

Generalised AVL Binary Search Trees

Moahan Murugappan

September 2024

This document introduces the idea of how the self-balancing AVL binary search trees (BST) can be generalised through rotation.

1 Recap: AVL binary search tree

A sequence of n data structures can be stored in a BST for quick search, where each data structure represents a *node* in the tree. Each data structure i of the set Ω needs only have the ‘less than’ operator $<$ defined on Ω . We begin with an $n = 3$ example. Figure. 1 shows a binary search tree for elements A , B and C , where $B < A < C$. Each of the cases represent a different sequence of insertion. In general, all nodes in the left sub-tree of a node has values smaller than that of the node’s, and all nodes in the right sub-tree has values equal or greater.

The *height* h of a node is minimum distance from a leave node, with the latter having height $h = 0$. Figure. 2 shows the node heights of the respective cases in Fig. 1. The *balance factor* b of a node, is defined by $b = h_L - h_R$, where h_L and h_R are the heights of the left and right children, respectively. As some nodes may have only one child, we assign the value $h = -1$ for the ‘missing’ child. Figure. 3 shows the node balance factors for the corresponding examples of Fig. 1.

An AVL binary search tree is one that has each of its node having

$$|b| \leq 1. \tag{1}$$

When such a condition is satisfied, the BST is termed *balanced*. A BST that maintains Eq. 1 after every element insertion (and deletion) is term ‘self-balancing’. From Fig. 3, we can see that only the example Fig. 1a is balanced. A balanced BST ensures that searching an element goes like $\mathcal{O}(\log n)$. The balancing mechanism, for the AVL BST involves the *rotation* of 3 nodes. The R (right) and LR (left-right) rotations are demonstrated via Figs. 4 and 5. The left-right mirror cases of L (left) and RL (right-left) rotations are easily inferred. In Figs. 4 and 5, the leading node (C) is chosen such that it is the node with $|b| = 2$, of minimum h , above the node that has just been inserted or deleted.

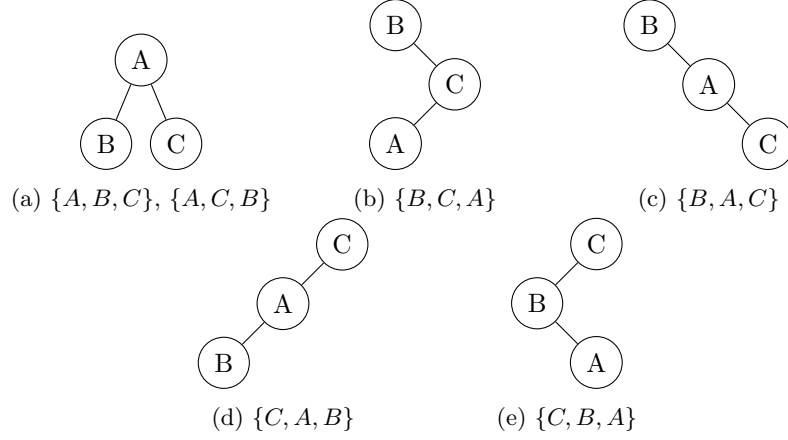


Figure 1: An $n = 3$ binary search tree example for elements A , B and C with $B < A < C$. Each cases represents different sequence of insertion, starting from the first element. For the example of Fig.1a, A is the parent node, with B and C its children. Also in Fig.1a, A is the root node, and B and C are the leaf nodes. Edges (lines connecting nodes) slanting to the left represent the child being ‘smaller’ $<$, than the parent, while to the right represent ‘greater or equal’.

