2 shows the summary of the results related to 2-3 trees using the overflow technique.

3.2. First Order Analysis

The analysis of the lowest level of the 2-3 tree to estimate $\bar{n}(N)$, $Pr\{0\ splits\}$, $Pr\{1\ or\ more\ splits\}$, $\bar{s}(N)$, and $E[s(N)]$ can be carried out in the following way. The tree collection shown in Figure 3.2.1 contains two members and its corresponding transition matrix is

$$H = \begin{bmatrix} -3 & 4 \\ 3 & -4 \end{bmatrix}$$

From Eq. 2.2-3 we have $Hp(N) = 0$, and therefore $p_1(\infty) = 4/7$, and $p_2(\infty) = 3/7$. Since the eigenvalues of H are 0 and -7, we observe that $p_1(N) = 4/7$ and $p_2(N) = 3/7$ for $N \geq 6$. To simplify notation $p_i(N)$ is written as p_i throughout the remainder of this thesis.

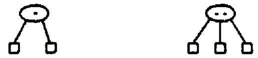


Fig. 3.2.1 Tree collection of 2-3 trees of height 1

Lemma 3.2.1. Let nl indicate the number of nodes at level l of a 2-3 tree with N keys. Then the number of nodes above level l, nal, is bounded by

$$\frac{nl-1}{2} \leq nal \leq nl-1$$

Proof : Consider the level l as being the $N+1$ leaves of a 2-3 tree with N keys. (Each leaf represents a node.) The minimum and the maximum number of nodes above the level l is obtained when each node above level l contains 2 keys and 1 key respectively. (That is $2nal = nl-1$ and $nal = nl-1$ respectively.) ▪

Lemma 3.2.1 and Eq. 2.2-5 lead to the following theorem:

Theorem 3.2.2. The expected number of nodes in a random 2-3 tree with N keys is bounded by

$$\left(1+\frac{1}{2}\right)\left[\frac{p_1}{L_1}+\frac{p_2}{L_2}\right](N+1)-\frac{1}{2} \leq \bar{n}(N) \leq 2\left[\frac{p_1}{L_1}+\frac{p_2}{L_2}\right](N+1)-1 \quad \text{for } N \geq 1$$

	First Order Analysis ($N \geq 6$)	Second Order † Analysis ($N \rightarrow \infty$)	Third Order ‡ Analysis ($N \rightarrow \infty$)
$\frac{\bar{n}(N)}{N}$	$\left[0.64 + 0.14/N,\ 0.86 - 0.14/N\right]$	$\left[0.70 + 0.20/N,\ 0.79 - 0.21/N\right]$	$\left[0.73 + 0.23/N,\ 0.77 - 0.23/N\right]$
$Pr\{0\ splits\}$	4/7	4/7	4/7
$Pr\{1\ or\ more\ splits\}$	3/7	3/7	3/7
$Pr\{1\ split\}$	-	0.25	0.25
$Pr\{2\ or\ more\ splits\}$	-	0.18	0.18
$Pr\{2\ splits\}$	-	-	0.10
$Pr\{3\ or\ more\ splits\}$	-	-	0.08
$\bar{s}(N)$	$\left[0.64 + 0.14/N - \lceil \log_3(N+1) \rceil / N,\ 0.86 - 0.14/N - \lfloor \log_2(N+1) \rfloor / N\right]$	$\left[0.70 + 0.20/N - \lceil \log_3(N+1) \rceil / N,\ 0.79 - 0.21/N - \lfloor \log_2(N+1) \rfloor / N\right]$	$\left[0.73 + 0.23/N - \lceil \log_3(N+1) \rceil / N,\ 0.77 - 0.23/N - \lfloor \log_2(N+1) \rfloor / N\right]$
$E[s(N)]$	$\left[0.43,\ 0.43 \lfloor \log_2(N+1) \rfloor\right]$	$\left[0.61,\ 0.25 + 0.18 \lfloor \log_2(N+1) \rfloor\right]$	$\left[0.69,\ 0.46 + 0.08 \lfloor \log_2(N+1) \rfloor\right]$
Upper bound on $\bar{h}(N)$	$\log_2(N+1) - 0.22$	$\log_2(N+1) - 0.46$	$\log_2(N+1) - 0.69$
$Pr\{dsn\ at\ 1^{st}\ lowest\ level\}$	4/7	4/7	4/7
$Pr\{dsn\ at\ 2^{nd}\ lowest\ level\}$	-	0.25	0.25
$Pr\{dsn\ at\ 3^{rd}\ lowest\ level\}$	-	-	0.10
$Pr\{dsn\ above\ 3^{rd}\ l.level\}$	-	-	0.08

† Results are approximated to $O(N^{-5.55})$

‡ Results are approximated to $O(N^{-4.37})$

Table 3.1.1 Summary of the 2-3 tree results

	Second Order Analysis ($N \rightarrow \infty$) †
$\frac{\bar{n}(N)}{N}$	$\left[0.63 + 0.13/N,\ 0.71 - 0.29/N\right]$
$Pr\{0\ splits\}$	0.61
$Pr\{1\ split\}$	0.23
$Pr\{2\ or\ more\ splits\}$	0.16
$\bar{s}(N)$	$\left[0.63 + 0.13/N - \lceil \log_3(N+1) \rceil / N,\ 0.71 - 0.29/N - \lfloor \log_2(N+1) \rfloor / N\right]$
$E[\ s(N)]$	$\left[0.55,\ 0.23 + 0.16 \lfloor \log_2(N+1) \rfloor\right]$
$Pr\{dsn\ at\ 1^{st}\ lowest\ level\}$	0.61
$Pr\{dsn\ at\ 2^{nd}\ lowest\ level\}$	0.23
$Pr\{dsn\ above\ 2^{nd}\ lowest\ level\}$	0.16

† Results are approximated to $O(N^{-6.61})$

Table 3.1.2 Summary of the 2-3 tree results using an overflow technique

Corollary. $\frac{9}{14}+\frac{1}{7N}\leq\frac{\bar{n}(N)}{N}\leq\frac{6}{7}-\frac{1}{7N}$ for $N\geq6$

The remaining results are contained in the lemmas that follow.

Lemma 3.2.3. The probability that no split occurs on the $(N+1)^{st}$ random insertion into a 2-3 tree with N keys is

$$Pr\{0\ splits\} = \frac{4}{7} \text{ for } N\geq6$$

Proof : An insertion into a type 1 tree shown in Figure 3.1.1 causes no split, and the probability that a random insertion into a random 2-3 tree falls into a type 1 tree is p_1. ▪

Lemma 3.2.4. The probability that 1 or more splits occur on the $(N+1)^{st}$ random insertion into a 2-3 tree with N keys is

$$Pr\{1\ or\ more\ splits\} = \frac{3}{7} \text{ for } N\geq6$$

Proof : Similar to the proof of Lemma 3.2.3. ▪

Lemma 3.2.5. Let $\bar{h}(N)$ denote the expected height of a random 2-3 tree with N keys. Then the expected number of splits is

$$\bar{s}(N) = \frac{\bar{n}(N)}{N}-\frac{\bar{h}(N)}{N}$$

Proof : From the insertion algorithm presented in Appendix B we can see that each time a node split occurs one new node is created, except when the node is a root, in which case two nodes are created. ▪

Lemma 3.2.6. The height of a 2-3 tree with N keys is bounded by

$$\lceil\log_3(N+1)\rceil\leq\bar{h}(N)\leq\lfloor\log_2(N+1)\rfloor$$

Proof : The lower bound and the upper bound on the height are obtained when each node of the 2-3 tree contains 2 and 1 key respectively. ▪

Lemmas 3.2.5 and 3.2.6 lead to the following theorem:

Theorem 3.2.7. The expected number of splits in a random 2-3 tree with N keys is bounded by

$$\frac{9}{14}+\frac{1}{7N}-\frac{\lfloor\log_2(N+1)\rfloor}{N}\leq \bar{s}(N) \leq \frac{6}{7}-\frac{1}{7N}-\frac{\lfloor\log_3(N+1)\rfloor}{N} \quad \text{for } N\geq 6$$

Lemma 3.2.8. A lower bound on the expected number of splits that will occur on the $(N+1)^{st}$ insertion into a random 2-3 tree with N keys is

$$E[s(N)] \geq Pr\{1 \textit{ or more splits}\}$$

Proof : Similar to the proof of Lemma 3.2.3. ▪

Corollary. $E[s(N)] \geq \frac{3}{7}$ for $N\geq 6$

Lemma 3.2.9. An upper bound on the expected number of splits that will occur on the $(N+1)^{st}$ insertion into a random 2-3 tree with N keys is

$$E[s(N)] \leq Pr\{1 \textit{ or more splits}\}\lfloor\log_2(N+1)\rfloor$$

Proof : The upper bound on $E[s(N)]$ is equal to the number of splits/insertion in the fringe plus all splits that might occur in the nodes above the lowest level, which might be equal to the height of the tree with all nodes binary but the nodes on the path of splitting. ▪

Lemmas 3.2.8 and 3.2.9 lead to the following theorem:

Theorem 3.2.10. The expected number of splits that will occur on the $(N+1)^{st}$ insertion into a random 2-3 tree with N keys is bounded by

$$\frac{3}{7}\leq E[s(N)] \leq \frac{3}{7}\lfloor\log_2(N+1)\rfloor \quad \text{for } N\geq 6$$

It is interesting to conjecture that the expected value for $E[s(N)]$ converges to the value of $\bar{s}(N)$. However, we cannot prove this; $E[s(N)]$ may oscillate between a lower bound and an upper bound, where the lower bound is the number of splits per insertion in the fringe, and the upper bound is the number of splits per insertion in the fringe plus the number of splits per insertion outside the fringe. (The upper bound is a function of $\log_2 N$.)

Lemma 3.2.11. The expected number of keys in the fringe of a 2-3 tree with N keys that corresponds to the tree collection shown in Figure 3.2.1 is

$$\bar{f}(N) = \left[\frac{p_1}{L_1}+2\frac{p_2}{L_2}\right](N+1)$$

Proof : The above expression is obtained by observing Figure 3.2.1 and by using Eq.2.2-5. ▪

Corollary. $\overline{f}(N) = \frac{4}{7}(N+1)$ for $N \geq 6$.

Theorem 3.2.12. The expected height of a 2-3 tree with N keys is bounded above by

$$\bar{h}(N) \leq \log_2(N+1) - 0.22239$$

Proof : Let ***nkal*** indicate the number of keys above the level l of a 2-3 tree. Considering the second lowest level (distance one from the leaves), and using Lemma 3.2.6 then the height $h(n)$ of a 2-3 tree with N keys is bounded by

$$\lceil \log_3(nkal+1) \rceil + 1 \leq h(N) \leq \lfloor \log_2(nkal+1) \rfloor + 1.$$

Considering the expected value of the right hand side of the above inequality then

$$\bar{h}(N) \leq E[\lfloor \log_2(nkal+1) \rfloor + 1] \leq E[\log_2(nkal+1)+1]$$

Using Jensen's inequality (Jensen, 1906, p.180) we obtain

$$\bar{h}(N) \leq \log_2 E[nkal+1]+1 \qquad (1)$$

But

$$E[nkal] = N - \overline{f}(N)$$

where $\overline{f}(N) = \frac{4}{7}(N+1)$ for $N \geq 6$ (see Lemma 3.2.11). Then

$$E[nkal] = \frac{3}{7}(N+1) - 1.$$

Substituting this equation into Eq.(1) we obtain

$$\bar{h}(N) \leq \log_2(N+1) - 0.22239 \quad ▪$$

3.3. Second Order Analysis

The analysis for the two lowest levels of 2-3 trees leads to better bounds for $\bar{n}(N)$, $\bar{s}(N)$, $E[s(N)]$, and exact results for $Pr\{1\ split\}$, and $Pr\{2\ or\ more\ splits\}$. Yao (1978) showed that there are 12 possible trees in the tree collection of 2-3 trees of height 2, which are grouped into 7 types, as shown in Figure 3.3.1. The corresponding transition matrix is shown in Table 3.3.1.

Again using Eq. 2.2-3 we obtain

$$\begin{aligned} p_1 &= 1656/7991 \\ p_2 &= 1980/7991 \\ p_3 &= 5472/55937 \\ p_4 &= 7128/55937 \\ p_5 &= 1575/7991 \\ p_6 &= 800/7991 \\ p_7 &= 180/7991. \end{aligned} \qquad (1)$$

Since the eigenvalues of H are $0, -6.55\pm6.25i, -7, -9.23\pm1.37i$, and -13.44, using Eq. 2.2-3 the asymptotic values of $p(N)$ obtained from Eq. 2.2-4 are approximated to the $O(N^{-6.55})$.

Lemma 3.2.1 and expression Eq. 2.2-5 lead to the following theorem:

Theorem 3.3.1. The expected number of nodes in a random 2-3 tree with N keys is bounded by

$$\left\{\left(3+\frac{1}{2}\right)\left[\sum_{i=1}^{3}\frac{p_i}{L_i}\right]+\left(4+\frac{1}{2}\right)\left[\sum_{i=4}^{7}\frac{p_i}{L_i}\right]\right\}(N+1)-\frac{1}{2}\le \bar{n}(N) \le \left\{4\left[\sum_{i=1}^{3}\frac{p_i}{L_i}\right]+5\left[\sum_{i=4}^{7}\frac{p_i}{L_i}\right]\right\}(N+1)-1$$

Corollary.

$$\frac{78501}{111874}+\frac{11282}{55937N}+O(N^{-6.55}) \le \frac{\bar{n}(N)}{N} \le \frac{44343}{55937}-\frac{11594}{55937N}+O(N^{-6.55})$$

To five place decimals we have

$$0.70169+\frac{0.20169}{N}+O(N^{-6.55}) \le \frac{\bar{n}(N)}{N} \le 0.79273-\frac{0.20727}{N}+O(N^{-6.55}).$$

Lemma 3.3.2. The probability that 1 split occurs on the $(N+1)^{st}$ random insertion into a 2-3 tree with N keys is

$$Pr\{1\ split\} = \frac{13788}{55937}+O(N^{-6.55})$$

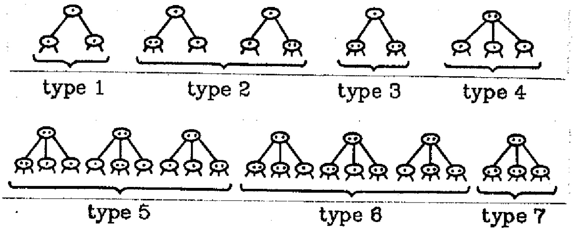


Figure 3.3.1 Tree collection of 2-3 trees of height 2
(stubs indicate leaves)

$$\begin{bmatrix}
-5 & & & & 8\times3/7 & 4\times6/8 & 4\times6/9 \\
5 & -6 & & & & 5\times6/8 & 5\times6/9 \\
 & 6\times2/5 & -7 & & & & 6\times6/9 \\
 & 6\times3/5 & & -7 & & & \\
 & & 7 & 7 & -8 & & \\
 & & & & 8\times4/7 & -9 & \\
 & & & & & 9\times2/8 & -10
\end{bmatrix}.$$

Table 3.3.1 Transition matrix corresponding to the tree collection of 2-3 trees of height 2 shown in Figure 3.3.1

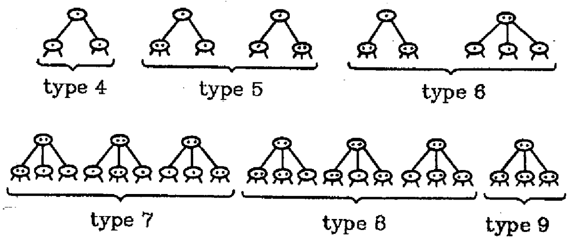


Figure 3.4.1 Tree collection of 2-3 trees of height 2 obtained by grouping type 3 and type 4 shown in Figure 3.3.1 into type 6 above

Proof : An insertion into the type 2 tree shown in Figure 3.3.1 causes one split 3/5 of the time, and an insertion into the type 3 shown in Figure 3.3.1 always causes one split. Since the probability that a random insertion into a random 2-3 tree falls into a type 2 or type 3 tree are p_2 and p_3 respectively, then $Pr\{1\ split\} = 3/5p_2+p_3$. ▪

Lemma 3.3.3. The probability that 2 or more splits occur on the $(N+1)^{st}$ random insertion into a 2-3 tree with N keys is

$$Pr\{2\ or\ more\ splits\} = \frac{1455}{7991}+O(N^{-6.55})$$

Proof : Similar to the proof of Lemma 3.3.2. ▪

Lemma 3.2.5 leads to the following theorem:

Theorem 3.3.4. The expected number of splits in a random 2-3 tree with N keys is bounded by

$$\frac{78501}{111874}+\frac{11282}{55937N}-\frac{\lfloor \log_2(N+1)\rfloor}{N}+O(N^{-6.55}) \le \bar{s}(N) \le$$

$$\frac{44343}{55937}-\frac{11594}{55937N}-\frac{\lceil \log_3(N+1)\rceil}{N}+O(N^{-6.55})$$

To five place decimals we have

$$0.70169+\frac{0.20169}{N}-\frac{\lfloor \log_2(N+1)\rfloor}{N}+O(N^{-6.55}) \le \bar{s}(N) \le$$

$$0.79273-\frac{0.20727}{N}-\frac{\lceil \log_3(N+1)\rceil}{N}+O(N^{-6.55}) .$$

Lemma 3.3.5. A lower bound on the expected number of splits that will occur on the $(N+1)^{st}$ insertion into a random 2-3 tree with N keys is

$$E[s(N)] \ge Pr\{1\ split\}+2Pr\{2\ or\ more\ splits\}$$

Proof : Similar to the proof of Lemma 3.2.3. ▪

Lemma 3.3.6. An upper bound on the expected number of splits that will occur on the $(N+1)^{st}$ insertion into a random 2-3 tree with N keys is

$$E[s(N)] \le Pr\{1\ split\}+Pr\{2\ or\ more\ splits\}\lfloor \log_2(N+1)\rfloor$$

Proof : Similar to the proof of Lemma 3.2.8. ▪

Lemmas 3.3.5 and 3.3.6 lead to the following theorem:

Theorem 3.3.7. The expected number of splits that will occur on the $(N+1)^{st}$ insertion into a random 2-3 tree with N keys is bounded by

$$\frac{34158}{55937}+O(N^{-6.55}) \le E[s(N)] \le \frac{13788}{55937}+\frac{1455}{7991}\lfloor \log_2(N+1)\rfloor+O(N^{-6.55})$$

To five place decimals we have

$$0.61065+O(N^{-6.55}) \le E[s(N)] \le 0.24649+0.18208\lfloor \log_2(N+1)\rfloor+O(N^{-6.55}) \ .$$

Lemma 3.3.8. The expected number of keys in the fringe of a 2-3 tree with N keys that corresponds to the tree collection shown in Figure 3.3.1 is

$$\bar{f}(N) = \left(3\frac{p_1}{L_1}+4\frac{p_2}{L_2}+5\frac{p_3}{L_3}+5\frac{p_4}{L_4}+6\frac{p_5}{L_5}+7\frac{p_6}{L_6}+8\frac{p_7}{L_7}\right)(N+1)$$

Proof : The above expression is obtained by observing Figure 3.3.1 and by using Eq.2.2-5. ▪

Corollary. $\bar{f}(N) = \frac{6536}{7991}(N+1)+O(N^{-6.55})$

Theorem 3.3.9. The expected height of a 2-3 tree with N keys is bounded above by

$$\bar{h}(N) \le \log_2(N+1)-0.45736$$

Proof : Similar to the proof of Theorem 3.2.12. ▪

3.4. Third Order Analysis

In this section we present the analysis of the three lowest levels of 2-3 trees. Brown (1979b) performed a three level analysis using a transition matrix of 978 × 978 elements, and obtained asymptotic values for the number of nodes with one key and the number of nodes with two keys at each of the three lowest levels. However an equivalent three level analysis can be performed on a smaller matrix by grouping trees into types, in the same way the two level matrix in the previous section was reduced from 12 × 12 to 7 × 7. If we consider combinations of the 7 types of the two level tree collection as subtrees of nodes with one and two keys then it is possible to obtain a three level tree collection with 224 types. This may be further reduced to 147 types as we shall see in the following.

The idea behind our approach is to group all trees with the same number of leaves into types. Thus the tree collection shown in Figure 3.3.1 is reduced from 7 types to 6 types by grouping the types 3 and 4 into one unique type, as shown in Figure 3.4.1. In this new tree collection the types are numbered sequentially from 4 to 9, where the type 4 tree has 4 leaves, the type 5 tree has 5 leaves, ..., and the type 9 tree has 9 leaves. Of course the probability related to the type 6 shown in Figure 3.4.1 is the sum of the probabilities related to the types 3 and 4 shown in Figure 3.3.1, and the probabilities of the other types remain as before. (Types 4, 5, 7, 8, and 9 shown in Figure 3.4.1 have the same probabilities as types 1, 2, 5, 6, and 7 shown in Figure 3.3.1 respectively.)

Lemma 3.4.1. The 6 types of the tree collection shown in Figure 3.4.1 can be used as subtrees of nodes with one or two keys in order to obtain a three level tree collection.

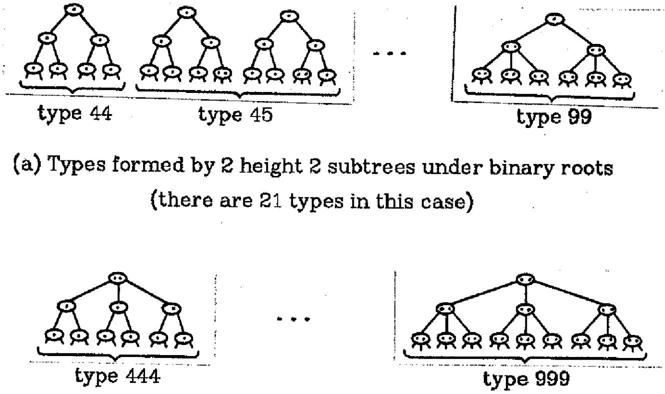
Proof : From the trees shown in Figure 3.3.1, the ones with the same number of leaves appear as subtrees of nodes with one or two keys having the same probability, simply because they belong to the same type. ▪

Lemma 3.4.2. The two level tree collection with 6 types shown in Figure 3.4.1 can be used to form a three level 2-3 tree collection with 147 types.

Proof : Following the notation presented in Figure 3.4.2, the 147 types of the three level tree collection are represented either as type ij ($4 \leq i \leq 9$ and $i \leq j \leq 9$) for the tree types with binary roots, or as type ijk ($4 \leq i \leq 9$, $4 \leq j \leq 9$, and $i \leq k \leq 9$) for the tree types with ternary roots. The number of tree types with binary roots is 21, and the number of tree types with ternary roots is 126, which gives a total of 147 types. ▪

Notice that the trees with ternary roots must have $4 \leq j \leq 9$ (and not $i \leq j \leq 9$ and $j \leq k \leq 9$). Consider for example types 459 and 495. These must be treated as different types because an insertion into the leftmost leaf of the middle subtree of type 459 gives types 44 and 56, and an insertion into the leftmost leaf of the middle subtree of type 495 gives types 45 and 46.

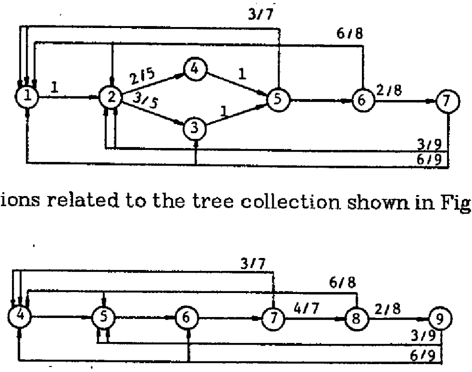
Lemma 3.4.3. The transitions related to the 6 types of the tree collection shown



(a) Types formed by 2 height 2 subtrees under binary roots
(there are 21 types in this case)

(b) Types formed by 3 height 2 subtrees under ternary roots
(there are 126 types in this case)

Figure 3.4.2 Tree collection of 2-3 trees of height 3 (type 44 is formed by two subtrees with 4 leaves each, type 45 is formed by two subtrees with 4 and 5 leaves each, etc)



(a) Transitions related to the tree collection shown in Figure 3.3.1

(b) Transitions related to the tree collection shown in Figure 3.4.1

Figure 3.4.3 Diagrams for transitions

in Figure 3.4.1 are equivalent to the transitions related to the 7 types of the tree collection shown in Figure 3.3.1 when both are used as subtrees of nodes with one or two keys in order to obtain a three level tree collection.

