8-1 AVL Tree Basic Concepts

An AVL tree is a BST that is guaranteed to always be balanced. We begin with a discussion of its basic structural differences and then discuss the four different balancing situations that can occur when data are inserted or deleted.

- **AVL Tree Balance Factor**
- **Balancing Trees**

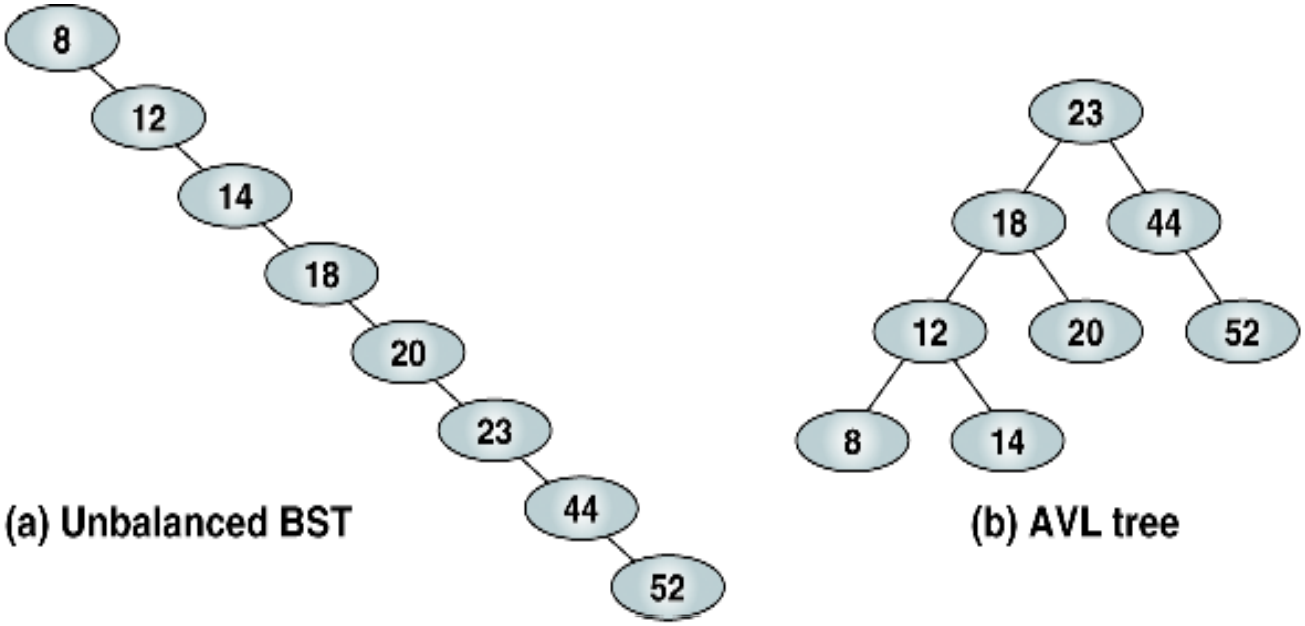
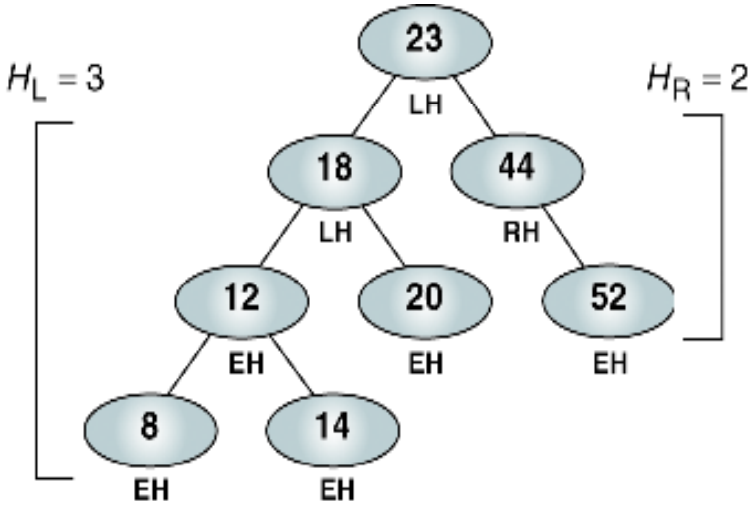
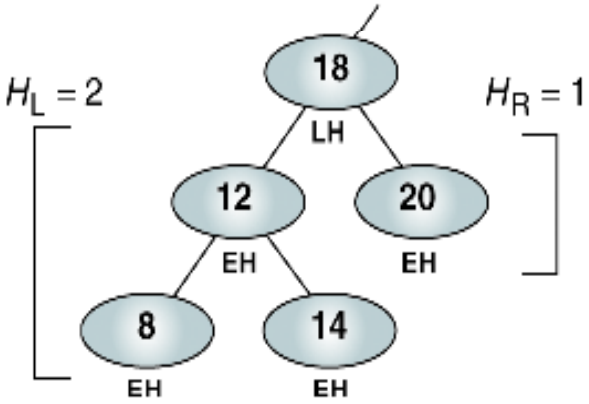