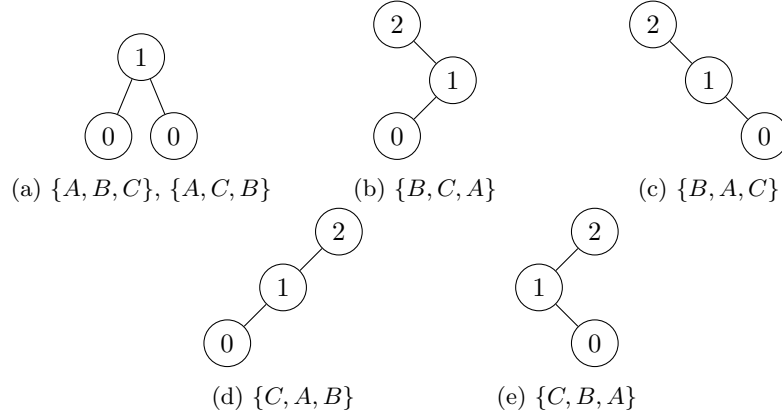


Figure 2: Node height h for the corresponding examples in Fig. 1.

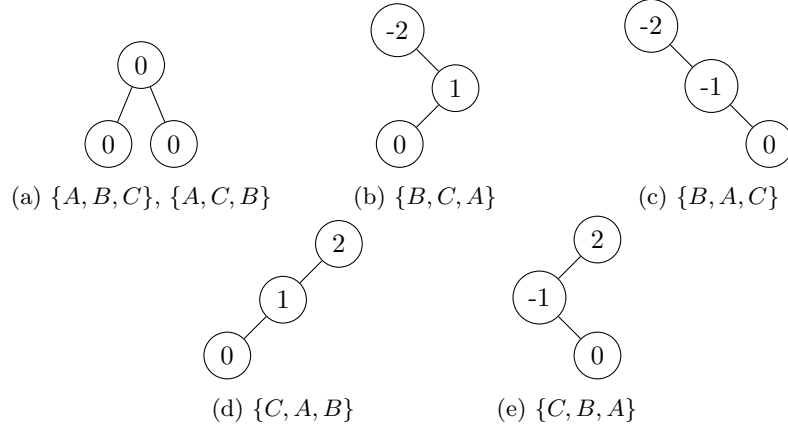


Figure 3: Balance factor $b = h_L - h_R$ for the corresponding examples in Fig. 1, with h_L and h_R being the node height values of the left and right child, respectively. ‘Missing’ child is assigned $h = -1$. Note that for $b < 0$, the tree slants to the left, and vice versa.

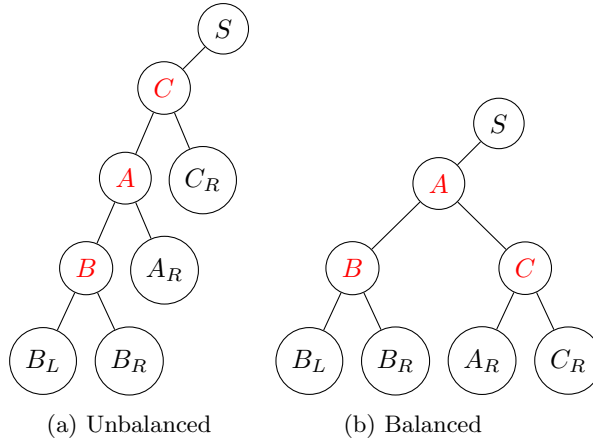


Figure 4: R rotation. S represents an arbitrary parent node. A_R , B_L , B_R and C_R represent sub-trees. The red 3-node sequence for the AVL BST case is chosen such that C has $b = 2$.

