An Alternative Approach to the Analysis of Late Binding Trees

Patricio V. Poblete [†]

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

Research Report CS-82-23
August 1982

ABSTRACT

Late Binding Trees were defined and studied by D.E. Knuth. We present a different method for analysing their behaviour. The approach taken in this paper is simpler and yields more information on the performance of this data structure.

† On leave from the University of Chile.

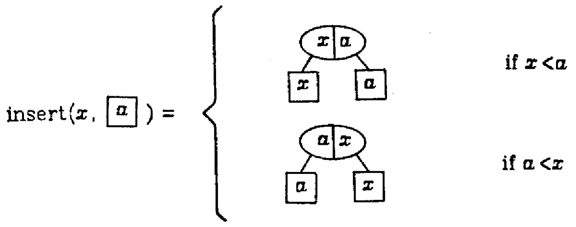
An Alternative Approach to the Analysis of Late Binding Trees

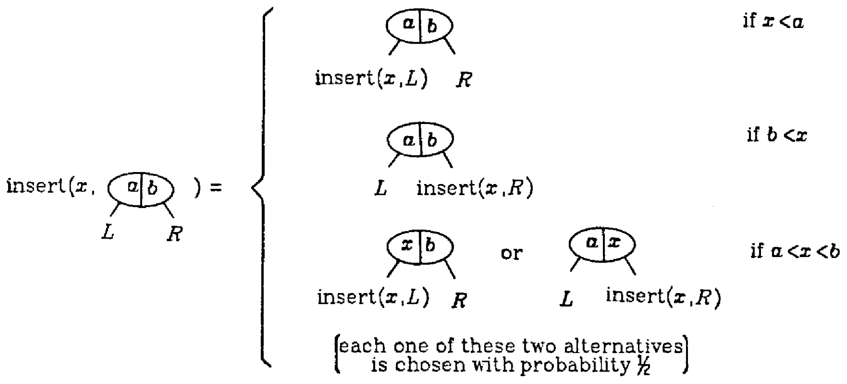
Patricio V. Poblete

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

Late Binding Trees[1] (or LBTs, for short) are a class of binary search trees constructed by using two types of nodes, *internal* and *external*. An internal node contains a pair of keys $(a \mid b)$ such that $a<b$; an external node contains a single key $\boxed{a}$.

An LBT with a single key a consists of just a leaf $\boxed{a}$. Larger LBTs are built by inserting elements according to the following rules:





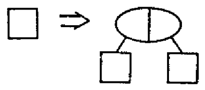
In [1] Knuth outlines a method to compute the expected external path length of an LBT built by inserting n keys $x_1, \ldots, x_n$, under the assumption that $x_1, \ldots, x_n$ is a random permutation of $\{1, \ldots, n\}$.

In this paper we study the average length of a root-to-leaf path (whose expected value is $1/n$ times the expected external path length) under the equivalent assumption that if $x_1 < x_2 < \cdots < x_n$ have already been inserted and x is a new key, then

$$\mathbf{P}\{x < x_1\} = \mathbf{P}\{x_1 < x < x_2\} = \cdots = \mathbf{P}\{x_{n-1} < x < x_n\} = \mathbf{P}\{x > x_n\} = \frac{1}{n+1}$$

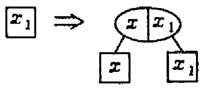
The method we use for this analysis is similar to the one used in [2] to study a fringe heuristic for binary search trees.

To carry out our analysis we will concentrate on the *shape* of the tree, and ignore whenever convenient the labels of the nodes. This can be done because the relative ranks of the elements stored in the tree are completely determined by the shape of it. From this point of view, the only effect of an insertion is that some external node becomes an internal node with two external nodes as sons:

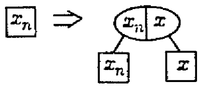


Consider the insertion of a random element x into a tree that already contains $x_1, \ldots, x_n$:

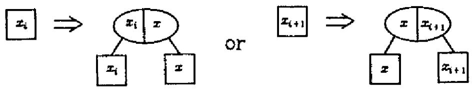
(1) If $x < x_1$ then



(2) If $x > x_n$ then

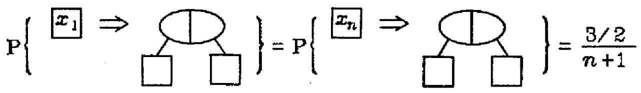


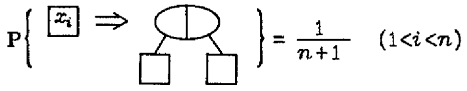
(3) If $x_i < x < x_{i+1}$ $(1 \le i < n)$ then



In case (3) we choose with probability ½ between the two alternatives and we ignore the necessary relabelling of one internal node, as it does not affect the shape of the tree.

