Height Balanced Trees - AVL SearchTrees

RM.Periakaruppan
Dept of AM&CS
PSG Tech

AVL Tree and AVL Search Tree

- AVL (Adelson-Velsky and Landis) trees

- An empty binary tree is AVL balanced.

If T is a nonempty binary tree with T_L and T_R as its left and right subtrees, then T is an AVL tree if

(1) T_L and T_R are AVL trees and

(2) $|h_L - h_R| \leq 1$ where h_L and h_R are the heights of TL and TR, respectively.

An AVL Search tree is a binary search tree that is also an AVL tree.

Normally when we refer as AVL tree, we mean AVL Search tree

Height of an AVL Tree

Let N_h be the minimum number of nodes in an AVL tree of height h.

In the worst case the height of one of the subtrees is h-1, and the height of the other is h-2. Both these subtrees are also AVL trees. Hence

$N_h = N_{h-1} + N_{h-2} + 1$, $N_0 = 0$, and $N_1 = 1$

Solving this recurrence we get the height of an AVL tree as

$N_h = 1.44 \log(n+2) = O(\log n)$ approximately

Representation of an AVL Tree

AVL trees are normally represented by using the linked representation scheme for binary trees.

To facilitate insertion and deletion, a balance factor *bf* is associated with each node.

The balance factor *bf (x)* of a node *x* is defined as

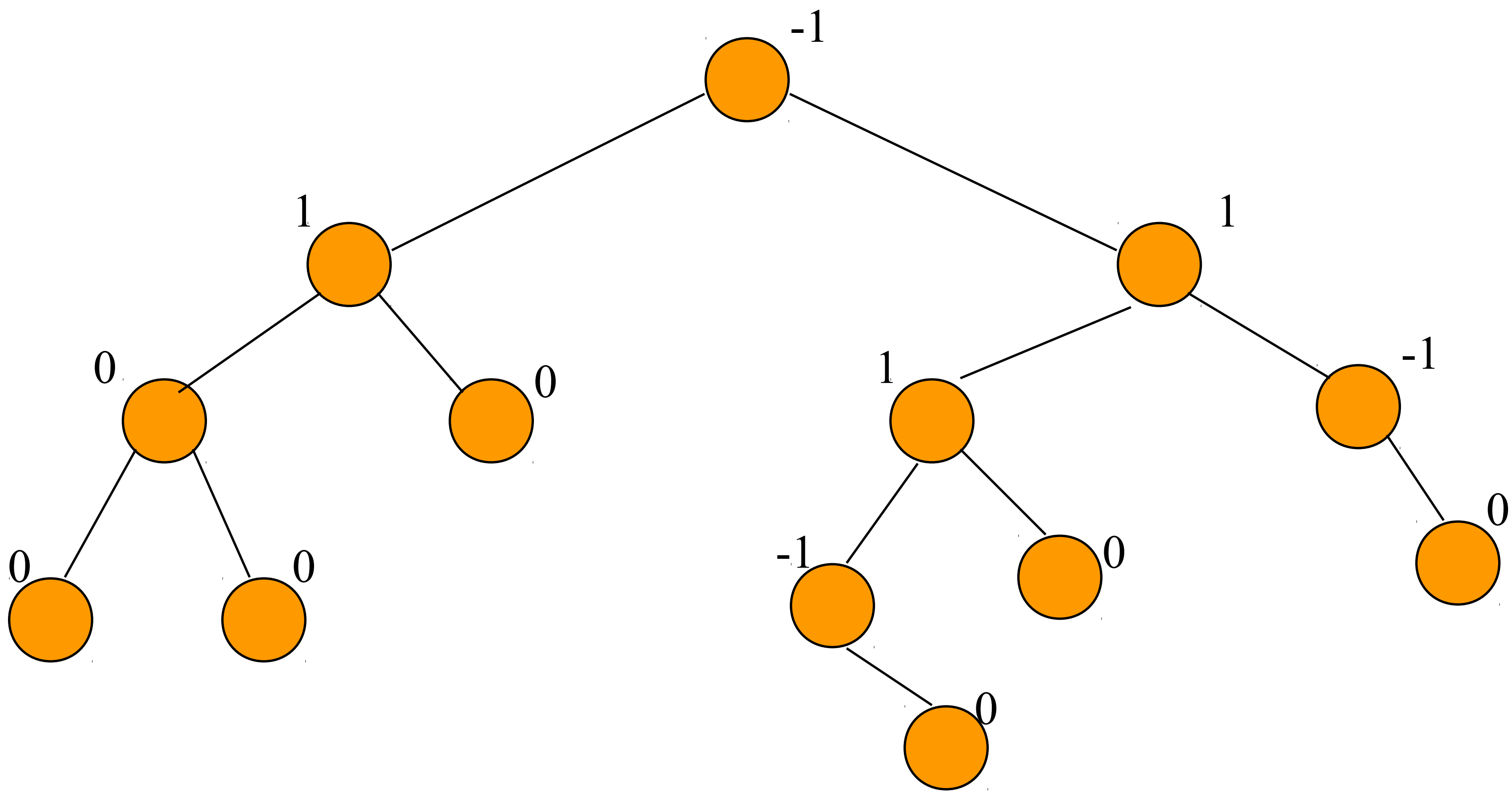
Height of left subtree of *x* - height of right subtree of *x*