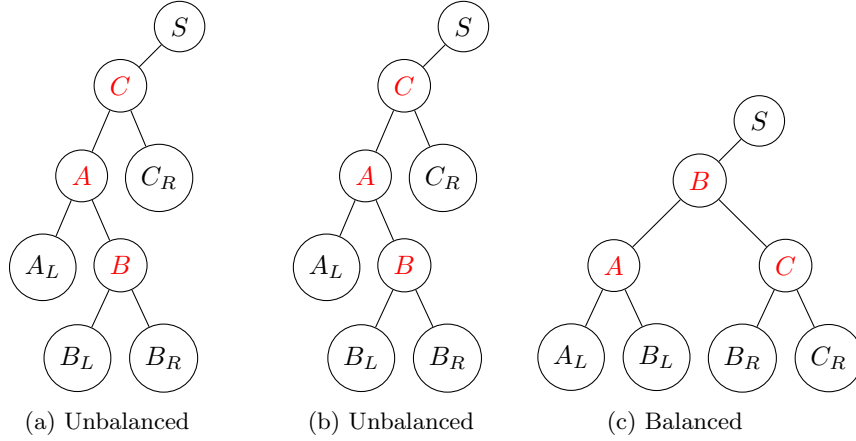


Figure 5: LR rotation. S represents an arbitrary parent node. A_L , B_L , B_R and C_R represent sub-trees. The red 3-node sequence for the AVL BST case is chosen such that C has $b = 2$.

2 Generalisation on rotation

Generalisation of the AVL BST balancing mechanism can be made by two ways. The first is to relax Eq. 1, i.e. the minimum absolute node balance factor value, above which the BST is considered unbalanced. That is,

$$|b| \leq m \quad (2)$$

for the BST to be balanced. $m = 1$ for AVL BST. The second is that for $m > 1$, the number of nodes considered for rotation can be increased. Define this value to be the *rotation length* l . Generally,

$$3 \leq l \leq m + 2.$$

The following subsections demonstrate the varying of parameters m and l via examples.

2.1 Balance factor threshold, m

For the example of this subsection, $l = 3$. Consider the sequence $\Omega = \{9, 7, 5, 3, 1\}$ to be inserted into a BST. The self-balancing of cases $m = 1$ and $m = 2$ are shown in Figs. 6 and 7, respectively.

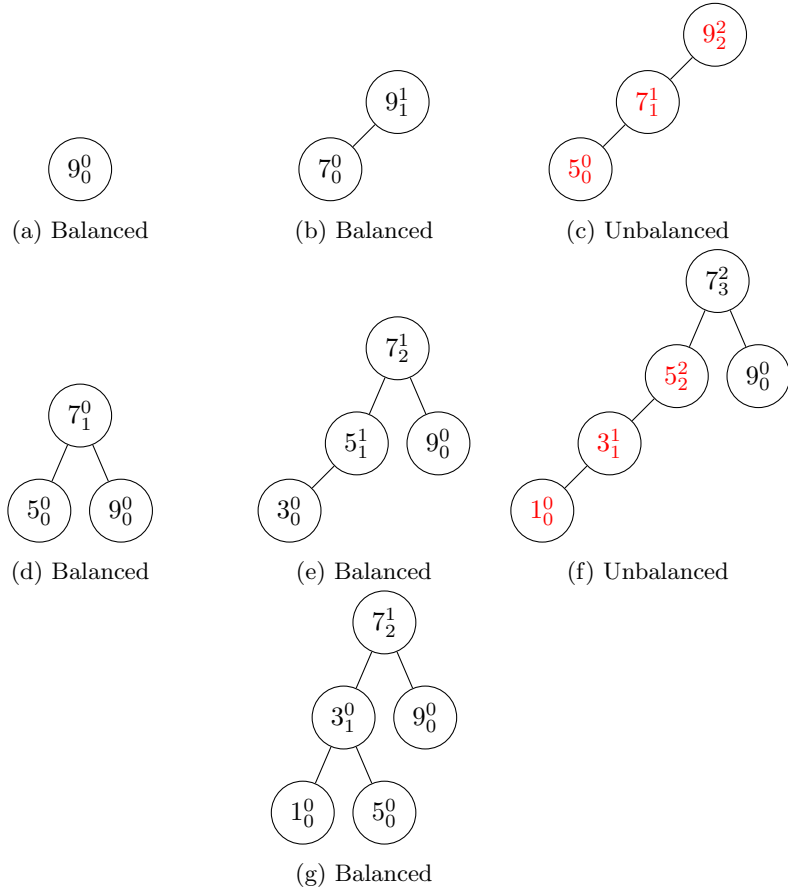


Figure 6: Balancing mechanism for AVL ($m = 1$, $l = 3$) BST for the sequence $\Omega = \{9, 7, 5, 3, 1\}$. Subscripts and superscripts indicate height h and balance factor b for each node. The ($l = 3$) red nodes are to be considered for rotation for the next step.