Proof : Figures 3.4.3(a) and 3.4.3(b) show the transitions related to the tree collections shown in Figure 3.3.1 and Figure 3.4.1 respectively. It is irrelevant whether we use the 6 types of the tree collection shown in Figure 3.4.1 or the 7 types of the tree collection shown in Figure 3.3.1 as subtrees of nodes with one or two keys. In the case we choose the former types we have to remember that (i) the type 6 shown in Figure 3.4.3(b) is composed by types 3 and 4 shown in Figure 3.4.3(a), and (ii) from Eq. 3.3-1 that types 3 and 4 shown in Figure 3.4.3(a) occur with probabilities 5472/55937 and 7128/55937 respectively. ▪

Using Eq. 2.2-3 for the 147 × 147 transition matrix T we obtain a linear system of 147 unknowns, which was solved using an algebraic manipulation language called MAPLE, developed by Geddes and Gonnet (1981). An advantage of using such a system is that we obtain rationals instead of real numbers, avoiding computational errors. The 147 p_i's obtained contain integer numbers in the numerator and in the denominator, with approximately 90 digits each. Since the eigenvalues of H are 0, $-4.37\pm 8.23i$, ..., $-31.49\pm 2.92i$, and -33.27, the asymptotic values for $p(N)$ obtained from Eq. 2.2-4 are approximated to the $O(N^{-4.37})$.

We shall see that the analysis for the three lowest levels of 2-3 trees leads to better results for $\bar{n}(N)$, $\bar{s}(N)$, $E[s(N)]$, and exact results for $Pr\{2\ splits\}$, and $Pr\{3\ or\ more\ splits\}$.

Lemma 3.4.4. Let $nn(i)$ indicate the number of nodes of the type i tree in the tree collection shown in Figure 3.4.1. Then

$$nn(i) = 3 \qquad \text{for } 4\le i\le 5$$

$$nn(6) = 3\times\frac{5472}{12600}+4\times\frac{7128}{12600}$$

$$nn(i) = 4 \qquad \text{for } 7\le i\le 9$$

Proof : For i=4,5,7,8,9, from Figure 3.4.1 the values for $nn(i)$ are immediate. For i = 6, consider the two trees of type 6 shown in Figure 3.4.1. We know from Eq. 3.3-1 that the tree with 3 nodes occur with probability 5472/55937, and the tree with 4 nodes occur with probability 7128/55937. Normalising the probabilities we obtain

$$nn(6) = 3\times\frac{5472}{12600}+4\times\frac{7128}{12600} \; ■$$

Let L_{ij} indicate the number of leaves of the type ij tree ($4\le i\le 9$, $i\le j\le 9$) shown in Figure 3.4.2. Let L_{ijk} indicate the number of leaves of the type ijk tree ($4\le i\le 9$, $4\le j\le 9$, $i\le k\le 9$) shown in Figure 3.4.2. The proof of the following theorem is similar to the proof of Theorems 3.2.2 and 3.3.1. Note that the double summation contains the number of nodes of type i ($4\le i\le 9$), plus the number of nodes of type j ($i\le j\le 9$), plus the binary root node (see Figures 3.4.1 and 3.4.2), plus 1/2 for the lower bound (1 for the upper bound) due to the number of nodes outside the fringe. (cf. Theorem 3.2.1.) The triple summation is similar.

Theorem 3.4.5. The expected number of nodes in a random 2-3 tree with N keys is bounded by

$$\left[\sum_{i=4}^{9}\sum_{j=i}^{9}\left(nn(i)+nn(j)+1+\frac{1}{2}\right)\left(\frac{p_{ij}}{L_{ij}}\right) + \right.$$

$$\left.\sum_{i=4}^{9}\sum_{j=4}^{9}\sum_{k=i}^{9}\left(nn(i)+nn(j)+nn(k)+1+\frac{1}{2}\right)\left(\frac{p_{ijk}}{L_{ijk}}\right)\right](N+1)-\frac{1}{2}$$

$$\le \bar{n}(N) \le \left[\sum_{i=4}^{9}\sum_{j=i}^{9}\left(nn(i)+nn(j)+2\right)\left(\frac{p_{ij}}{L_{ij}}\right) + \right.$$

$$\left.\sum_{i=4}^{9}\sum_{j=4}^{9}\sum_{k=i}^{9}\left(nn(i)+nn(j)+nn(k)+2\right)\left(\frac{p_{ijk}}{L_{ijk}}\right)\right](N+1)-1$$

Corollary. †

$$0.72683+\frac{0.22683}{N}+O(N^{-4.37}) \le \frac{\bar{n}(N)}{N} \le 0.76556-\frac{0.23444}{N}+O(N^{-4.37})$$

Experimental results show that $\bar{n}(N)$ is approximately $0.75N$. The

† All the results of this section are presented as real numbers because the exact rationals are too long to be printed. As a curiosity, the exact lower bound on $\bar{n}(N)$ is

$$\frac{779859931429091308052840727221956234622563673252979381819376884206537337452971355745773 4066}{10729604856083907760988691252514032168069885375054364827047705340026365840593873897782021229}$$

$$= 0.72683\ 00574\ 80536\ \cdots$$

minimum and the maximum number of internal nodes possible in any 2-3 tree with N keys are 0.5 N and N respectively.

Lemma 3.4.6. The probability that 2 splits occur on the $(N+1)^{st}$ random insertion into a 2-3 tree with N keys is

$$Pr\{2 \ splits\} = 0.10462+O(N^{-4.37})$$

Proof : Similar to the proof of Lemma 3.3.2. ▪

Lemma 3.4.7. The probability that 3 or more splits occur on the $(N+1)^{st}$ random insertion into a 2-3 tree with N keys is

$$Pr\{3 \ or \ more \ splits\} = 0.07745 + O(N^{-4.37})$$

Proof : Similar to the proof of Lemma 3.3.2. ▪

Lemma 3.2.5 leads to the following theorem:

Theorem 3.4.8. The expected number of splits in a random 2-3 tree with N keys is bounded by

$$0.72683+\frac{0.22683}{N}-\frac{\lfloor \log_2(N+1) \rfloor}{N}+O(N^{-4.37})$$

$$\leq \bar{s}(N) \leq 0.76556-\frac{0.23444}{N}-\frac{\lceil \log_3(N+1) \rceil}{N}+O(N^{-4.37})$$

Lemma 3.4.9. A lower bound on the expected number of splits that will occur on the $(N+1)^{st}$ insertion into a random 2-3 tree with N keys is

$$E[s(N)] \geq Pr\{1 \ split\}+2Pr\{2 \ splits\}+3Pr\{3 \ or \ more \ splits\}$$

Proof : Similar to the proof of Lemma 3.2.3. ▪

Lemma 3.4.10. An upper bound on the expected number of splits that will occur on the $(N+1)^{st}$ insertion into a random 2-3 tree with N keys is

$$E[s(N)] \leq Pr\{1 \, split\}+2Pr\{2 \, splits\}+Pr\{3 \, or \ more \ splits\} \lfloor \log_2(N+1) \rfloor$$

Proof : Similar to the proof of Lemma 3.2.8. ▪

Lemmas 3.4.9 and 3.4.10 lead to the following theorem:

Theorem 3.4.11. The expected number of splits that will occur on the $(N+1)^{st}$ insertion into a random 2-3 tree with N keys is bounded by

$$0.68810+O(N^{-4.37}) \leq E[s(N)] \leq 0.45575 + 0.07745\lfloor\log_2(N+1)\rfloor+O(N^{-4.37})$$

Lemma 3.4.12. The expected number of keys in the fringe of a 2-3 tree with N keys that corresponds to the tree collection shown in Figure 3.4.2 is

$$\overline{f}(N) = \left[\sum_{i=4}^{9}\sum_{j=i}^{9}(i+j-1)\left(\frac{p_{ij}}{L_{ij}}\right) + \sum_{i=4}^{9}\sum_{j=4}^{9}\sum_{k=i}^{9}(i+j+k-1)\left(\frac{p_{ijk}}{L_{ijk}}\right)\right](N+1)$$

Proof : The above expression is obtained by observing Figure 3.4.2 and by using Eq.2.2-5. ▪

Corollary. $\overline{f}(N) = 0.92255(N+1)+O(N^{4.37})$

Theorem 3.4.13. The expected height of a 2-3 tree with N keys is bounded above by

$$\overline{h}(N) \leq \log_2(N+1)-0.69054$$

Proof : Similar to the proof of Theorem 3.2.12. ▪

It is important to note that the values for $\overline{n}(N)$, $\overline{s}(N)$, $E[s(N)]$, *Pr{j splits}*, and *Pr{j or more splits}* for one and two level analysis can be obtained using the 147 probabilities we obtained from the three level analysis. Among other verifications, this is what we did in order to check the results of this section.

3.5. 2-3 Trees with Overflow Technique

The overflow technique was first presented by Bayer and McCreight (1972, p.183). The idea, when applied to 2-3 trees, is the following: Assume that a key must be inserted in a node already full because it contains 2 keys; instead of splitting it, we look first at its brother node on the right. If this node has only one key, a simple rearrangement of keys makes splitting unnecessary. If the right brother node is also full (or does not exist), we can look at its left brother in essentially the same way.

The object of this section is to present a second order analysis of the 2-3 tree insertion algorithm using an overflow technique that is simpler than the one proposed by Bayer and McCreight. In order to make the analysis possible we restrict the overflow technique to the lowest level, and moreover, we only split a node when an insertion is performed in a full node and its closest brother is also full; otherwise a rearrangement of keys is performed and the closest non-full brother node will accommodate one more key. Figure 3.5.1 shows the two level tree collection, and Table 3.5.1 shows its corresponding transition matrix.

Using Eq. 2.2-3 we obtain

$p_1 = 1584/15949$
$p_2 = 2970/15949$
$p_3 = 3600/15949$
$p_4 = 3150/15949$
$p_5 = 2000/15949$
$p_6 = 800/15949$
$p_7 = 45/389$

Since the eigenvalues of H are $0, -6.81 \pm 5.96i, -8.51 \pm 2.97i, -9.0$, and -14.37, the asymptotic values of p(N) obtained from Eq. 2.2-4 are approximated to the $O(N^{-6.81})$.

Lemma 3.2.1 and expression Eq. 2.2-5 lead to the following theorem:

Theorem 3.5.1. The expected number of nodes in a random 2-3 tree with N keys is bounded by

$$\left\{\left(3+\frac{1}{2}\right)\left[\sum_{i=1}^{3}\frac{p_i}{L_i}\right]+\left(4+\frac{1}{2}\right)\left[\sum_{i=4}^{7}\frac{p_i}{L_i}\right]\right\}(N+1)-\frac{1}{2} \le \bar{n}(N) \le \left\{4\left[\sum_{i=1}^{3}\frac{p_i}{L_i}\right]+5\left[\sum_{i=4}^{7}\frac{p_i}{L_i}\right]\right\}(N+1)-1$$

Corollary.

$$\frac{20175}{31898}+\frac{2113}{15949N}+O(N^{-6.81}) \le \frac{\bar{n}(N)}{N} \le \frac{11385}{15949}-\frac{4564}{15949N}+O(N^{-6.81})$$

To five place decimals we have

$$0.63248+\frac{0.13248}{N}+O(N^{-6.81}) \le \frac{\bar{n}(N)}{N} \le 0.71384-\frac{0.28616}{N}+O(N^{-6.81}),$$

which should be compared to the

$$0.72683+\frac{0.22683}{N}+O(N^{-4.37}) \le \frac{\bar{n}(N)}{N} \le 0.76556-\frac{0.23444}{N}+O(N^{-4.37}),$$

which are the third order approximation of $\frac{\bar{n}(N)}{N}$ for the non-overflow algorithm.

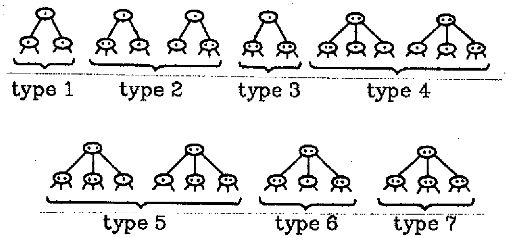


Figure 3.5.1 Tree collection of 2-3 trees of height 2 using overflow technique

$$\begin{bmatrix}
-5 & & & & 4\times3/8 & & 4\times6/9 \\
5 & -6 & & & 5\times3/8 & & 10\times3/9 \\
 & 6 & -7 & & & & 6\times6/9 \\
 & & 7 & -8 & & & \\
 & & & 8\times5/7 & -9 & & \\
 & & & 8\times2/7 & & -9 & \\
 & & & & 9\times5/8 & 9 & -10
\end{bmatrix}$$

Table 3.5.1 Transition matrix corresponding to the tree collection of 2-3 trees of height 2 shown in Figure 3.5.1

Lemma 3.5.2. The probabilities that no split, 1 split, and 2 or more splits occur on the $(N+1)^{st}$ insertion into a 2-3 tree with N keys using an overflow technique are, respectively

$$(a)\ \ Pr\{0\ splits\} = \frac{9754}{15949} + O(N^{-6.81})$$

$$(b)\ \ Pr\{1\ split\} = \frac{3600}{15949} + O(N^{-6.81})$$

$$(c)\ \ Pr\{2\ or\ more\ splits\} = \frac{2595}{15949} + O(N^{-6.81})$$

Proof : The proofs of (a), (b), and (c) are similar to those of Lemmas 3.2.3, 3.3.2, and 3.3.3, respectively. ▪

Lemma 3.2.5 leads to the following theorem:

Theorem 3.5.3. The expected number of splits in a random 2-3 tree with N keys using an overflow technique is bounded by

$$\frac{20175}{31898} + \frac{2113}{15949N} - \frac{\lfloor \log_2(N+1) \rfloor}{N} + O(N^{-6.81}) \le \bar{s}(N) \le$$

$$\frac{11385}{15949} - \frac{4564}{15949N} - \frac{\lceil \log_3(N+1) \rceil}{N} + O(N^{-6.81})$$

To five place decimals we have

$$0.63248 + \frac{0.13248}{N} - \frac{\lfloor \log_2(N+1) \rfloor}{N} + O(N^{-6.81}) \le \bar{s}(N) \le$$

$$0.71384 - \frac{0.28616}{N} - \frac{\lceil \log_3(N+1) \rceil}{N} + O(N^{-6.81}) ,$$

which should be compared to the bounds

$$0.72683 + \frac{0.22683}{N} - \frac{\lfloor \log_2(N+1) \rfloor}{N} + O(N^{-4.37}) \le \bar{s}(N) \le$$

$$0.76556 - \frac{0.23444}{N} - \frac{\lceil \log_3(N+1) \rceil}{N} + O(N^{-4.37}) ,$$

which are the third order approximation of $\bar{s}(N)$ for the non-overflow algorithm.

Lemma 3.5.4. A lower bound on the expected number of splits that will occur on the $(N+1)^{st}$ insertion into a random 2-3 tree with N keys using an overflow technique is

$$E[s(N)] \geq Pr\{1\ split\} + 2Pr\{2\ or\ more\ splits\}$$

Proof : Similar to the proof of Lemma 3.2.3. ▪

Lemma 3.5.5. An upper bound on the expected number of splits that will occur on the $(N+1)^{st}$ insertion into a random 2-3 tree with N keys using an overflow technique is

$$E[s(N)] \leq Pr\{1\ split\} + Pr\{2\ or\ more\ splits\} \lfloor \log_2(N+1) \rfloor$$

Proof : Similar to the proof of Lemma 3.2.8. ▪

Lemmas 3.5.4 and 3.5.5 lead to the following theorem:

Theorem 3.5.6. The expected number of splits that will occur on the $(N+1)^{st}$ insertion into a random 2-3 tree with N keys using an overflow technique is bounded by

$$\frac{8790}{15949} + O(N^{-6.81}) \leq E[s(N)] \leq \frac{3600}{15949} + \frac{2595}{15949} \lfloor \log_2(N+1) \rfloor + O(N^{-6.81})$$

To five place decimals we have

$$0.55113 + O(N^{-6.81}) \leq E[s(N)] \leq 0.22572 + 0.16270 \lfloor \log_2(N+1) \rfloor + O(N^{-6.81}) .$$

3.6. Concurrency of Operations on 2-3 Trees

A 2-3 tree node is insertion safe if it contains only one key. When considering concurrency of operations on 2-3 trees, one possible technique to permit simultaneous access to the tree by more than one process is to lock the deepest safe node on the insertion path. (A safe node is the deepest one in a particular insertion path if there are no safe nodes below it.) The object of this section is to give a probability distribution of the depth of the deepest safe node.

3.6.1. Deepest Safe Node in 2-3 Trees with Normal Insertion Algorithm

In the following lemma we use the p's obtained in Sections 3.2, 3.3, and 3.4.

Lemma 3.6.1.1. The probabilities that the deepest safe node is located at the 1^{st}, the 2^{nd}, and the 3^{rd} lowest level, and above the 3^{rd} lowest level of a 2-3 tree with N keys are, respectively (*a*) $Pr\{dsn\ at\ 1^{st}\ lowest\ level\} = \frac{4}{7}$

(*b*) $Pr\{dsn\ at\ 2^{nd}\ lowest\ level\} = \frac{13788}{55937} + O(N^{-6.55})$

(*c*) $Pr\{dsn\ at\ 3^{rd}\ lowest\ level\} = 0.10462 + O(N^{-4.37})$

(*d*) $Pr\{dsn\ above\ 3^{rd}\ lowest\ level\} = 0.07745 + O(N^{-4.37})$

Proof : It is not difficult to see that the probability that the deepest safe node is located at j^{th} ($j \geq 1$) lowest level is equal to the probability that exactly $j-1$ splits occur on the $(N+1)^{st}$ random insertion (see Lemmas 3.2.3, 3.3.2, 3.4.6, and 3.4.7 for the proof of items (a), (b), (c), and (d) respectively). ▪

From Lemma 3.6.1.1, item (d), we can see that in only 8% of the time the deepest safe node is above the 3^{rd} lowest level of a random 2-3 tree. In other words by locking the deepest safe node on the insertion path we lock at most height 3 fringe subtrees 92% of the time.

3.6.2. Deepest Safe Node in 2-3 Trees with Overflow Technique

In the following lemma we use the p's obtained in Section 3.5.

Lemma 3.6.2.1. The probabilities that the deepest safe node is located at the 1^{st} and the 2^{nd} lowest level, and above the 2^{nd} lowest level of a 2-3 tree with N keys using an overflow technique are, respectively

(*a*) $Pr\{dsn\ at\ 1^{st}\ lowest\ level\} = \frac{9754}{15949} + O(N^{-6.81})$

(*b*) $Pr\{dsn\ at\ 2^{nd}\ lowest\ level\} = \frac{3600}{15949} + O(N^{-6.81})$

(*c*) $Pr\{dsn\ above\ 2^{nd}\ lowest\ level\} = \frac{2595}{15949} + O(N^{-6.81})$

Proof : Similar to the proof of Lemma 3.6.1.1 (see Lemma 3.5.2 in Section 3.5 for the proof of items (*a*), (*b*), and (*c*)) ▪

2.7. Higher Order Analysis

Yao (1978, p. 165) predicted that an analysis for the k lowest levels would be difficult to carry out for $k=3$ and virtually impossible to carry out for $k \geq 4$. However, if we apply the same technique used to obtain the three level tree collection with 147 types then it might be possible to think about fourth order analysis.

In order to obtain a four level tree collection we define a 20 types three level tree collection containing trees with 8, 9, 10, ... , 27 leaves, in a way similar to the way we obtained the 6 types two level tree collection shown in Figure 3.4.1. This three level tree collection can be used to obtain a four level tree collection with 4410 types, by considering combinations of the 20 types as subtrees of nodes with one and two keys. Thus the fourth order analysis will require the solution of a 4410×4410 linear system.

Again if we apply the same technique it is possible to obtain a five level tree collection with 148137 types, which is practically impossible to handle nowadays. Table 3.7.1 shows the sizes of the tree collections used by Yao, Brown, and in this thesis, for various levels of analysis.

Analysis	Brown	Yao	Ours
First order	2	2	2
Second order	9	7	6
Third order	978	224	147
Fourth order	3.3×10^9	5.67×10^6	4410
Fifth order	-	$\approx 9.11 \times 10^{19}$	148137

Table 3.7.1 Sizes of the tree collections used by Brown (1979a,p.57), Yao (1978, p.165), and in this thesis

Finally, we want to say something more about the expected height of 2-3 trees.

Lemma 3.7.1. Let l_j indicate the number of nodes at the j^{th} $(j \geq 1)$ lowest level of a random 2-3 tree with N keys.Then

(*i*) $l_1 = N+1$

(*ii*) $l_2 = \left[\frac{p_1}{L_1} + \frac{p_2}{L_2}\right](N+1)$

(*iii*) $l_3 = \left[\sum_{i=1}^{7} \frac{p_i}{L_i}\right](N+1)$

(*iv*) $l_4 = \left[\sum_{i=4}^{9}\sum_{j=i}^{9} \frac{p_{ij}}{L_{ij}} + \sum_{i=4}^{9}\sum_{j=4}^{9}\sum_{k=i}^{9} \frac{p_{ijk}}{L_{ijk}}\right](N+1)$

Proof : Case (*i*) is obvious: the number of external nodes is equal to the number of keys in the tree plus one. In cases (*ii*) to (*iv*) we just count the number of trees in the fringe that corresponds to the tree collection of Figure 3.2.1, Figure 3.3.1, and Figure 3.4.2, respectively. ▪

Corollary.

(*i*) $l_1 = N+1$

(*ii*) $l_2 = \frac{3}{7}(N+1)$ for $N \geq 6$

(*iii*) $l_3 = \frac{1455}{7991}(N+1) + O(N^{-6.55})$

(*iv*) $l_4 = 0.07745(N+1) + O(N^{-4.37})$

Table 3.7.2 shows the ratio of the expected numbers of nodes at two consecutive levels for the four lowest levels of a random 2-3 tree with N keys.

Lowest level	$l_i (1 \leq i \leq 4)$	$\frac{l_j}{l_{j-1}} (2 \leq j \leq 4)$
4th	$0.07745(N+1)$	0.42538
3rd	$\frac{1455}{7991}(N+1)$	0.42485
2nd	$\frac{3}{7}(N+1)$	0.42857
1st	$N+1$	

Table 3.7.2 Ratio of the expected numbers of nodes at two consecutive levels

Assuming that this ratio is approximately the same for the other levels of the tree, we derive the following conjecture:

Conjecture 3.7.2. The expected height of a random 2-3 tree with N keys is

$$\bar{h}(N) \approx \log_{7/3}(N+1)$$

4. AN ANALYSIS OF B-TREES

4.1. Motivation

B-trees were presented by Bayer and McCreight (1972) as a dictionary structure primarily for secondary storage. In a B-tree of order m each node has between $m+1$ and $2m+1$ subtrees, and all external nodes appear at the same level. According to this, a 2-3 tree is a B-tree of order $m=1$. The definition of B-trees, the insertion algorithm, and the transformations (called splits) necessary to keep the tree balanced are presented in Appendix C.

The interest in B-trees has grown in the recent years to the extent that Comer (1979) referred to them as ubiquitous. In spite of this interest, no analytical results were known about the performance of B-trees prior to the pioneering work of Yao (1978). The main result in Yao's paper is related to the expected number of internal nodes in B-trees, when the assumptions of the model are met. For completeness we derive again Yao's results using the technique presented in Chapter 2.

