Reprinted from INFORMATION AND CONTROL
All Rights Reserved by Academic Press, New York and London

Vol. 55, Nos. 1–3, October/November/December 1982
Printed in Belgium

The Theory of Fringe Analysis and Its Application to 2–3 Trees and B-Trees

BERNHARD EISENBARTH

Universität des Saarlandes, Fachbereich 10/D-6600 Saarbrucken, West Germany

NIVIO ZIVIANI*

*Departamento de Ciencia da Computação, UFMG,
Belo Horizonte MG 30000, Brazil*

GASTON H. GONNET†

*Department of Computer Science, University of Waterloo,
Waterloo, Ontario N2L 3G1, Canada*

KURT MEHLHORN

Universität des Saarlandes, Fachbereich 10/D-6600 Saarbrucken, West Germany

AND

DERICK WOOD‡

*Department of Computer Science, University of Waterloo,
Waterloo, Ontario N2L 3G1, Canada*

A fringe analysis method based on a new way of describing the composition of a fringe in terms of tree collections is presented. It is shown that the derived matrix recurrence relation converges to the solution of a linear system involving the transition matrix, even when the transition matrix has eigenvalues with multiplicity greater than one. As a consequence, bounds and some exact results 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, on the expected height of 2–3 trees are obtained, and improvements of the bounds on the expected number of nodes in 2–3 trees are derived also. Bounds and some exact results for 2–3 trees and B-trees using an overflow technique are obtained.

* Work was supported by a Brazilian Coordenação do Aperfeiçoamento de Pessoal de Nivel Superior Contract 4799/77 and by the University of Waterloo.

† Work supported by a Natural Sciences and Engineering Research Council of Canada Grant A-3353.

‡ Work supported by a Natural Sciences and Engineering Research Council of Canada Grant A-7700.

0019-9958/82 $2.00

Copyright © 1982 by Academic Press, Inc.
All rights of reproduction in any form reserved.

1. INTRODUCTION

Balanced search trees provide an efficient means of storing information. B-trees, 2–3 trees, 1–2 brother trees, symmetric binary *B*-trees, AVL trees, weight-balanced trees, etc, are examples of balanced search trees. These structures have been known for many years; for example, AVL trees appeared in 1962 and 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.

The fringe analysis technique is based on a method that considers only the bottom part or fringe of a tree. By considering only part of the nodes of a tree one is able to obtain bounds on most complexity measures and also some exact results. 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, whereas Yao (1978) requires that the eigenvalues be pairwise distinct.

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 the external nodes appear at the same level. The interest in B-trees has grown in the recent years to the extent that Comer (1979a) referred to them as ubiquitous. Comer (1979a, 1979b) described several systems which use B-trees.

The 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. According to this, a 2–3 tree is a B-tree of order $m = 1$, as shown in Fig. 1.1. Unlike B-trees, 2–3 trees are more appropriate for use in primary rather than secondary storage. For this reason they became equal contenders with AVL trees, often being the preferred data structure (Aho, Hopcroft, and Ullman, 1974; Huddleston and Mehlhorn 1982).

Consider a B-tree T with N keys and consequently $N + 1$ external nodes.

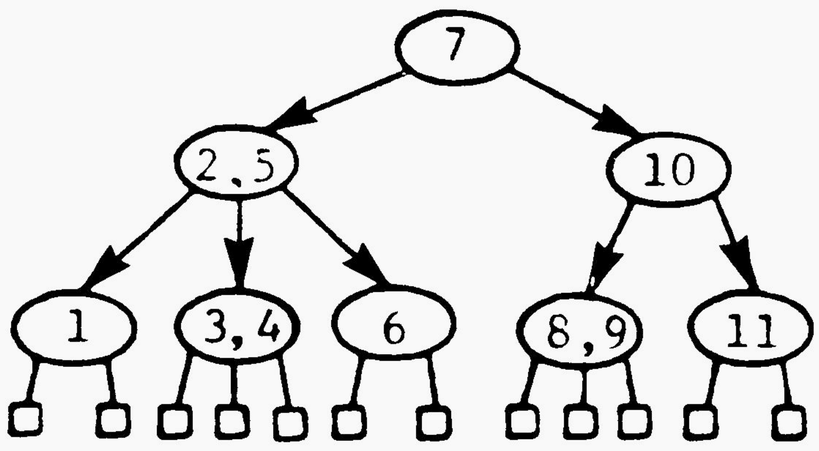

FIG. 1.1. A 2–3 tree with 11 keys.

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 these $N + 1$ intervals. A *random* B-*tree* with N keys is a B-tree tree constructed by making N successive random insertions into an initially empty tree. In this paper we assume that all trees are random trees. Random 2–3 trees are random B-trees of order 1.

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

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

(ii) When considering insertions, the most expensive 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 nth random insertion, on the average,...". We present the first partial analysis of this measure for 2–3 trees and B-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, Sect. 6.2.4). In the overflow technique, instead of splitting an overfull node, we look first as its sibling nodes and rearrange the keys when possible. The effect of the overflow technique is to produce trees with fewer internal nodes on the average, giving a better storage utilization. We present an analysis of 2–3 trees and B-trees created 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 B-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 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 on 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.

Part of the results about 2–3 trees and B-trees presented in this paper appeared in Gonnet, Ziviani, and Wood (1981), and part of the results

presented in Section 2 appeared in Eisenbarth (1981). Finally, most of the results presented in this paper appeared also in Ziviani (1982).

In Section 2 we present a fringe analysis theory containing a general analysis of the matrices that appear in fringe analysis problems. In Section 3 we present the analysis of 2–3 trees related to the four questions considered above, while in Section 4 we present a parallel but briefer analysis of B-trees. Finally in Section 5 we discuss some open problems.

2. A General Investigation of Matrices in Fringe Analysis Problems

In the first part of this section 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.

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.

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. 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.

We say that a tree collection C is *closed* if

(i) for all T in C, an insertion into T always leads to one or more members of C, and

(ii) the effect on an arbitrary tree, of an insertion, on the composition of the fringe is determined solely by the subtree of the fringe in which the insertion is performed.

FIG. 2.1.1. The tree collection of 2–3 trees of height 1.

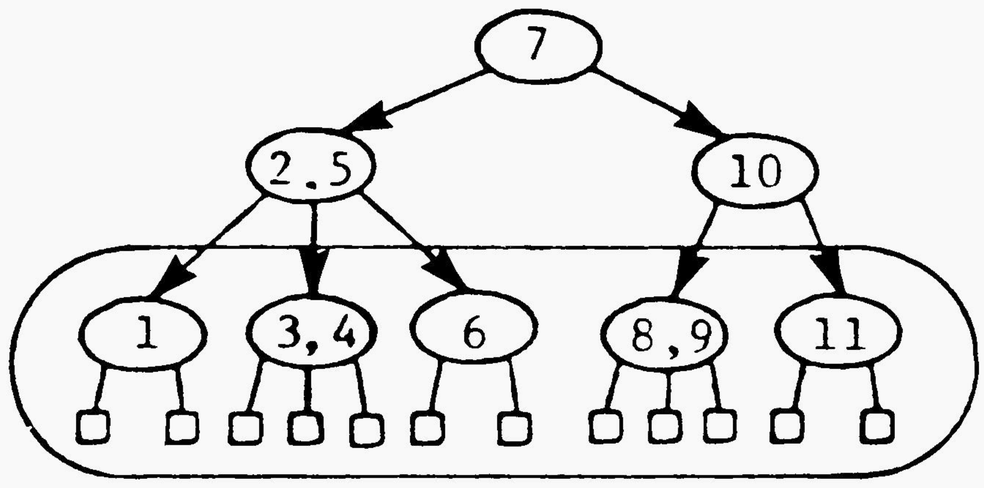

FIG. 2.1.2. A 2–3 tree and its fringe of height 1 subtrees.

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. We say a leaf is of *type i* if it belongs to a tree of type *i*. In other words, the probability that a leaf is of type *i* in a (random) 2–3 tree of $N+1$ leaves is

$$p_i(N) = \frac{\text{Expected number of leaves of type } i \text{ in an } N-\text{key tree}}{N+1}. \quad (2.1.1)$$

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. (2.1.1). As we shall see our description of the composition of the fringe simplifies the notation necessary to present the fringe analysis technique, and also eases the task of determining 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 a type 1 tree, see Fig. 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 a type 2 tree three leaves of type 2 are lost and four leaves of type 1 are obtained as a result of node splitting.

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

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

for all $i, j, j_1, \ldots, j_{N-1} \in S$. 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 we consider the tree collection of 2–3 trees of height 1 shown in Fig. 2.1.1. In this context, let X_N and Y_N be the numbers of type 1 and type 2 leaves, respectively, after the Nth insertion into an, initially empty, 2–3 tree. 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.

Since $\{X_N\}$ and $\{Y_N\}$ are Markov chains we can easily compute their transition probabilities. Consider an insertion in an N-leaf 2–3 tree, that is, the Nth insertion, then it falls into either a type 1 or a type 2 tree. This implies that X_{N-1} is either reduced by two or increased by four (by the remarks above) to give X_N. Now the probability that the former occurs is

$$\frac{\text{the number of leaves type 1}}{\text{the number of leaves}} = \frac{X_{N-1}}{N}$$

and the probability that the latters occurs is $(N - X_{N-1})/N = Y_{N-1}/N$. Thus we obtain the conditional transition probabilities

$$\Pr(X_N = X_{N-1} - 2 \mid X_{N-1}) = \frac{X_{N-1}}{N},$$

$$\Pr(X_N = X_{N-1} + 4 \mid X_{N-1}) = \frac{Y_{N-1}}{N},$$

$$\Pr(Y_N = Y_{N-1} - 3 \mid Y_{N-1}) = \frac{X_{N-1}}{N},$$

$$\Pr(Y_N = Y_{N-1} + 3 \mid Y_{N-1}) = \frac{Y_N}{N},$$

or, as they are more usually written

$$\begin{aligned} \Pr(X_N = i \mid X_{N-1} = j) &= j/N && \text{if} \quad i = j - 2, \\ &= (N - j)/N && \text{if} \quad i = j + 4 \end{aligned}$$

and

$$\begin{aligned} \Pr(Y_N = i \mid Y_{N-1} = j) &= j/N && \text{if} \quad i = j - 3, \\ &= (N - j)/N && \text{if} \quad i = j + 3. \end{aligned}$$

Now we wish to obtain the expected values of X_N and Y_N, that is, $E(X_N)$ and $E(Y_N)$. First observe, from Eq. (2.1.1), that

$$p_1(N) = E(X_N)/N + 1 \qquad \text{and} \qquad p_2(N) = E(Y_N)/N + 1.$$

Now the expected values of X_N and Y_N conditional on X_{N-1} and Y_{N-1}, can be expressed as

$$E(X_N \mid X_{N-1}, Y_{N-1}) = \frac{X_{N-1}}{N}(X_{N-1} - 2) + \frac{Y_{N-1}}{N}(X_{N-1} + 4)$$

and

$$E(Y_N \mid X_{N-1}, Y_{N-1}) = \frac{Y_{N-1}}{N}(Y_{N-1} - 3) + \frac{X_{N-1}}{N}(Y_{N-1} + 3).$$