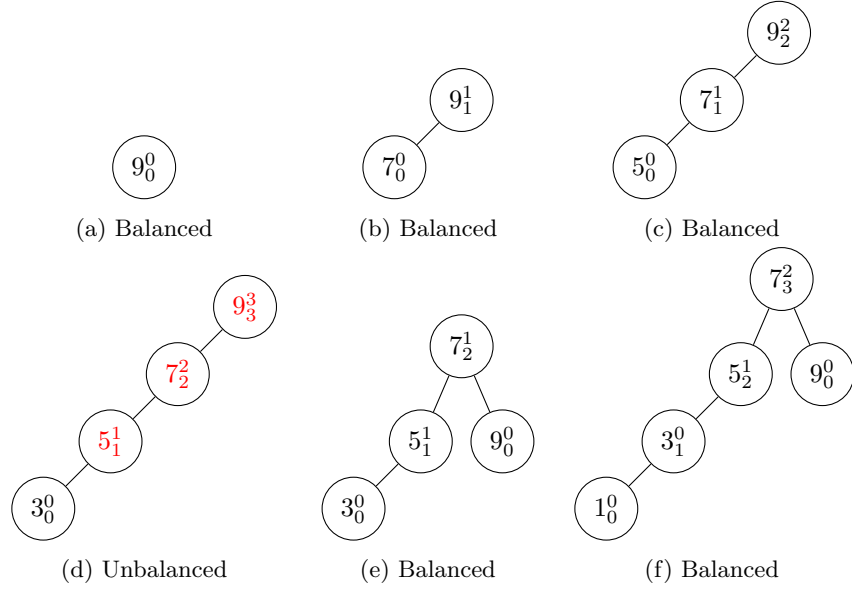


Figure 7: Balancing mechanism for $m = 2, l = 3$, BST for the sequence $\Omega = \{9, 7, 5, 3, 1\}$. Subscripts and superscripts indicate height h and balance factor b for each node. The ($l = 3$) red nodes are to be considered for rotation in the next step.

2.2 Rotation length, l

Consider again the sequence $\Omega = \{9, 7, 5, 3, 1\}$, but now with $m = 2$ and $l = 4$. The self balancing mechanism is shown in Fig. 8. Note that, when compared to the ($m = 2, l = 3$) case of Fig. 7, this ($m_2, l = 4$) case has a different BST height, i.e. height of root node.

As a final example, consider the sequence $\Omega = \{40, 20, 10, 25, 30, 22, 28\}$, to be inserted into the BST with $m = 3$ and $l = 4$. This case demonstrates the self balancing mechanism when the nodes being considered for rotation are not ‘straight’, like that of Fig. 3c or Fig. 3d (but of length $l = 4$). As shown in Fig. 9, the idea is to straighten the l nodes, and then bend $\lfloor l/2 \rfloor$ of them via sequence of 3-node rotations. The sense of straightening depends on the b value of the leading node considered for rotation, i.e. for $b < 0$, the nodes descend to the left, and vice versa. **Comparing with the standard AVL ($m = 1, l = 3$) balancing for the same sequence in Fig. 10, the ($m = 3, l = 4$) balancing BST saves 1 rotation step, i.e. 5 against 4.**