AVL Tree Balance Requirements

- *Left High(balance factor +1)* *:* The left-sub tree is one level taller than the right-sub tree.
- *Balanced (balance factor 0)* : The left and right sub-trees are of the same heights.
- *Right High (balance factor -1):* The right sub-tree is one level taller than the left-sub tree.

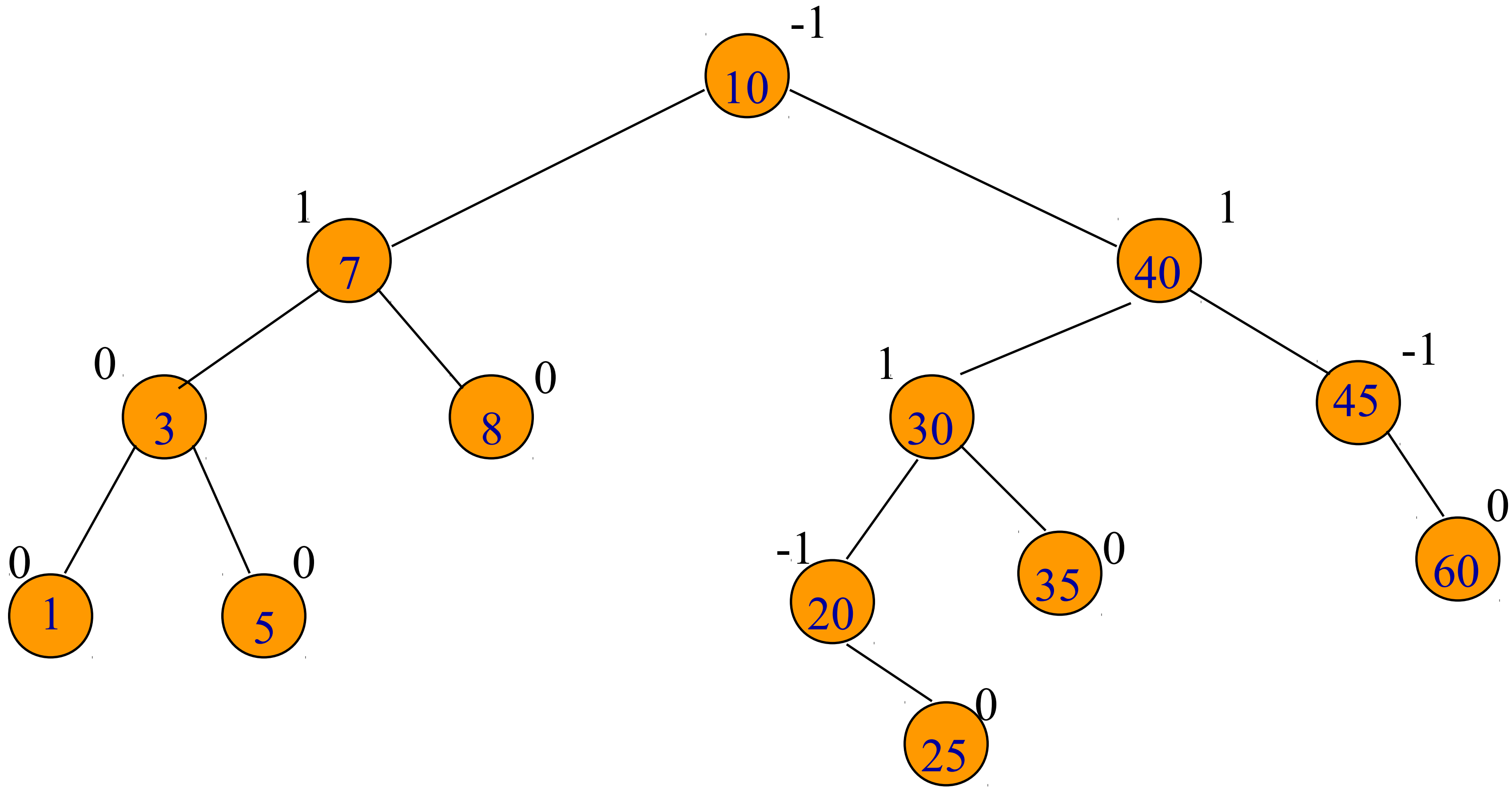
Balance factor of every node x is -1, 0, or 1