Many questions of interest about B-trees were left open. Some of these questions are:

(i) As in the 2-3 tree case, the operation of splitting an overfull node is the costliest one when considering insertions. We present the first partial analysis of this measure for B-trees;

(ii) The overflow technique, as described for 2-3 trees, is also studied for the general B-tree. We present an analysis of the expected number of internal nodes and the expected number of splitting operations for B-trees using a special case of the overflow technique presented by Bayer and McCreight (1972, p.183);

(iii) Considering the fact that B-trees are primarily designed for secondary storage, the concurrency of operations on B-trees is of crucial importance. Our analysis enables us to provide some results on the expected depth of the deepest safe node in an insertion path.

The complexity measures used in this chapter are exactly the same complexity measures defined for 2-3 trees in Section 3.1. They are written in this chapter with a subscript m. The only new complexity measure is:

Let $\overline{n}_m(N)/[N/(2m)]$ be the *storage used* by a B-tree T of order m, where $N/(2m)$ represents the number of nodes when all the nodes of T contain $2m$ keys.

In section 4.2 we shall derive exact values for $Pr\{0\ splits\}_m$, $Pr\{1\ or\ more\ splits\}_m$, and bounds on $\bar{n}_m(N)$ by considering the lowest level of a random N key B-tree of order m obtained using the insertion algorithm described above. In section 4.3 we shall derive exact values for $Pr\{0\ splits\}_m$, $Pr\{1\ split\}_m$, $Pr\{1\ or\ more\ splits\}_m$, $Pr\{2\ or\ more\ splits\}_m$, and bounds on $\bar{n}_m(N)$ for an insertion algorithm for B-trees that uses an overflow technique, by considering the lowest two levels of a random N key B-tree of order m. In Section 4.4 we shall derive exact values for $Pr\{dsn\ at\ 1^{st}\ lowest\ level\}_m$ and $Pr\{dsn\ above\ 1^{st}\ lowest\ level\}_m$ for the normal insertion algorithm, and $Pr\{dsn\ at\ 1^{st}\ lowest\ level\}_m$, $Pr\{dsn\ at\ 2^{nd}\ lowest\ level\}_m$, and $Pr\{dsn\ above\ 2^{nd}\ lowest\ level\}_m$ for the insertion algorithm using an overflow technique.

Table 4.1.1 shows the summary of the results related to B-trees using the normal insertion algorithm, and Table 4.1.2 shows the summary of the results related to B-trees using an overflow technique.

4.2. First Order Analysis

The tree collection of B-trees of order m and height 1 contains $m+1$ types. Figure 4.2.1 shows the one level tree collection of B-trees of order m=3.

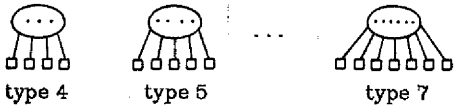


Figure 4.2.1 Tree collection of B-trees of order m=3 and height 1

The transition matrix H corresponding to the one level tree collection of B-trees of order m is

$$H = \begin{bmatrix} -(m+2) & & & & & 2(m+1) \\ m+2 & -(m+3) & & & & \\ & m+3 & -(m+4) & & & \\ & & \ddots & \ddots & & \\ & & & \ddots & \ddots & \\ & & & & 2m+1 & -(2m+2) \end{bmatrix}$$

Let H_n denote the Harmonic numbers, $H_n = \sum_{i=1}^{n} \frac{1}{i}$, for $n \geq 1$. From Eq. 2.2-3 we have $(H)p(N) = 0$, and therefore

$$p_{m+1} = \frac{1}{(m+2)\left[H_{2m+2} - H_{m+1}\right]}$$

	First order analysis ($N \to \infty$)
$\frac{\overline{n}_m(N)}{N}$	$\left[\frac{1}{(2\ln 2)m} + \left(\frac{1}{8\ln 2} - \frac{1}{4}\right)\frac{1}{(\ln 2)m^2} + O(m^{-3})\right.,$ $\left.\frac{1}{(2\ln 2)m} + \frac{1}{8(\ln 2)^2 m^2} + O(m^{-3})\right]$
$Pr\{0\ splits\}_m$	$1 - \frac{1}{(2\ln 2)m} - \left(\frac{1}{8\ln 2} - \frac{1}{2}\right)\frac{1}{(\ln 2)m^2} + O(m^{-3})$
$Pr\{1\ or\ more\ splits\}_m$	$\frac{1}{(2\ln 2)m} + \left(\frac{1}{8\ln 2} - \frac{1}{2}\right)\frac{1}{(\ln 2)m^2} + O(m^{-3})$
Storage used	$\frac{1}{\ln 2} + O(m^{-1})$
$Pr\{dsn\ at\ 1^{st}\ l.\ level\}_m$	$1 - \frac{1}{(2\ln 2)m} - \left(\frac{1}{8\ln 2} - \frac{1}{2}\right)\frac{1}{(\ln 2)m^2} + O(m-3)$
$Pr\{dsn\ above\ 1^{st}\ l.\ level\}_m$	$\frac{1}{(2\ln 2)m} + \left(\frac{1}{8\ln 2} - \frac{1}{2}\right)\frac{1}{(\ln 2)m^2} + O(m^{-3})$

Table 4.1.1 Summary of the B-tree results

	Second order analysis ($N \to \infty$)
$\frac{\overline{n}_m(N)}{N}$	$\left[\frac{1}{2m} + \left(\frac{3}{8\ln 2} - \frac{1}{4}\right)\frac{1}{m^2} + O(m^{-3})\right.,$ $\left.\frac{1}{2m} + \left(\frac{3}{8\ln 2} - \frac{1}{4}\right)\frac{1}{m^2} + O(m^{-3})\right]$
$Pr\{0\ splits\}_m$	$1 - \frac{1}{2m} - \left(\frac{1}{8\ln 2} - \frac{1}{4}\right)\frac{1}{m^2} + O(m^{-3})$
$Pr\{1\ split\}_m$	$\frac{1}{2m} + \left(-\frac{1}{8\ln 2} - \frac{1}{4}\right)\frac{1}{m^2} + O(m^{-3})$
$Pr\{1\ or\ more\ splits\}_m$	$\frac{1}{2m} + \left(\frac{1}{8\ln 2} - \frac{1}{4}\right)\frac{1}{m^2} + O(m^{-3})$
$Pr\{2\ or\ more\ splits\}_m$	$\frac{1}{(4\ln 2)m^2} + O(m^{-3})$
Storage used	$1 + \left(\frac{3}{4\ln 2} - \frac{1}{2}\right)\frac{1}{m} + O(m^{-2})$
$Pr\{dsn\ at\ 1^{st}\ lowest\ level\}_m$	$1 - \frac{1}{2m} - \left(\frac{1}{8\ln 2} - \frac{1}{4}\right)\frac{1}{m^2} + O(m^{-3})$
$Pr\{dsn\ at\ 2^{nd}\ lowest\ level\}_m$	$\frac{1}{2m} + \left(-\frac{1}{8\ln 2} - \frac{1}{4}\right)\frac{1}{m^2} + O(m^{-3})$
$Pr\{dsn\ above\ 2^{nd}\ lowest\ level\}_m$	$\frac{1}{(4\ln 2)m^2} + O(m^{-3})$

Table 4.1.2 Summary of the B-tree results using an overflow technique

$$p_{m+2} = \frac{1}{(m+3)\left[H_{2m+2}-H_{m+1}\right]} \tag{1}$$

$$\vdots$$

$$p_{2m+1} = \frac{1}{(2m+2)\left[H_{2m+2}-H_{m+1}\right]}$$

Lemma 4.2.1. The probability that 1 or more splits occur on the $(N+1)^{st}$ random insertion into a B-tree of order m with N keys is

$$Pr\{1 \text{ or more splits}\}_m = \frac{1}{(2m+2)\left[H_{2m+2}-H_{m+1}\right]}$$

Proof : In the lowest level of a B-tree of order m a split occurs when an insertion happens in a node with $2m$ keys, and such nodes correspond to the type $2m+1$ of the tree collection of B-trees of order m and height 1. Thus, $Pr\{1 \text{ or more splits}\}_m = p_{2m+1}$ ▪

Lemma 4.2.2. The probability that no split occurs on the $(N+1)^{st}$ random insertion into a B-tree of order m with N keys is

$$Pr\{0 \text{ splits}\}_m = 1-\frac{1}{(2m+2)\left[H_{2m+2}-H_{m+1}\right]}$$

Proof : $Pr\{0 \text{ splits}\}_m = 1-Pr\{1 \text{ or more splits}\}_m$. ▪

It is well known that $H_m = \ln m + \gamma + \frac{1}{2m} - \frac{1}{12m^2} + O(m^{-4})$, where $\gamma = 0.57721...$ is Euler's constant (Knuth, 1968, § 1.2.7). Then

Corollary.

$$Pr\{1 \text{ or more splits}\}_m = \frac{1}{(2\ln 2)\, m} + \left[\frac{1}{8\ln 2} - \frac{1}{2}\right]\frac{1}{(\ln 2)\, m^2} + O(m^{-3})$$

Lemma 4.2.3. Let nl_m be the number of nodes at level l of an order m B-tree. Then the number of nodes above the level l, nal_m, is bounded by

$$\frac{nl_m-1}{2m} \le nal_m \le \frac{nl_m-1}{m}$$

Proof : Consider the level l as being the $N+1$ leaves of a B-tree with N keys. (Each leaf represents a node.) The *minimum* and the *maximum* number of nodes above the level l is obtained when each node above the level l contains $2m$ and m keys respectively. (That is $2m \times nal_m = nl_m - 1$ and $m \times nal_m = nl_m - 1$ respectively.) ▪

Lemma 4.2.3 and Eq. 2.2-5 lead to the following theorem:

Theorem 4.2.4. The expected number of nodes in a random B-tree of order m with N keys is bounded by

$$\left(1+\frac{1}{2m}\right)\left[\sum_{i=m+1}^{2m+1}\frac{p_i}{L_i}\right](N+1)-\frac{1}{2} \le \bar{n}_m(N) \le \left(1+\frac{1}{m}\right)\left[\sum_{i=m+1}^{2m+1}\frac{p_i}{L_i}\right](N+1)-1$$

Corollary. $$\left[\frac{2m+1}{(4m^2+4m)(H_{2m+2}-H_{m+1})}\right]\left(1-\frac{1}{N}\right)-\frac{1}{2N}+O(N^{Re(\lambda_2)}) \le \frac{\bar{n}_m(N)}{N} \le$$

$$\left[\frac{1}{2m(H_{2m+2}-H_{m+1})}\right]\left(1-\frac{1}{N}\right)-\frac{1}{N}+O(N^{Re(\lambda_2)})$$

where $Re(\lambda_2) < 0$.

Corollary. $$\frac{1}{(2\ln 2)m}+\left[\frac{1}{8\ln 2}-\frac{1}{4}\right]\frac{1}{(\ln 2)m^2}+O(m^{-3}) \le \frac{\bar{n}_m(N)}{N} \le$$

$$\frac{1}{(2\ln 2)m}+\frac{1}{8(\ln 2)^2 m^2}+O(m^{-3})$$

Corollary. Storage used = $\frac{1}{\ln 2}+O(m^{-1})$

The values obtained for the storage used (cf. definition of storage used in Section 4.1) are between 1 and 2. The value 1 corresponds to the B-tree with all nodes with $2m$ keys, and the value 2 corresponds to the B-tree with all nodes with m keys. Yao (1978) used a different measure. He defined *storage utilisation* as $[N/(2m)]/\bar{n}_m(N)$, where $N/(2m)$ represents the number of nodes when all the nodes contain $2m$ keys. However, it is known that, in general,

$$E\left(\frac{1}{X}\right) \ne \frac{1}{E(X)}$$

for a random variable X. Furthermore, by using the Kantorovich inequality (see Clausing, 1982, pp. 314-330) we have

$$1 \leq E(X) \times E\left(\frac{1}{X}\right) \leq \frac{9}{8} \qquad (2)$$

Then

Corollary. The *storage utilisation* for a random B-tree of order m with N keys is bounded by

$$\ln 2 + O(m^{-1}) \leq \textit{storage utilisation} \leq \frac{9}{8}\ln 2 + O(m^{-1})$$

4.3. B-trees with Overflow Technique

In this section we present a second order analysis of the B-tree insertion algorithm using the following overflow technique. We restrict the overflow technique to the lowest level, and moreover, we only split a node when an insertion is performed in a full node and all its brothers are also full; otherwise a rearrangement of keys is performed and the closest non-full brother node will accommodate one more key.

Any tree collection of B-trees of order m using the overflow technique described above contains $(m+1)(2m+1)$ types. Figure 4.3.1 shows the transition diagram corresponding to the two level tree collection of B-trees of order m=2. The transition matrix H corresponding to the two level tree collection of B-trees of order m using the overflow technique described above is shown in Table 4.3.1.

In order to obtain the vector $p(N)$ from Eq. 2.2-3, we make $p_{(2m+1)(2m+1)}=1$ † and solve for all the other p's. After this we normalise the p's by dividing each one by their sum. Then

$$p_{(2m+1)(2m+1)} = 1$$

$$p_{(2m)+2m(2m+1)} = \frac{(2m+1)(2m+1)+1}{(2m+1)(2m+1)} = \frac{4m^2+4m+2}{(2m+1)(2m+1)}$$

$$p_{(2m-1)+2m(2m+1)} = \frac{4m^2+4m+2}{(2m)+2m(2m+1)}$$

† $p_{(2m+1)(2m+1)}$ means $p_{(2m+1)+(2m+1)+...+(2m+1)}$, where $(2m+1)$ appear $2m+1$ times. Applying this notation to the B-tree of order m=2 shown in Figure 4.3.1, p_{55555} is equivalent to $p_{(2m+1)(2m+1)}$, p_{335} is equivalent to $p_{(m+1)+(m+1)+(m-1)(2m+1)}$, etc.

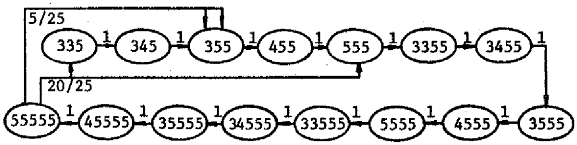


Fig. 4.3.1 Transition diagram representing the two level tree collection for B-trees of order m=2 using overflow technique (e.g. type 335 corresponds to the height 2 type tree containing a root node with 3 descendants, the first one with 3 leaves, the second one also with 3 leaves, and the third one with 5 leaves)

$$
\left[\begin{array}{ccccccccccc}
-[(m+1)+(m+1)+(m-1)(2m+1)]-1 & & & & & & & & & & 2m/(2m+1)\times[(m+1)+(m+1)+(m-1)(2m+1)] \\
 & \ddots & & & & & & & & & \\
 & & -[(m+1)+2m+(m-1)(2m+1)]-1 & & & & & & & & \\
 & & [(m+1)+m(2m+1)] & -[(m+1)+m(2m+1)]-1 & & & & & & & 2/(2m+1)[(m+1)+m(2m+1)] \\
 & & & & \ddots & & & & & & \\
 & & & & & -[2m+m(2m+1)]-1 & & & & & \\
 & & & & & [(m+1)\times(2m+1] & -[(m+1)\times(2m+1)]-1 & & & & 2m/(2m+1)\times[(m+1)(2m+1)] \\
 & & & & & & [(m+1)+(m+1)+m(2m+1)] & & & & \\
 & & & & & & & \ddots & & & \\
 & & & & & & & & [2m+2m(2m+1)] & -[2m+2m(2m+1)]-1 & \\
 & & & & & & & & & (2m+1)(2m+1) & -(2m+1)(2m+1)-1
\end{array}\right]
$$

Table 4.3.1 Transition matrix corresponding to the tree collection of B-trees of order m and height 2 using an overflow technique

$$\vdots$$

$$p_{2m(2m+1)} = \frac{4m^2+4m+2}{2m(2m+1)+1}$$

$$p_{(2m)+(2m-1)(2m+1)} = \frac{4m^2+4m+2}{2m(2m+1)} \qquad (1)$$

$$\vdots$$

$$p_{(m+1)(2m+1)} = \frac{4m^2+4m+2}{(m+1)(2m+1)+1}$$

$$p_{(2m)+m(2m+1)} = \frac{1}{(m+1)(2m+1)}\left[4m^2+4m+2-\frac{2m}{2m+1}(m+1)(2m+1)\right]$$

$$= \frac{2m^2+2m+2}{(m+1)(2m+1)}$$

$$\vdots$$

$$p_{(m+1)+m(2m+1)} = \frac{2m^2+2m+2}{(m+2)+m(2m+1)}$$

$$p_{(m+1)+(2m)+(m-1)(2m+1)} = \frac{1}{(m+1)+m(2m+1)}\times$$

$$\left[2m^2+2m+2-\frac{2}{2m+1}(m+1+m(2m+1))\right]$$

$$= \frac{(4m^3+2m^2+2m)/(2m+1)}{(m+1)+m(2m+1)}$$

$$\vdots$$

$$p_{(m+1)+(m+2)+(m-1)(2m+1)} = \frac{(4m^3+2m^2+2m)/(2m+1)}{(m+1)+(m+3)+(m-1)(2m+1)}$$

$$p_{(m+1)+(m+1)+(m-1)(2m+1)} = \frac{(4m^3+2m^2+2m)/(2m+1)}{(m+1)+(m+2)+(m-1)(2m+1)}$$

Let S be the sum of all p's above. Then

$$S = \left(\frac{4m^3+2m^2+2m}{2m+1}\right)\left[H_{2m^2+2m+1}-H_{2m^2+m+1}\right] +$$

$$(2m^2+2m+2)\left[H_{2m^2+3m+1}-H_{2m^2+2m+1}\right] + \qquad (2)$$

$$\left((2m+1)(2m+1)+1\right)\left[H_{4m^2+4m+2}-H_{2m^2+3m+1}\right]$$

To obtain the final probabilities all the above p's have to be divided by S.

Let $\psi(Z)$ be the Psi function $\psi(Z)=\frac{\Gamma'(Z)}{\Gamma(Z)}$ (Abramowitz and Stegun, 1972, § 6.3.1).

Lemma 4.3.1. The probability that 1 or more splits occur on the $(N+1)^{st}$ random insertion into an N-key random B-tree of order m using an overflow technique is

$$Pr\{1 \textit{ or more splits}\}_m =$$

$$\frac{1}{S}\left[\frac{(2m+1)(2m+1)+1}{2m+1}\right]\left[\psi\left(2m+2+\frac{1}{2m+1}\right)-\psi\left(m+1+\frac{1}{2m+1}\right)\right]$$

where S is as defined in Eq.(2).

Proof : $Pr\{1 \textit{ or more splits}\}_m = p_{(m+1)(2m+1)}+p_{(m+2)(2m+1)}+\cdots+p_{(2m+1)(2m+1)}$

$$= \frac{1}{S}\left[\frac{(2m+1)(2m+1)+1}{2m+1}\sum_{i=1}^{m+1}\frac{1}{(m+i)+\frac{1}{2m+1}}\right]$$

where $\sum_{i=1}^{m+1}\frac{1}{(m+i)+\frac{1}{2m+1}} = \psi\left(2m+2+\frac{1}{2m+1}\right)-\psi\left(m+1+\frac{1}{2m+1}\right)$ ▪

It is well known (Abramowitz and Stegun, 1972, § 6.3.18) that

$$\psi(m) = \ln m-\frac{1}{2m}-\frac{1}{12m^2}+O(m^{-4}) \cdot$$

Corollary. $Pr\{1 \textit{ or more splits}\}_m = \frac{1}{2m}+\left[\frac{1}{8\ln 2}-\frac{1}{4}\right]\frac{1}{m^2}+O(m^{-3})$

Lemma 4.3.2. The probability that 1 split occurs on the $(N+1)^{st}$ random insertion into an N-key random B-tree of order m using an overflow technique is

$$Pr\{1 \textit{ split}\}_m = \frac{1}{S}\left[\frac{(2m+1)(2m+1)+1}{2m+1}\right]\left[\psi\left(2m+1+\frac{1}{2m+1}\right)-\psi\left(m+1+\frac{1}{2m+1}\right)\right]$$

where S is as defined in Eq.(2).

Proof : The only difference from the proof of lemma 4.3.1 is that

$Pr\{1\ split\}_m = p_{(m+1)(2m+1)} + p_{(m+2)(2m+1)} + \cdots + p_{(2m)+(2m)(2m+1)}$ ■

Corollary. $Pr\{1\ split\}_m = \frac{1}{2m} + \left[-\frac{1}{8\ln 2} - \frac{1}{4} \right] \frac{1}{m^2} + O(m^{-3})$

Lemma 4.3.4. The probability that 2 or more splits occur on the $(N+1)^{st}$ random insertion into an N-key random B-tree of order m using an overflow technique is

$$Pr\{2\ or\ more\ splits\}_m = \frac{1}{S}$$

where S is as defined in Eq.(2).

Proof : $Pr\{2\ or\ more\ splits\}_m = Pr\{1\ or\ more\ splits\}_m - Pr\{1\ splits\}_m$

$$= \frac{1}{S} \left[\frac{(2m+1)(2m+1)+1}{2m+1} \right] \left[\psi\left(2m+2+\frac{1}{2m+1}\right) - \psi\left(2m+1+\frac{1}{2m+1}\right) \right] = \frac{1}{S} \text{ ■}$$

Corollary. $Pr\{2\ or\ more\ splits\}_m = \frac{1}{(4\ln 2)\, m^2} + O(m^{-3})$

Lemma 4.3.5. The probability that no split occurs on the $(N+1)^{st}$ random insertion into an N-key random B-tree of order m using an overflow technique is

$Pr\{0\ splits\}_m =$

$$1 - \frac{1}{S} \left[\frac{(2m+1)(2m+1)+1}{2m+1} \right] \left[\psi\left(2m+2+\frac{1}{2m+1}\right) - \psi\left(m+1+\frac{1}{2m+1}\right) \right]$$

where S is as defined in Eq.(2).

Proof : $Pr\{0\ splits\}_m = 1 - Pr\{1\ or\ more\ splits\}_m$ ■

Corollary. $Pr\{0\ splits\}_m = 1 - \frac{1}{2m} - [\frac{1}{8\ln 2} - \frac{1}{4}] \frac{1}{m^2} + O(m^{-3})$

Lemma 4.2.3 and Eq.2.2-5 lead to the following theorem:

Theorem 4.3.6. The expected number of nodes in a random B-tree of order m with N keys using an overflow technique is bounded by

$$A(2m)(N+1) - \frac{1}{2} \le \bar{n}_m(N) \le A(m)(N+1) - 1$$

where

$$A(X) = \frac{1}{S}\Big\{\Big(m+2+\frac{1}{X}\Big)\Big[\frac{p_{(m+1)+(m+1)+(m-1)(2m+1)}}{(m+1)+(m+1)+(m-1)(2m+1)}+$$

$$\frac{p_{(m+1)+(m+2)+(m-1)(2m+1)}}{(m+1)+(m+2)+(m-1)(2m+1)}+\cdots+\frac{p_{(m+1)(2m+1)}}{(m+1)(2m+1)}\Big]+$$

$$\Big(m+3+\frac{1}{X}\Big)\Big[\frac{p_{(m+1)+(m+1)+m(2m+1)}}{(m+1)+(m+1)+m(2m+1)}+\frac{p_{(m+1)+(m+2)+m(2m+1)}}{(m+1)+(m+2)+m(2m+1)}+\cdots+$$

$$\frac{p_{(m+2)(2m+1)}}{(m+2)(2m+1)}\Big]+\cdots+\Big(2m+2+\frac{1}{X}\Big)\Big[\frac{p_{(m+1)+(m+1)+(2m-1)(2m+1)}}{(m+1)+(m+1)+(2m-1)(2m+1)}+$$

$$\frac{p_{(m+1)+(m+2)+(2m-1)(2m+1)}}{(m+1)+(m+2)+(2m-1)(2m+1)}+\cdots+\frac{p_{(2m+1)(2m+1)}}{(2m+1)(2m+1)}\Big]\Big\}$$

and S is as defined in Eq.(2).