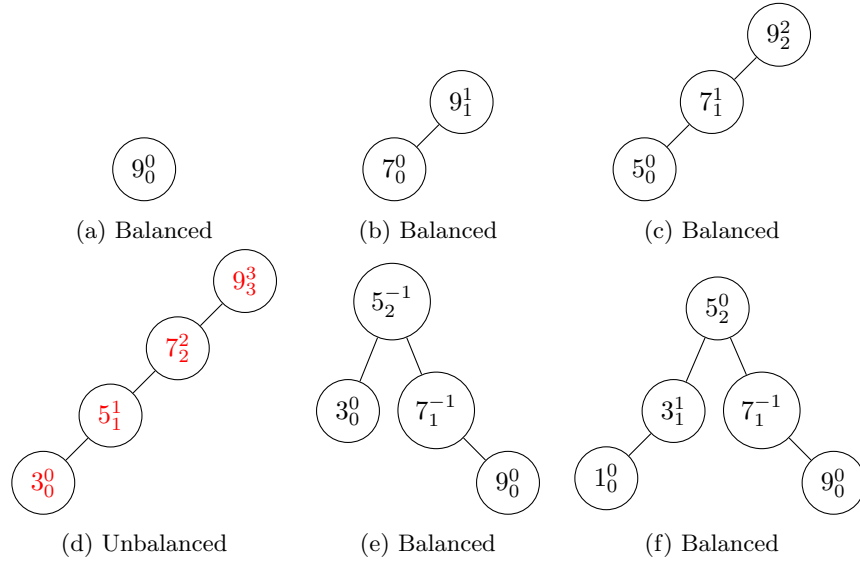


Figure 8: Balancing mechanism for $m = 2$, $l = 3$, BST for the sequence $\Omega = \{9, 7, 5, 3, 1\}$. Subscripts and superscripts indicate height h and balance factor b for each node. The ($l = 3$) red nodes are to be considered for rotation in the next step.