Example - Balance Factors



This is an AVL tree.

Example - AVL Search Tree



Inserting in an AVL Tree

Nodes are initially inserted into AVL Trees leaf nodes.

After insertion, however, the insertion algorithm for an AVL Tree travels back along the path it took to find the point of insertion, and checks the balance at each node on the path. If all nodes are balanced, then no need to rebalance.

If a node is found that is unbalanced (i.e., it has a balance factor of either -2 or +2), then a rotation is performed based on the inserted nodes position relative to the node being examined (the unbalanced node).

At the most one rotation (single or double) is needed to balance the tree after an insert operation.

Inserting in an AVL Tree

Tree Rotations used to restore the balance

If an insertion cause an imbalance, which nodes can be affected?

Nodes on the path of the inserted node.

Let A be the node nearest to the inserted one which has an imbalance.

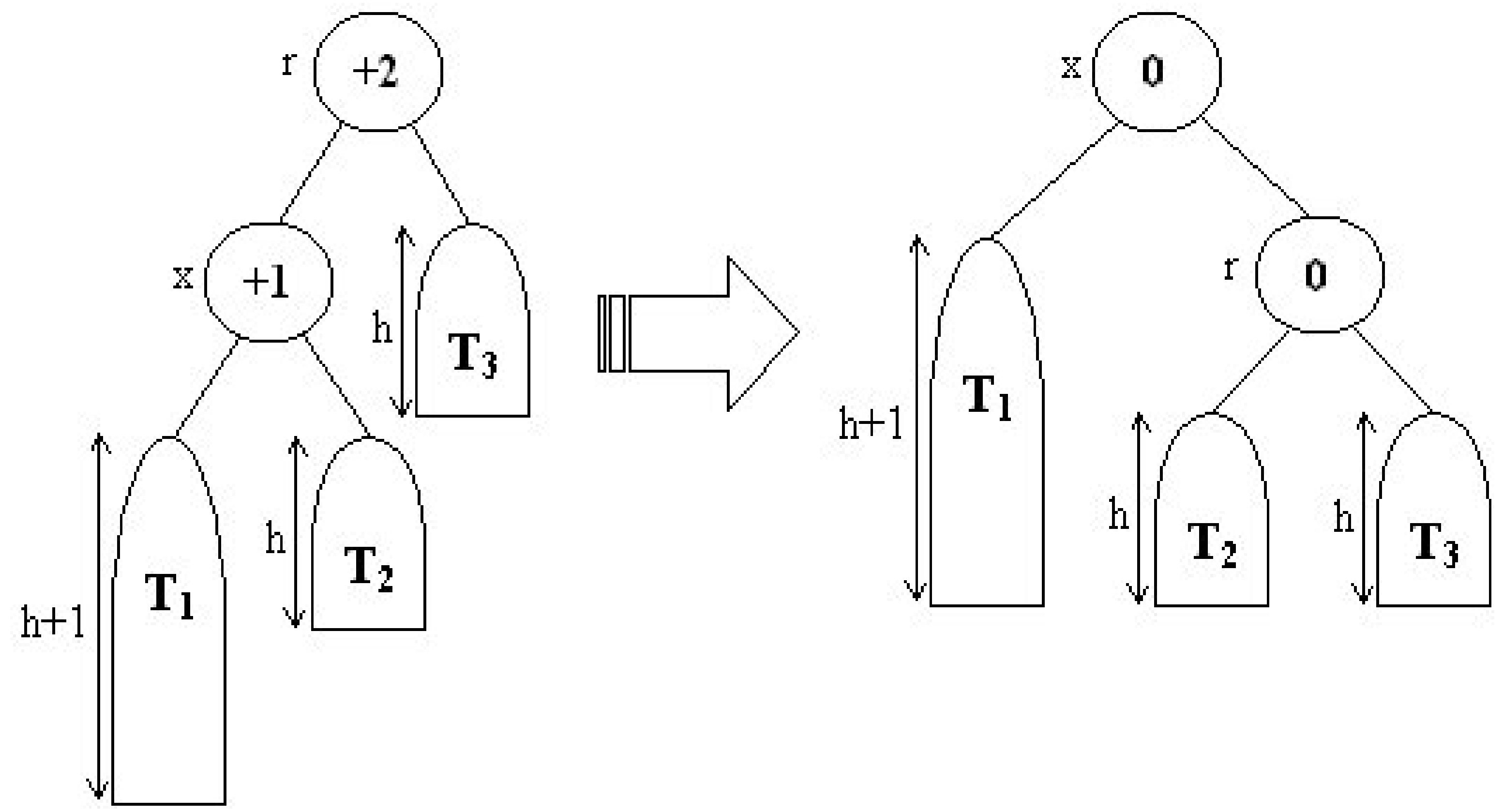
insertion in the left subtree of the left child of A(LL)

insertion in the right subtree of the left child of A(LR)

insertion in the left subtree of the right child of A(RL)

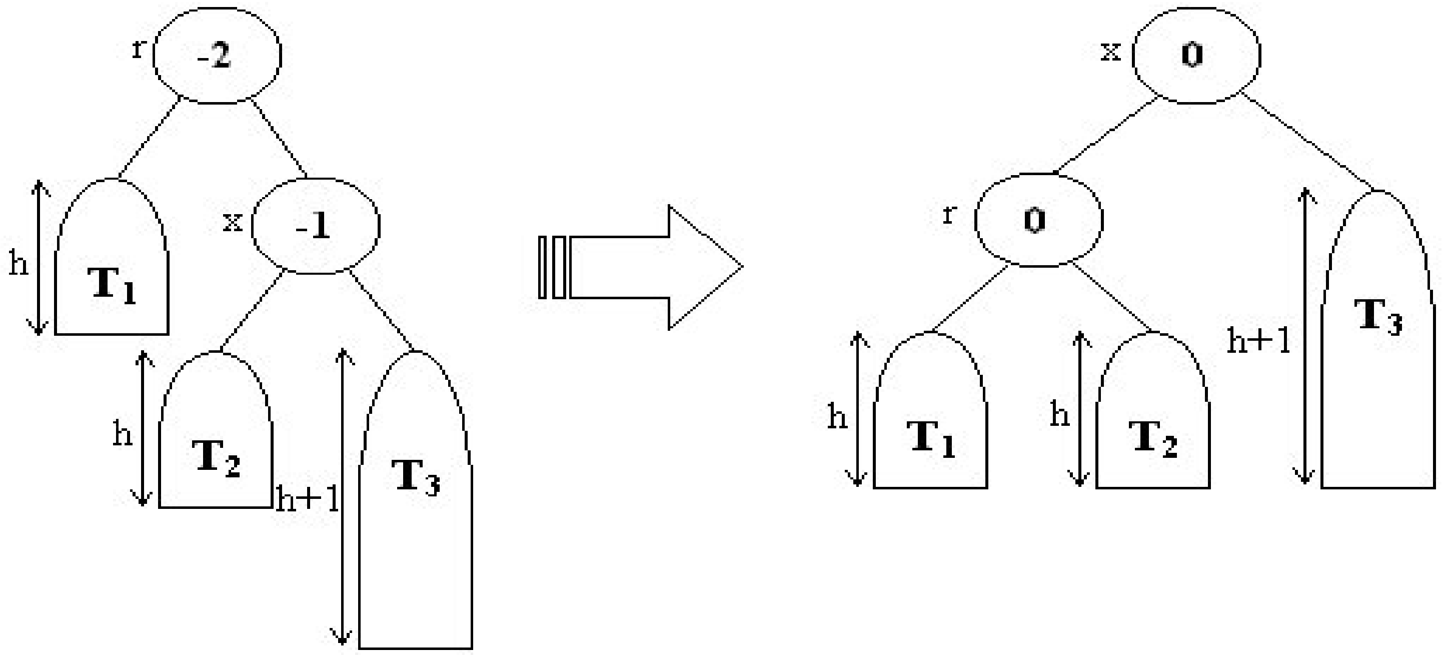
insertion in the right subtree of the right child of A(RR)

