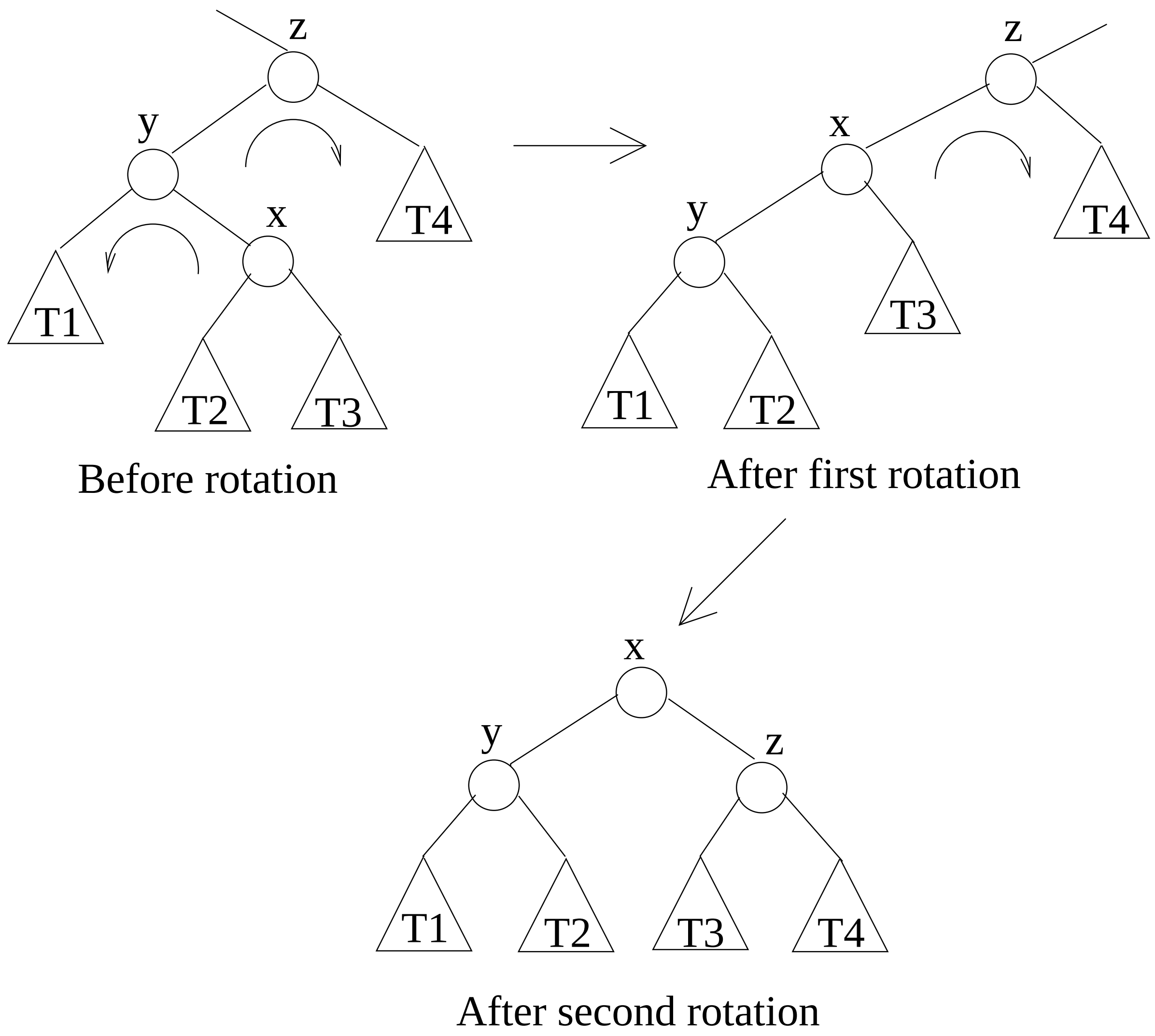
Double rotation

- A double rotation is equivalent to a pair of single rotation applied one after the other.
- Clearly, the pair of single rotation defining the double rotation can involve a combination of single rotation types.
- The access path consists of a left and right or a right and left pair as first two edges on the access path.
- We can identify a single rotation as **clockwise/right** or as **anti-clockwise/left** according to the direction of rotation.
- Implying a double rotation consists of:
  - A pair (**anti-clockwise**, **clockwise**) rotation, or
  - A pair (**clockwise**, **anti-clockwise**) rotation

Double rotation



Insertion to AVL tree

- Only one rotation is required to rebalance the tree after an insertion.

  Proof: The inserted item say k is placed as leaf.

  - The nodes whose heights get affected due to insertion are those on the access path for k.
  - More specifically the heights of these nodes either remain same or increase by 1 unit
  - Consider node n on access path. Two possibilities are either (i) height of n remains same, or (ii) height of n increases.

Insertion to AVL tree

- **Proof(contd.)**
  - If n height does not change due to insertion, then no rotation is needed at n. However, imbalance may be at a node above n along the access path.
  - Note that we only consider the *critical node* on the access path. The critical node is the root of the smallest subtree which becomes non AVL due to insertion.
  - Let height of n increase by 1. Then a rotation is needed at n, and after (a double/single) rotation n's height is changed to previous value.
  - Since rotation decrements the height of the root of smallest non-AVL subtree the height of nodes above this on the access path also decremented by 1.

Properties of AVL tree

The number of internal nodes in an AVL tree of height h bounded below by $F_{h+2} - 1$ where F_h is the fibonacci number.

- Let T_h denote the AVL tree of height h with minimum number of nodes.
- T_h must consists of a root, one subtree of height $h-2$, and another of height $h-1$
- Clearly, N_h must satisfy the following equation:

$$N_h = \begin{cases} 1 & \text{for } h = 0 \\ 2 & \text{for } h = 1 \\ N_{h-1} + N_{h-2} + 1 & \text{for } h \geq 1 \end{cases}$$

- $N_0 = 1, F_2 = 1 \implies N_0 \geq F_2 - 1$
- $N_1 = 2, F_3 = 2 \implies N_0 \geq F_3 - 1$
- By induction $N_{h+1} = N_{h-1} + N_h + 1 \geq F_{h+2} - 1 + F_{h+1} = F_{h+3} - 1$

Analysis of AVL tree

The height h of an AVL tree with n internal nodes satisfies

$$\log(n+1) < h \leq 1.440\log(n+2) - 0.328$$

- Fibonacci number are given by
$$F_n = \frac{1}{\sqrt{5}}(\Phi^n - \bar{\Phi}^n)$$
-
$$\begin{aligned} N_h \geq F_{h+2} - 1 \;\Rightarrow\; & N_h \geq \Phi^{h+2}\sqrt{5} - 2 \\ \Rightarrow\; & (N_h + 2)\sqrt{5} \geq \Phi^{h+2} \\ \Rightarrow\; & \log_{\Phi}((N_h + 2)\sqrt{5}) \geq h + 2 \\ \Rightarrow\; & h \leq \log_{\Phi}((N_h + 2)\sqrt{5}) - 2 \\ \Rightarrow\; & h \leq 1.440\log(N_h + 2) - 0.328 \end{aligned}$$
- The left side inequality follows from the fact that with a particular height with maximum number of nodes in an AVL tree occurs when the tree is fully balanced. So the lower bound on height is given by $\log(n+1)$.