We wish to obtain the unconditional values of X_N and Y_N, that is,

$$E(X_N) = E(E(X_N \mid X_{N-1}, Y_{N-1}))$$

and

$$E(Y_N) = E(E(Y_N \mid X_{N-1}, Y_{N-1})),$$

from which we derive

$$E(X_N) = E\left(\frac{X_{N-1}}{N}(X_{N-1} - 2) + \frac{Y_{N-1}}{N}(X_{N-1} + 4)\right),$$

which simplifies to

$$E(X_N) = E(X_{N-1}) - \frac{2E(X_{N-1})}{N} + \frac{4E(Y_{N-1})}{N}.$$

Dividing each side by $N+1$ and replacing the expectations with probabilities gives

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

and, similarly, we can derive

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

In matrix notation

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

or

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

where

$$H = \begin{pmatrix} -3 & 4 \\ 3 & -4 \end{pmatrix} \quad \text{and} \quad I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

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

$$p(N) = \left(I + \frac{H}{N+1}\right) p(N-1), \tag{2.1.2}$$

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 $\{\mathbf{X}_N\}$, 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 kth 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

$$\frac{1}{L_j} \sum_{i=1}^{m} \sum_{k=1}^{L_j} l_{ij}(k) = L_j + 1 \qquad \text{for} \quad 1 \leqslant j \leqslant m. \tag{2.2.1}$$

This leads to

DEFINITION 2.2.1. A *fringe analysis problem of site m* consists of

(i) m integers $L_1,..., L_m$,

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

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

Let $p_i(N)$ be defined as in Eq. (2.1.1). Then Eq. (2.1.2) can be written as

$$p(N) = \left(I + \frac{H_2 - H_1 - I}{N+1}\right) p(N-1), \tag{2.2.2}$$

where

$$H_2 = \left(\frac{L}{L_j} \sum_{k=1}^{L_j} l_{ij}(k)\right)_{1 \leqslant i,j \leqslant m}, \qquad H_1 = \operatorname{diag}(L_1,..., L_m),$$

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

DEFINITION 2.2.2. Consider a fringe analysis problem. Equation (2.2.2) is the associated recurrence equation, where $H = H_2 - H_1 - I = (h_{ij})$ is its transition matrix and

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

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.

DEFINITION 2.2.3. A fringe analysis is *connected* if there is an $l \in [1 \cdots m]$ such that $\det(H_u) \neq 0$, where H_u is matrix H with the lth column and lth 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 an $m \times m$ transition matrix H. Let $\lambda_1,\ldots,\lambda_m$ be the eigenvalues of H. Then they can be ordered so that $\lambda_1 = 0$ and $0 > \operatorname{Re}\lambda_2 \geqslant \operatorname{Re}\lambda_3 \geqslant \cdots \geqslant \operatorname{Re}\lambda_m$.*

Proof. Consider the sum of the elements in the jth column of H:

$$\sum_{i=1}^{m} h_{ij} = \sum_{i=1}^{m} \left(\frac{1}{L_j} \sum_{k1}^{L_j} l_{ij}(k) - \delta_{ij} L_j - \delta_{ij} \right)$$

$$= \frac{1}{L_j} \sum_{i=1}^{m} \sum_{k=1}^{L_j} l_{ij}(k) - (L_j + 1) \qquad \text{by Eq. (2.2.1)}$$

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

From Gerschgorin's theorem (see Wilkinson, 1965, Chap. 2, Sect. 13) it is known that all eigenvalues of H are contained in the union of the disks with center h_{ii} and radius $\sum_{j \neq 1} |h_{ij}|$. Since the sum of the elements in any column of H is zero, the diagonal elements are negative, and the off-diagonal elements non-negative, then

$$h_{ii} + \sum_{j \neq i} |h_{ij}| = h_{ii} + \sum_{j \neq i} h_{ij} = 0,$$

that is, each disk does not extedn into the positive half-plane for x, therefore all eigenvalues of H have non-positive real part.

From $\sum_{i=1}^{m} h_{ij} = 0$, for $1 \leqslant j \leqslant m$, 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

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

where S_q is the sum of the determinants of the principal minors of order q of the matrix H, $q = 1, 2,\ldots, m$ (see Gantmacher, 1959, Chap. 3, Sect. 7). Since $\lambda = 0$ is a solution, this implies $S_m = \det(H) = 0$, and

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

where H_{ii} is the matrix H with the ith row and the ith column deleted. H_{ii} is an $(m-1)\times(m-1)$ matrix and the Gershgorin criterion shows that all eigenvalues of H_{ii} have non-positive real part. Thus $\det(H_{ii}) = (-1)^{m-1}\,|\det(H_{ii})|$. Hence S_{m-1}, the linear term of the characteristic polynomial, is zero if and only if $\det(H_{ii}) = 0$ for all i. But $\det(H_{ii}) \neq 0$ for some i because we are dealing with a connected fringe analysis problem. Thus the linear term of the characteristic polynomial of H is non-zero, which implies that 0 is an eigenvalue of multiplicity 1. ∎

DEFINITION 2.2.4. Let $T_j \to T_i$ if $\sum_{k=1}^{L_j} l_{ij}(k) > 0$, that is 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 $j \in [1 \cdots m]$.*

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, that is $\det(H_{ii}) = 0$. Let $\mathbf{u} = (u_1,\ldots, u_{i-1}, u_{i+1},\ldots, u_m)$ be a left eigenvector of H_{ii} corresponding to eigenvalue 0. Let u_q be a component of maximal absolute value in $\mathbf{u}$ (without loss of generality $u_q = 0$) and let $J = \{j; u_j = u_q\} \subseteq [1 \cdots m] - \{i\}$. 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 $\sum_{l=1}^{m} h_{ij} = 0$ (cf. proof of Theorem 2.2.1) we have

$$\begin{aligned}\sum_{\substack{l=1\\ l\neq j}}^{m} u_i h_{ij} &= \sum_{l\in J} u_i h_{ij} + \sum_{\substack{l\notin J\\ l\neq i}} u_i h_{ij}\\ &\geqslant \sum_{l\in J} u_q h_{ij} - \sum_{\substack{l\notin J\\ l\neq i}} |u_i|\, h_{ij}\\ &> \sum_{l\neq i} h_{ij} \geqslant 0, \qquad \text{a contradiction.}\end{aligned}$$

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

Assume $\det(H_{ii}) \neq 0$. We will show $T_j \overset{*}{\to} T_i$ for all j. Assume otherwise. Then there is some j such that $T_j \overset{*}{\to} T_i$. Let $J = \{l; T_j \overset{*}{\to} T_l\}$. Then $\varnothing \neq J \neq [1 \cdots m]$ and $h_{kl} = 0$ for all $k \notin J$ and $l \in J$. We may assume without loss of generality that $J = \{1,\ldots, |J|\}$. Then H has the form

$$H = \begin{pmatrix} H' & H'' \\ 0 & H''' \end{pmatrix},$$

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 ith column and ith row deleted. But H' comes from the tran-

sition 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. (2.2.2) for connected fringe analysis problems. In a previous version of the proof of the convergence of the matrix recurrence relation (Gonnet *et al.*, 1981, Lemma 2.1, p. 4) the eigenvalues of the transition matrix are assumed to be pairwise distinct. The following theorem (Eisenbarth, 1981) extends the proof of 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 $\lambda_1 = 0 > \operatorname{Re}\lambda_2 \geqslant \operatorname{Re}\lambda_3 \geqslant \cdots \geqslant \operatorname{Re}\lambda_m$, and let x_1 be the right eigenvector of H corresponding to $\lambda_1 = 0$. Then there is a c such that for every vector $p(N)$*

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

where $p(N)$ is defined by Eq. (2.2.2).

Proof. By the ordering of the eigenvalues $N^{\operatorname{Re}\lambda_2}$ is larger than $N^{\operatorname{Re}\lambda_i}$, $i \geqslant 3$. Note also that $N^{\operatorname{Re}\lambda_1} = N^0 = 1$, for all N. Thus proving that $|p(N) - cx_1| = O(N^{\operatorname{Re}\lambda_2})$ proves that $p(N)$, as given by Eq. (2.2.2), converges to cx_1.

The proof proceeds as follows. First Eq. (2.2.2) which can be written as

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

can be further rewritten in terms of the function $f_N(x)$ defined as $\prod_{i=1}^{N} (1 + (x/i))$ yielding

$$p(N) = f_N(H)\, p(0).$$

Now the limiting value of $p(N)$, denoted by $p(\infty)$, can be expressed as

$$p(\infty) = f(H)\, p(0),$$

where $f(x)$ is the limiting value of $f_N(x)$. Moreover we show that $|f_N(x) - f(x)| = O(N^{\operatorname{Re}x})$ for $\operatorname{Re}x < 0$. To prove that $p(N)$ converges we now compute an upper bound on the value of $|p(N) - p(\infty)|$ by computing upper bounds for the elements of the matrices $f(H)$ and $f_N(H)$. In particular we find that

$$f(H) = T^{-1} \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 0 & & \\ & & \ddots & \\ 0 & \cdot & \cdots & 0 \end{pmatrix} T$$

and

$$f_N(H) = T^{-1} \begin{pmatrix} 1+\varepsilon(N) & \varepsilon(N) & \cdots & \varepsilon(N) \\ & & \cdots & \varepsilon(N) \\ & & & \varepsilon(N) \end{pmatrix} T,$$

where $\varepsilon(N) = O(N^{\mathrm{Re}\,\lambda_2})$ and T and T^{-1} are the matrices transforming H into Jordan form. We also prove that $Hp(\infty) = 0$, implying that $p(\infty)$ is a multiple of x_1, cx_1 say, since $Hx_1 = \lambda_1 x_1 = 0$ and $\lambda_1 = 0$. Combining these three facts we have

$$\begin{aligned} p(N) - p(\infty) &= p(N) - cx_1 \\ &= (f_N(H) - f(H))\, p(0) \\ &= T^{-1} \begin{pmatrix} \varepsilon(N) & \cdots & \varepsilon(N) \\ & & \vdots \\ & & \varepsilon(N) \end{pmatrix} Tp(0) \\ &= \begin{pmatrix} \delta(N) \\ \vdots \\ \delta(N) \end{pmatrix}, \end{aligned}$$

where $\delta(N) = O(N^{\mathrm{Re}\,\lambda_2})$, since $p(0)$ and T are constants. Thus $|p(N) - cx_1| = O(N^{\mathrm{Re}\,\lambda_2})$ as desired.