Substituting Eq.(1) in the expression of Theorem 4.3.6 gives:

Corollary.

$$B(2m)\Big(1-\frac{1}{N}\Big)-\frac{1}{2N}+O(N^{Re(\lambda_2)}) \le \frac{\overline{n}_m(N)}{N} \le$$

$$B(m)\Big(1-\frac{1}{N}\Big)-\frac{1}{N}+O(N^{Re(\lambda_2)})\ ,\ \ Re(\lambda_2)<0$$

where

$$B(X) = \frac{1}{S}\Big\{\Big(m+2+\frac{1}{X}\Big)\Big[\Big(\frac{4m^3+2m^2+2m}{2m+1}\Big)\times$$

$$\Big(\frac{1}{(m+1)+(m+1)+(m-1)(2m+1)}-\frac{1}{(m+1)+m(2m+1)}\Big)+$$

$$\Big(2m^2+2m+2\Big)\Big(\frac{1}{(m+1)+m(2m+1)}-\frac{1}{(m+1)(2m+1)}\Big)+$$

$$\Big(4m^2+4m+2\Big)\Big(\frac{1}{(m+1)(2m+1)}\times\frac{1}{(m+1)(2m+1)+1}\Big)\Big]+$$

$$\Big(4m^2+4m+2\Big)\Big[\frac{m+3+\frac{1}{X}}{(m+1)+(m+1)+m(2m+1)}-\frac{2m+2+\frac{1}{X}}{(2m+1)(2m+1)+1}+$$

$$\frac{\psi\left(2m+1+\frac{1}{2m+1}\right)-\psi\left(m+2+\frac{1}{2m+1}\right)}{2m+1}\Bigg]\Bigg]\Bigg\}$$

or

$$B(X) = \frac{1}{S}\left\{\frac{1}{X}+\frac{8m^2+10m+6}{2m^2+3m+2}+\right.$$

$$\left.\frac{4m^2+4m+2}{2m+1}\left[\psi\left(2m+1+\frac{1}{2m+1}\right)-\psi\left(m+2+\frac{1}{2m+1}\right)\right]\right\}$$

and S is as defined in Eq.(2).

Corollary.

$$\frac{1}{2m}+\left[\frac{3}{8\ln 2}-\frac{1}{4}\right]\frac{1}{m^2}+\left[-\frac{9}{32\ln 2}+\frac{1}{8}\right]\frac{1}{m^3}+O(m^{-4}) \leq \frac{\overline{n}_m(N)}{N} \leq$$

$$\frac{1}{2m}+\left[\frac{3}{8\ln 2}-\frac{1}{4}\right]\frac{1}{m^2}+\left[-\frac{5}{32\ln 2}+\frac{1}{8}\right]\frac{1}{m^3}+O(m^{-4})$$

Corollary. Storage used $= 1+\left(\frac{3}{4\ln 2}-\frac{1}{2}\right)\frac{1}{m}+O(m^{-2})$

Corollary. The *storage utilisation* for a random B-tree of order m with N keys using an overflow technique is bounded by

$$1-\left(\frac{3}{4\ln 2}-\frac{1}{2}\right)\frac{1}{m}+O(m^{-2}) \leq \textit{storage utilisation} \leq \frac{9}{8}-\left(\frac{3}{4\ln 2}-\frac{1}{2}\right)\frac{9}{8m}+O(m^{-2})$$

Proof : The above bounds are obtained by using Eq.4.2-2 and the result of the previous corollary. ▪

4.4. Concurrency of Operations on B-trees

A node of a B-tree of order m is insertion safe if it contains fewer than $2m$ keys. A safe node is the deepest one in a particular insertion path if there are no safe nodes below it. The object of this section is to derive probabilities related to the depth of the deepest safe node.

4.4.1. Deepest Safe Node in B-trees with Normal Insertion Algorithm

Lemma 4.4.1.1. The probabilities that the deepest safe node is located at the 1st lowest level and above the 1st lowest level of an N-key random B-tree of order m are, respectively

$$(a)\ Pr\{dsn\ at\ 1^{st}\ lowest\ level\}_m = 1 - \frac{1}{(2m+2)\left[H_{2m+2}-H_{m+1}\right]}$$

$$(b)\ Pr\{dsn\ above\ 1^{st}\ lowest\ level\}_m = \frac{1}{(2m+2)\left[H_{2m+2}-H_{m+1}\right]}$$

Proof : Similar to the proof of Lemma 3.6.1.1. ▪

Corollary.

(a) $Pr\{dsn\ at\ 1^{st}\ lowest\ level\}_m =$

$$1 - \frac{1}{(2\ln 2)\ m} - \left(\frac{1}{8\ln 2} - \frac{1}{2}\right)\frac{1}{(\ln 2)\ m^2} + O(m^{-3})$$

(b) $Pr\{dsn\ above\ 1^{st}\ lowest\ level\}_m =$

$$\frac{1}{(2\ln 2)\ m} + \left(\frac{1}{8\ln 2} - \frac{1}{2}\right)\frac{1}{(\ln 2)\ m^2} + O(m^{-3})$$

This analysis shows that complicated solutions for the use of concurrency of operations on B-trees are rarely of benefit, since the solution analysed in this thesis will lock height 1 fringe subtrees most of the time.

4.4.2. Deepest Safe Node in B-trees with Overflow Technique

Lemma 4.4.2.1. The probabilities that the deepest safe node is located at the 1st and the 2nd lowest level, and above the 2nd lowest level of an N-key random B-tree of order m using an overflow technique are, respectively

(a) $Pr\{dsn\ at\ 1^{st}\ lowest\ level\}_m =$

$$1 - \frac{1}{S}\left[\frac{(2m+1)(2m+1)+1}{2m+1}\right]\left[\psi\left(2m+2+\frac{1}{2m+1}\right) - \psi\left(m+1+\frac{1}{2m+1}\right)\right]$$

(b) $Pr\{dsn\ at\ 2^{nd}\ lowest\ level\}_m =$

$$\frac{1}{S}\left[\frac{(2m+1)(2m+1)+1}{2m+1}\right]\left[\psi\left(2m+1+\frac{1}{2m+1}\right)-\psi\left(m+1+\frac{1}{2m+1}\right)\right]$$

(c) $Pr\{dsn\ above\ 2^{nd}\ lowest\ level\}_m = \frac{1}{S}$

where S is as defined in Eq.4.3-2.

Proof : Similar to the proof of Lemma 3.6.1.1. ▪

Corollary.

(a) $Pr\{dsn\ at\ 1^{st}\ lowest\ level\}_m = 1-\frac{1}{2m}-\left[\frac{1}{8\ln 2}-\frac{1}{4}\right]\frac{1}{m^2}+O(m^{-3})$

(b) $Pr\{dsn\ at\ 2^{nd}\ lowest\ level\}_m = \frac{1}{2m}+\left[-\frac{1}{8\ln 2}-\frac{1}{4}\right]\frac{1}{m^2}+O(m^{-3})$

(c) $Pr\{dsn\ above\ 2^{nd}\ lowest\ level\}_m = \frac{1}{(4\ln 2)\ m^2}+O(m^{-3})$

5. AN ANALYSIS OF AVL TREES

5.1. Motivation

AVL trees were introduced by Adel'son-Vel'skii and Landis (1962). A binary search tree is AVL if the height of the subtrees at each node differ by at most one. The description of the insertion algorithm and the transformations (called rotations) necessary to keep the tree balanced are presented in Appendix A.

Bayer (1972) showed that the class of AVL trees is a proper subclass of symmetric binary B-trees, the object of the following chapter. The first analytical results on the expected behaviour of AVL trees were obtained by Brown (1979a), and an improvement on Brown's results was obtained by Mehlhorn (1979a). The main complexity measure studied by Brown and the only one studied by Mehlhorn is the expected number of balanced nodes in an AVL tree randomly generated.

The use of larger AVL tree collections represents a complex problem. An inherent difficulty posed by the transformations necessary to maintain the AVL trees balanced makes its fringe analysis quite difficult. (cf. Section 5.3.) In Section 5.4 we present a technique to cope with this difficulty which permits us to obtain bounds on the expected number of balanced nodes and the expected number of rotations per insertion.

We now define certain complexity measures:

(i) Let $\bar{b}(N)$ be the expected number of balanced nodes in an AVL tree after the random insertion of N keys into the initially empty tree;

(ii) Let $r(N)$ be the expected number of rotations required during the insertion of the $(N+1)^{st}$ key into a random AVL tree with N keys;

(iii) Let $Pr\{no\ rotation\}$ be the probability that no rotation occurs during the $(N+1)^{st}$ random insertion into a random AVL tree with N keys;

(iv) Let $m(N)$ be the maximum number of rotations that may occur outside the fringe of an AVL tree during the insertion of the $(N+1)^{st}$ key into a random AVL tree with N keys;

(v) Let $\bar{u}(N)$ be the expected number of unbalanced nodes in an AVL tree after the random insertion of N keys into the initially empty tree;

(vi) Let $\bar{f}(N)$ be the expected number of nodes in the fringe of an AVL tree after the random insertion of N keys into the initially empty tree.

In Section 5.2 we present a new closed AVL tree collection which permits an improvement in the lower bound on the expected number of balanced nodes. In Section 5.3 we study weakly-closed AVL tree collections. In Section 5.4 we present a technique to deal with weakly-closed AVL tree collections. In Section 5.5 we present larger weakly-closed AVL tree collections and discuss the problems involved in their analyses.

Table 5.1.1 shows the summary of the results related to AVL trees.

Tree Collection		$\bar{J}(N)$	$r(N)$	$\frac{\bar{b}(N)}{N}$
Size	Characteristic			
2	closed	$0.57N$ for $N \geq 6$	$[0.29, 0.86]$ for $N \geq 6$	$[0.48+0.48/N, 0.86-0.14/N]$ for $N \geq 6$
3	closed ambiguous	$0.66N$ for $N \geq 6$	$[0.29, 0.86]$ for $N \geq 6$	$[0.51+0.51/N, 0.86-0.14/N]$ for $N \geq 6$
4 †	weakly-closed ambiguous	$0.69N$	$[0.29, 0.81]$	$[0.51+0.51/N, 0.81-0.19/N]$

† Results are approximated to $O(N^{-5})$

Table 5.1.1 Summary of AVL tree results

5.2. Closed AVL Tree Collections

The only previously known closed tree collection for AVL trees is the one composed of trees with three leaves or less. This tree collection is studied in Section 5.2.1. In Section 5.2.2 we present a new closed tree collection for AVL trees composed of trees with four leaves or less.

5.2.1. Tree Collection of AVL Trees with Three Leaves or Less

The tree collection of AVL trees with three leaves or less is shown in Figure 5.2.1.1. Brown(1979a) proved that this tree collection is closed, obtained bounds on the expected number of balanced nodes, and gave a lower bound on the expected number of rotations. For completeness we derive again the results obtained by Brown and present also an upper bound on the expected number of rotations.

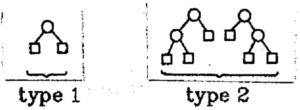


Fig. 5.2.1.1 Tree collection of AVL trees with three leaves or fewer

For the AVL tree collection shown in Figure 5.2.1.1 $H = \begin{pmatrix} -3 & 4 \\ 3 & -4 \end{pmatrix}$. From Eq.2.2-3 we have $Hp(N) = 0$, and therefore $p_1(\infty) = 4/7$, and $p_2(\infty) = 3/7$. Since the eigenvalues of H are 0 and -7, we observe that $p_1(N) = 4/7$, and $p_2(N) = 3/7$, for $N \geq 6$. (As before $p_1(N)$ and $p_2(N)$ are written as p_1 and p_2, respectively.)

Lemma 5.2.1.1. The expected number of rotations in a random AVL tree with N keys is bounded above by

(i) $r(N) = 1 - Pr\{no\ rotation\}$

and

(ii) $r(N) \leq r(N)$ in the fringe $+m(N)$

Proof : For case (i) it is known that the maximum number of rotations per insertion in an AVL tree is 1. For case (ii) $r(N)$ must be less than or equal to the number of rotations per insertion in the fringe plus all possible rotations per insertion that may occur outside the fringe. ▪

Theorem 5.2.1.2. The expected number of rotations in a random AVL tree with N keys is bounded by

(i) $\frac{2}{3}p_2 \leq r(N) \leq 1 - \frac{1}{3}p_2$ for $N \geq 1$

and

(ii) $\frac{2}{3}p_2 \leq r(N) \leq \frac{2}{3}p_2 + p_1$ for $N \geq 1$

Proof : The left hand side of (i) and (ii) are obtained by observing Figure 5.2.1.1. The right hand side of (i) and (ii) are obtained by using Lemma 5.2.1.1. ▪

Corollary. $\frac{2}{7} \leq r(N) \leq \frac{6}{7}$ for $N \geq 6$

Lemma 5.2.1.3. The expected number of single rotations ($sr(N)$) in a random AVL tree with N keys is bounded by

$$\frac{1}{3}p_2 \le sr(N) \le 1-\frac{1}{3}p_2 \quad \text{for } N\ge 1$$

Proof : The above expression can be obtained by observing Figure 5.2.1.1 and by using Lemma 5.2.1.1. ▪

Corollary. $\frac{1}{7} \le sr(N) \le \frac{6}{7}$ for $N\ge 6$

Lemma 5.2.1.4. The expected number of double rotations ($dr(N)$) in a random AVL tree with N keys is bounded by

$$\frac{1}{3}p_2 \le dr(N) \le 1-\frac{1}{3}p_2 \quad \text{for } N\ge 1$$

Proof : Similar to the proof of Lemma 5.2.1.3. ▪

Corollary. $\frac{1}{7} \le dr(N) \le \frac{6}{7}$ for $N\ge 6$

Lemma 5.2.1.5. The expected number of nodes in the fringe of an AVL tree with N keys that corresponds to the tree collection of Figure 5.2.1.1 is

$$\bar{f}(N) = \left[\frac{p_1}{L_1}+2\frac{p_2}{L_2}\right](N+1) \quad \text{for } N\ge 1$$

Proof : From Eq.2.2-5 we have $\bar{f}(N) = A_1(N)+2A_2(N)$. ▪

Corollary. $\bar{f}(N) = \frac{4}{7}N+\frac{4}{7}$ for $N\ge 6$

Lemma 5.2.1.6. The expected number of balanced nodes in a random AVL tree with N keys is bounded above by

(*i*) $\bar{b}(N) = N-\bar{u}(N)$ for $N\ge 1$

and

(*ii*) $\bar{b}(N) \le \bar{b}(N)$ in the fringe + $[N-\bar{f}(N)]$ for $N\ge 1$

Proof : For case (i) $\bar{b}(N)+\bar{u}(N) = N$. For case (ii) $\bar{b}(N)$ must be less than or equal to the number of balanced nodes in the fringe plus all nodes outside the fringe. ▪

Theorem 5.2.1.7. The expected number of balanced nodes in a random AVL tree with N keys is bounded by

$$\left[\frac{p_1}{L_1}+\frac{p_2}{L_2}\right](N+1) \le \bar{b}(N) \le N-\frac{p_2}{L_2}(N+1) \quad \text{for } N\ge 1$$

Proof : The left hand side is obtained by observing Figure 5.2.1.1 and by using Eq.2.2-5. The right hand side is obtained by using Lemma 5.2.1.6, by observing Figure 5.2.1.1, and by using Eq.2.2-5. ▪

Corollary. $\frac{3}{7}+\frac{3}{7N}\le\frac{\bar{b}(N)}{N}\le\frac{6}{7}-\frac{1}{7N}$ for $N\ge 6$

Brown (1979a, p.40) showed that an improvement on the lower bound of the result of Theorem 5.2.5 can be obtained by observing that, when the number of type 1 trees is greater than the number of type 2 trees, then at least $(\frac{p_1}{L_1}-\frac{p_2}{L_2})(N+1)/3$ balanced nodes lie outside the fringe. Thus

$$\bar{b}(N) \ge \left[\frac{p_1}{L_1}+\frac{p_2}{L_2}\right](N+1)+\frac{1}{3}\left[\frac{p_1}{L_1}-\frac{p_2}{L_2}\right](N+1) \quad \text{for } N\ge 1$$

or

$$\bar{b}(N) \ge \frac{10}{21}N+\frac{10}{21}$$

5.2.2. Tree Collection of AVL Trees with Four Leaves or Less

To improve the results obtained in the previous section we need bigger tree collections. A tree collection with three types is shown in Figure 5.2.2.1. The first step necessary to perform the analysis is to show that the AVL tree collection of Figure 5.2.2.1 is closed. (cf. Definition 2.3.2.)

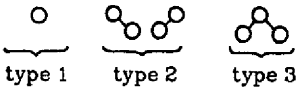
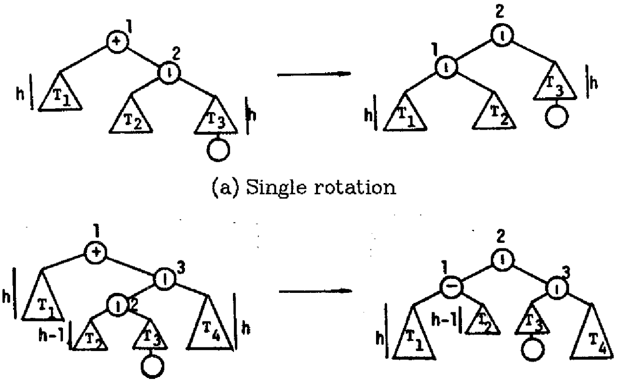


Fig. 5.2.2.1 Tree collection of AVL trees with four leaves or fewer

Theorem 5.2.2.1. The AVL tree collection shown in Figure 5.2.2.1 is closed.

Proof : An insertion into the type 1 tree always leads to a type 2 tree, and an insertion into the type 2 tree always leads to a type 3 tree. An insertion into the type 3 tree may cause a transformation higher in the tree, since the root of a type 3 tree is balanced. By inspecting Figure 5.2.2.2 we can see that a transformation has no effect on the nodes which are outside the transformed subtree. Furthermore, if the fringe of the transformed subtree is entirely contained in the subtrees T_1, T_2, T_3 and T_4 of Figure 5.2.2.2(b) then the transformation has no effect on the composition of the fringe. (i.e. T_1, T_2, T_3 and T_4 are moved without change by the transformation.)

However, there are six cases in which the fringe of the transformed subtree is moved with change by the transformation, as shown in Figure 5.2.2.3. In all six cases the number of type 3 trees decreases by one and the number of type 1 and type 2 trees increases by one. Note that each one of the three transformed trees shown in Figure 5.2.2.3(a and b) contains one 3-nodes subtree which is not considered as a type 3 tree, but as a subtree composed of two type 1 trees. ▪



(a) Single rotation

(b) Double rotation

Fig.5.2.2.2 AVL tree transformations (symmetric transformations occur)

Theorem 5.2.2.1 says that the transitions in the tree collection of Figure 5.2.2.1 are well-defined, so the theorems of Chapter 2 can be applied. Thus

$$H = \begin{bmatrix} -3 & 0 & 2 \\ 3 & -4 & 3 \\ 0 & 4 & -5 \end{bmatrix}.$$

From Eq.2.2.-3 we have $Hp(N) = 0$, and therefore $p_1(\infty) = 8/35$, $p_2(\infty) = 15/35$, and $p_3(\infty) = 12/35$. Since the eigenvalues of H are 0, −5, and −7, we observe that $p_1(N) = 8/35$, $p_2(N) = 15/35$, and $p_3(N) = 12/35$, for $N \geq 6$.

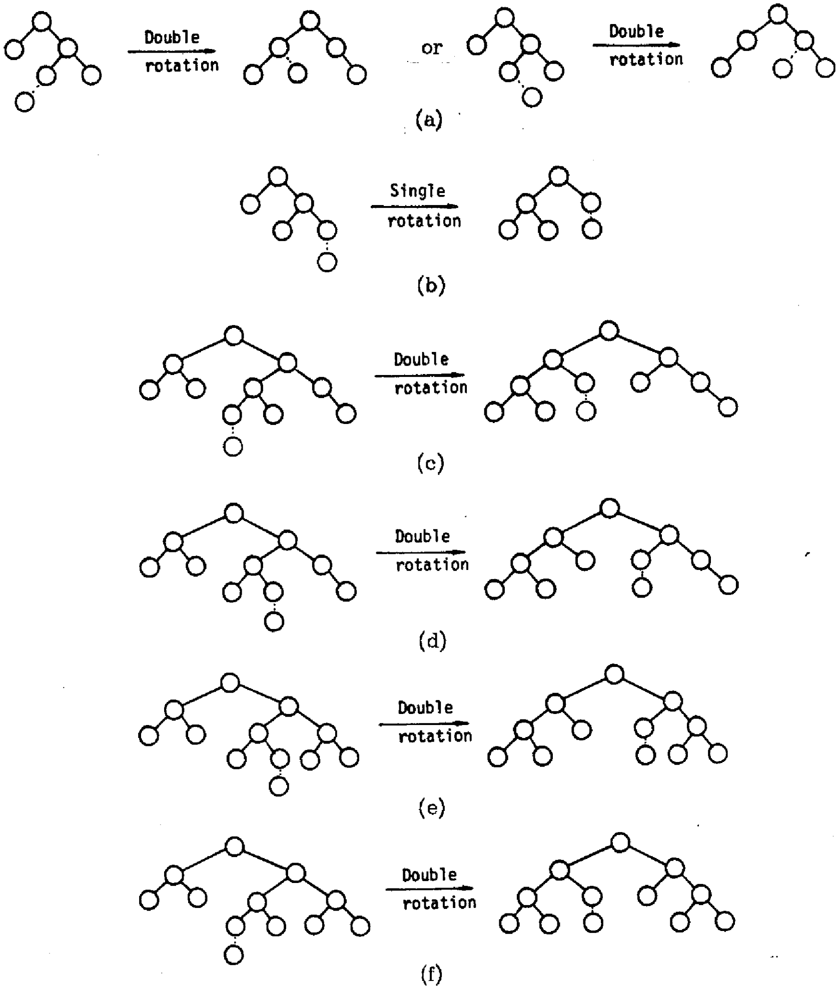


Fig. 5.2.2.3 Cases in which the fringe of the transformed subtree is moved with change (symmetric transformations occur)

Theorem 5.2.2.2. The expected number of balanced nodes in a random AVL tree with N keys is bounded by

$$\left(\frac{p_1}{L_1}+\frac{p_2}{L_2}+3\frac{p_3}{L_3}\right)(N+1) \leq \bar{b}(N) \leq N-\frac{p_2}{L_2}(N+1)$$

Proof : The left hand side is obtained by observing Figure 5.2.2.1 and by using Eq.2.2-5. The right hand side is obtained by using Lemma 5.2.1.6, by observing Figure 5.2.2.1, and by using Eq.2.2-5. ▪

Corollary. $\frac{18}{35}+\frac{18}{35N} \leq \frac{\bar{b}(N)}{N} \leq \frac{6}{7}-\frac{1}{7N}$, for $N \geq 6$

Lemma 5.2.2.3. The expected number of nodes in the fringe of an AVL tree with N keys that corresponds to the tree collection of Figure 5.2.2.1 is

$$\bar{f}(N) = \left(\frac{p_1}{L_1}+2\frac{p_2}{L_2}+3\frac{p_3}{L_3}\right)(N+1)$$

Proof : From Eq.2.2-5 we have $\bar{f}(N) = A_1(N)+2A_2(N)+3A_3(N)$. ▪

Corollary. $\bar{f}(N) = \frac{23}{35}(N+1)$, for $N \geq 6$

The results on the expected number of rotations derived in the previous section cannot be improved by the use of this tree collection. This tree collection corresponds to the tree collection used in the previous section augmented by the type 3 tree, and the type 3 tree does not contain any information about rotations.

5.3. Weakly-closed AVL Tree Collections

If the effect of an insertion on the composition of the fringe is determined not only by the subtree of the fringe where the insertion is performed, but by some other transformation that may happen outside the fringe, then the tree collection is weakly-closed (Definition 2.3.2). We will show that the tree collection of AVL trees with five or less leaves shown in Figure 5.3.1 is not closed.