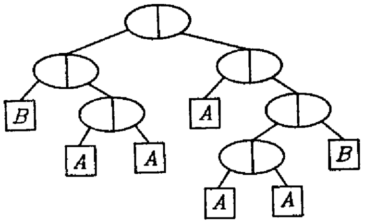
These rules imply that



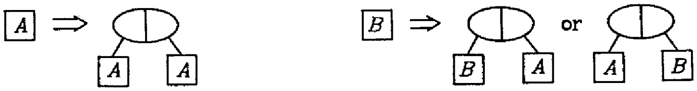


Therefore, an external node that is either the leftmost or the rightmost one of the tree is 50% more likely to be expanded as the result of an insertion than is one of the remaining external nodes.

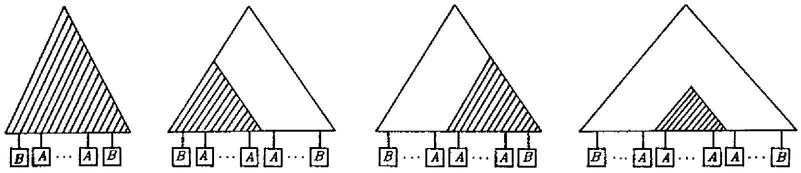
Let us say that all external nodes are of type A, except for the leftmost and rightmost ones, which are of type B. With this convention, an LBT looks like



and the possible effects of an insertion are



We can immediately observe that, from the point of view of their statistical behaviour, there are three types of subtrees: the full tree (which contains two B nodes), a leftmost or rightmost subtree (which contains only one B node) and a middle tree (which contains only A nodes). A direct consequence of this observation is that a middle subtree with n elements has the same shape as an ordinary binary search tree with $n-1$ elements. In particular, the average length of a root-to-leaf path in a middle subtree with n elements has expected value $2H_n-2$.



Let us now assign levels to the tree, starting from the root, which is at level zero. Let

$A_{n,k}$ = average number of A nodes at level k

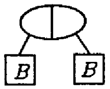
$B_{n,k}$ = average number of B nodes at level k

$$P_{n,k} = \frac{A_{n,k} + B_{n,k}}{n}$$

$P_{n,k}$ is the probability of finding an external node (of any type) at level k in a tree with n elements.

We also use the associated generating functions

$$A_n(z) = \sum_{k\geq 0} A_{n,k} z^k, \quad B_n(z) = \sum_{k\geq 0} B_{n,k} z^k, \quad P_n(z) = \sum_{k\geq 0} P_{n,k} z^k$$

Initially (n=2) the only possible tree is , so

$A_{2,k} = 0$ for all k

$B_{2,1} = 2;\ B_{2,k} = 0$ for all $k \neq 1$

For $n \geq 2$

$$A_{n+1,k} = A_{n,k} - \frac{A_{n,k}}{n+1} + 2\frac{A_{n,k-1}}{n+1} + \frac{\frac{3}{2}B_{n,k-1}}{n+1}$$

$$B_{n+1,k} = B_{n,k} - \frac{\frac{3}{2}B_{n,k}}{n+1} + \frac{\frac{3}{2}B_{n,k-1}}{n+1}$$

Introducing generating functions we have

$$A_{n+1}(z) = A_n(z) + \frac{1}{n+1}\left[(2z-1)A_n(z) + \frac{3}{2}zB_n(z)\right]; \quad A_2(z) = 0$$

$$B_{n+1}(z) = B_n(z) + \frac{1}{n+1}\,\frac{3}{2}(z-1)B_n(z); \quad B_2(z) = 2z$$

Or, using matrix notation,

$$\begin{bmatrix} A_{n+1}(z) \\ B_{n+1}(z) \end{bmatrix} = (I + \frac{1}{n+1}H(z)) \begin{bmatrix} A_n(z) \\ B_n(z) \end{bmatrix} ; \quad \begin{bmatrix} A_2(z) \\ B_2(z) \end{bmatrix} = \begin{bmatrix} 0 \\ 2z \end{bmatrix}$$

where

$$H(z) = \begin{bmatrix} 2z-1 & \frac{3}{2}z \\ 0 & \frac{3}{2}(z-1) \end{bmatrix}$$