For $N \in \mathbf{N}$ let $f_N: \mathbf{C} \to \mathbf{C}$, where $\mathbf{C}$ is the complex plane, be given by the polynomial $f_N(x) = \prod_{i=1}^{N} (1 + (x/i))$. Let $f(x) = \lim_{N\to\infty} f_N(x)$. Then $f(0) = 1$, $f(x) = 0$, for $\mathrm{Re}\, x < 0$, and $|f(x) - f_N(x)| = O(N^{\mathrm{Re}\, x})$ for $\mathrm{Re}\, x < 0$, because

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

Furthermore, $p(N) = (I + (H/(N+1)))\, p(N-1) = f_N(H)\, p(0)$, and $p(\infty) = \lim_{N\to\infty} p(N) = f(H)\, p(0)$. (cf. Gantmacher, 1959, Chap. 5). Let

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

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 zero matrix. Also

$$J_i=\begin{pmatrix}\lambda_i & 1 & & \\ & \ddots & \ddots & \\ & & \ddots & 1\\ & & & \lambda_i\end{pmatrix} \qquad \text{with} \quad \mathrm{Re}(\lambda_i)<0,$$

where λ_i is an eigenvalue of multiplicity r_i. Since $f_N(x)$ is a polynomial in x then

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

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

$$f(J_l)=\begin{pmatrix}f(\lambda_i) & \dfrac{f(\lambda_i)}{1!} & \cdots & \dfrac{f^{(r_l-1)}(\lambda_l)}{(r_l-1)!}\\ & \ddots & \ddots & \vdots\\ & & \ddots & \dfrac{f(\lambda_i)}{1!}\\ 0 & & & f(\lambda_l)\end{pmatrix},$$

where r_l is the multiplicity of λ_l, and $f^{(k)}$ is the kth derivative of f. Hence $f(J_1)=[1]$, the 1×1 unit matrix, since $f(0)=1$, and $f(J_l)=[0]$, the $r_l\times r_l$ zero matrix, since $f(x)=0$ for $\mathrm{Re}\,x<0$.

Thus $f(H)=T^{-1}QT$, where

$$Q=\begin{pmatrix}1 & 0 & \cdots & 0\\ 0 & 0 & & \\ \vdots & & \ddots & \\ 0 & & \cdots & 0\end{pmatrix}$$

and

$$Hp(\infty)=Hf(H)\,p(0)=T^{-1}THT^{-1}QTp(0)=T^{-1}JQTp(0)$$

$$=T^{-1}0Tp(0)=\begin{pmatrix}0\\ \vdots\\ 0\end{pmatrix}$$

since $JQ=[0]$, the zero matrix.

This shows that $p(\infty)$ is a multiple of x_1, say $p(\infty)=cx_1$, because $Hx_1=\lambda_1x_1$, or $Hx_1=0$ for $\lambda_1=0$, and $Hp(\infty)=0$. Furthermore

$$f_N(H)=T^{-1}\begin{pmatrix}f_N(J_1) & & \\ & \ddots & \\ & & f_N(J_k)\end{pmatrix}T=T^{-1}\begin{pmatrix}1+\varepsilon(N) & \varepsilon(N) & \cdots & \varepsilon(N)\\ & \ddots & \varepsilon(N) & \cdots & \varepsilon(N)\\ & & \ddots & \ddots & \vdots\\ & & & \ddots & \varepsilon(N)\end{pmatrix}T,$$

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

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

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. (2.2.2), converges to the solution of

$$Hq = 0 \qquad \text{as} \quad N \to \infty, \tag{2.2.3}$$

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

$$p(N) = \alpha_1 x_1 + O(N^{\mathrm{Re}\,\lambda_2}), \tag{2.2.4}$$

where x_1 is the right eigenvector of H corresponding to eigenvalue $\lambda_1 = 0$. 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. (2.2.1) can be written as

$$p_i(N) = \frac{A_i(N)\, L_i}{N+1}. \tag{2.2.5}$$

3. An Analysis of 2–3 Trees

3.1. *Motivation*

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

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 $\Pr\{j \text{ splits}\}$ be the probability that j splits occur on the $(N+1)$th random insertion into a random 2–3 tree with N keys.

(iii) Let $\Pr\{j \text{ or more splits}\}$ be the probability that j or more splits occur on the $(N+1)$th 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 $(N+1)$th insertion into a random 2–3 tree with N keys.

(vi) Let Pr(dsn at jth lowest level} be the probability that the deepest safe node on a random search is located at the jth ($j \geqslant 1$) lowest level of a random 2–3 tree with N keys.

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

In Subsections 3.2, 3.3, and 3.4 we shall derive exact values for $\Pr\{0 \text{ splits}\}$, $\Pr\{1 \text{ split}\}$, $\Pr\{2 \text{ splits}\}$, $\Pr\{3 \text{ or more splits}\}$, and bounds on $\bar{s}(N)$, $E[s(N)]$, and improve Yao's previous results on $\bar{n}(N)$. In Subsection 3.5 we shall derive exact values for $\Pr\{0 \text{ splits}\}$, $\Pr\{1 \text{ split}\}$, 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 Subsection 3.6 we shall derive exact values for Pr{dsn at 1st lowest level}, Pr{dsn at 2nd lowest level}, Pr{dsn at 3rd lowest level}, and Pr{dsn above 3rd lowest level} for the normal insertion algorithm, and Pr{dsn at 1st lowest level}, Pr{dsn at 2nd lowest level}, and

TABLE 3.1.1.

Summary of the 2–3 Tree Results

	First Order Analysis ($N \geqslant 6$)	Second Order[a] Analysis ($N \to \infty$)	Third Order[b] Analysis ($N \to \infty$)
$\bar{n}(N)$	$[0.64N + 0.14, 0.86N - 0.14]$	$[0.70N + 0.20, 0.79N - 0.21]$	$[0.73N + 0.23, 0.77N - 0.23]$
Pr{0 splits}	$\frac{4}{7}$	$\frac{4}{7}$	$\frac{4}{7}$
Pr{1 or more splits}	$\frac{3}{7}$	$\frac{3}{7}$	$\frac{3}{7}$
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
$\bar{s}(N)$	$[0.64 + 0.14/N - \lceil \log_3(N+1) \rceil / N,$ $0.86 - 14/N - \lfloor \log_2(N+1) \rfloor / N]$	$[0.70 + 0.20/N - \lceil \log_3(N+1) \rceil / N,$ $0.79 - 0.21/N - \lfloor \log_2(N+1) \rfloor / N]$	$[0.73 + 0.23/N - \lceil \log_3(N+1) \rceil / N,$ $0.77 - 0.23/N - \lfloor \log_2(N+1) \rfloor / N]$
$E[s(N)]$	$[0.43, 0.43\lfloor \log_2(N+1) \rfloor]$	$[0.61, 0.25 + 0.18\lfloor \log_2(N+1) \rfloor]$	$[0.69, 0.46 + 0.08\lfloor \log_2(N+1) \rfloor]$
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 1st lowest level}	$\frac{4}{7}$	$\frac{4}{7}$	$\frac{4}{7}$
Pr{dsn at 2nd lowest level}	—	0.25	0.25
Pr{dsn at 3rd lowest level}	—	—	0.10
Pr{dsn above 3rd l. level}	—	—	0.08

[a] Results are approximated to $O(N^{-6.55})$.

[b] Results are approximated to $O(N^{-4.37})$.

TABLE 3.1.2

Summary of the 2–3 Tree Results Using an Overflow Technique

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

[a] Results are approximated to $O(N^{-6.81})$.

Pr{dsn above 2nd lowest level} for the insertion algorithm using an overflow technique. In Subsection 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 Fig. 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 $p_1(\infty) = \frac{4}{7}$, and $p_2(\infty) = \frac{3}{7}$. Since the eigenvalues of H are 0 and -7, we observe that

FIG. 3.2.1. The tree collection of 2–3 trees of height 1.

$p_1(N) = \frac{4}{7}$ and $p_2(N) = \frac{3}{7}$ for $N \geqslant 6$. To simplify notation $p_i(N)$ is written as p_i throughout the remainder of this paper.

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*

$$\frac{\text{nl} - 1}{2} \leqslant \text{nal} \leqslant \text{nl} - 1.$$

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 $2\text{nal} = \text{nl} - 1$ and $\text{nal} = \text{nl} - 1$ respectively.) ∎

Lemma 3.2.1 and Eq. (2.2.5) lead to

THEOREM 3.2.2. *The expected number of nodes in a random* 2–3 *tree with N keys is bounded by*

$$\left(1 + \frac{1}{2}\right)\left[\frac{p_1}{L_1} + \frac{p_2}{L_2}\right](N + 1) - \frac{1}{2}$$
$$\leqslant \bar{n}(N) \leqslant 2\left[\frac{p_1}{L_1} + \frac{p_2}{L_2}\right](N + 1) - 1 \qquad \textit{for} \quad N \geqslant 1$$

that is,

$$\frac{9N}{14} + \frac{1}{7} \leqslant \bar{n}(N) \leqslant \frac{6N}{7} - \frac{1}{7} \qquad \textit{for} \quad N \geqslant 6.$$

The remaining results are contained in the lemmas that follow.

LEMMA 3.2.3. *The probability that no split occurs on the* $(N + 1)$th *random insertion into a* 2–3 *tree with N keys is*

$$\Pr\{0 \text{ splits}\} = \tfrac{4}{7} \qquad \textit{for} \quad N \geqslant 6.$$

Proof. An insertion into a type 1 tree shown in Fig. 3.2.1 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* $(N + 1)$th *random insertion into a* 2–3 *tree with N keys is*

$$\Pr\{1 \text{ or more splits}\} = \tfrac{3}{7} \qquad \textit{for} \quad N \geqslant 6.$$

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 Section 4 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*

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

Proof. Two lower bound and the upper bound on the height are obtained when each node of the 2–3 tree contains 2 keys and 1 key, respectively. ∎

Lemmas 3.2.5 and 3.2.6 lead to

THEOREM 3.2.7. *The expected number of splits in a random 2–3 tree with N keys is bounded by*

$$\frac{9}{14} + \frac{1}{7N} - \frac{\lfloor \log_2(N+1) \rfloor}{N} \leqslant \bar{s}(N) \leqslant \frac{6}{7} - \frac{1}{7N} - \frac{\lceil \log_3(N+1) \rceil}{N} \quad \text{for} \quad N \geqslant 6.$$

LEMMA 3.2.8. *A lower bound on the expected number of splits that occur on the $(N+1)$th insertion into a random 2–3 tree with N keys is*

$$E[s(N)] \geqslant \Pr\{1 \text{ or more splits}\}.$$

Proof. Similar to the proof of Lemma 3.2.3. ∎

COROLLARY. $E[s(N)] \geqslant \frac{3}{7}$ *for* $N \geqslant 6$.

LEMMA 3.2.9. *An upper bound on the expected number of splits that occur on the $(N+1)$th insertion into a random 2–3 tree with N keys is*

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

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

THEOREM 3.2.10. *The expected number of splits that occur on the* $(N+1)$th *insertion into a random 2–3 tree with N keys is bounded by*

$$\tfrac{3}{7} \leqslant E[s(N)] \leqslant \tfrac{3}{7}\lfloor \log_2(N+1) \rfloor \qquad \textit{for} \quad N \geqslant 6.$$

One may be tempted to conjecture that the expected value for $E[s(N)]$ converges to the value of $\bar{s}(N)$. However, we cannot prove this. For example, $E[s(N)]$ could 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.

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 Fig.* 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 Fig. 3.2.1 and by using Eq. (2.2.5). ∎

COROLLARY. $\bar{f}(N) = \tfrac{4}{7}(N+1)$ *for* $N \geqslant 6$.