Lemma 5.3.1. If the trees shown in Figure 5.3.1 form the fringe of a random AVL tree with N keys and $N \rightarrow \infty$, then an insertion into a leaf of a type 3 tree (i) decreases by one the number of type 3 trees and increases by one the number

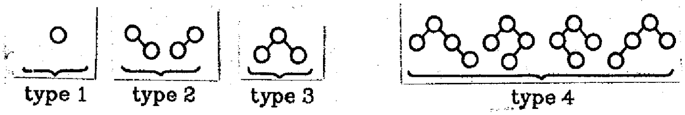


Fig. 5.3.1 Tree collection of AVL trees with five leaves or fewer

of type 4 trees; or (ii) decreases by one the number of type 3 trees and increases by one the number of type 1 and type 2 trees; or (iii) decreases by one the number of type 1 trees and increases by one the number of type 2 trees.

Proof : We will denote the probability of the second of these alternatives by s_N, the probability of the third by t_N, and the probability of the first by $1-s_N-t_N$.

Case (i): This case is obvious: the type 3 tree is transformed in a type 4 tree. If there is no transformation higher in the tree then this is the transition.

Cases $(ii)$and(iii): If a transformation takes place higher in the tree, which is possible since the root of a type 3 tree is balanced, side-effects on the composition of the fringe will occur. By inspecting Figure 5.2.2.2 we can see that a transformation has no effect on the nodes which are outside of the transformed subtree. Furthermore, if the fringe of the transformed subtree is entirely contained in the subtrees T_1, T_2, T_3, and T_4 of Figure 5.2.2.2 then the transformation has no effect on the composition of the fringe, because T_1, T_2, T_3, and T_4 are moved without change by the transformation.

Figure 5.3.2 shows the five cases in which the fringe of the tree to be transformed is moved with change by the transformation, and this change produces side-effects on the composition of the fringe.

In cases (a) and (b) of Figure 5.3.2 the number of type 1 trees decreases by one and the number of type 2 trees increases by one. This case occurs with an unknown probability we call t_N.

In cases (c), (d), and (e) of Figure 5.3.2 the number of type 3 trees decreases by one and the number of type 1 and type 2 trees increases by one. This case occurs with an unknown probability we call s_N. ▪

Lemma 5.3.1 tells us that any AVL tree collection that contains the types 3 and 4 shown in Figure 5.3.1 is not closed. (i.e. it is weakly-closed.) In fact it is not difficult to show that every AVL tree type that contains more than one internal node and has its root node balanced suffers from the same type of misbehaviour that occurs with type 3 (i.e. consider the AVL tree with six nodes). Consequently an AVL tree collection that contains a tree type with the root node balanced and

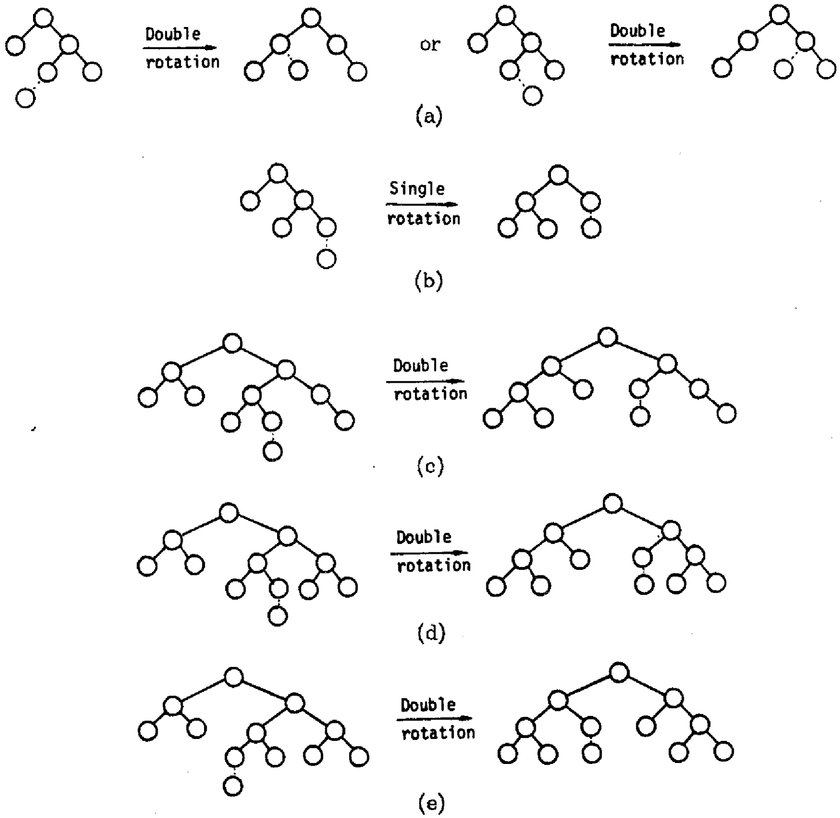


Fig. 5.3.2 Cases in which transformations change the fringe
(symmetric transformations occur)

has more than three types is weakly-closed.

We know from Lemma 5.3.1 that an insertion into the type 3 tree shown in Figure 5.3.1, when it belongs to a fringe of an AVL tree with N keys, produces a transition that is not well defined; the transition depends on two unknown probabilities s_N and t_N which also depend on N. First of all let us give a more precise meaning to s_N and t_N. Let I be the expected number of leaves in an AVL tree with N keys such that an insertion in one of the I leaves causes one of the three transformations shown in Figure 5.3.2(c, d, and e). In a similar way let J be the expected number of leaves such that an insertion in one of the J leaves causes one of the two transformations shown in Figure 5.3.2(a and b). Thus

$$s_N = \frac{I}{N+1}$$

and

$$t_N = \frac{J}{N+1} \ .$$

Although the probabilities s_N and t_N are unknown they cannot assume arbitrary values between 0 and 1.

Lemma 5.3.2. The probability t_N is bounded by $\quad 0 \le t_N \le \frac{1}{3}$

Proof :

Case (*i*): Let q_1 be the probability that an insertion is made into any of the subtrees of Figure 5.3.3. Let $1-q_1$ be the probability that an insertion is made into any of the subtrees of Figure 5.3.4.

Consider a N-key AVL tree with all subtrees in the fringe being of the type shown in Figure 5.3.3, the type shown in Figure 5.3.4, or a mixture of the two. Let us consider one tree of Figure 5.3.3 and one tree of Figure 5.3.4, as shown in Figure 5.3.5. The arcs show the probabilities of two possible transitions. Then

$$\frac{3(1-q_1)}{5N} = \frac{q_1}{N}$$

or $q_1 = \frac{3}{8}$.

If $\Delta t_N \uparrow$ is the increment in t_N then

$$\Delta t_N \uparrow \le \frac{3-8q_1}{5N} \ ,$$

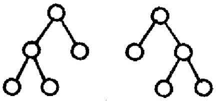


Fig. 5.3.3

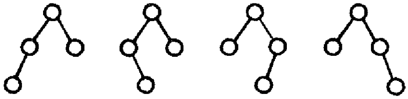


Fig. 5.3.4

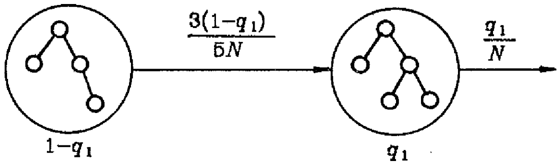


Fig.5.3.5

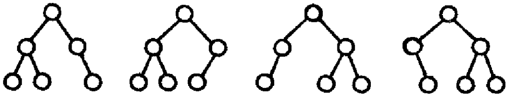


Fig. 5.3.6

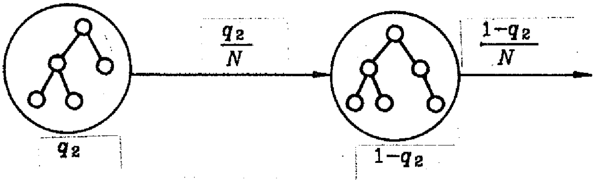


Fig. 5.3.7

so t_N cannot increment beyond $q_1 = 3/8$ and $q_1 = 3/8$ is the maximum value for q_1. By definition $t_N = \frac{2q_1}{3}$, which gives

$$0 \le t_N \le \frac{1}{4} \cdot$$

Case (*ii*): Let q_2 be the probability that an insertion is made into any of the subtrees of Figure 5.3.3. Let $1-q_2$ be the probability that an insertion is made into any of the subtrees of Figure 5.3.6.

Consider a N-key AVL tree with all subtrees in the fringe being of the type shown in Figure 5.3.3, or the type shown in Figure 5.3.6, or a mixture of both. Let us consider one tree of Figure 5.3.3 and one tree of Figure 5.3.6, as shown in Figure 5.3.7. The arcs show the probabilities of two possible transitions. Then

$$\frac{q_2}{N} = \frac{1-q_2}{N}$$

or $q_2 = \frac{1}{2}$, where $q_2 = 1/2$ is the maximum value for q_2. By definition $t_N = \frac{2q_2}{3}$, which gives

$$0 \le t_N \le \frac{1}{3} \quad ■$$

Lemma 5.3.3. The probability s_N is bounded by $0 \le s_N \le \frac{1}{6}$

Proof :

Case (*i*): Let r_1 be the probability that an insertion is made into a tree of the type shown in Figure 5.3.8. Let $1-r_1$ be the probability that an insertion happens into any tree of the types shown in Figure 5.3.9. Notice that an insertion into a tree of Figure 5.3.9 gives a tree of Figure 5.3.8 with probability 3/11. Furthermore, it is not difficult to see that the trees of Figure 5.3.9 represent the main source of subtrees that under a new insertion are transformed into a tree of the type shown in Figure 5.3.8. (The trees of Figure 5.3.8 may be obtained from other sources by performing rotations on bigger subtrees, but the probabilities in these cases are smaller than the probabilities related to the trees shown in Figure 5.3.9.)

Consider a N-key AVL tree with all subtrees in the fringe being of the type shown in Figure 5.3.8, type shown in Figure 5.3.9, or a mixture of the two. Let us con-

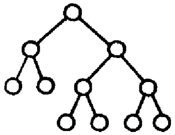


Fig. 5.3.8 (Symmetric cases occur)

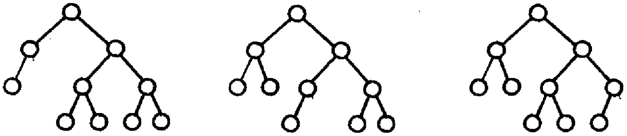


Fig. 5.3.9 (Symmetric cases occur)

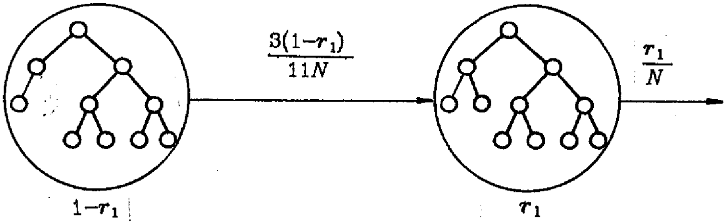


Fig. 5.3.10 .

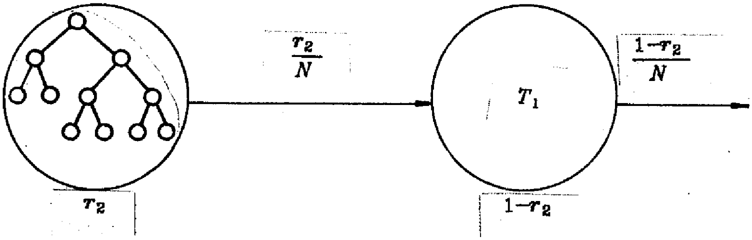


Fig. 5.3.11 (T_1 is a tree obtained from an insertion into a tree of Figure 5.3.8)

sider one tree of Figure 5.3.8 and one tree of Figure 5.3.9, as shown in Figure 5.3.10. The arcs show the probabilities of two possible transitions. Then

$$\frac{3(1-r_1)}{11N} = \frac{r_1}{N}$$

or $r_1 = \frac{3}{14}$.

If $\Delta s_N \uparrow$ is the increment in s_N then

$$\Delta s_N \uparrow \leq \frac{3-14r_1}{11n} .$$

so s_N cannot increment beyond $r_1 = 3/14$ and $r_1 = 3/14$ is the maximum value for r_1. By definition $s_N = \frac{r_1}{3}$, which gives

$$0 \leq s_N \leq \frac{1}{14} .$$

Case (*ii*): Let r_2 be the probability that an insertion is made into any of the trees of Figure 5.3.8. Let $1-r_2$ be the probability that an insertion is made in one of the trees one may obtain by inserting into a tree of Figure 5.3.8.

Consider a N-key AVL tree with all subtrees in the fringe being of the type shown in Figure 5.3.8, the type one may obtain by inserting into a tree of Figure 5.3.8, or a mixture of the two. Let us consider the two trees shown in Figure 5.3.11. The arcs show the probabilities of two possible transitions. Then

$$\frac{r_2}{N} = \frac{1-r_2}{N}$$

or $r_2 = \frac{1}{2}$, where $r_2 = 1/2$ is the maximum value for r_2. By definition $s_N = \frac{r_2}{3}$, which gives

$$0 \leq s_N \leq \frac{1}{6} .$$

In the following section we present a technique to deal with weakly-closed tree collections, in which unknown probabilities appear in the transition matrix.

5.4. Coping with Weakly-closed AVL Tree Collections

Consider again the tree collection of AVL trees with five or less leaves shown in Figure 5.3.1. As shown in Section 5.3 this tree collection is weakly-closed. In this section we present a technique to deal with weakly-closed AVL tree collections.

From the results of Lemma 5.3.1 we can examine the insertion process and obtain

$$H(s_N,t_N) = \begin{bmatrix} -3 & 0 & 2(s_N-t_N) & 6/5 \\ 3 & -4 & 3(s_N+t_N) & 12/5 \\ 0 & 4 & -5+4t_N & 12/5 \\ 0 & 0 & 5(1-s_N-t_N) & -6 \end{bmatrix} \qquad (1)$$

where s_N and t_N depend on N. Figure 5.4.1 shows how the values of column three of $H(s_N,t_N)$ in Eq.(1) were obtained.

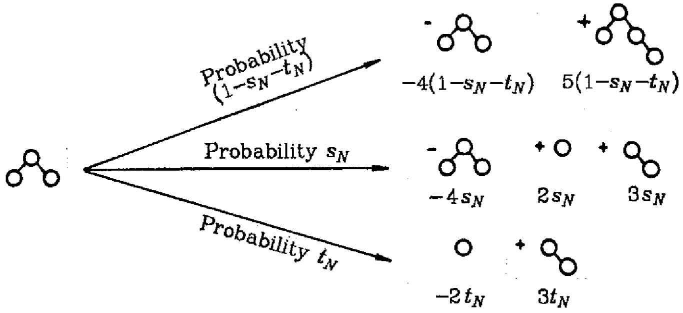


Fig. 5.4.1

The characteristic polynomial of $H(s_N,t_N)$ is

$$\det\big(H(s_N,t_N)-\lambda I\big) = \lambda^4+(18-4t_N)\,\lambda^3+(107-52t_N)\,\lambda^2+(210-168t_N)\lambda,$$

the eigenvalues are $\lambda_1 = 0$, $\lambda_2 = -5+4t_N$, $\lambda_3 = -6$, $\lambda_4 = -7$,

and the eigenvectors are

$$x_1(s_N,t_N) = \frac{1}{35-28t_N}\begin{bmatrix} 4(1+s_N-3t_N) \\ 3(3+2s_N-2t_N) \\ 12 \\ 10(1-s_N-t_N \end{bmatrix}, \text{ considering } p_1+p_2+p_3+p_4=1$$

$$x_2(s_N,t_N) = \frac{1}{1+4t_N}\begin{pmatrix} -3+2(s_N-t_N) \\ 3(-1+s_N+t_N) \\ 1+4t_N \\ 5(1-s_N-t_N \end{pmatrix}$$

$$x_3 = \frac{1}{5}\begin{pmatrix} 2 \\ 3 \\ 0 \\ -5 \end{pmatrix}$$

$$x_4(s_N,t_N) = \frac{1}{2}\begin{pmatrix} 3-4s_N-2t_N \\ 5-6s_N-8t_N \\ 2 \\ 10(-1+s_N+t_N) \end{pmatrix}$$

where $x_1(s_N,t_N)$, $x_2(s_N,t_N)$, x_3, and $x_4(s_N,t_N)$ correspond to the eigenvalues λ_1, $\lambda_2(s_N,t_N)$, λ_3, and λ_4 respectively.

If the matrix $H = H(s_N,t_N)$ is independent of N, has one eigenvalue equal to zero, and the others have negative real part, then $p(N)$, the solution of Eq.2.2-2 converges to the solution of $Hq = 0$ (cf. Theorem 2.2.3). However the matrix $H(s_N,t_N)$ in Eq.(1) contains the unknown probabilities s_N and t_N that depend on N, and consequently $H(s_N,t_N)$ depends on N. For this reason we have to prove the following result:

Theorem 5.4.1. Let $p(N)$ be defined by

$$p(N) = \left[I + \frac{H(s_N,t_N)}{N+1}\right]p(N-1), \quad N>4 \tag{2}$$

and $p(4) = (0,0,0,1)^T$ (an AVL tree with four nodes is the type 4 tree shown in Figure 5.3.1, with probability 1), where $\{s_N,t_N\}_{N>4}$ is a given sequence of probabilities. Then there exists a sequence $\{s'_N,t'_N\}_{N>4}$, such that $p(N)$ converges to $q(N)$, the solution of

$$H(s'_N,t'_N)q(N) = 0 \cdot \tag{3}$$

Proof : We will construct a sequence $\{s'_N,t'_N\}$ and, in each iteration, express $p(N)$ in the basis of eigenvectors of $H(s'_N,t'_N)$. In this basis

$$p(N) = \alpha_1(N)x_1(s_N,t_N)+\alpha_2(N)x_2(s_N,t_N)+\alpha_3(N)x_3+\alpha_4(N)x_4(s_N,t_N)$$

where $\alpha_1(N) = 1$. Because $p_1+p_2+p_3+p_4 = 1$ the components of $x_1(s_N,t_N)$ add to 1. The initial vector is

$$\begin{bmatrix}\alpha_1(4)\\ \alpha_2(4)\\ \alpha_3(4)\\ \alpha_4(4)\end{bmatrix} = E^{-1}(s_4',t_4')p(4)$$

where $E^{-1}(s_4',t_4')$ is the matrix that produces the spectral decomposition for $H(s_4',t_4')$. ($E(s,t)$ is the matrix of eigenvectors of $H(s,t)$.)

We want to prove that $(\alpha_1(N), \alpha_2(N), \alpha_3(N), \alpha_4(N))^T$ converges to $(1, 0, 0, 0)^T$, as $N\to\infty$.

Assume at step $N-1$ we have $(\alpha_1(N-1),\alpha_2(N-1),\alpha_3(N-1),\alpha_4(N-1))^T$ and we have already constructed $\{s_4',t_4'; \cdots ;s_{N-1}',t_{N-1}'\}$. In the next step we compute the effect of applying $(I+\frac{H(s_N,t_N)}{N+1})$ to $(\alpha_1(N-1),\alpha_2(N-1),\alpha_3(N-1),\alpha_4(N-1))^T$ and at the same time express the new probability vector in a different basis of eigenvectors.

This is equivalent to the effect of one random insertion into the tree, i.e. going from $N-1$ to N nodes. To compute this we apply $(I+\frac{H(s_N,t_N)}{N+1})$ to $x_1(s_{N-1}',t_{N-1}')$, $x_2(s_{N-1}',t_{N-1}')$, x_3, and $x_4(s_{N-1}',t_{N-1}')$, and obtain the spectral decomposition in each case. Then

$$\begin{bmatrix}\alpha_1(N)\\ \alpha_2(N)\\ \alpha_3(N)\\ \alpha_4(N)\end{bmatrix} = C(s_N,t_N;s_{N-1}',t_{N-1}')\begin{bmatrix}\alpha_1(N-1)\\ \alpha_2(N-1)\\ \alpha_3(N-1)\\ \alpha_4(N-1)\end{bmatrix}$$

where $C(s_N,t_N;s_{N-1}',t_{N-1}')$ is the matrix that operates the transformation with parameters s_N and t_N due to one insertion on the basis $x_1(s_{N-1}',t_{N-1}')$, $x_2(s_{N-1}',t_{N-1}')$, x_3, $x_4(s_{N-1}',t_{N-1}')$. Then

$$C(s_N,t_N;s_{N-1}',t_{N-1}') = \begin{bmatrix} 1 & 0 & 0 & 0 \\ C[2,1] & 1-\frac{5-4t_N}{N+1} & 0 & C[2,4] \\ C[3,1] & C[3,2] & 1-\frac{6}{N+1} & C[3,4] \\ 0 & 0 & 0 & 1-\frac{7}{N+1} \end{bmatrix}$$

where

$$C[2,1] = \frac{-48(t_N - t'_{N-1})}{7(4t'_{N-1}-5)(N+1)}$$

$$C[2,4] = \frac{4(t_N - t'_{N-1})}{N+1}$$

$$C[3,1] = \frac{60(4s'_{N-1}t_N - 5t_N - 4s_N t'_{N-1} + 5t'_{N-1} - s_N + s'_{N-1})}{7(4t'_{N-1}+1)(4t'_{N-1}-5)(N+1)}$$

$$C[3,2] = \frac{-5(4s'_{N-1}t_N - 5t_N - 4s_N t'_{N-1} + 5t'_{N-1} - s_N + s'_{N-1})}{(4t'_{N-1}+1)(N+1)}$$

$$C[3,4] = \frac{-5(4s'_{N-1}t_N - 5t_N - 4s_N t'_{N-1} + 5t'_{N-1} - s_N + s'_{N-1})}{(4t'_{N-1}+1)(N+1)}$$

At this point the new probability vector is still expressed in the basis of eigenvectors of $H(s'_{N-1}, t'_{N-1})$. Now we will change to a new basis of eigenvectors, for suitably chosen s'_N, t'_N. Let $B(s'_N, t'_N; s'_{N-1}, t'_{N-1})$ be the matrix that changes basis. In this case

$$\begin{bmatrix} \alpha_1(N) \\ \alpha_2(N) \\ \alpha_3(N) \\ \alpha_4(N) \end{bmatrix} = B(s'_N, t'_N; s'_{N-1}, t'_{N-1}) \begin{bmatrix} \alpha_1(N-1) \\ \alpha_2(N-1) \\ \alpha_3(N-1) \\ \alpha_4(N-1) \end{bmatrix}$$

Notice again that $\alpha_1(N) = \alpha_1(N-1) = 1$. Then the matrix B is

$$B(s'_N, t'_N; s'_{N-1}, t'_{N-1}) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ B[2,1] & 1 & 0 & B[2,4] \\ B[3,1] & B[3,2] & 1 & B[3,4] \\ 0 & 0 & 0 & B[4,4] \end{bmatrix}$$

where

$$B[2,1] = \frac{-48(t'_N - t'_{N-1})}{7(4t'_N-5)(4t'_{N-1}-5)}$$

$$B[2,4] = \frac{2(t'_N - t'_{N-1})}{2t'_N+1}$$

$$B[3,1] = \frac{-10(4s'_{N-1}t'_N - 5t'_N - 4s'_N t'_{N-1} + 5t'_{N-1} - s'_N + s'_{N-1})}{7(4t'_N+1)(4t'_{N-1}-5)}$$

$$B[3,2] = \frac{5(4s'_{N-1}t'_N - 5t'_N - 4s'_N t'_{N-1} + 5t'_{N-1} - s'_N + s'_{N-1})}{(4t'_N+1)(4t'_{N-1}+1)}$$

$$B[3,4] = \frac{-5(4s'_{N-1}t'_N-5t'_N-4s'_Nt'_{N-1}+5t'_{N-1}-s'_N+s'_{N-1})}{4t'_N+1}$$

$$B[4,4] = \frac{2t'_{N-1}+1}{2t'_N+1}$$

The combined effect of these two transformations is the matrix

$$A(s'_N,t'_N;s_N,t_N;s'_{N-1},t'_{N-1}) = B(s'_N,t'_N;s'_{N-1},t'_{N-1}) \times C(s_N,t_N;s'_{N-1},t'_{N-1})$$