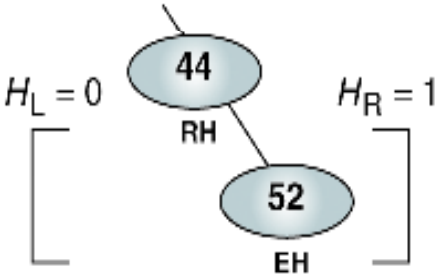
FIGURE 8-1 Two Binary Trees



(a) Tree 23 appears balanced: $H_L - H_R = 1$



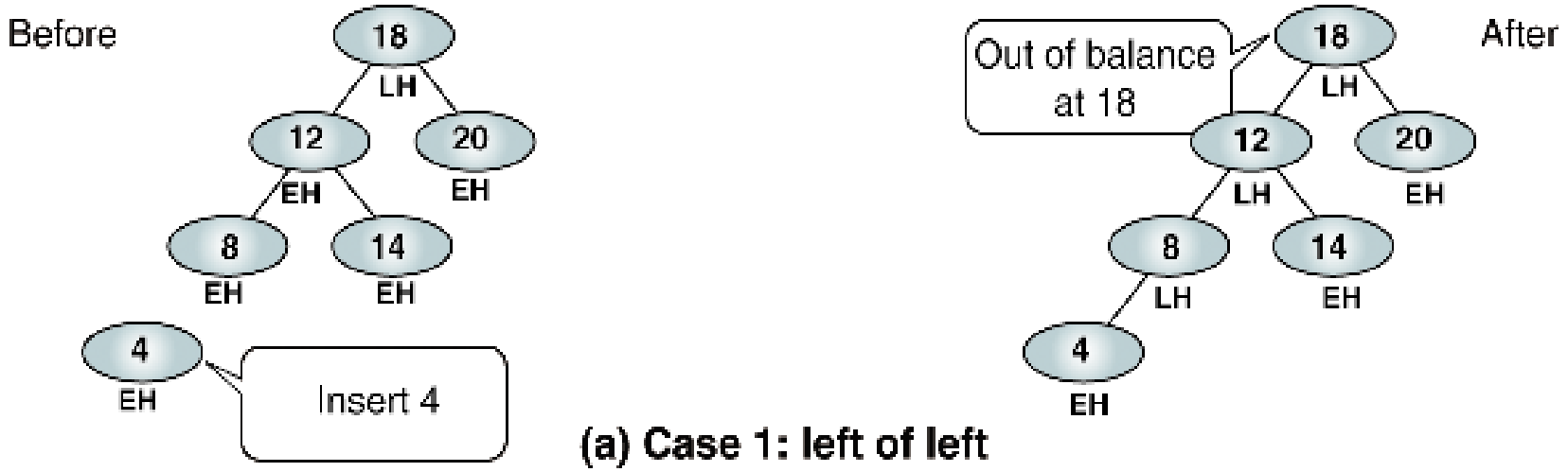
(b) Subtree 18 appears balanced: $H_L - H_R = 1$



