Why Our AVL Rebalancing Algorithms Works
Remember that Z is the node that fails the AVL condition, Y is Z’s tallest child, X is Y’s tallest child.

Case 1: Y is Z’s left child, X is Y’s left child

One of nodes t1 and t2 must have height h; the other has height h-1. t3 has height h.
Putting these heights into the tree we build we see that every node satisfies the AVL property:

One of nodes t1 and t2 has height h; the other has height h-1. t3 has height h.
Furthermore, t1 and t2 have the same relation to a as before, all of the values in t3 are greater than the value of b and less than the value of c, and t4 remains the right child of c. This means that the new tree is a Binary Search tree that satisfies the AVL property.
The other three cases are similar; we will go through them more quickly.
Case 2: Y is Z’s right child, X is Y’s right child. This is symmetric to case 1.
Case 3: Y is Z’s left child and X is Y’s right child:

One of nodes t2 and t3 has height h; the other has height h-1.
Case 4: Y is Z’s right child and X is Y’s left child; this is the mirror image of Case 3:

One of nodes t2 and t3 has height h; the other has height h-1.