THEOREM 3.2.12. *The excepted height of a* 2–3 *tree with N keys is bounded above by*

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

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

$$\lceil \log_3(\text{nkal}+1) \rceil + 1 \leqslant h(N) \leqslant \lfloor \log_2(\text{nkal}+1) \rfloor + 1.$$

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

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

Using Jensen's inequality (Feller, 1966, p. 152) we obtain

$$\bar{h}(N) \leqslant \log_2 E[\text{nkal}+1] + 1. \tag{3.2.1}$$

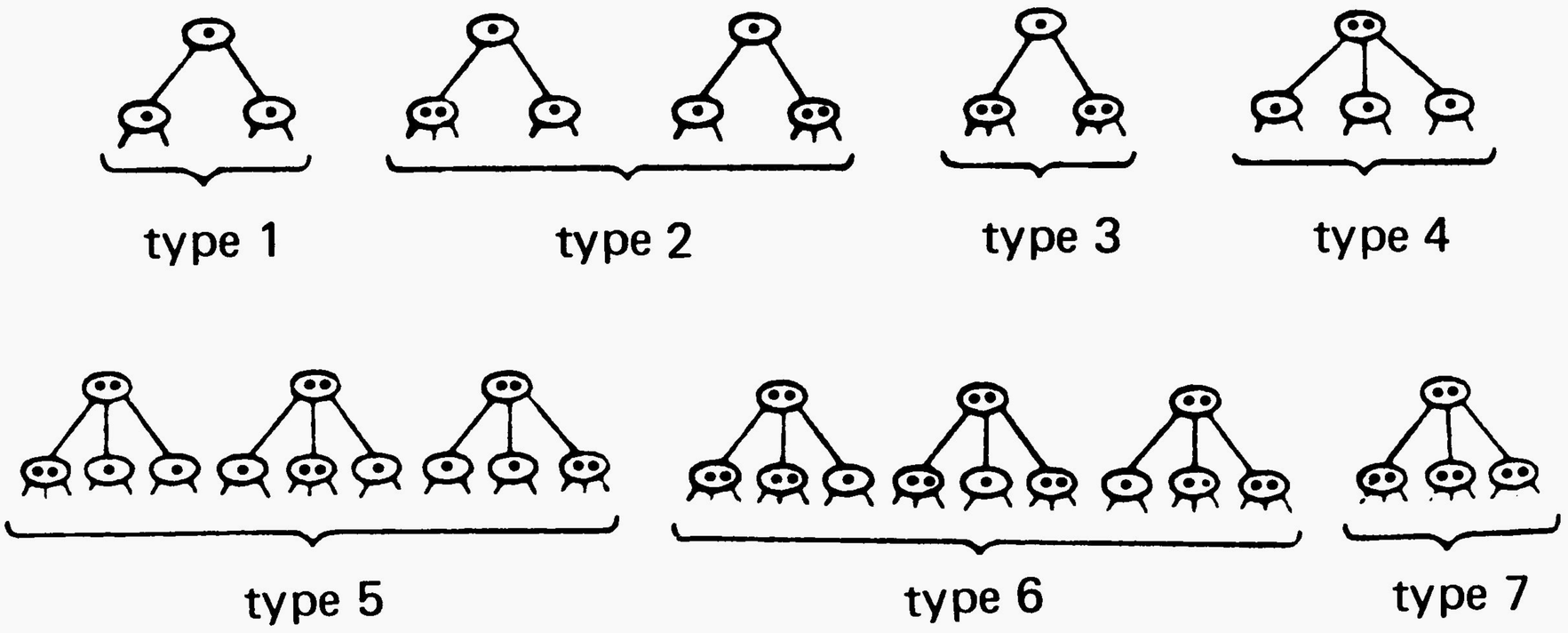

FIG. 3.3.1. The tree collection of 2–3 trees of height 2; stubs indicate leaves.

But

$$E[\text{nkal}] = N - \bar{f}(N),$$

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

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

Substituting this equation into Eq. (3.2.1) we obtain

$$\bar{h}(N) \leqslant \log_2(N+1) - 0.22239. \quad \blacksquare$$

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.

TABLE 3.3.1

The Transition Matrix Corresponding to the Tree Collection of 2–3 Trees of Height 2 of Fig. 3.3.1

$$\begin{vmatrix}
-5 & & & & 8\times 3/7 & 4\times 6/8 & 4\times 6/9 \\
5 & -6 & & & & 5\times 6/8 & 5\times 6/9 \\
 & 6\times 2/5 & -7 & & & & 6\times 6/9 \\
 & 6\times 3/5 & & -7 & & & \\
 & & 7 & 7 & & & \\
 & & & & 8\times 4/7 & -9 & \\
 & & & & & 9\times 2/8 & -10
\end{vmatrix}$$

Again using Eq. (2.2.3) we obtain

$$\begin{aligned} p_1 &= 1656/7991, & p_5 &= 1575/7991, \\ p_2 &= 1980/7991, & p_6 &= 800/7991, \\ p_3 &= 5472/55937, & p_7 &= 180/7991. \\ p_4 &= 7128/55937, \end{aligned} \tag{3.3.1}$$

Since the eigenvalues of H are 0, $-6.55 \pm 6.25i$, -7, $-9.23 \pm 1.37i$, and -13.44, using Eq. (2.2.3) the asymptotic values of $p(N)$ obtained from Eq. (2.2.4) are approximated to the $O(N^{-6.55})$.

THEOREM 3.3.1. *The expected number of nodes in a random* 2–3 *tree with N keys is bounded, to five decimal places, by*

$$0.70169N + 0.20169 + O(N^{-5.55}) \leqslant \bar{n}(N) \leqslant 0.79273N - 0.20727 + O(N^{-5.55}).$$

Proof. Lemma 3.2.1 and Eq. (2.2.5) lead to

$$\left\{\left(3+\frac{1}{2}\right)\left(\sum_{i=1}^{3}\frac{p_i}{L_i}\right)+\left(4+\frac{1}{2}\right)\left(\sum_{i=4}^{7}\frac{p_i}{L_i}\right)\right\}(N+1)-\frac{1}{2}$$

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

or, alternatively

$$\frac{78501N}{111874}+\frac{11282}{55937}+O(N^{-5.55})$$

$$\leqslant \bar{n}(N) \leqslant \frac{44343N}{55937}-\frac{11594}{55937}+O(N^{-5.55}). \quad \blacksquare$$

LEMMA 3.3.2. *The probability that* 1 *split occurs on the* $(N+1)$th *random insertion into a* 2–3 *tree with N keys is*

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

Proof. An insertion into the type 2 tree shown in Fig. 3.3.1 causes one split $\frac{3}{5}$ of the time, and an insertion into the type 3 shown in Fig. 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 $\Pr\{1 \text{ split}\} = 3/5p_2 + p_3$. $\blacksquare$

LEMMA 3.3.3. *The probability that* 2 *or more splits occur on the* $(N+1)$th *random insertion into a* 2–3 *tree with* N *keys is*

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

Proof. Similar to the proof of Lemma 3.3.2. ∎

Lemma 3.2.5 leads to

THEOREM 3.3.4. *The expected number of splits in a random* 2–3 *tree with* N *keys is bounded by*

$$\tfrac{78501}{111874} + \tfrac{11282}{55937N} - (\lfloor \log_2(N+1) \rfloor / N) + O(N^{-6.55}) \leqslant \bar{s}(N)$$
$$\leqslant \tfrac{44343}{55937} - \tfrac{11594}{55937N} - (\lceil \log_3(N+1) \rceil / N) + O(N^{-6.55}),$$

and to five decimal places we have

$$0.70169 + \frac{0.20169}{N} - \frac{\lfloor \log_2(N+1) \rfloor}{N} + O(N^{-6.55}) \leqslant \bar{s}(N)$$
$$\leqslant 0.79273 - \frac{0.20727}{N} - \frac{\lceil \log_3(N+1) \rceil}{N} + O(N^{-6.55}).$$

LEMMA 3.3.5. *A lower bound on the expected number of splits that will occur on the* $(N+1)$th *insertion into a random* 2–3 *tree with* N *keys is*

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

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* $(N+1)$th *insertion into a random* 2–3 *tree with* N *keys is*

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

Proof. Similar to the proof of Lemma 3.2.8. ∎

Lemmas 3.3.5 and 3.3.6 lead to

THEOREM 3.3.7. *The expected number of splits that will occur on the* $(N+1)$th *insertion into a random* 2–3 *tree with* N *keys is bounded by*

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

and to five decimal places we have

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

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 Fig.* 3.3.1 *is* $\bar{f}(N) = \frac{6536}{7991}(N+1) + O(N^{-6.55})$.

Proof. The equation

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

is obtained from Fig. 3.3.1 and Eq. (2.2.5). ∎

THEOREM 3.3.9. *The expected height of a* 2–3 *tree with* N *keys is bounded above by*

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

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 (1979) 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. Obviously solving the recurrence for an H which is 147×147 is preferable to solving it for an H which is 978×978.

The idea behind our approach is to group all trees with the same number of leaves into types. Thus the tree collection shown in Fig. 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 Fig. 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 Fig. 3.4.1 is the sum of the probabilities

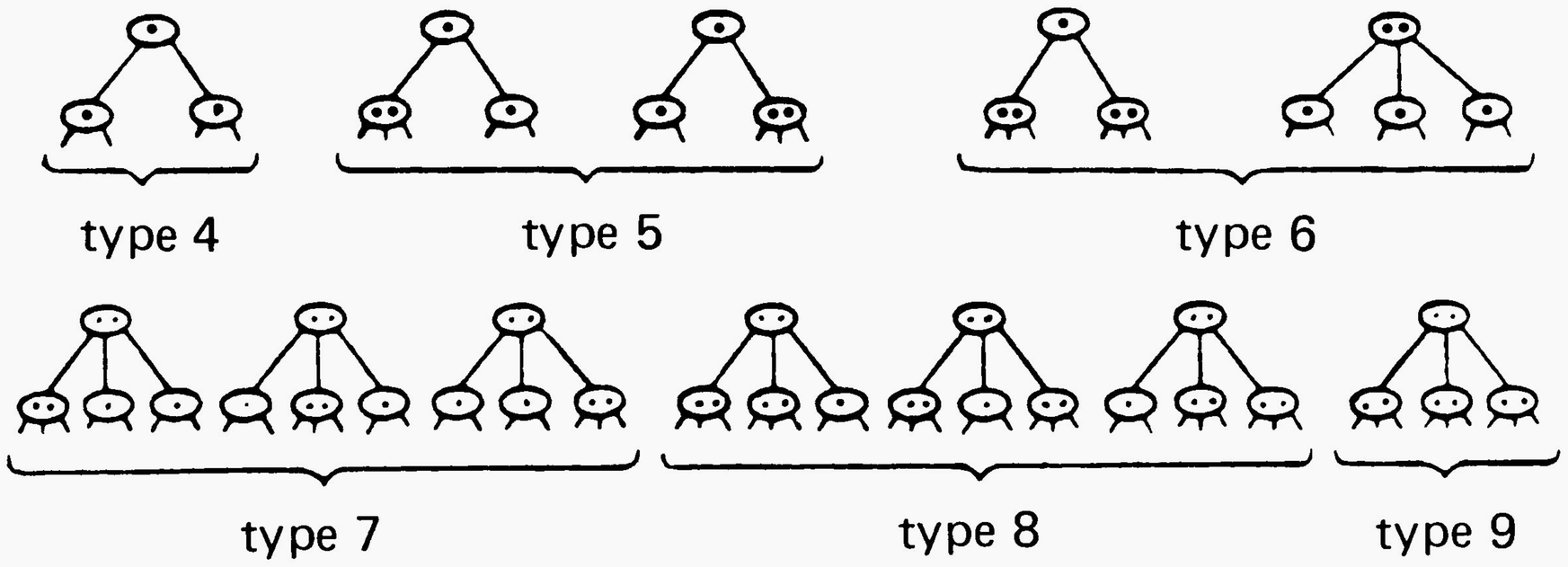

FIG. 3.4.1. A tree collection of 2–3 trees of height 2 obtained by grouping types 3 and 4 trees of Fig. 3.3.1 into type 6.

related to the types 3 and 4 shown in Fig. 3.3.1, and the probabilities of the other types remain as before. (Types 4, 5, 7, 8, and 9 shown in Fig. 3.4.1 have the same probabilities as types 1, 2, 5, 6, and 7 shown in Fig. 3.3.1, respectively.)