Then

$$A(s'_N,t'_N;s_N,t_N;s'_{N-1},t'_{N-1}) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ A[2,1] & 1-\frac{5-4t_N}{N+1} & 0 & A[2,4] \\ A[3,1] & A[3,2] & 1-\frac{6}{N+1} & A[3,4] \\ 0 & 0 & 0 & A[4,4] \end{pmatrix}$$

where

$$A[2,1] = \frac{-48(t'_N-t'_{N-1})}{7(4t'_N-5)(4t'_{N-1}-5)} + \frac{-48(t_N-t'_N)}{7(4t'_{N-1}-5)(N+1)}$$

$$A[2,4] = \frac{4(t_N-t'_{N-1})}{N+1} + \left(1-\frac{7}{N+1}\right)\left(\frac{2(t'_N-t'_{N-1})}{2t'_N}\right)$$

$$A[3,1] = \frac{-10(4s'_{N-1}t'_N-5t'_N-4s'_Nt'_{N-1}+5t'_{N-1}-s'_N+s'_{N-1})}{7(4t'_N+1)(4t'_{N-1}-5)}$$

$$+\frac{-48(t_N-t'_{N-1})}{7(4t'_{N-1}-5)(N+1)}\left(\frac{5(4s'_{N-1}t'_N-5t'_N-4s'_Nt'_{N-1}+5t'_{N-1}-s'_N+s'_{N-1})}{(4t'_N+1)(4t'_{N-1}+1)}\right)$$

$$+\frac{60(4s'_{N-1}t_N-5t_N-4s_Nt'_{N-1}+5t'_{N-1}-s_N+s'_{N-1})}{7(4t'_{N-1}+1)(4t'_{N-1}-5)(N+1)}$$

$$A[3,2] = \frac{-5(4s'_{N-1}t_N-5t_N-4s_Nt'_{N-1}+5t'_{N-1}-s_N+s'_{N-1})}{(4t'_{N-1}+1)(N+1)}$$

$$+\left(1-\frac{5-4t_N}{N+1}\right)\left(\frac{5(4s'_{N-1}t'_N-5t'_N-4s'_Nt'_{N-1}+5t'_{N-1}-s'_N+s'_{N-1})}{(4t'_N+1)(4t'_{N-1}+1)}\right)$$

$$A[3,4] = \frac{-5(4s'_{N-1}t_N-5t_N-4s_Nt'_{N-1}+5t'_{N-1}-s_N+s'_{N-1})}{(4t'_{N-1}+1)(N+1)}$$

$$+\frac{4(t_N-t'_{N-1})}{N+1}\left[\frac{5(4s'_{N-1}t'_N-5t'_N-4s'_Nt'_{N-1}+5t'_{N-1}-s'_N+s'_{N-1})}{(4t'_N+1)(4t'_{N-1}+1)}\right]$$

$$+\left(1-\frac{7}{N+1}\right)\left(\frac{-5(4s'_{N-1}t'_N-5t'_N-4s'_Nt'_{N-1}+5t'_{N-1}-s'_N+s'_{N-1})}{4t'_N+1}\right)$$

$$A[4,4]=\left(1-\frac{7}{N+1}\right)\left(\frac{2t_N+1}{2t'_N}\right)$$

Let us show what happens when we go step by step. Recall that s'_N, t'_N represent the current values of the auxiliary sequence, and s_N, t_N represent the current values of the unknown probabilities in the transition matrix. Then

$$\begin{pmatrix}1\\ \alpha_2(N)\\ \alpha_3(N)\\ \alpha_4(N)\end{pmatrix}=\begin{pmatrix}\alpha_1(N)\\ \alpha_2(N)\\ \alpha_3(N)\\ \alpha_4(N)\end{pmatrix}=A(s'_N,t'_N;s_N,t_N;s'_{N-1},t'_{N-1})\times\cdots\times$$

$$A(s'_6,t'_6;s_6,t_6;s'_5,t'_5)\times A(s'_5,t'_5;s_5,t_5;s'_4,t'_4)E^{-1}(s'_4,t'_4)p(4)\,.$$

To prove that $(\alpha_1(N),\alpha_2(N),\alpha_3(N),\alpha_4(N))^T$ converges to $(1,0,0,0)^T$ it is enough to prove that

$$\lim_{N\to\infty}\prod_{k=5}^{N}H(s'_k,t'_k;s_k,t_k;s'_{k-1},t'_{k-1})=\begin{pmatrix}1&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{pmatrix}. \tag{4}$$

This would mean that

$$\begin{pmatrix}1\\ \alpha_2(N)\\ \alpha_3(N)\\ \alpha_4(N)\end{pmatrix}=\begin{pmatrix}1&0&0&0\\0&0&0&0\\0&0&0&0\\0&0&0&0\end{pmatrix}\cdot E^{-1}(s'_4,t'_4)p(4)=\begin{pmatrix}1\\0\\0\\0\end{pmatrix},$$

or that the probabilities converge to an eigenvector associated to $\lambda=0$ of $H(s'_N,t'_N)$. Notice that this is independent of $p(4)$ and of the choice of s'_4,t'_4.

The entries $A[2,1]$ and $A[3,1]$ are the critical ones for the convergence of the $\prod_N A(s'_N,t'_N;s_N,t_N;s'_{N-1},t'_{N-1})$ to the matrix (4). Let us solve $A[2,1]=0$ for t'_N. Then

$$t'_N=\frac{(N+1)t'_{N-1}+5(t_N-t'_{N-1})}{N+1+4(t_N-t'_{N-1})} \tag{5}$$

Now we substitute t_N' in $A[3,1] = 0$ and solve for s_N'. Then

$$s_N' = \frac{5(t_N - t_{N-1}') + 6(s_N - s_{N-1}') + (N+1)s_{N-1}'}{N+1+4(t_N - t_{N-1}')} \tag{6}$$

Before we go ahead to compute the $\prod\limits_N A(s_N', t_N'; s_N, t_N; s_{N-1}', t_{N-1}')$ with $A[2,1] = 0$ and $A[3,1] = 0$ we have to show that t_N' in Eq.(5) and s_N' in Eq.(6) are bounded.

Proposition 5.4.2. For $0 \leq t_{N-1}', t_N \leq \frac{1}{3}$ the value of t_N' in Eq.(5) is bounded by

$$0 \leq t_N' \leq \frac{1}{3}$$

Proof :

Case (*i*): $t_N' \geq 0$ is equivalent to

$$(N+1)t_{N-1}' + 5(t_N - t_{N-1}') \geq 0 \ ,$$

which is true.

Case (*ii*): $t_N' \leq \frac{1}{3}$ is equivalent to

$$(N+1)t_{N-1}' + 5(t_N - t_{N-1}') \leq \frac{1}{3}[N+1+4(t_N - t_{N-1}')]$$

or

$$\left[N+1-5+\frac{4}{3}\right]t_{N-1}' + \left[5-\frac{4}{3}\right]t_N \leq \frac{1}{3}(N+1)$$

The left hand side of the above expression is maximum when $t_N = t_{N-1}' = \frac{1}{3}$. Then

$$\frac{1}{3}(N+1) \leq \frac{1}{3}(N+1) \quad \blacksquare$$

Proposition 5.4.3. For $0 \leq s_N \leq \frac{1}{6}$, $-\frac{5}{14} \leq s_{N-1}' \leq \frac{4}{11}$, and $0 \leq t_{N-1}', t_N \leq \frac{1}{3}$, the value of s_N' in Eq.(6) is bounded by

$$-\frac{5}{14} \leq s_N' \leq \frac{4}{11}$$

Proof :

Case (*i*): $s'_N \geq -\frac{5}{14}$ is equivalent to

$$5(t_N - t'_{N-1}) + 6(s_N - s'_{N-1}) + (N+1)s'_{N-1} \geq -\frac{5}{14}[N+1+4(t_N - t'_{N-1})]$$

or

$$\frac{90}{14}(t_N - t'_{N-1}) + (N-5)s'_{N-1} + 6s_N \geq -\frac{5}{14}(N+1) \ .$$

The left hand side of the above expression is minimum when $t_N = 0$, $t'_{N-1} = \frac{1}{3}$, $s_N = 0$, and $s'_{N-1} = -\frac{5}{14}$. Then

$$-\frac{5}{14}(N+1) \geq -\frac{5}{14}(N+1)$$

Case (*ii*): $s'_N \leq \frac{4}{11}$ is equivalent to

$$5(t_N - t'_{N-1}) + 6(s_N - s'_{N-1}) + (N+1)s'_{N-1} \leq \frac{4}{11}[N+1+4(t_N - t'_{N-1})]$$

or

$$\frac{39}{11}(t_N - t'_{N-1}) + (N-5)s'_{N-1} + 6s_N \leq \frac{4}{11}(N+1) \ .$$

The left hand side of the above expression is maximum when $t_N = \frac{1}{3}$, $t'_{N-1} = 0$, $s'_{N-1} = \frac{4}{11}$, and $s_N = \frac{1}{6}$. Then

$$\frac{4}{11}(N+1) \leq \frac{4}{11}(N+1) \quad \blacksquare$$

Now we compute $\prod_N A(s'_N, t'_N; s_N, t_N; s'_{N-1}, t'_{N-1})$:

$$A(s'_N, t'_N; s_N, t_N; s'_{N-1}, t'_{N-1}) \times \prod_{k=5}^{N-1} A(s'_k, t'_k; s_k, t_k; s'_{k-1}, t'_{k-1}) =$$

$$= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & u_N & 0 & v_N \\ 0 & w_N & x_N & y_N \\ 0 & 0 & 0 & z_N \end{pmatrix} \times \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & a_{N-1} & 0 & b_{N-1} \\ 0 & c_{N-1} & d_{N-1} & e_{N-1} \\ 0 & 0 & 0 & f_{N-1} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & a_N & 0 & b_N \\ 0 & c_N & d_N & e_N \\ 0 & 0 & 0 & f_N \end{pmatrix}$$

where

$$a_N = a_{N-1}u_N$$

$$b_N = b_{N-1}u_N + f_{N-1}v_N$$

$$c_N = c_{N-1}x_N + a_{N-1}w_N$$

$$d_N = d_{N-1}x_N$$

$$e_N = b_{N-1}w_N + e_{N-1}x_N + f_{N-1}y_N$$

$$f_N = f_{N-1}z_N$$

and

$$u_N = 1 - \frac{5-4t_N}{N+1}$$

$$v_N = A[2,4]$$

$$w_N = A[3,2]$$

$$x_N = 1 - \frac{6}{N+1}$$

$$y_N = A[3,4]$$

$$z_N = A[4,4]$$

Notice that:

(i) $d_N = 0$ for $N \geq 5$

(ii) $f_N = 0$ for $N \geq 6$

(iii) $a_N = \prod_{k=5}^{N}\left(1 - \frac{5-4t_k}{k+1}\right) \leq \prod_{k=5}^{N}\left(1 - \frac{5-\frac{4}{3}}{k+1}\right) = O(N^{-11/3})$

(iv) $b_N = b_{N-1}u_N$ since $f_N = 0$ for $N \geq 6$. Then b_N is like a_N, or $b_N = O(N^{-11/3})$.

The recurrences for c_N and e_N are the remaining ones. Considering that $f_N = 0$ for $N \geq 6$, and that a_N and b_N have the same type of recurrence we conclude that c_N and e_N are similar.

Let us look at c_N. The solution for c_N can be found in Sedgewick (1975, pp. 297-298). Then

$$c_N = \prod_{j=7}^{N} x_j \left(c_6 + \sum_{k=7}^{N} w_k a_{k-1} \prod_{j=1}^{k} \frac{1}{x_j} \right)$$

$$= \left(\prod_{j=7}^{N} x_j \right) c_6 + \sum_{k=7}^{N} w_k a_{k-1} \left(\prod_{j=k+1}^{N} x_j \right)$$

Considering that

$$\prod_{j=7}^{N} x_j = \Theta(N^{-6}) \, ,$$

$$c_6 = O(1) \, ,$$

$$a_N = O(N^{-11/3}) \, ,$$

$$w_N = O(1),$$

$$\prod_{j=k+1}^{N} x_j = O(1),$$

then

$$c_N = O(N^{-8/3}) \; .$$

We conclude that if the value of t_N' is selected according to Eq.(5) and the value of s_N' is selected according to Eq.(6) then $\prod_N A(s_N',t_N';s_N,t_N;s_{N-1}',t_{N-1}')$ converges to $\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$. ■

The theorem we just proved tell us that the solution of Eq.(2) converges to the solution of $H(s_N,t_N)p(N) = 0$, where $H(s_N,t_N)$ is as in Eq.(1). Then

$$p_1 = \frac{4(1+s_N-3t_N)}{35-28t_N}$$

$$p_2 = \frac{3(3+2s_N-2t_N)}{35-28t_N}$$

$$p_3 = \frac{12}{35-28t_N}$$

$$p_4 = \frac{10(1-s_N-t_N)}{35-28t_N}$$

for some value of s_N,t_N, according to Proposition 5.4.2 and Proposition 5.4.3.

Theorem 5.4.4. The expected number of rotations in a random AVL tree with N

keys is bounded by

$$(i)\ 2\frac{p_2}{L_2}+2\frac{p_4}{L_4} \le r(N) \le 1-(\frac{p_2}{L_2}+3\frac{p_4}{L_4})$$

and

$$(ii)\ 2\frac{p_2}{L_2}+2\frac{p_4}{L_4} \le r(N) \le 2\frac{p_2}{L_2}+2\frac{p_4}{L_4}+p_1+p_3$$

Proof : The left hand side of cases (i) and (ii) can be obtained by observing Figure 5.3.1. The right hand side of cases (i) and (ii) can be obtained by using Lemma 5.2.1.1 and by observing Figure 5.3.1. ▪

In the following corollary the bounds for $r(N)$ are obtained to hold for any values of s_N and t_N in the range given by Proposition 5.4.2 and Proposition 5.4.3:

Corollary. $\frac{2}{7}+O(N^{-5}) \le r(N) \le \frac{98}{121}+O(N^{-5})$

Lemma 5.4.5. The expected number of nodes in the fringe of an AVL tree with N keys corresponding to the tree collection of Figure 5.3.1 is

$$\overline{f}(N) = \left[\frac{p_1}{L_1}+2\frac{p_2}{L_2}+3\frac{p_3}{L_3}+4\frac{p_4}{L_4}\right](N+1)$$

Proof : The above expression can be obtained by observing Figure 5.3.1 and by using Eq.2.2-5. ▪

In the following corollary the value for $\overline{f}(N)$ is obtained to hold for any values of s_N and t_N in the range given by Proposition 5.4.2 and Proposition 5.4.3:

Corollary. $\overline{f}(N) = \frac{267}{385}(N+1)+O(N^{-5})$

Lemma 5.4.6. The expected number of unbalanced nodes outside the fringe of a random AVL tree with N keys is at least $\frac{p_1}{L_1}(N+1)$.

Proof : The above expression is obtained as follows: a type 1 tree shown in Figure 5.3.1 must always have a type 3 tree as brother, otherwise it constitutes a type 3 or a type 4 tree. Thus the father node of a type 1 tree is always unbalanced, and the number of trees in this situation is $\frac{p_1}{L_1}$. ▪

Theorem 5.4.7. The expected number of balanced nodes in a random AVL tree with N keys is bounded by

$$\left(\frac{p_1}{L_1}+\frac{p_2}{L_2}+3\frac{p_3}{L_3}+2\frac{p_4}{L_4}\right)(N+1)\le\bar{b}(N)\le N-\left(\frac{p_1}{L_1}+\frac{p_2}{L_2}+2\frac{p_4}{L_4}\right)(N+1)$$

Proof : The left hand side of the above expression is obtained by observing Figure 5.3.1 and by using Eq.2.2-5. The right hand side is obtained by using Lemma 5.2.1.6, Lemma 5.4.5, and Lemma 5.4.6. ▪

In the following corollary the bounds for $\bar{b}(N)$ are obtained to hold for any values of s_N and t_N in the range given by Proposition 5.4.2 and Proposition 5.4.3:

Corollary. $\frac{18}{35}+\frac{18}{35N}+O(N^{-5})\le\frac{\bar{b}(N)}{N}\le\frac{62}{77}-\frac{15}{77N}+O(N^{-5})$

Experimental results show that $r(N)\approx 0.47$ (Ziviani and Tompa, 1980), and $\bar{b}(N)\approx 0.68N$ (Knuth, 1973).

5.5. Larger Weakly-closed AVL Tree Collections

In Section 5.3 we showed that any AVL tree collection that contains a tree type with its root node balanced and has more than three types is weakly-closed. This happens because every AVL tree type that contains more than one internal node and has its root node balanced suffers from the same type of misbehaviour that occurs with type 3 of Figure 5.3.2, as described in Lemma 5.3.1.

It is easy to prove a lemma similar to Lemma 5.3.1 for the tree collection shown in Figure 5.5.1. The only difference in the proof of such lemma is that the trees shown in Figure 5.3.2(a and b) do not occur, and consequently the unknown probability t_N does not exist. The transition matrix corresponding to the tree collection shown in Figure 5.5.1 involves one unknown probability s_N, as follows

$$H(s_N)=\begin{bmatrix}-4 & 3s_N & 12/5 & 3\\ 4 & -5-4s_N & 0 & 4\\ 0 & 5(1-s_N) & -6 & 0\\ 0 & 6s_N & 18/5 & -7\end{bmatrix} \qquad (1)$$

The transition matrix in Eq.(1) contains only one unknown probability, and the corresponding tree collection shown in Figure 5.5.1 contains more information than the tree collection used in the previous section. Now comes the question: Is it possible to apply Theorem 5.4.1 to this tree collection? Unfortunately we were not able to show convergence in this case. We feel that a similar proof

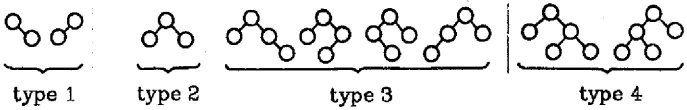


Fig. 5.5.1 Tree collection of AVL trees with more than 2 and less than 7 leaves (leaves not shown)

may exist for the tree collection shown in Figure 5.5.1. In fact the ideal situation is to prove a general theorem about matrix recurrence relations involving unknown probabilities, but it seems too difficult to obtain.

What can we say about s as a function of N ?

Unfortunately we cannot say much about s_N. For trees of size N = 10 and N = 11 we are able to obtain s_N exactly (5/154 and 3/77, respectively). Table 5.3.1 shows simulation results for bigger trees, obtained with a 95% confidence interval. From Table 5.3.1 the value of s_N seems to converge to 8/900 when N is large, but we are not able to prove it. Moreover s_N may oscillate smoothly, in such a way that simulations cannot detect. (e.g. consider $s_N = \cos(\ln N)/100$.)

Tree Size	Number of Trees	s_N (percent)		
		Fig.5.3.4(a)	Fig.5.3.4(b-c)	Total
49	10000	0.4204 ± 0.0250	0.5088 ± 0.0388	0.9292 ± 0.0454
99	5000	0.4368 ± 0.0257	0.4960 ± 0.0383	0.9328 ± 0.0459
499	5000	0.4201 ± 0.0113	0.4936 ± 0.0173	0.9138 ± 0.0204
999	2000	0.4212 ± 0.0125	0.4944 ± 0.0193	0.9156 ± 0.0230
2999	1500	0.4154 ± 0.0083	0.4740 ± 0.0127	0.8893 ± 0.0152
4999	1000	0.4101 ± 0.0076	0.4810 ± 0.0119	0.8910 ± 0.0141
9999	1000	0.4132 ± 0.0055	0.4806 ± 0.0086	0.8938 ± 0.0101
14999	200	0.4092 ± 0.0106	0.4773 ± 0.0153	0.8865 ± 0.0183
19999	300	0.4086 ± 0.0070	0.4836 ± 0.0109	0.8922 ± 0.0126

Table 5.3.1 Results for s_N

It is also possible to prove a lemma similar to Lemma 5.3.1 for the tree collection containing 10 types shown in Figure 5.5.2. The corresponding transition matrix, which involves eight unknown probabilities s, s_1, s_2, s_3, s_4, s_5, t and u, is shown in Figure 5.5.3.

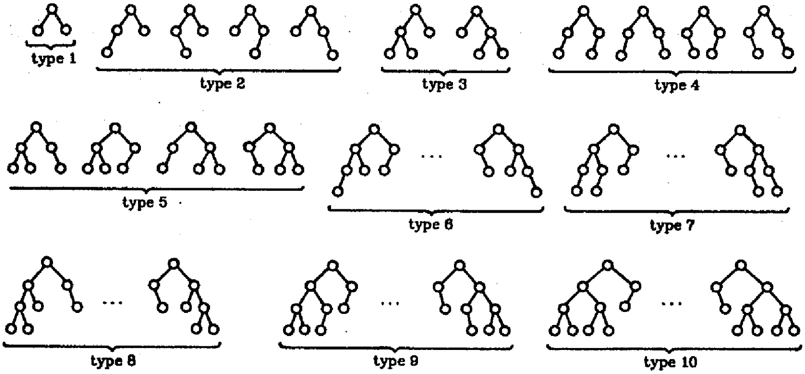


Fig. 5.5.2 Tree collection of AVL trees with 10 types (leaves not shown)

$$
\begin{pmatrix}
-5+8u & & & & \frac{24}{7}-\frac{32}{7}ss_5-\frac{16}{7}t & \frac{3}{2} & \frac{4}{3} & \frac{4}{3} & \frac{6}{5} & \frac{36}{11} \\
5(1-u) & -6 & & & \frac{20}{7}(s-ss_1)+\frac{15}{7}t & \frac{15}{8} & & & 2 & \frac{30}{11} \\
6u & \frac{18}{5} & -7 & & -\frac{24}{7}ss_2+\frac{6}{7}t & & & 2 & & \frac{12}{11} \\
 & \frac{12}{5} & & -7 & -\frac{24}{7}ss_3+\frac{6}{7}t & & 2 & & \frac{12}{7} & \frac{12}{11} \\
7u & & 7 & 7 & -8-4ss_4+3t & & & & \frac{21}{10} & \frac{42}{11} \\
 & & & & \frac{32}{7}\left(1-s+ss_1-t\right) & -9 & & & & \\
 & & & & \frac{36}{7}ss_3 & \frac{9}{4} & -10 & & & \\
 & & & & \frac{36}{7}ss_2 & \frac{27}{8} & & -10 & & \\
 & & & & \frac{40}{7}ss_4 & & \frac{20}{3} & \frac{20}{3} & -11 & \\
 & & & & \frac{44}{7}ss_5 & & & & \frac{33}{10} & -12
\end{pmatrix}
$$

Fig. 5.5.3 Transition matrix corresponding to the tree collection of AVL trees shown in Figure 5.5.2

When the number of unknown probabilities involved in the transition matrix is greater than one the problem of dealing with these unknown probabilities becomes a mathematical programming problem. This fact is important because the bounds for any complexity measure are obtained from the minimum over all possible values of the unknown probabilities in the transition matrix.

Assuming that a convergence theorem exists for (i) the AVL tree collection containing 4 types shown in Figure 5.5.1, (ii) the AVL tree collection containing 10 types shown in Figure 5.5.2, and (iii) the AVL tree collection containing 15 types shown in Figure 5.5.4, then the solution of

$$p(N) = \left\{I + \frac{H(N)}{N+1}\right\} p(N-1)$$

converges to the solution of

$$H(N)p(N) = 0. \qquad (2)$$

Solving Eq.(2) for the three AVL tree collections just mentioned, and taking the minimum over all possible values of the unknown probabilities for each complexity measure considered, we obtain the results shown in Table 5.5.1.