The solution of this recurrence can be found by expressing the unknown vector in a basis of eigenvectors of the matrix $H(z)$. The eigenvalues of $H(z)$ are $\lambda_1(z)=2z-1$ and $\lambda_2(z)=\frac{3}{2}(z-1)$, and the solution is

$$\begin{bmatrix} A_n(z) \\ B_n(z) \end{bmatrix} = E(z) \begin{bmatrix} \pi^{(1)} & 0 \\ 0 & \pi^{(2)} \end{bmatrix} E^{-1}(z) \begin{bmatrix} A_2(z) \\ B_2(z) \end{bmatrix}$$

where

$$E(z) = \begin{bmatrix} 1 & 1 \\ 0 & -\frac{z+1}{3z} \end{bmatrix}, \quad E^{-1}(z) = \begin{bmatrix} 1 & \frac{3z}{z+1} \\ 0 & -\frac{3z}{z+1} \end{bmatrix},$$

$$\pi^{(1)} = \prod_{2<j\le n} \left(\frac{j+2z-1}{j}\right), \quad \pi^{(2)} = \prod_{2<j\le n} \left(\frac{j+\frac{3}{2}(z-1)}{j}\right)$$

Therefore,

$$\begin{bmatrix} A_n(z) \\ B_n(z) \end{bmatrix} = 2z \begin{bmatrix} \frac{3z}{z+1}(\pi^{(1)}-\pi^{(2)}) \\ \pi^{(2)} \end{bmatrix}$$

The probability generating function for the average length of a root-to-leaf path is

$$P_n(z) = \frac{2z}{n}\left[\frac{3z}{z+1}\pi^{(1)} + (1-\frac{3z}{z+1})\pi^{(2)}\right]$$

$$= \frac{2z}{n}\left[\frac{3z}{z+1}\prod_{2<j\le n}\left(\frac{j+2z-1}{j}\right) + (1-\frac{3z}{z+1})\prod_{2<j\le n}\left(\frac{j+\frac{3}{2}(z-1)}{j}\right)\right]$$

An equivalent form, which may be more suitable for differentiation, is

$$P_n(z) = \frac{4z}{n\Gamma(n+1)}\left[\frac{3z}{z+1}\,\frac{\Gamma(n+2z)}{\Gamma(2+2z)} + (1-\frac{3z}{z+1})\frac{\Gamma(n+1+\frac{3}{2}(z-1))}{\Gamma(3+\frac{3}{2}(z-1))}\right]$$

From this generating function we can compute the expected value of the average length of a root-to-leaf path

$$\mathrm{mean}(P_n(z)) = (2+\frac{1}{2n})H_n - \frac{13}{6} - \frac{5}{12n}$$

and its variance

$$\mathrm{var}(P_n(z)) = (2+\frac{35}{12n}+\frac{5}{12n^2})H_n - \frac{1}{4n}(1+\frac{1}{n})(H_n)^2$$
$$-(4+\frac{7}{4n})H_n^{(2)} + \frac{55}{36} - \frac{5}{8n} - \frac{25}{144n^2}$$

Note that the same analysis applies to leftmost, rightmost and middle subtrees, by changing only the initial conditions. For leftmost or rightmost subtrees we have $A_2(z)=z$, $B_2(z)=z$; for middle subtrees, $A_2(z)=2z$, $B_2(z)=0$.

Unfortunately, the detailed knowledge about the performance of LBTs we have just obtained does not support their use as a replacement for "naive" binary search trees. Neither the mean nor the variance show any significant improvement that could justify the increased complication in the algorithms that handle the tree. This does not mean, of course, that there is no simple way to improve the performance of a binary search tree by delaying the decision about which element should be at the root: the heuristics analysed in [2] are an example of a successful application of that idea.

References

1. Knuth, D.E., Problem 82-3, *J. of Algorithms* 3,2(June 1982), 178-180.
2. Poblete, P.V. and Munro, J.I., "The Analysis of a Fringe Heuristic for Binary Search Trees," Research Report CS-82-22, Dept. of Computer Science, U. of Waterloo, Canada, July 1982.