LEMMA 3.4.1. *The two level tree collection of Fig.* 3.4.1 *can be used to obtain a three level tree collection which is closed.*

Proof. Simply consider the trees obtained by hanging the two-level trees of Fig. 3.4.1 from a binary or ternary node. The resulting collection is clearly closed. ∎

LEMMA 3.4.2. *The two level tree collection of Fig.* 3.4.1 *can be used to form a closed three level* 2–3 *tree collection with* 147 *types.*

Proof. Following the notation presented in Fig. 3.4.2, the 147 types of

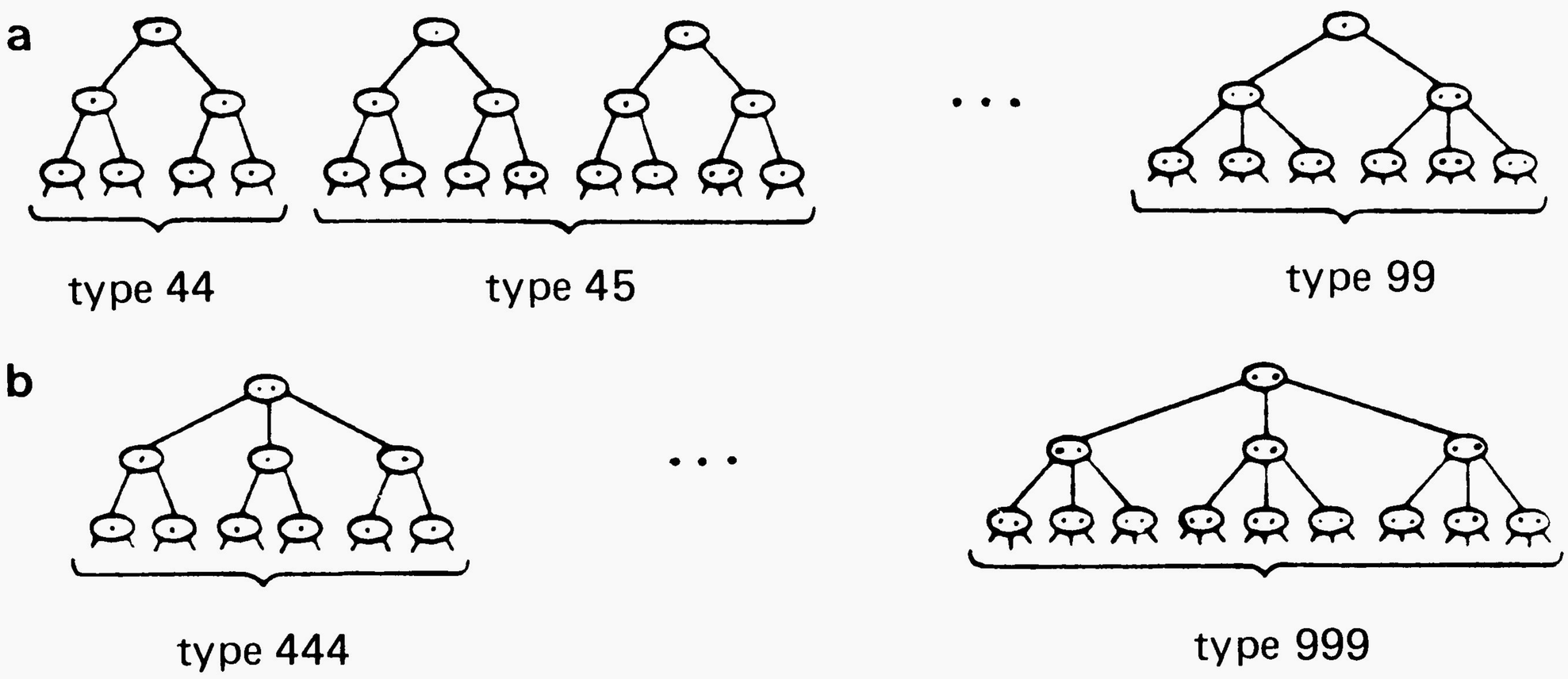

FIG. 3.4.2. A tree collection of 2–3 trees of height 3 (type 44 is formed by two subtrees with 4 leaves each, type 45 is formed by two subtrees with 4 and 5 leaves each, etc.) (a) Types formed by two height 2 subtrees under binary roots; there are 21 types in this case. (b) Types formed by three height 2 subtrees under ternary roots; there are 126 types in this case.

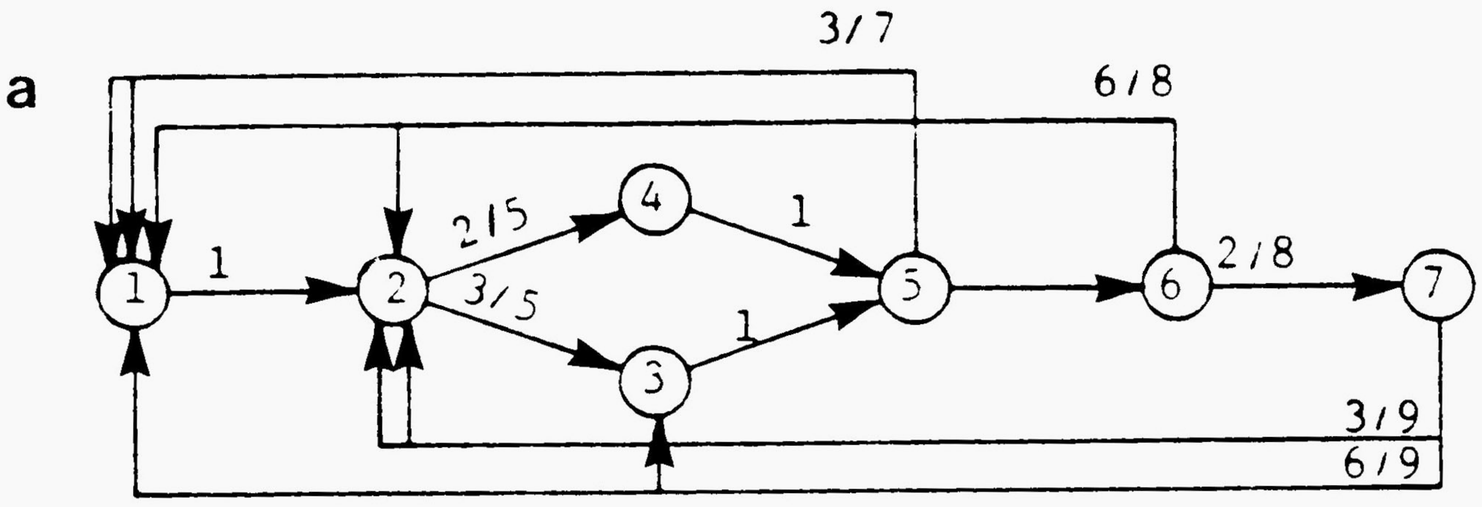

FIG. 3.4.3. Diagrams for transitions. (a) Transitions related to the tree collection shown in Fig. 3.3.1. (b) Transitions related to the tree collection shown in Fig. 3.4.1.

the three level tree collection are represented either as type ij ($4 \leqslant i \leqslant 9$ and $i \leqslant j \leqslant 9$) for the tree types with binary roots, or as type ijk ($4 \leqslant i \leqslant 9$, $4 \leqslant j \leqslant 9$, and $i \leqslant k \leqslant 9$) 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 $4 \leqslant j \leqslant 9$ (and not $i \leqslant j \leqslant 9$ and $j \leqslant k \leqslant 9$). 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 495 gives types 44 and 56, and an insertion into the leftmost leaf of the right subtree of type 459 gives types 45 and 46.

LEMMA 3.4.3. *The transitions related to the* 6 *types of the tree collection shown in Fig.* 3.4.1 *are equivalent to the transitions related to the* 7 *types of the tree collection shown in Fig.* 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 3.4.3a and b show the transitions related to the tree collections shown in Figs. 3.3.1 and 3.4.1, respectively. It is irrelevant whether we use the 6 types of the tree collection shown in Fig. 3.4.1 or the 7 types of the tree collection shown in Fig. 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 Fig. 3.4.3b is composed by types 3 and 4 shown in Fig. 3.4.3a, and (ii) from Eq. (3.3.1) that types 3 and 4 shown in Fig. 3.4.3a occur with probabilities 472/55937 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 Fig.* 3.4.1. *Then*

$$\mathrm{nn}(i) = 3 \qquad \textit{for} \quad 4 \leqslant i \leqslant 5,$$

$$\mathrm{nn}(6) 3 \times \tfrac{5472}{12600} + 4 \times \tfrac{7128}{12600},$$

$$\mathrm{nn}(i) = 4 \qquad \textit{for} \quad 7 \leqslant i \leqslant 9.$$

Proof. For $i = 4, 5, 7, 8, 9$, from Fig. 3.4.1 the values for nn(i) are immediate. For $i = 6$, consider the two trees of type 6 shown in Fig. 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

$$\mathrm{nn}(6) = 3 \times \tfrac{5472}{12600} + 4 \times \tfrac{7128}{12600} \quad \blacksquare$$

Let L_{ij} indicate the number of leaves of the type ijk tree ($4 \leqslant i \leqslant 9$, $i \leqslant j \leqslant 9$) shown in Fig. 3.4.2. Let L_{ijk} indicate the number of leaves of the type ijk tree ($4 \leqslant i \leqslant 9$, $4 \leqslant j \leqslant 9$, $i \leqslant k \leqslant 9$) shown in Fig. 3.4.2. The proof of Theorem 3.4.5 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 ($4 \leqslant i \leqslant 9$), plus the number of nodes of type j ($i \leqslant j \leqslant 9$), plus the binary root node (see Figs. 3.4.1 and 3.4.2), plus $\frac{1}{2}$ 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*[1]

$$0.72683N + 0.22683 + O(N^{-3.37})$$

$$\leqslant \bar{n}(N) \leqslant 0.76556N - 0.23444 + O(N^{-3.37})$$

[1] 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

$$\frac{77985993142909130805284072722195623462256367325297938181937688420653733745297135574577340 66}{10729604856083907760988691252514032168089885375054384827047705340026365840593873897782021229}$$

$$= 0.72683\ 00574\ 80536\ldots .$$

Proof. The above remarks lead to

$$\left[\sum_{i=4}^{9}\sum_{j=i}^{9}(\mathrm{nn}(i)+\mathrm{nn}(j)+1+\tfrac{1}{2})(p_{ij}/L_{ij})\right.$$
$$\left.+\sum_{i=4}^{9}\sum_{j=4}^{9}\sum_{k=i}^{9}(\mathrm{nn}(i)+\mathrm{nn}(j)+\mathrm{nn}(k)+1+\tfrac{1}{2})(p_{ijk}/L_{ijk})\right](N+1)-\tfrac{1}{2}$$
$$\leqslant \bar{n}(N) \leqslant \left[\sum_{i=4}^{9}\sum_{j=i}^{9}(\mathrm{nn}(i)+\mathrm{nn}(j)+2)(p_{ij}/L_{ij})\right.$$
$$\left.+\sum_{i=4}^{9}\sum_{j=4}^{9}\sum_{k=i}^{9}(\mathrm{nn}(i)+\mathrm{nn}(j)+\mathrm{nn}(k)+2)(p_{ijk}/L_{ijk})\right](N+1)-1. \quad \blacksquare$$

Experimental results show that $\bar{n}(N)$ is approximately $0.75N$. The maximum and the maximum number of internal nodes possible in any 2–3 tree with N keys are $0.5N$ and N, respectively.