Tree Collection		$\bar{f}(N)$	$r(N)$	$\frac{\delta(N)}{N}$
Size	Characteristic			
4	weakly-closed	$0.75N$	$[0.38, 0.74]$	$[0.53 + 0.53/N, 0.78 - 0.22/N]$
10	weakly-closed	$0.83N$	$[0.40, 0.72]$	$[0.58 + 0.58/N, 0.76 - 0.24/N]$
15	weakly-closed	$0.86N$	$[0.43, -]$	$[0.60 + 0.60/N, 0.74 - 0.26/N]$

Table 5.5.1

5.6. Application to Other Binary Search Trees

Weight-balanced trees ($WB[\alpha]$) were introduced by Nievergelt and Reingold (1973). A binary search tree is $WB[\alpha]$ if the number of leaves in the left subtree of the root node over the total number of leaves in the tree is in the interval $[\alpha, 1-\alpha]$. The root balance α of a complete binary search tree is 1/2. Like AVL trees, $WB[\alpha]$ trees are balanced by single and double rotations.

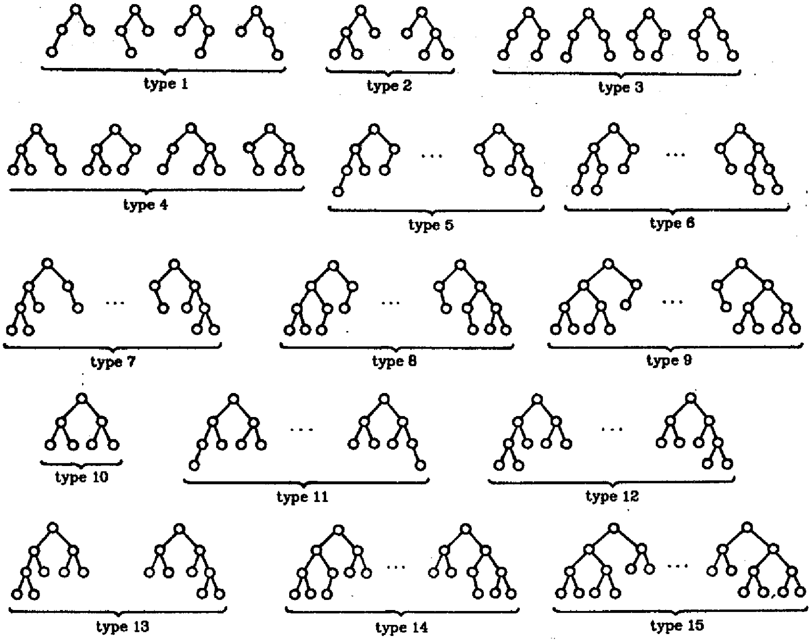


Fig. 5.5.4 Tree collection of AVL trees with 15 types (leaves not shown)

Another class of weight-balanced trees were introduced by Baer (1975) and also Gonnet (1982). They derived an algorithm that can be described as a counterpart of the AVL trees: perform single or double rotations whenever these rotations can reduce the total internal path of the subtree.

The closed AVL tree collections of Figure 5.2.1.1 and Figure 5.2.2.1 are also closed weight-balanced tree collections. Consequently, the AVL results shown in Table 5.1.1 for these tree collections are exactly the same results one would obtain in the analysis of weight-balanced trees using these same tree collections.

6. AN ANALYSIS OF SYMMETRIC BINARY B-TREES

6.1. Motivation

Bayer (1971) proposed a binary representation for 2-3 trees, as shown in Figure 6.1.1. Note that the binary representation for 2-3 trees has an asymmetry: the left edges always point to a node at the next level, while the right edges either point to a node at the same level or point to a node at the next level. Removing the asymmetry of the binary B-trees leads to the symmetric binary B-trees, abbreviated as SBB trees (Bayer, 1972).

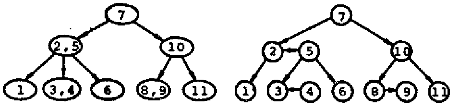


Fig. 6.1.1 A 2-3 tree and the corresponding binary B-tree (leaves not shown)

Figure 6.1.2 shows a graphic representation of an SBB tree. For SBB trees two kinds of heights need to be distinguished: the vertical height h (called h-height), required for the uniform height constraint and calculated by counting only vertical edges in any path from root to leaf, and the ordinary height k (called k-height), required to determine the maximum number of key comparisons and calculated by counting all edges in a maximal path from root to leaf. The formal definition of SBB trees, the description of the insertion algorithm and the transformations (called splits) necessary to keep the tree balanced are presented in Appendix D.

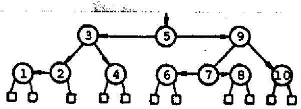


Fig. 6.1.2 SBB tree of h-height 2 and k-height 4

An SBB tree can also be seen as a binary representation for a 2-3-4 tree as defined by Guibas and Sedgewick (1978), in which "supernodes" may contain up to three keys and four sons. Such a "supernode" (with keys 3, 5, and 9 and sons containing keys 2, 4, and 7, 10) can be seen in the SBB tree of Figure 6.1.2.

In 1972 Bayer (1972) introduced the trees and the maintenance algorithms, and showed that the class of AVL trees is a proper subset. Later Wirth (1976) presented an implementation of the insertion algorithm using Pascal. Huddleston and Mehlhorn (1981) showed that the number of nodes revisited to restore the tree property, counted from father of the node inserted into the tree to the node at which the retreat terminated, is constant. In another paper by Huddleston and Mehlhorn (1980), SBB trees are used as a basic data structure for representing linear lists. The University of Washington's ESP text editor developed by Fisher, Ladner, Robertson and Sandberg (Ladner, 1980) uses SBB trees as a basic data structure.

Olivié (1980a, 1980b) presented a relationship between SBB trees and son-trees (Ottmann and Six, 1976) and a new insertion algorithm which needs less restructuring per insertion and produces SBB trees with smaller height than the original algorithms proposed by Bayer (1972). Ziviani and Tompa (1980) showed experimentally that, on the average, SBB trees perform approximately as well as AVL trees. Using the set of transformations suggested by Olivié (1980b) to preserve the balance of the tree, the experimental results show that SBB trees require less work than AVL trees to maintain balance, and the search time is only slightly longer. One main observation in Ziviani and Tompa's paper is that SBB trees are a practical structure for representing dictionaries.

We now define certain complexity measures:

(i) Let $\bar{b}(N)$ be the expected number of completely k-balanced † nodes in an SBB tree after the random insertion of N keys into the initially empty tree;

(ii) Let $s(N)$ be the expected number of splits ‡ required during the insertion of the $(N+1)^{st}$ key into a random SBB tree with N keys;

(iii) Let $Pr\{0\ hi(N)\}$ be the probability that zero height-increase transformations ‡ occur during the insertion of the $(N+1)^{st}$ key into a random SBB tree with N keys;

(iv) Let $Pr\{1\ \mathit{or\ more}\ hi(N)\}$ be the probability that one or more height-increase transformations occur during the insertion of the $(N+1)^{st}$ key into a random SBB tree with N keys;

(v) Let $m(N)$ be the maximum number of splits that may occur outside the fringe in an SBB tree during the insertion of the $(N+1)^{st}$ key into a random SBB

† A node in an SBB tree is k-balanced when the k-height of the left subtree is equal to the k-height of the right subtree.

‡ See Appendix D for the definition of the transformations in SBB trees.

tree with N keys;

(vi) Let $\overline{f}(N)$ be the expected number of nodes in the fringe of an SBB tree after the random insertion of N keys into the initially empty tree.

In Section 6.2 a small tree collection of SBB trees of h-height 1 is studied. In Section 6.3 a bigger tree collection of SBB trees of h-height 2 is proved to be a closed collection, and results on the complexity measures are obtained.

Table 6.1.1 shows a summary of the results obtained for SBB trees.

	First Order Analysis †	Second Order Analysis ‡
Tree Collection Size	3	30
$\frac{\overline{b}(N)}{N}$	$\left[0.51 + \frac{0.51}{N}, 0.86 - \frac{0.14}{N}\right]$	$\left[0.57 + \frac{0.57}{N}, 0.72 - \frac{0.28}{N}\right]$
$\overline{s}(N)$	$[0.29, 0.63]$	$[0.36, 0.56]$
$Pr\{0\ hi(N)\}$	0.66	0.66
$Pr\{1\ or\ more\ hi(N)\}$	0.34	0.34
$\overline{f}(N)$	$0.66N+0.66$	$0.85N+0.85$

† For $N \geq 6$

‡ Results are approximated to $O(N^{-5})$

Table 6.1.1 Summary of the SBB tree results

6.2. First Order Analysis

The analysis of the lowest level of the SBB tree can be carried out by considering the tree collection of SBB trees with four or less leaves and h-height 1, as shown in Figure 6.2.1. The first step necessary to perform the first order analysis is to show that the SBB tree collection of Figure 6.2.1 is closed (Definition 2.3.2).

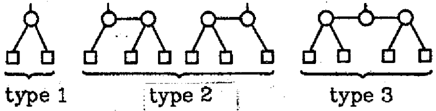


Fig. 6.2.1 Tree collection of SBB trees with four or less leaves and h-height 1

Theorem 6.2.1. The SBB tree collection shown in Figure 6.2.1 is closed.

Proof : An insertion into type 1 always leads to a type 2 tree. An insertion into a type 2 tree always leads to a type 3 tree, and in this case a split transformation occurs with probability 2/3. The only case where a height-increase transformation occurs is after an insertion into type 3, and in this case a split transformation may take place higher in the tree. But even if this split transformation takes place higher it causes no problem because the smallest subtrees that may be moved around are exactly type 1 and type 2 subtrees, which are the types one would obtain anyway if an insertion is performed into a type 3 tree. ▪

Theorem 6.2.1 says that all the transitions in the tree collection of Figure 6.2.1 are well-defined, so that the theorems presented in Chapter 2 can be applied. Thus

$$H = \begin{bmatrix} -3 & 0 & 2 \\ 3 & -4 & 3 \\ 0 & 4 & -5 \end{bmatrix} .$$

From Eq.2.2-3 we have $Hp(N) = 0$, and therefore $p_1(\infty) = 8/35$, $p_2(\infty) = 15/35$, and $p_3(\infty) = 12/35$. Since the eigenvalues of H are 0, −5, ,−7, we observe that $p_1(N) = 8/35$, $p_2(N) = 15/35$, and $p_3(N) = 12/35$, for $N \geq 6$.

Theorem 6.2.2. The expected number of completely k-balanced nodes in a random SBB tree with N keys is bounded by

$$\left[\frac{p_1}{2}+\frac{p_2}{3}+3\frac{p_3}{4}\right](N+1) \leq \bar{b}(N) \leq N-\left[\frac{p_2}{3}\right](N+1)$$

Proof : The lower bound is obtained by using Eq.2.2-5 and observing Figure 6.2.1. The upper bound comes from the fact that $\bar{b}(N)$ plus the expected number of k-unbalanced nodes is equal to N, and the expected number of k-unbalanced nodes in this case is $\frac{p_3}{3}(N+1)$. ▪

Corollary. $\frac{18}{35}+\frac{18}{35N} \leq \frac{\bar{b}(N)}{N} \leq \frac{6}{7}-\frac{1}{7N}$ for $N \geq 6$

Theorem 6.2.3. The expected number of splits in a random SBB tree with N keys is bounded by

$$\frac{2}{3}p_2 \leq s(N) \leq 1-(p_1+\frac{1}{3}p_2)$$

Proof : The lower bound is obtained by observing that a split transformation happens when an insertion is performed into the type 2 shown in Figure 6.2.1, with probability 2/3. The upper bound is obtained by observing that the maximum number of splits per insertion is 1. ▪

Corollary. $\frac{2}{7} \leq s(N) \leq \frac{22}{35}$ for $N \geq 6$

Lemma 6.2.4. The probability that no height-increase and one or more height-increases occur on the $(N+1)^{st}$ random insertion into a random SBB tree with N keys are, respectively

$$Pr\{0\ hi(N)\} = p_1 + p_2$$

$$Pr\{1\ or\ more\ hi(N)\} = p_3$$

Proof : By observing Figure 6.2.1. ▪

Corollary. $Pr\{0\ hi(N)\} = \frac{23}{35}$ for $N \geq 6$

$Pr\{1\ or\ more\ hi(N)\} = \frac{12}{35}$ for $N \geq 6$

Lemma 6.2.5. The probability of a split occurring outside the fringe during the insertion of the $(N+1)^{st}$ key into a random SBB tree with N keys is $m(N) = p_3$. Furthermore, no more than one split may occur.

Proof : An insertion into type 3 shown in Figure 6.2.1 causes a height-increase transformation, which may cause a split higher in the tree. ▪

Corollary. $m(N) = \frac{12}{35}$ for $N \geq 6$

Lemma 6.2.6. The expected number of nodes in the fringe of an SBB tree with N keys that corresponds to the tree collection of Figure 6.2.1 is

$$\bar{f}(N) = \left[\frac{p_1}{2} + 2\frac{p_2}{3} + 3\frac{p_3}{4}\right](N+1)$$

Proof : By observing Figure 6.2.1. ▪

Corollary. $\bar{f}(N) = \frac{23}{35}N + \frac{23}{35}$ for $N \geq 6$

6.3. Second Order Analysis

We can improve the bounds obtained in the previous section by considering a larger tree collection. Figure 6.3.1 shows a tree collection of SBB tree with 30 types and h-height 2.

Theorem 6.3.1. The SBB tree collection of Figure 6.3.1 is closed.

Proof : When an insertion into an SBB tree causes no split transformation then there is no problem. When an insertion does cause a split transformation somewhere in the tree then we consider two possible cases:

(i) The split transformation occurs at a node that belongs to one of the trees of the tree collection shown in Figure 6.3.1. This case obviously causes no problem.

(ii) The split transformation occurs at a node outside the fringe. By examining Figure 6.3.1 we can see that there is no type that contains two opposite horizontal pointers at the second level. This fact implies that in order to have a split transformation out of the fringe we must (a) have at least any two subtrees from the tree collection of Figure 6.3.1 sharing the same root; and (b) an insertion into one of these two subtrees must cause a height-increase transformation which will cause the split transformation higher in the tree. However this split transformation has no effect on the composition of the fringe because the fringe of the transformed subtree is entirely contained in the subtrees that are moved around during the transformation.

Figure 6.3.2 illustrates this fact: Assume that T_1, T_2, T_3, and T_4 each contain a subtree that belongs to the tree collection of Figure 6.3.1. Suppose that an insertion into T_2 (or T_3) causes a height-increase transformation (resulting in subtree T_2'), which will cause a left-right split transformation at the next level. Clearly the subtrees T_1, T_2', T_3, and T_4 are moved around without any modification in their structures. ▪

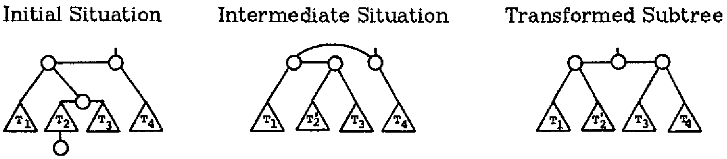


Fig. 6.3.2 Left-right split transformation (an insertion into T_2 transforms it into T_2')

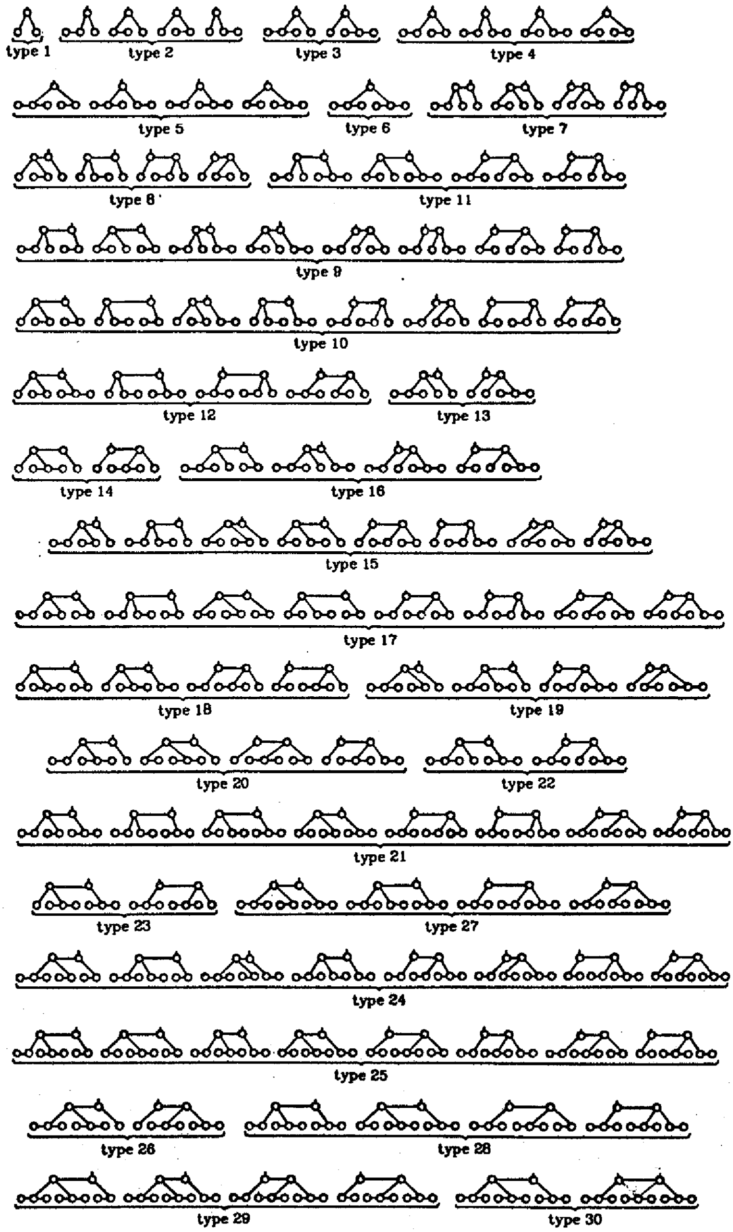


Fig. 6.3.1 Tree collection of SBB trees with 30 types (leaves not shown)

The matrix H can be easily obtained by observing Figure 6.3.1. From Eq.2.2-3 we have $H p(N) = 0$, and therefore

$$
\begin{aligned}
p_1 &= 874463196/49525503055 \\
p_2 &= 1620896463/9905100611 \\
p_3 &= 1102942170/9905100611 \\
p_4 &= 783174402/9905100611 \\
p_5 &= 176215347/1415014373 \\
p_6 &= 516201576/9905100611 \\
p_7 &= 183823695/5660057492 \\
p_8 &= 183823695/5660057492 \\
p_9 &= 394971148/9905100611 \\
p_{10} &= 394971148/9905100611 \\
p_{11} &= 7311869433/198102012220 \\
p_{12} &= 7311869433/198102012220 \\
p_{13} &= 122549130/9905100611 \\
p_{14} &= 122549130/9905100611 \\
p_{15} &= 163398840/9905100611 \\
p_{16} &= 1608763167/99051006110 \\
p_{17} &= 1072508778/49525503055 \\
p_{18} &= 1608763167/99051006110 \\
p_{19} &= 330882651/39620402444 \\
p_{20} &= 330882651/39620402444 \\
p_{21} &= 4946862/230351177 \\
p_{22} &= 7420293/460702354 \\
p_{23} &= 7420293/460702354 \\
p_{24} &= 114212043/9905100611 \\
p_{25} &= 114212043/9905100611 \\
p_{26} &= 50133735/9905100611 \\
p_{27} &= 340437933/28300287460 \\
p_{28} &= 340437933/28300287460 \\
p_{29} &= 360039042/49525503055 \\
p_{30} &= 78121098/9905100611
\end{aligned}
$$

Since the eigenvalues of H are 0, -5, $-5.96 \pm 7.03i$, ..., -13.73, $-14.80 \pm 4.22i$, the asymptotic values of $p(N)$ obtained from Eq.2.2-4 are approximated to $O(N^{-5.96})$.

Theorem 6.3.2. The expected number of completely k-balanced nodes in a random SBB tree with N keys is bounded by

$$0.57097 + \frac{0.57097}{N} + O(N^{-5.96}) \leq \frac{\bar{b}(N)}{N} \leq 0.71632 - \frac{0.28368}{N} + O(N^{-5.96})$$

Proof : Similar to the proof of Theorem 6.2.2. ▪

Theorem 6.3.3. The expected number of splits in a random SBB tree with N keys is bounded by

$$0.35921+O(N^{-5.96}) \leq s(N) \leq 0.55672+O(N^{-5.96})$$

Proof : Similar to the proof of Theorem 6.2.3. ▪

Experimental results show that $s(N) \approx 0.39$ (Ziviani and Tompa, 1980).

Lemma 6.3.4. The probability of a split occurring outside the fringe during the insertion of the $(N+1)^{st}$ key into a random SBB tree with N keys is $m(N) = 0.19751+O(N^{-5.96})$.

Proof : Similar to the proof of Theorem 6.2.5. ▪

Lemma 6.3.5. The expected number of nodes in the fringe of an SBB tree with N keys that corresponds to the tree collection of Figure 6.3.1 is

$$\overline{f}(N) = 0.85465N + 0.85465 + O(N^{-5.96})$$

Proof : Similar to the proof of Theorem 6.2.6. ▪

We end this chapter with the following remarks:

(i) The results on height-increase transformations obtained in the first order analysis cannot be improved in the second order analysis, because there is no possibility of a height-increase transformation at the second level of any type in the tree collection shown in Figure 6.3.1. (This fact is the key point that permits the proof of Theorem 6.3.1.)

(ii) A third order analysis seems difficult to obtain because of the large number of types involved in a tree collection of SBB trees with h-height 3. A tree collection of SBB trees of h-height 3 which have as subtrees the 30 types of the h-height 2 tree collection of Figure 6.3.1 contains $n\left(\frac{n(n+1)}{2}\right) = 13950$ types, since $n=30$.

7. EXPECTED HEIGHT OF BINARY SEARCH TREES

7.1. Motivation

In this chapter we study random binary search trees † constructed with no balance constraints. Knuth (1973, § 6.2.2, p.427) showed that the tree search requires about $2\ln N$ comparisons if the keys are inserted into the tree in random order. Robson (1979) showed that the expected depth of the deepest node (i.e. the height) is approximately twice the expected depth of a random node in the tree. In fact Robson presented a proof of the existence of a logarithm upper bound of approximately $4.31\ \ln N$. He also presented a sequence of lower bounds by analysing heuristics for finding nodes of near maximum depth, which the best lower bound computed is about $3.63\ln N$. Gonnet (1981, p.47) conjectured that an improved upper bound of

$$4.31...\ln N - 2.81...\ln\ln N + O(1)$$

is tight.

The expected height of random binary search trees is the maximum size of a stack which may be required by a recursive algorithm operating on the tree. The worst case is $O(N)$, but it happens very rarely. Sedgewick (1975, pp.38-41) also pointed out that this height is the stack size necessary to sort N keys using the straightforward version of quicksort, in which the recursion is applied to both subfiles after the partitioning of the file.

The object of this chapter is to show how to apply the fringe analysis technique presented in Chapter 2 to obtain a lower bound on the expected height of binary search trees. This application differs considerably from the previous applications of fringe analysis shown in this thesis. The lower bound obtained is not as good as the one obtained by Robson (1979), but the method used to obtain the result is interesting in its own right.

The complexity measures used in this chapter are as follows:

(i) Let $\overline{h}(N)$ be the expected height of a random binary search tree with N keys;

(ii) Let $\overline{t}(N)$ be the expected number of external nodes at the lowest level of a random binary search tree with N keys.

† See Section 1.4 for the definition of a random binary search tree

7.2. Fringe versus Height

Figure 7.2.1 shows an open tree collection of binary search trees. (cf. Definition 2.3.4.) Notice that an insertion into a type 2 tree shown in Figure 7.2.1 gives one type 1 tree and one other node which we ignore.

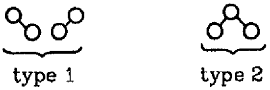


Fig. 7.2.1 (Leaves not shown)

Unlike the fringe of a balanced search tree, the *fringe* of a binary search tree with N keys consists of one subtree that is isomorphic to one type in a tree collection C. Figure 7.2.2 shows some instances of the fringe of a binary search tree corresponding to the tree collection of Figure 7.2.1 when random insertions are performed into the tree. Note that our fringe "traces" one subtree of the total tree. An insertion into the type 1 tree causes a height increase by 1 with probability 2/3, and an insertion into the type 2 tree always cause a height increase by 1. Of course the actual height of the tree may be given by some other subtree, but the one under analysis will give a lower bound on the height.