AVL Tree Rotations



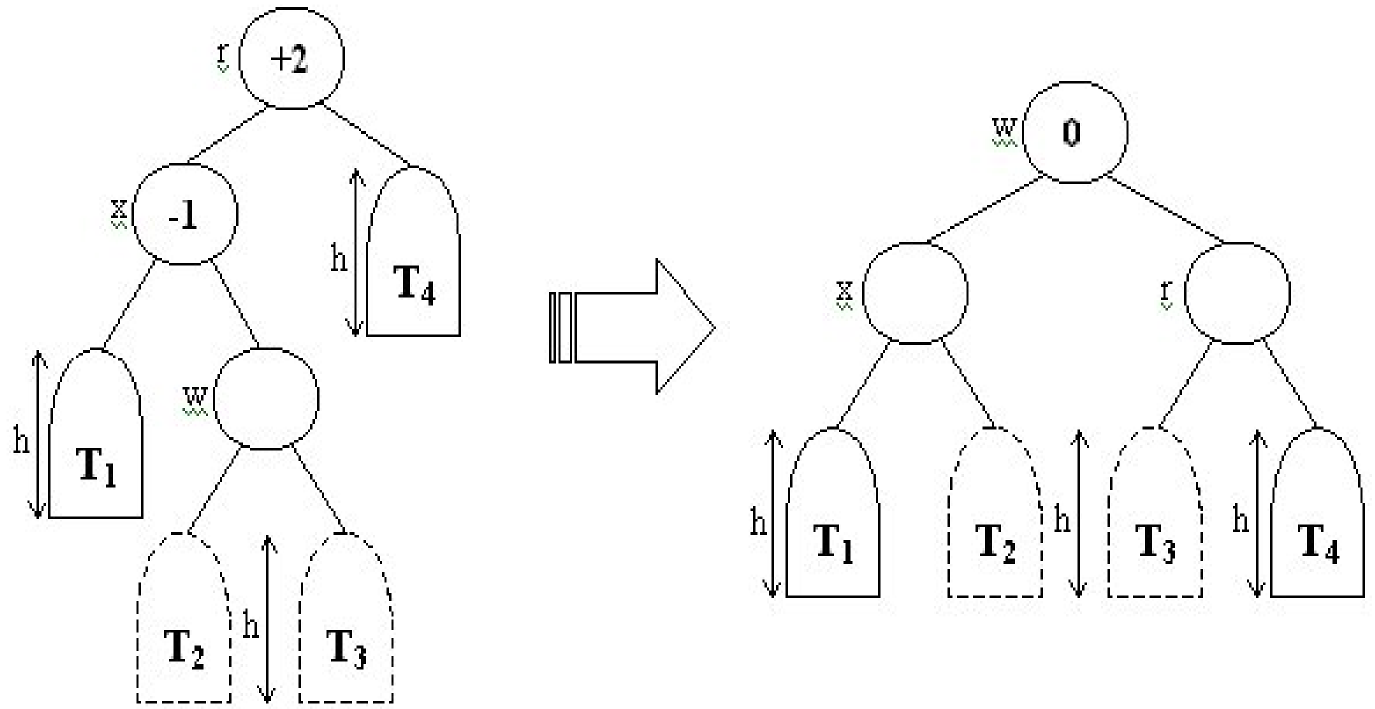
(a) LL Rotation

AVL Tree Rotations