LEMMA 3.4.6. *The probability that* 2 *splits occur on the* $(N+1)$th *random insertion into a* 2–3 *tree with* N *keys is*

$$\Pr\{2 \textit{ splits}\} = 0.1046 + O(N^{-4.37}).$$

Proof. Similar to the proof of Lemma 3.3.2. ∎

LEMMA 3.4.7. *The probability that* 3 *or more splits occur on the* $(N+1)$th *random insertion into a* 2–3 *tree with* N *keys is*

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

Proof. Similar to the proof of Lemma 3.3.2. ∎

Lemma 3.2.5 leads to

THEOREM 3.4.8. *The expected number of splits in a random* 2–3 *tree with* N *keys is bounded by*

$$0.72683 + \frac{0.22683}{N} - \frac{\lfloor \log_2(N+1) \rfloor}{N} + O(N^{-4.37})$$
$$\leqslant \bar{s}(N) \leqslant 0.76556 - \frac{0.23444}{N} - \frac{\lceil \log_3(N+1) \rceil}{N} + O(N^{-4.37}).$$

LEMMA 3.4.9. *A lower bound on the expected number of splits that will occur on the* $(N+1)$th *insertion into a random* 2–3 *tree with* N *keys is*

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

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* $(N+1)$th *insertion into a random 2–3 tree with N keys is*

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

Proof. Similar to the proof of Lemma 3.2.8. ∎

Lemmas 3.4.9. and 3.4.10 lead to

THEOREM 3.4.11. *The expected number of splits that will occur on the* $(N+1)$th *insertion into a random 2–3 tree with N keys is bounded by*

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

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 Fig. 3.4.2 is* $\bar{f}(N) = 0.92255(N+1) + O(N^{4.37})$.

Proof.

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

is obtained from Fig. 3.4.2 and Eq. (2.2.5). ∎

THEOREM 3.4.13. *The expected height of a 2–3 tree with N keys is bounded above by*

$$\bar{h}(N) \leqslant \log_2(N+1) - 0.69054.$$

Proof. Similar to the proof of Theorem 3.2.12. ∎

It is important to note that the values for $\bar{n}(N)$, $\bar{s}(N)$, $E[s(N)]$, $\Pr\{j \text{ splits}\}$, and $\Pr\{j \text{ or more splits}\}$ 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.

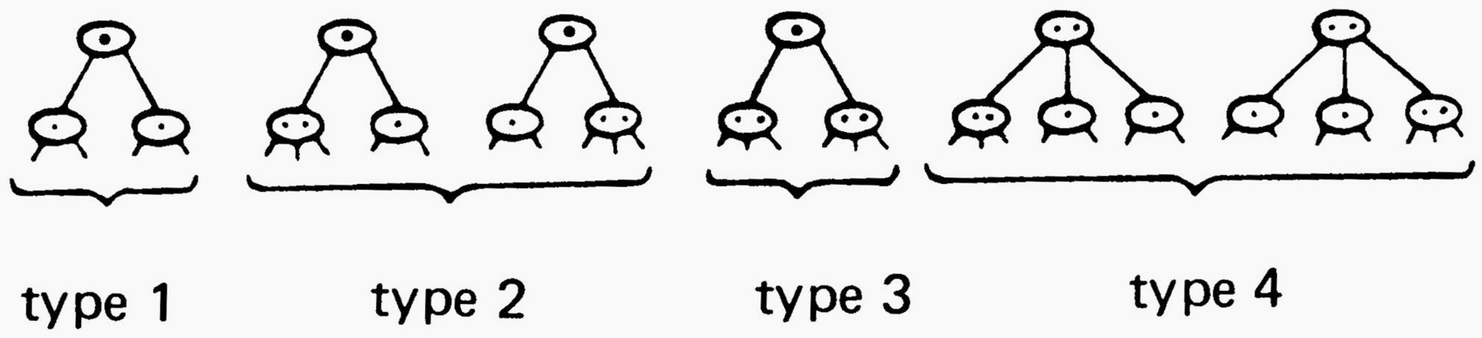

FIG. 3.5.1. Tree collection of 2–3 trees of height 2 using overflow technique.

3.5. 2–3 *Trees with an 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. If this node is the middle node of a ternary subtree then the closest non-full brother may be located either to the right or to the left of it. 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.

TABLE 3.5.1

Transition Matrix Corresponding to the Tree Collection of 2–3 Trees of Height 2 Shown in Fig. 3.5.1

−5					$4 \times 3/8$		$4 \times 6/9$
5	−6				$5 \times 3/8$		$10 \times 3/9$
	6	−7					$6 \times 6/9$
		7	−8				
			$8 \times 5/7$	−9			
			$8 \times 2/7$			−9	
					$9 \times 5/8$	9	−10

Using Eq. (2.2.3) we obtain

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

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 $O(N^{-6.81})$.

THEOREM 3.5.1. *The expected number of nodes in a random 2–3 tree with N keys is bounded by*

$$0.63248N + 0.13248 + O(N^{-5.81}) \leqslant \bar{n}(N) \leqslant 0.71384N - 0.28616 + O(N^{-5.81}).$$

Proof. Lemma 3.2.1 and expression Eq. (2.2.5) lead to

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

which in turn gives

$$\tfrac{20175N}{31989} + \tfrac{2113}{15949} + O(N^{-5.81}) \leqslant \bar{n}(N) \leqslant \tfrac{11385N}{15949} - \tfrac{4564}{15949} - O(N^{-5.81}). \quad \blacksquare$$

This estimate should be compared to

$$0.72683N + 0.22683 + O(N^{-3.37})$$
$$\leqslant \bar{n}(N) \leqslant 0.76556N - 0.23444 + O(N^{-3.37}),$$

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

LEMMA 3.5.2. *The probabilities that no split, 1 split, and 2 or more splits occur on the* $(N+1)$th *insertion into a 2–3 tree with N keys using an overflow technique are, respectively,*

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

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

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

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

Lemma 3.2.5 leads to

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*

$$\tfrac{20175}{31898} + \tfrac{2113}{15949N} - (\lfloor \log_2(N+1) \rfloor / N) + O(N^{-6.81}) \leqslant \bar{s}(N) \leqslant \tfrac{11385}{15949} - \tfrac{4564}{15949N} - (\lceil \log_3(N+1) \rceil / N) + O(N^{-6.81}).$$

To five place decimals we have

$$0.63248 + \frac{0.13248}{N} - \frac{\lfloor \log_2(N+1) \rfloor}{N} + O(N^{-6.81}) \leqslant \bar{s}(N) \leqslant 0.71384 - \frac{0.28616}{N} - \frac{\lceil \log_3(N+1) \rceil}{N} + O(N^{-6.81}),$$

which should be compared to the bounds

$$0.72683 + \frac{0.22683}{N} - \lfloor \log_2 \frac{(N+1)}{N} + O(N^{-4.37}) \leqslant \bar{s}(N) \leqslant 0.76556 - \frac{0.23444}{N} - \frac{\lceil \log_3(N+1) \rceil}{N} + O(N^{-4.37}),$$

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* $(N+1)$th *insertion into a random* 2–3 *tree with N keys using an overflow technique is*

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

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* $(N+1)$th *insertion into a random* 2–3 *tree with N keys using an overflow technique is*

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

Proof. Similar to the proof of Lemma 3.2.8. ∎

Lemmas 3.5.4 and 3.5.5 lead to

THEOREM 3.5.6. *The expected number of splits that will occur on the* $(N+1)$th *insertion into a random* 2–3 *tree with* N *keys using an overflow technique is bounded by*

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

To five place decimals we have

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

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 Subsections 3.2, 3.3, and 3.4.

LEMMA 3.6.1.1. *The probabilities that the deepest safe node is located at the* 1st, *the* 2nd, *and the* 3rd *lowest level, and above the* 3rd *lowest level of a* 2–3 *tree with* N *keys are, respectively,*

(a) $\Pr\{\text{dsn } at \text{ 1st } lowest\ level\} = \frac{4}{7}$,

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

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

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

Proof. It is not difficult to see that the probability that the deepest safe node is located at jth $(j \geqslant 1)$ lowest level is equal to the probability that exactly $j-1$ splits occur on the $(N+1)$th random insertion (see Lemmas 3.2.3, 3.3.2, 3.4.6, and 3.4.7 for the proof of items (a)–(d), respectively). ∎

From Lemma 3.6.1.1(d), we can see that in only 8% of the time the deepest safe node is above the 3rd 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 Subsection 3.5.

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

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

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

(c) $\Pr\{\text{dsn } above \text{ 2nd } lowest\ level\} = \frac{2595}{15949} + 1 \quad (N^{-6.81})$.

Proof. Similar to the proof of Lemma 3.6.1.1 (see Lemma 3.5.2 in Subsection 3.5 for the proof of items (a)–(c)). ∎

3.7. *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 $k \geqslant 4$. However, if we apply the same technique used to obtain the three level tree collection with 147 types then it is possible to consider a fourth order analysis.

In order to obtain a four level tree collection we define a 20 type 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 Fig. 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. Table 3.7.1 compares the sizes of the tree collections used by Yao, Brown, and ourselves, for various levels of analysis.

TABLE 3.7.1

Sizes of the Tree Collections Used by Brown (1979, p. 57), Yao (1978, p. 165), and in this Paper

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

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 jth $(j \geqslant 1)$ lowest level of a random 2–3 tree with N keys. Then*

(i) $l_1 = N + 1$,

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

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

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

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)–(iv) we just count the number of trees in the fringe that correspond to the three collection of Figs. 3.2.1, 3.3.1, and 3.4.2, respectively. These yield

$$l_2 = \left(\frac{p_i}{L_1} + \frac{p_2}{L_2}\right)(N + 1), \qquad l_3 = \left(\sum_{i=1}^{7} \frac{p_i}{L_i}\right)(N + 1),$$

$$l_4 = \left(\sum_{i=4}^{9} \sum_{j=i}^{9} \frac{p_{ij}}{L_{ij}} + \sum_{i=4}^{9} \sum_{j=4}^{9} \sum_{k=i}^{9} \frac{p_{ijk}}{L_{ijk}}\right)(N + 1). \quad \blacksquare$$

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

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

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

TABLE 3.7.2

Ratio of the Expected Numbers of Nodes at Two Consecutive Levels

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

4. An Analysis of B-Trees

4.1. *Motivation*

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.[2] 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.