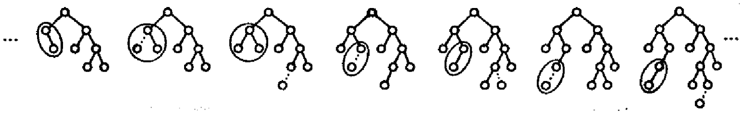


Fig. 7.2.2 Instances of the fringe of a binary search tree when random insertions are performed (fringe is encircled)

The following theorem relates information about the fringe (as defined above) with the expected height of random binary search trees.

Theorem 7.2.1. If $c_0 \le \bar{t}(N) \le c_1$ then

$$c_0 \ln N + O(1) \le \bar{h}(N) \le c_1 \ln N + O(1)$$

Proof : The expected increase in height after an insertion is $\frac{\bar{t}(N)}{N+1}$. By summation we obtain

$$c_0 H_{N+1} \le \bar{h}(N) = \sum_{i=0}^{N} \frac{\bar{t}(i)}{i+1} \le c_1 H_{N+1}$$

where $H_N = \sum_{i=1}^{N} \frac{1}{i}$, for $N \geq 1$, and $H_N = \ln N + 0.577... + O(N^{-1})$ (Knuth, 1968, § 1.2.7).

Thus $\qquad c_0 \ln N + O(1) \leq \bar{h}(N) \leq c_1 \ln N + O(1)$ ▪

In the next paragraphs we show how to obtain several values of c_0.

The definition of the fringe presented above differs considerably from the definition of the fringe used in other parts of this thesis. The fringe now is composed of one subtree. Consequently, $p_i(N)$ is defined as the probability that the tree we are tracing is of type i.

Let us consider one tree of type 1 and the type 2 tree of Figure 7.2.1 as shown in Figure 7.2.3. The arcs show the probabilities of different transitions. The transitions marked with a * produce a height increase of 1.

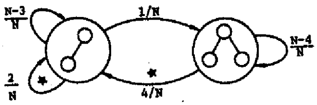


Fig. 7.2.3

Then

$$\begin{bmatrix} p_1(N) \\ p_2(N) \end{bmatrix} = \begin{bmatrix} \frac{N-1}{N} & \frac{4}{N} \\ \frac{1}{N} & \frac{N-4}{N} \end{bmatrix} \begin{bmatrix} p_1(N-1) \\ p_2(N-1) \end{bmatrix} = \begin{bmatrix} I + \frac{H}{N} \end{bmatrix} \begin{bmatrix} p_1(N-1) \\ p_2(N-1) \end{bmatrix}$$

where $H = \begin{bmatrix} -1 & 4 \\ 1 & -4 \end{bmatrix}$.

Let $p(N)$ be an m-component column vector containing $p_i(N)$. Then

$$p(N) = \begin{bmatrix} I + \frac{H}{N} \end{bmatrix} p(N-1) \tag{1}$$

where I is an $m \times m$ identity matrix.

Eq.(1) is of the form of Eq.2.1-2, which means that the theorems of Section 2.2 apply. Thus the solution of Eq.(1) converges to the solution of $Hp(N)$. Since the eigenvalues of H are 0 and -5 then $p_1(N) = 4/5$ and $p_2(N) = 1/5$, for $N \geq 5$.

Lemma 7.2.2. Let $\Delta\bar{h}(N)$ be the expected increase in height due to one insertion into an N-key binary search tree. Then

$$\Delta\bar{h}(N) = \left[\frac{2}{N}, \frac{4}{N}\right]\left[\begin{matrix}p_1(N)\\ p_2(N)\end{matrix}\right] = \frac{12}{5N} \quad \text{for } N\geq 6$$

Proof : An insertion into 2 of the 3 external nodes of the type 1, and an insertion into any of the 4 external nodes of type 2 shown in Figure 7.2.1 increases the height by 1. ▪

Lemma 7.2.2 leads to the following theorem:

Theorem 7.2.3. The expected height of a random binary search tree with N keys that corresponds to the tree collection shown in Figure 7.2.1 is

$$\bar{h}(N) \geq \frac{12}{5}\ln N + O(1)$$

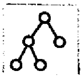
Proof : Similar to the proof of Theorem 7.2.1. ▪

7.3. Larger Tree Collections

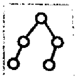
In order to improve the lower bound on the expected height of binary search trees obtained in the previous section we have to consider larger tree collections. The following notation is useful in this case. We represent a type as

$$[\,a,\, b\,,\, c\,,\, d\,,\ldots]$$

where a, b, c, d,... represent the number of external nodes at the lowest level, at the second lowest level, at the third lowest level, at the fourth lowest level,... . Several trees may be grouped with this notation. There is no need to differentiate trees other than by the number of external nodes at different levels. For example



and



are grouped as type [4,2].

This notation applied to the tree collection shown in Figure 7.2.1 appears as [2,1] for type 1 and [4,0] for type 2. An insertion into the first element of type [2,1] gives [2,1], an insertion into the second element of type [2,1] gives [4,0], an insertion into the first element of type [4,0] gives [2,3], which is reduced to [2,1] by ignoring one node at the second lowest level, and an insertion into the second element of type [4,0] is not possible because it contains zero external nodes.

Thus the tree collection constituted by types [2,1] and [4,0] gives

$$p[2,1] = \frac{4}{5}$$

$$p[4,0] = \frac{1}{5}$$

where type [2,3], obtained when an insertion is performed into type [4,0], is reduced to type [2,1]. Every time a reduction is performed information about nodes at higher levels is lost. However, the reductions are necessary to have a finite system.

Now comes the question: Which is the best way to obtain bigger tree collections and improve the lower bound on the height ? It is not difficult to see that tree collections containing types with height 4 or more may contain a large number of types. We arrived at the conclusion that one good strategy is to minimise the losses in the reductions. (i.e. try to avoid reductions at the lowest levels as much as possible.)

Two examples of tree collections containing four types are:

type [2,1,1]		type [2,1,1]
type [2,3,0]	and	type [2,3,0]
type [4,0,1]		type [2,1,3]
type [4,2,0]		type [4,0,0]

Among tree collections with four types the best that we found is the following:

type [2,1,1,1]
type [2,1,3,0] (1)
type [2,3,0,0]
type [4,0,0,0]

with the following reductions:

[2,3,0,1] is reduced to [2,3,0,0]
[2,3,2,0] is reduced to [2,3,0,0]
[4,0,1,1] is reduced to [4,0,0,0]

$[4,0,3,0]$ is reduced to $[4,0,0,0]$

$[4,2,0,0]$ is reduced to $[4,0,0,0]$

Using Eq.7.2-1 in the tree collection shown in (1) we get the matrix:

$$H = \begin{bmatrix} -3 & 2 & 0 & 0 \\ 1 & -6 & 2 & 0 \\ 1 & 3 & -5 & 4 \\ 1 & 1 & 3 & -4 \end{bmatrix}$$

which has the solution:

$$p[2,1,1,1] = 8/81$$

$$p[2,1,3,0] = 12/81$$

$$p[2,3,0,0] = 32/81$$

$$p[4,0,0,0] = 29/81$$

Considering the fact that

$$\Delta\bar{h}(N) = \left(\frac{2}{N}, \frac{2}{N}, \frac{2}{N}, \frac{4}{N}\right) \cdot p(N)$$

then

$$\bar{h}(N) \geq \frac{220}{81}\ln N + O(1) = 2.716\ln N + O(1).$$

Table 7.3.1 shows the best values obtained for c_0 for different tree collections. Each tree collection is composed of types with four levels. The results for c_0 may not be optimal, but they are the best we obtained after trying many different tree collections for each size.

From Table 7.3.1 we can see that the values obtained for c_0 improve at first, but not too much when the size of the tree collection gets bigger. The best result was obtained for a tree collection with 33 types, where

$$\bar{h}(N) \geq 3.179\ln N + O(1).$$

We suspect that one needs a very large tree collection (probably on the order of thousands of types) in order to improve the above result significantly. Nevertheless, it is a surprising application of fringe analysis to deduce bounds on the height, which is a global measure.

Tree Collection Size	Constant c_0
4	2.716
5	2.748
6	2.813
7	2.861
8	2.909
9	2.949
10	2.965
11	2.984
12	2.996
13	3.013
18	3.081
22	3.125
32	3.160
33	3.179

Table 7.3.1 Constant term in the lower bound for the height

8. CONCLUSIONS

In Chapter 2 we show that the matrix recurrence relation related to fringe analysis problems converges to the solution of a linear system involving the transition matrix, even when the transition matrix has eigenvalues with multiplicity greater than one (i.e., the eigenvalues of the transition matrix do not need to be pairwise distinct). This fact makes the fringe analysis theory presented there flexible and general enough to permit its application in the analysis of many different search trees.

In Chapter 3 an analysis for the three lowest levels of 2-3 trees is accomplished. It is indicated that if one applies the same technique used to obtain the three level tree collection for 2-3 trees, then it might be possible to carry out an analysis for the four lowest levels, which would imply the solution of a 4410 × 4410 linear system.

In Chapter 4 an analysis of B-trees is performed. Information about the operation of splitting an overfull node and the concurrency of operations are some of the results presented there.

In Chapter 5 we present a closed AVL tree collection containing three types. We also show that an AVL tree collection containing four types is not closed. An inherent difficult posed by the rotations necessary to keep the AVL tree balanced forces the introduction of two unknown probabilities s_N and t_N into the transition matrix. In the main theorem of Chapter 5 we prove convergence of the matrix recurrence relation involving the unknown probabilities s_N and t_N.

Like AVL trees, weight-balanced trees are balanced by single and double rotations (Knuth, 1973, § 6.2.3). For this reason only small tree collections of weight-balanced trees are closed. For large tree collections we find the same type of difficulties showed in Chapter 5 for AVL trees. Consequently, the technique presented there is also suitable for the analysis of weight-balanced trees.

The main merit of Chapter 6 is to present a higher order analysis of a balanced binary search tree, the symmetric binary B-tree. A closed tree collection containing 30 types is obtained and some results are derived from it.

In Chapter 7 a theorem relating the fringe of a binary search tree with its height is presented. This result permits us to obtain a lower bound on the expected height of unbalanced binary search trees.

REFERENCES

Abramowitz, M. and Stegun, I.A. *Handbook of Mathematical Functions*, (New York: Dover, 1972).

Adel'son-Vel'skii, G.M. and Landis, E.M. "An Algorithm for the Organization of Information," *Doklady Akademia Nauk* USSR 146, 2 (1962), 263-266. English translation in *Soviet Math. Doklay* 3 (1962), 1259-1263.

Aho, A.V., Hopcroft, J.E. and Ullman, J.D. *The Design and Analysis of Computer Algorithms*, Addison Wesley, 1974.

Baer, J.-L. "Weight Balanced Trees," *Proc. AFIPS National Computer Conference 44* (1975), 467-472.

Bayer, R., "Binary B-trees for Virtual Memory," *Proc. 1971 ACM SIGFIDET Workshop*, San Diego (1971), 219-235.

Bayer, R., "Symmetric Binary B-trees: Data Structure and Maintenance Algorithms," *Acta Informatica 1*, 4(1972), 290-306.

Bayer, R. and McCreight, E. "Organization and Maintenance of Large Ordered Indexes," *Acta Informatica 1*, 3 (1972), 173-189.

Bayer, R. and Schkolnick, M. "Concurrency of Operations on B-trees," *Acta Informatica 9*, 1 (1977), 1-21.

Brown, M. "A Partial Analysis of Random Height-Balanced Trees," *SIAM Journal of Computing 8*, 1 (Feb 1979a), 33-41.

Brown, M. "Some Observations on Random 2-3 Trees," *Information Processing Letters 9*, 2 (Aug 1979b), 57-59.

Clausing, A. "Kantorovich-Type Inequalities," *The American Mathematical Monthly 89*, 5 (May 1982), 314-330.

Chvatal, V., Klarner, D.A. and Knuth, D.E. "Selected Combinatorial Research Problems," *Report STAN-CS-72-292*, Computer Science Department, Stanford University, 1972.

Comer, D. "The Ubiquitous B-tree," *Computing Surveys 11*, 2 (Jun 1979a), 121-137.

Cox, D.R. and Miller, H.D. *The Theory of Stochastic Processes* (London: Chapman and Hall Ltd., 1965).

Eisenbarth, B. and Mehlhorn, K. "Allgemeine Fringe-analysis und ihre Anwendung zur Untersuchung der Overflowtechnik bei B-Bäumen," Universität des

Saarlandes, Saarbrucken, West Germany, Oct 1980.

Feller, W. *An Introduction to Probability Theory and Its Applications*, Vol. 1 (New York: John Wiley and Sons, 1968).

Gantmacher, F.R. *The Theory of Matrices*, Vol. 1 (New York: Chelsea Publishing Company, 1959).

Geddes, K.O. and Gonnet, G.H. "MAPLE User's Manual," *Report CS-81-25*, Department of Computer Science, University of Waterloo, Waterloo, Canada, Jul 1981.

Gonnet, G.H. "Handbook of Algorithms and Data Structures," *Report CS-80-23*, Department of Computer Science, University of Waterloo, Jun 1981.

Gonnet, G.H. "Balancing Binary Trees by Internal Path Reduction," *Report CS-82-07*, Department of Computer Science, University of Waterloo, Waterloo, Canada, 1982.

Gonnet, G.H., Ziviani, N. and Wood, D. "An Analysis of 2-3 Trees and B-trees," *Report CS-81-21*, Department of Computer Science, University of Waterloo, Canada, Jun 1981.

Guibas, L.J. and Sedgewick, R. " A Dichromatic Framework for Balanced Trees," *19th Annual Symposium on Foundations of Computer Science*, 1978.

Huddleston, S. and Mehlhorn, K. "A New Data Structure for Representing Sorted Lists," Universität des Saarlandes, Saarbrucken, West Germany, Dec 1980.

Huddleston, S. and Mehlhorn, K., "Robust Balancing in B-trees," *5th GI-Conference on Theoretical Informatics*, Karlsruhe, March 1981.

Jensen, L.W.V. "Sur les Fonctions Convexes et les Inégalités Entre les Valeurs Moyennes," *Acta Mathematica 30* (1906), 175-193.

Knuth, D.E. *The Art of Computer Programming*, Vol. 1 (Reading, Mass.: Addison-Wesley, 1968).

Knuth, D.E. *The Art of Computer Programming*, Vol. 3 (Reading, Mass.: Addison-Wesley, 1973).

Kwong, Y.S. and Wood, D. "Approaches to Concurrency in B-trees," *Lecture Notes in Computer Science 88*, Springer-Verlag (1980), 402-413.

Ladner, R. Private Communication, 1980.

Mehlhorn, k. "A Partial Analysis of Height-Balanced Trees," *Report A 79/13*, Universität des Saarlandes, Saarbrucken, West Germany, 1979a.

Mehlhorn, K. "A Partial Analysis of Height-Balanced Trees under Random

Insertions and Deletions," *Report A 79/21*, Universität des Saarlandes, Saarbrucken, West Germany, 1979b.

Mehlhorn, K. Private Communication, 1981.

Nievergelt, J. and Reingold, E.M. "Binary Search Trees of Bounded Balance," *SIAM Journal of Computing 2*, 1 (Mar 1973), 33-43.

Olivié, H. "On the Relationship Between Son-Trees and Symmetric Binary B-trees," *Information Processing Letters 10*, 1 (Feb 1980a), 4-8.

Olivié, H. "Symmetric Binary B-trees Revisited," *Report 80-01*, Interstedelijke Industriële Hogeschool Antwerpen-Mechelen, Antwerp, Belgium, (1980b).

Ottmann, Th. and Six, H.-W. "Eine neue Klasse von ausgeglichenen Binärbäumen," *Augewandte Informatik 9* (1976), 395-400.

Ottmann, Th. and Stucky, W. "Higher Order Analysis of Random 1-2 Brother Trees," *BIT 20*, (1980), 302-314.

Ottmann, Th. and Wood, D. "1-2 Brother Trees or AVL Trees Revisited," *The Computer Journal 23*, 3 (1980), 248-255.

Robson, J.M. "The Height of Binary Search Trees," *The Australian Computer Journal 11*, 4 (Nov 1979), 151-153.

Sedgewick, R. "Quicksort," *Report STAN-CS-492*, Computer Science Department, Stanford University, May 1975.

Wilkinson, J.H. *The Algebraic Eigenvalue Problem*, (London: Oxford University Press, 1965).

Wirth, N. *Algorithms + Data Structures = Programs*, (New Jersey: Prentice-Hall, 1976).

Yao, A. "On Random 2-3 Trees," *Acta Informatica 9*, (1978), 159-170.

Ziviani, N. and Tompa, F. "A Look at Symmetric Binary B-trees," *Report CS-80-51*, Department of Computer Science, University of Waterloo, Canada, Nov 1980 (to appear in *INFOR - Canadian Journal of Operational Research and Information Processing*,).

APPENDIX A. AVL Trees

Adel'son-Vel'skii and Landis (1962) presented the AVL trees. A binary search tree is AVL if, for every node, the difference between the height of the left subtree and the height of the right subtree is at most one. A balance field in each node can indicate this with two bits: +1, higher right subtree; 0, equal heights; − 1, higher left subtree.

The process of insertion of a new key consists of three parts:

(i) Follow the search path until it is verified that the key is not in the tree (i.e., find the place of insertion).

(ii) Insert the new node and set the balance field to 0.

(iii) Retreat along the search path and check the balance field at each node. At this point a transformation may be necessary, as described below.

Phases 1 and 2 are similar to the search and insertion in a binary search tree, as described in Section 1.4, except by the balance field consideration. In phase 3 balancing occurs if the balance field indicates that the node becomes more unbalanced with the insertion (occurs when the direction of the search path and the present balance coincide). In this case a single or double rotation occurs, depending on the balance field of the node and on the balance field of its son, which is along the search path. Figure 5.3.3, in Chapter 5, illustrates the AVL tree transformations. As the height of the rotated subtree is the same as the height of the subtree before the insertion, at most one rotation per insertion is necessary. Of course if the balance field indicates that the subtree becomes less unbalanced, a modification of the balance field is sufficient.

APPENDIX B. 2-3 Trees

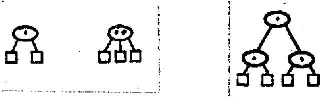
In a 2-3 tree every internal node contains either 1 or 2 keys, and all external nodes appear at the same level. The class of 2-3 trees is a special class of B-trees, and they are more appropriate for primary store.

The process of insertion of a new key consists of:

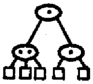
(i) Follow the search path until it is verified that the key is not in the tree (i.e., find the place of insertion).

(ii) Insert the new key into the node. To insert into a node that contains only one key, we insert it as the second key. If the node already contains two keys, we split it into two one-key nodes, and insert the middle key into the parent node. This process may propagate up if the parent node already contains two keys. When there is no node above we create a new root node to insert the middle key.

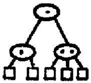
Following the notation presented by Chvatal et al. (1972, Problem 37), where the dots indicate keys, the first three steps in the growth of a 2-3 tree are



and the fourth step is either



or



APPENDIX C. B-Trees

According to Bayer and McCreight (1972) a *B−tree* of order m is a balanced multiway tree with the following properties: (a) The leaves are null nodes which all appear at the same depth. (b) Every node has at most $2m+1$ sons. (c) Every node except the root and the leaves has at least $m+1$ sons; the root is either a leaf or has at least two sons †. Consequently, a 2-3 tree is a B-tree of order $m = 1$.

The process of insertion of a new key starts with the search for the place of insertion, followed by the insertion of the key into a node. To insert a new key into a node that contains less than $2m$ keys we just insert it into the other keys. If the node already contains $2m$ keys, we split it into two m-keys nodes, and insert the middle key into the parent node, repeating the process again with the parent node. When there is no node above we create a new root node to insert the middle key.

† Knuth (1973, p. 473) presented a slightly different definition of B-trees. In Knuth's definition every node in a B-tree of order m has at most $m-1$ keys and at least $\lceil m/2-1 \rceil$ keys. Knuth's definition considers B-trees of order $2i$, $i \geq 2$ (B-trees containing at least i keys and at most $2i-1$ keys), while the above definition does not consider such trees. However, these trees present a disadvantage: the split operation divides the node into two nodes with a different number of keys in each one, which implies that a decision about which node will contain more keys has to be taken.

APPENDIX D. Symmetric Binary B-trees

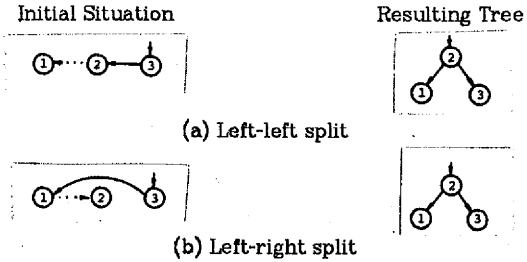
Symmetric binary B-trees, abbreviated as SBB trees, are binary trees with two kinds of edges, namely vertical edges and horizontal edges, such that (cf. Bayer, 1972):

(i) All paths from the root to every leaf node have the same number of vertical edges, and

(ii) There are no successive horizontal edges.

Figure 6.1.2 (Chapter 6) shows a graphic representation of an SBB tree. As mentioned in Chapter 6, SBB have two kinds of heights: the vertical height h, calculated by counting only vertical edges in any path from root to leaf, and the ordinary height k, calculated by counting all edges in a maximal path from root to leaf.

The algorithm to construct and maintain SBB trees uses local transformations on the path of insertion to preserve the balance of the trees. The key to be inserted is always inserted after the lowest vertical pointer in the tree. Depending on the tree's status prior to insertion two successive horizontal pointers may result, and a transformation may become necessary. If a transformation is performed, the number of vertical pointers from the root to the new leaf may be altered, thus requiring further transformations to obtain a uniform height. Figure D1 shows the transformations proposed by Bayer (1972). Symmetric transformations (i.e. right-right and right-left) also may occur.


Initial Situation

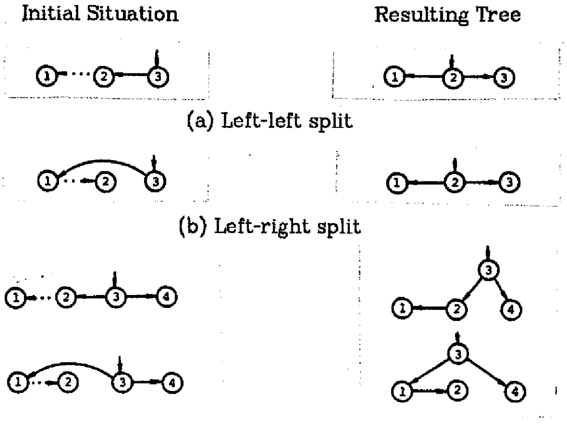
Resulting Tree

(a) Left-left split

(b) Left-right split

Fig. D1 The two transformations as proposed by Bayer (1972)

A revised set of transformations has been proposed by Olivié (1980b). The insertion algorithm using the new transformations produces SBB trees with smaller height than does the original algorithm, and it needs less transformations to build the tree. Guibas and Sedgewick (1978) have also defined similar transformations. Figure D2 shows the new transformations. The left-left split and the left-right split require the modification of 3 and 5 pointers respectively, and the height-increase transformation requires only the modification of two bits. Symmetric transformations may also occur.



(a) Left-left split

(b) Left-right split

(c) Height-increase

Fig. D2 The new transformations as proposed by Olivié (1980b)

When a height-increase transformation occurs, the height of the transformed subtree is one more than the height of the original subtree, and thus the node rearrangement may cause other transformations along the search path up to the root. Usually the retreat along the search path terminates when either a vertical pointer is found or a split transformation is performed. As the height of the split subtree is the same as the height of the original subtree, at most one split transformation per insertion may be performed.