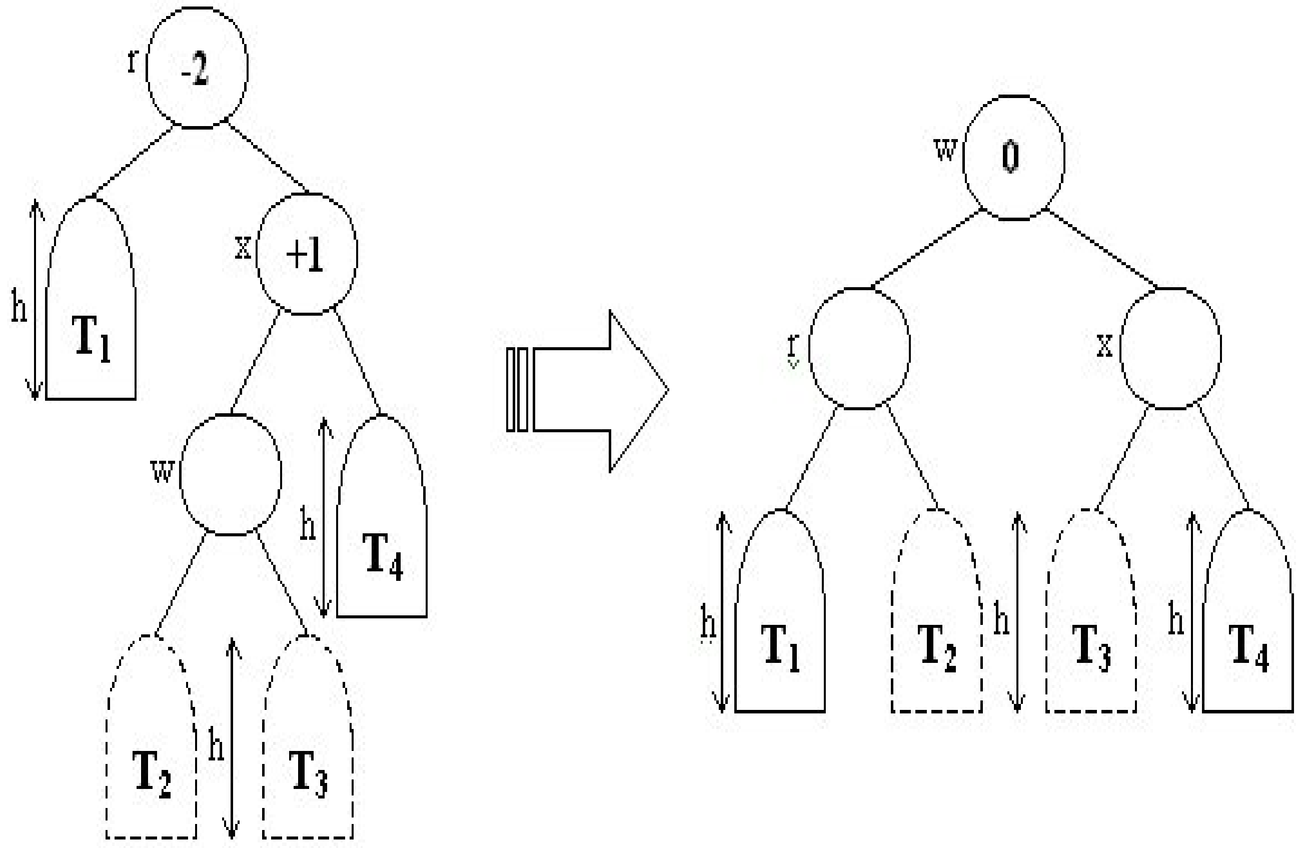
(b) RR Rotation

AVL Tree Rotations



(c) LR Rotation

AVL Tree Rotations



(d) RL Rotation