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

Let $\bar{n}_m(N)/[N/(2m)]$ 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 Subsection 4.2 we shall derive exact values for $\Pr\{0 \text{ splits}\}_m$, $\Pr\{1 \text{ or more splits}\}_m$, 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. It is convenient throughout Section 4 to refer to $n_m(N)/N$ rather than $\bar{n}_m(N)$ alone. In Subsection 4.3 we shall derive exact values for $\Pr\{0 \text{ splits}\}_m$, $\Pr\{1 \text{ split}\}_m$, $\Pr\{1 \text{ or more splits}\}_m$, $\Pr\{2 \text{ or more splits}\}_m$, 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 Subsection 4.4 we shall derive exact values for $\Pr\{\text{dsn at 1st lowest level}\}_m$ and $\Pr\{\text{dsn above 1st lowest level}\}_m$ for the normal insertion algorithm, and $\Pr\{\text{dsn at 1st lowest level}\}_m$, $\Pr\{\text{dsn at 2nd lowest level}\}_m$, and $\Pr\{\text{dsn above 2nd lowest level}\}_m$ for the insertion algorithm using an overflow technique.

[2] 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 $m - 1$ and at least $\lceil m/2 - 1 \rceil$ keys. Knuth's definition considers B-trees of order $2i$, $i \geqslant 2$ (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.

TABLE 4.1.1
Summary of the B-Tree Results

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

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 $m = 3$.

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

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

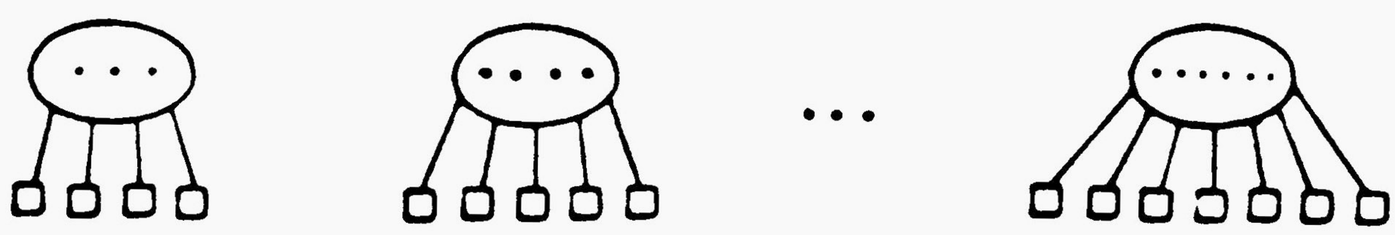

FIG. 4.2.1. Tree collection of B-tree of order $m = 3$ and height 1.

TABLE 4.1.2

Summary of the B-Tree Results Using an Overflow Technique

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

Let H_n denote the harmonic numbers, $H_n = \sum_{i=1}^{n} 1/i$, for $n \geqslant 1$. From Eq. (2.2.3) we have $Hp(N) = 0$, and therefore

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

$$p_{m+2} = \frac{1}{(m+3)(H_{2m+2} - H_{m+1})} \tag{4.2.1}$$

$$\vdots$$

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

LEMMA 4.2.1. *The probability that* 1 *or more splits occur on the* $(N+1)$th *random insertion into a* B*-tree of order* m *with* N *keys is*

$$\Pr\{1 \textit{ or more splits}\}_n = \frac{1}{(2m+2)(H_{2m+2} - H_{m+1})}.$$

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* $(N+1)$th *random insertion into a* B*-tree of order m with N keys is*

$$\Pr\{0 \text{ splits}\}_m = 1 - \frac{1}{(2m+2)(H_{2m+2} - H_{m+1})}.$$

Proof. $\Pr\{0 \textit{ splits}\}_m = 1 - \Pr\{1 \textit{ or more splits}\}_m$. ∎

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

COROLLARY.

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

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*

$$\frac{\mathrm{nl}_m - 1}{2m} \leqslant \mathrm{nal}_m \leqslant \frac{\mathrm{nl}_m - 1}{m}.$$

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, $2m \times \mathrm{nal}_m = \mathrm{nl}_m - 1$ and $m \times \mathrm{nal}_m = \mathrm{nl}_m - 1$, respectively.) ∎

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

$$\frac{1}{(2 \ln 2)m} + \left(\frac{1}{8 \ln 2} - \frac{1}{4}\right) \frac{1}{(\ln 2)\, m^2} + O(m^{-3}) \leqslant \frac{\bar{n}_n(N)}{N}$$

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

and the storage used is $(1/\ln 2) + O(m^{-1})$.

Proof. Lemma 4.2.3 and Eq. (2.2.5) leads to

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

and

$$\left(\frac{2m+1}{(4m^2+4m)(H_{2m+2}-H_{m+1})}\right)\left(1-\frac{1}{N}\right)-\frac{1}{2N}+O(N^{\mathrm{Re}\,\lambda_2})\leqslant\frac{\bar{n}_m(N)}{N}$$
$$\leqslant\left(\frac{1}{2m(H_{2m+2}-H_{m+1})}\right)\left(1-\frac{1}{N}\right)-\frac{1}{N}+O(N^{\mathrm{Re}\,\lambda_2}),$$

where $\mathrm{Re}\,\lambda_2<0$. ∎

The values obtained for the storage used (cf. definition of storage used in Subsection 4.1) are between 1 and 2. The value 1 corresponds to the B-tree with all nodes having $2m$ keys, and the value 2 corresponds to the B-tree with all nodes having m keys. Yao (1978) used a different measure. He defined *storage utilization* 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 know that, in general,

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

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

$$1\leqslant E(X)\times E\left(\frac{1}{X}\right)\leqslant\frac{9}{8} \tag{4.2.2}$$

which yields:

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

$$\ln 2+O(m^{-1})\leqslant \textit{storage utilization}\ \leqslant\tfrac{9}{8}\ln 2+O(m^{-1}).$$

4.3. B-*Trees with an Overflow Technique*

In this subsection 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

TABLE 4.3.1

Transition Matrix Corresponding to the Tree Collection of B-Trees of Order m and Height 2 Using an Overflow Technique

$$
\begin{pmatrix}
-[(m+1)+(m+1)+(m-1)(2m+1)]-1 & & & & 2m/(2m+1)\times[(m+1)+(m+1)+(m-1)(2m+1)] \\
\ddots & & & & \\
-[(m+1)+2m+(m-1)(2m+1)]-1 & & & & \\
[(m+1)+m(2m+1)] & -[(m+1)+m(2m+1)]-1 & & & 2/(2m+1|(m+1)+m(2m+1)] \\
 & \ddots & & & \\
 & -[2m+m(2m+1)]-1 & & & \\
 & [(m+1)\times(2m+1)] & -[(m+1)\times(2m+1)]-1 & & 2m/(2m+1)\times[(m+1)(2m+1)] \\
 & & [(m+1)+(m+1)+m(2m+1)] & & \\
 & & \ddots & & \\
 & & [2m+2m(2m+1)] & -[2m+2m(2m+1)]-1 & \\
 & & & (2m+1)(2m+1) & -(2m+1)(2m+1)-1
\end{pmatrix}
$$

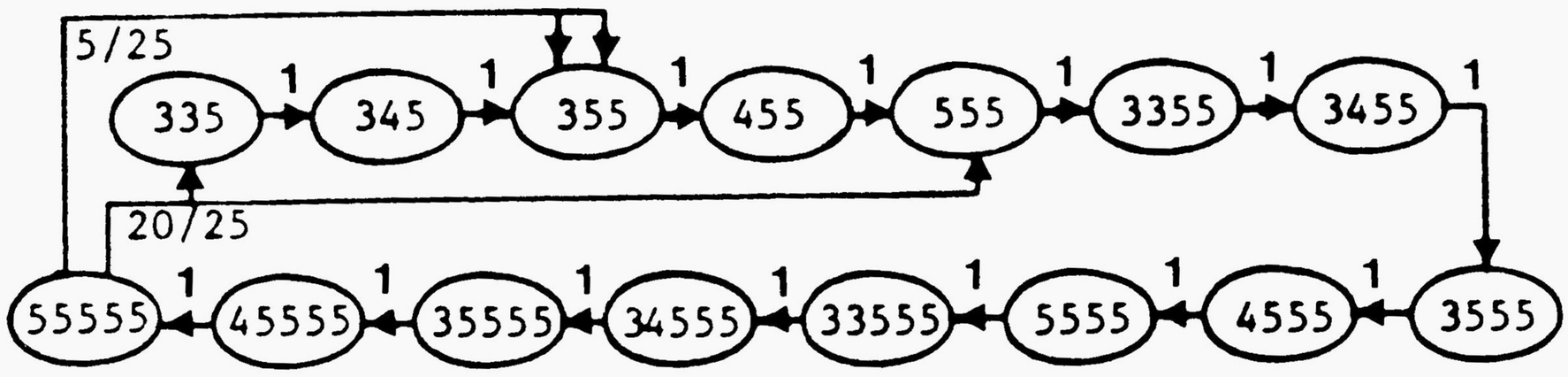

FIG. 4.3.1. Transition diagram representing the two level tree collection for B-trees of order $m = 2$ using overflow technique (e.g., type 335 corresponds to the height 2 type tree containing a root node with 3 descendants, the first one with 3 leaves, the second one also with 3 leaves, and the third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

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[3] $p_{(2m+1)(2m+1)} = 1$ and solve for all the other p's. After this we normalise the p's by dividing each one by their sum. Then

$$
\begin{aligned}
p_{(2m+1)(2m+1)} &= 1, \\
p_{(2m)+2m(2m+1)} &= \frac{(2m+1)(2m+1)+1}{(2m+1)(2m+1)} = \frac{4m^2+4m+2}{(2m+1)(2m+1)}, \\
p_{(2m-1)+2m(2m+1)} &= \frac{4m^2+4m+2}{(2m)+2m(2m+1)}, \\
&\vdots \\
p_{2m(2m+1)} &= \frac{4m^2+4m+2}{2m(2m+1)+1}, \\
p_{(2m)+(2m-1)(2m+1)} &= \frac{4m^2+4m+2}{2m(2m+1)}, \\
&\vdots
\end{aligned}
\tag{4.3.1}
$$

[3] $p_{(2m+1)(2m+1)}$ means $p_{(2m+1)+(2m+1)+\cdots+(2m+1)}$, where $(2m+1)$ appear $2m+1$ times. Applying this notation to the B-tree of order $m = 2$ shown in Fig. 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.

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

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

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

$$\vdots$$

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

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

$$(4m^3+2m^2+2m)/(2m+1)$$

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

$$\vdots$$

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

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

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

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

$$+ (2m^2+2m+2)[H_{2m^2+3m+1} - H_{2m^2+2m+1}] \tag{4.3.2}$$

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

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

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

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

Proof.

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

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

where $\sum_{i=1}^{m+1} 1/[(m+i)+(1/(2m+1))] = \psi(2m+2+(1/(2m+1))) - \psi(m+1+(1/(2m+1)))$, where S is as defined in Eq. (4.3.2) and $\psi(Z)$ is the Psi function $\Gamma'(Z)/\Gamma(Z)$ (Abramowitz and Stegun, 1972, Sect. 6.3.1).

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

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

Which yields the result. ∎

LEMMA 4.3.2. *The probability that* 1 *split occurs on the* $(N+1)$th *random insertion into an N-key random* B*-tree of order m using an overflow technique is*

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

where S is as defined in Eq. (4.3.2).