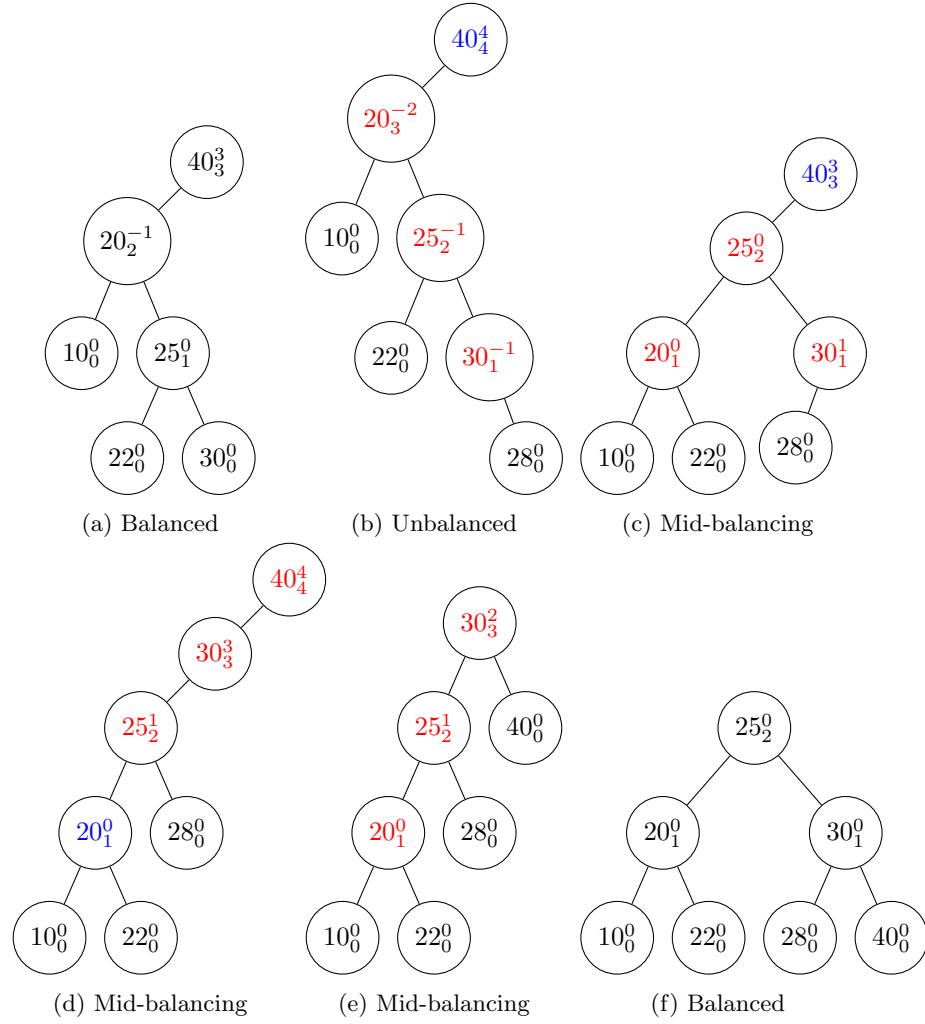


Figure 9: Balancing mechanism for $m = 3$, $l = 4$, BST for the sequence $\Omega = \{40, 20, 10, 25, 30, 22, 28\}$. Subscripts and superscripts indicate height h and balance factor b for each node. Red indicates the nodes considered for rotation in the next step. The blue nodes are to be consider for rotation only in subsequent steps. Each rotation step involves only 3 (red) nodes. A total of 4 rotations have been executed for this case.

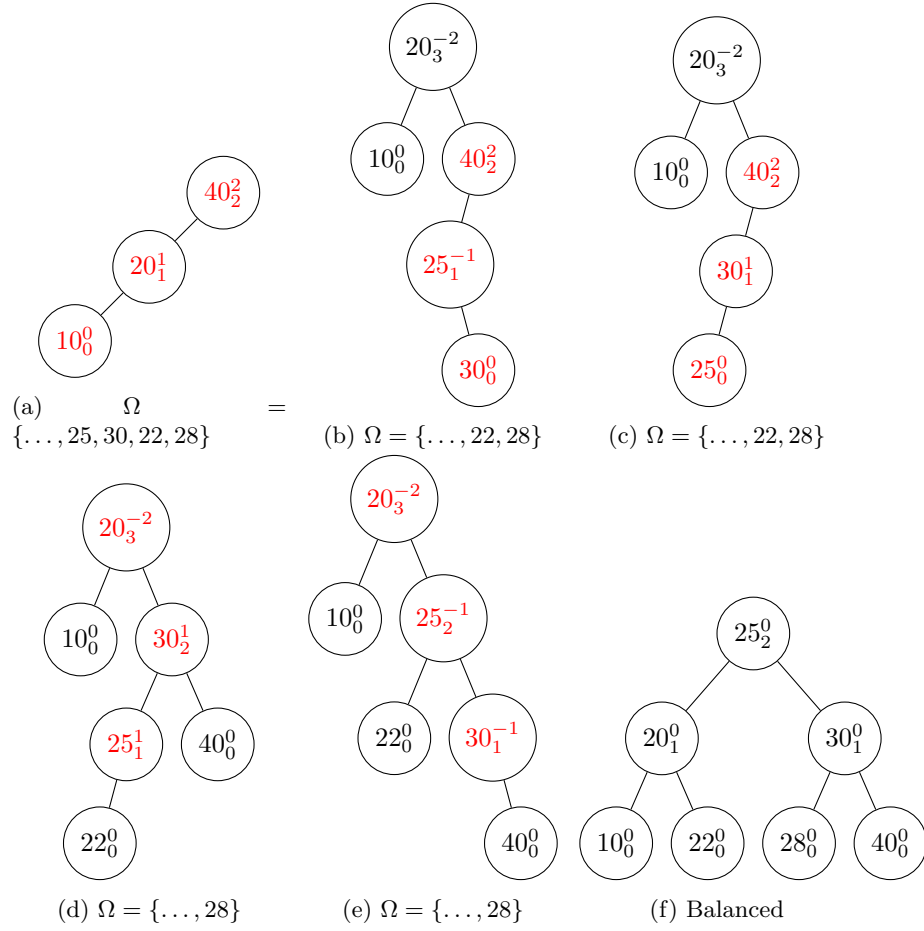


Figure 10: Balancing mechanism for $m = 1$, $l = 3$, (AVL) BST for the sequence $\Omega = \{40, 20, 10, 25, 30, 22, 28\}$. Subscripts and superscripts indicate height h and balance factor b for each node. Red indicates the nodes considered for rotation in the next step. A total of 5 rotations have been executed for this case.