Proof. The only difference from the proof of Lemma 4.3.1 is that

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

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

LEMMA 4.3.3. *The probability that* 2 *or more splits occur on the* $(N+1)$th *random insertion into an N-key random* B*-tree of order m using an overflow technique is*

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

Proof.

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

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

$$\times \left[\psi\left(2m+2+\frac{1}{2m+1}\right)\right.$$

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

where S is as defined in Eq. (4.3.2). ∎

LEMMA 4.3.4. *The probability that no split occurs on the* $(N+1)$th *random insertion into an N-key random* B*-tree of order m using an overflow technique is*

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

Proof.

$$\Pr\{0 \text{ splits}\}_m = 1 - \Pr\{1 \text{ or more splits}\}_m$$

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

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

where S is as defined in Eq. (4.3.2). ∎

THEOREM 4.3.5. *The expected number of nodes in a random* B-tree *of order m with N keys using an overflow technique is bounded by*

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

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

and the storage used $= 1 + ((3/4\ln 2) - \frac{1}{2})(1/m) + O(m^{-2})$.

Proof. Lemma 4.2.3 and Eq. (2.2.5) lead to

$$A(2m)(N+1) - \tfrac{1}{2} \leqslant \bar{n}_m(N) \leqslant A(m)(N+1) - 1,$$

where

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

$$\left. + \frac{p_{(m+1)+(m+2)+(m-1)(2m+1)}}{(m+1)+(m+2)+(m-1)(2m+1)} + \cdots + \frac{p_{(m+1)(2m+1)}}{(m+1)(2m+1)}\right]$$

$$+ \left(m+3+\frac{1}{X}\right)\left[\frac{p_{(m+1)+(m+1)+m(2m+1)}}{(m+1)+(m+1)+m(2m+1)}\right.$$

$$\left. + \frac{p_{(m+1)+(m+2)+m(2m+1)}}{(m+1)+(m+2)+m(2m+1)} + \cdots + \frac{p_{(m+2)(2m+1)}}{(m+2)(2m+1)}\right]$$

$$+ \cdots + \left(2m+2+\frac{1}{X}\right)\left[\frac{p_{(m+1)+(m+1)+(2m-1)(2m+1)}}{(m+1)+(m+1)+(2m-1)(2m+1)}\right.$$

$$\left. + \frac{p_{(m+1)+(m+2)+(2m-1)(2m+1)}}{(m+1)+(m+2)+(2m-1)(2m+1)}\right.$$

$$\left.\left. + \cdots + \frac{p_{(2m+1)(2m+1)}}{(2m+1)(2m+1)}\right]\right\}$$

and S is as defined in Eq. (4.3.2).

Substituting Eq. (4.3.1) in the above expression gives

$$B(2m)\left(1-\frac{1}{N}\right) - \frac{2}{2N} + O(N^{\mathrm{Re}\,\lambda_2}) \leqslant \frac{\bar{n}_m(N)}{N}$$

$$\leqslant B(m)\left(1-\frac{1}{N}\right) - \frac{1}{N} + O(N^{\mathrm{Re}\,\lambda_2}), \qquad \mathrm{Re}\,\lambda_2 < 0,$$

where

$$B(X) = \frac{1}{S}\left\{\left(m+2+\frac{1}{X}\right)\left[\left(\frac{4m^3+2m^2+2m}{2m+1}\right)\right.\right.$$

$$\times \left(\frac{1}{(m+1)+(m+1)+(m-1)(2m+1)} - \frac{1}{(m+1)+m(2m+1)}\right)$$

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

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

$$+ (4m^2 + 4m + 2)\left[\frac{m + 3 + (1/X)}{(m+1)+(m+1)+m(2m+1)} - \frac{2m + 2 + (1/X)}{(2m+1)(2m+1)+1} + \frac{\psi(2m + 1 + (1/(2m+1))) - \psi(m + 2 + (1/(2m+1)))}{2m+1}\right]\Bigg\}$$

or

$$B(X) = \frac{1}{S}\left\{\frac{1}{X} + \frac{8m^2 + 10m + 6}{2m^2 + 3m + 2} + \frac{4m^2 + 4m + 2}{2m+1} \times \psi\left(2m + 1 + \frac{1}{2m+1}\right) - \psi\left(m + 2 + \frac{1}{2m+1}\right)\right)\Bigg\}. \quad \blacksquare$$

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

$$1 - \left(\frac{3}{4 \ln 2} - \frac{1}{2}\right)\frac{1}{m} + O(m^{-2})$$

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

Proof. The above bounds are obtained by using Eq. (4.2.2) and the result of the previous corollary. ∎

Notice that the expected storage utilization is essentially one for large m, when the overflow technique is used.

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. *The Deepest Safe Node in* B*-Trees with the 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) $\Pr\{dsn\ at\ 1st\ lowest\ level\}_m = 1 - (1/(2 \ln 2)m) - ((1/8 \ln 2) - \frac{1}{2})(1/(\ln 2)\, m^2) + O(m^{-3})$,

(b) $\Pr\{dsn\ above\ 1st\ lowest\ level\}_m = (1/(2 \ln 2)m) + ((1/8 \ln 2) - \frac{1}{2})(1/(\ln 2)\, m^2) + O(m^{-3})$.

Proof. Similar to the proof of Lemma 3.6.1.1. we obtain

(a) $\Pr\{\text{dsn at 1st lowest level}\}_m = 1 - (1/(2m+2)(H_{2m+2} - H_{m+1}))$,

(b) $\Pr\{\text{dsn above 1st lowest level}\}_m = (1/(2m+2)(H_{2m+2} - H_{m+1}))$. ∎

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 paper will lock height 1 fringe subtrees most of the time.

4.4.2. *The Deepest Safe Node in* B*-Trees with an 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) $\Pr\{dsn\ at\ 1st\ lowest\ level\}_m = 1 - (1/2m) - ((1/8 \ln 2) - \frac{1}{4})(1/m^2) + O(m^{-3})$,

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

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

Proof. Similar to the proof of Lemma 3.6.1.1. we obtain

(a) $\Pr\{\text{dsn at 1st lowest level}\}_m = 1 - (1/S)|((2m+1)(2m+1) + 1)/(2m+1)||\psi(2m+2+(1/(2m+1))) - \psi(m+1+(1/(2m+1)))|$,

(b) $\Pr\{\text{dsn at 2nd lowest level}\}_m = (1/S)|((2m+1)(2m+1)\ 11)/(2m+1)||\psi(2m+1+(1/(2m+1))) - \psi(m+1+(1/(2m+1)))|$,

(c) $\Pr\{\text{dsn above 2nd lowest level}\}_m = 1/S$, where S is as defined in Eq. (4.3.2). ∎

5. CONCLUSIONS AND OPEN PROBLEMS

In Section 2 we have shown 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 makes our fringe

analysis theory flexible and general enough to permit its application in the analysis of many different classes of search trees.

In Section 3 an analysis for the three lowest levels of 2–3 is accomplished. We have discussed, in Section 3, how the same techniques might be extended to enable an analysis for the four lowest levels to be carried out. This will require the solution of a 4410×4410 linear system.

In Section 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 particular for large order B-trees it is shown that the storage utilization is, essentially, 1, when using the described overflow technique.

Clearly a central open problem is to analyze the behaviour of balanced trees under both random insertions and deletions. Whether or not fringe analysis techniques can be extended to accomplish this remains to be seen. The basic obstacle is that deletions do not preserve randomness, although a first step has been made by Melhorn (1982).

Finally, the original problem, namely, carry out a true analysis of 2–3 trees under random insertions, is still open. Our analysis is merely an approximation to the true analysis which can be viewed as an infinite order fringe analysis. Whether or not fringe analysis theory can be extended to this limiting case is also open.

REFERENCES

ABRAMOWITZ, M., AND STEGUN, I. A., (1972), "Handbook of Mathematical Functions," Dover, New York.

AHO, A. V., HOPCROFT, J. E., AND ULLMAN, J. D. (1974), "The Design and Analysis of Computer Algorithms,"Addison–Wesley, Reading, Mass.

BAYER, R., AND MCCREIGHT, E. (1972), Organization and maintenance of large ordered indexes, *Acta Inform.* **1** (3), 173–189.

BAYER, R., AND SCHKOLNICK, M. (1977), Concurrency of operations on B-trees, *Acta Informa.* **9** (1), 1–21.

BROWN, M. (1979), Some observations on random 2–3 trees, *Inform. Process. Lett.* **9** (2), 57–59.

CLAUSING, A. (1982), Kantorovich-type inequalities, *Amer. Math. Monthly* **89** (5), 314–330.

CHVATAL, V., KLARNER, D. A., AND KNUTH, D. E. (1972), "Selected Combinatorial Research Problems," report STAN-CS-72-292, Computer Science Department, Stanford Univ.

COMER, D. (1979a), The ubiquitous B-tree, *Comput. Surveys* **11** (2), 121–137.

COMER, D. (1979b), The tree branches, *Comput. Surveys* **11** (4), 412.

COX , D. R., AND MILLER, H. D. (1965), "The Theory of Stochastic Processes, Chapman & Hall, London.

EISENBARTH, B. (1981), "Allgemeine Fringe-analysis und ihre Anwendung zur Untersuchung der Overflowtechnik bei B-Bäumen," Diplomarbeit, Universität des Saarlandes, Saarbrucken, West Germany.

FELLER, W. (1968), "An Introduction to Probability Theory and Its Applications," Vol. 1, Wiley, New York.

FELLER, W. (1966), "An Introduction to Probability Theory and Its Applications," Vol. 2, Wiley, New York.

GANTMACHER, F. R. (1959), "The Theory of Matrices," Vol. 1, Chelsea, New York.

GEDDES, K. O., AND GONNET, G. H. (1981), "MAPLE User's Manual," Report CS-81-25, Department of Computer Science, University of Waterloo, Canada.

GONNET, G. H., ZIVIANI, N., AND WOOD, D. (1981), "An Analysis of 2–3 Trees and B-trees," Report CS-81-21, Department of Computer Science, University of Waterloo, Canada.

HUDDLESTON, S., AND MEHLHORN, K. (1982), A new data structure for representing sorted lists, *Acta Inform.* **17**, 157–184.

KNUTH, D. E. (1968), "The Art of Computer Programming," Vol. 1, Addison–Wesley, Reading, Mass.

KNUTH, D. E. (1973), "The Art of Computer Programming," Addison–Wesley, Reading, Mass.

KWONG, Y. S., AND WOOD, D. (1980), "Approaches to Concurrency in B-trees," Lecture Notes in Computer Science, No. 88, pp. 402–413, Springer-Verlag, Berlin.

MEHLHORN, K. (1982), A partial analysis of height-balanced trees under random insertions and deletions, *SIAM J. Comput.* **11**, 748–760.

WILKINSON, J. H. (1965), "The Algebraic Eigenvalue Problem," Oxford Univ. Press, London.

YAO, A. (1978), On random 2–3 trees, *Acta Inform.* **9**, 159–170.

ZIVIANI, N. (1982), "The Fringe Analysis of Search Trees," Ph.D. thesis, Report CS-82-15, Department of Computer Science, University of Waterloo, Canada.

ZIVIANI, N., AND TOMPA, F. (1982), "A Look at Symmetric Binary B-trees," *INFOR Canad. J. Oper. Res. Inform. Process.* **20** 65–81.

Printed by the St. Catherine Press Ltd., Tempelhof 41, Bruges, Belgium