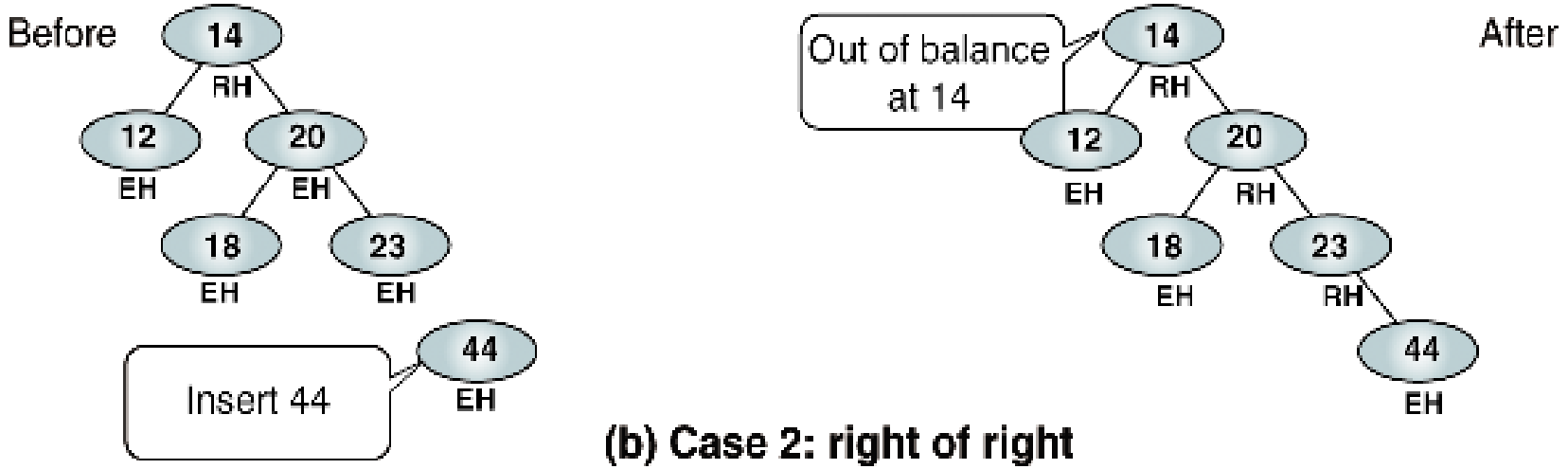
(c) Subtree 44 is balanced: $| H_L - H_R | = 1$

FIGURE 8-2 AVL Tree

FIGURE 8-3 Out-of-balance AVL Trees

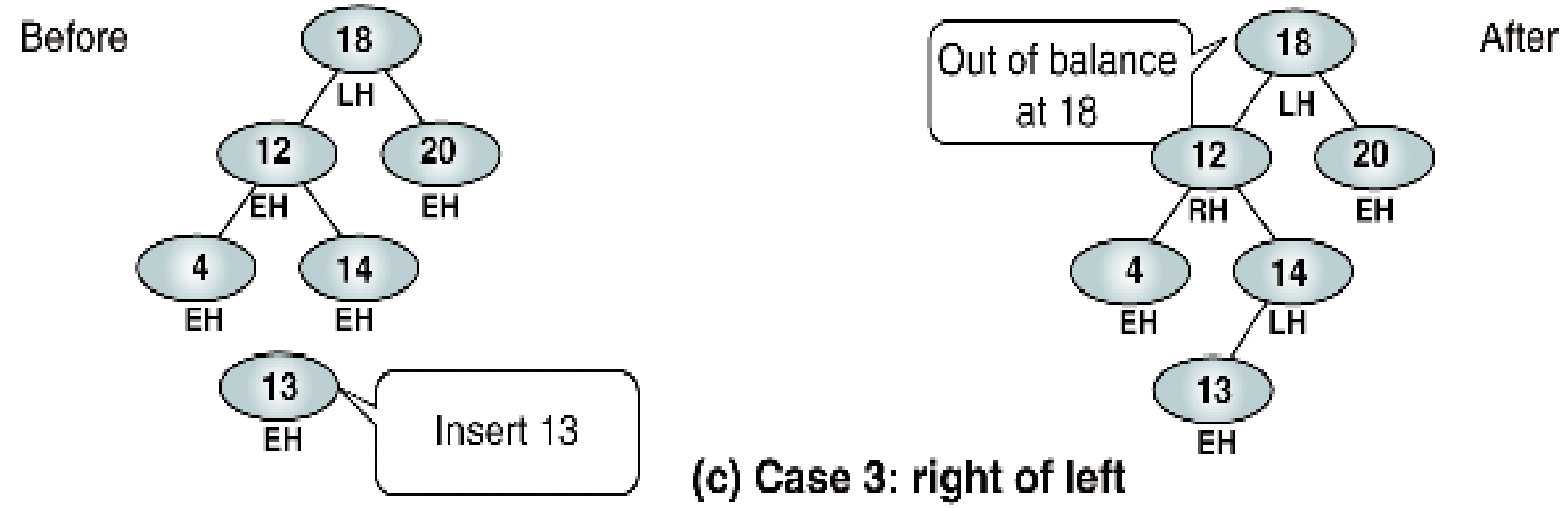


(a) Case 1: left of left

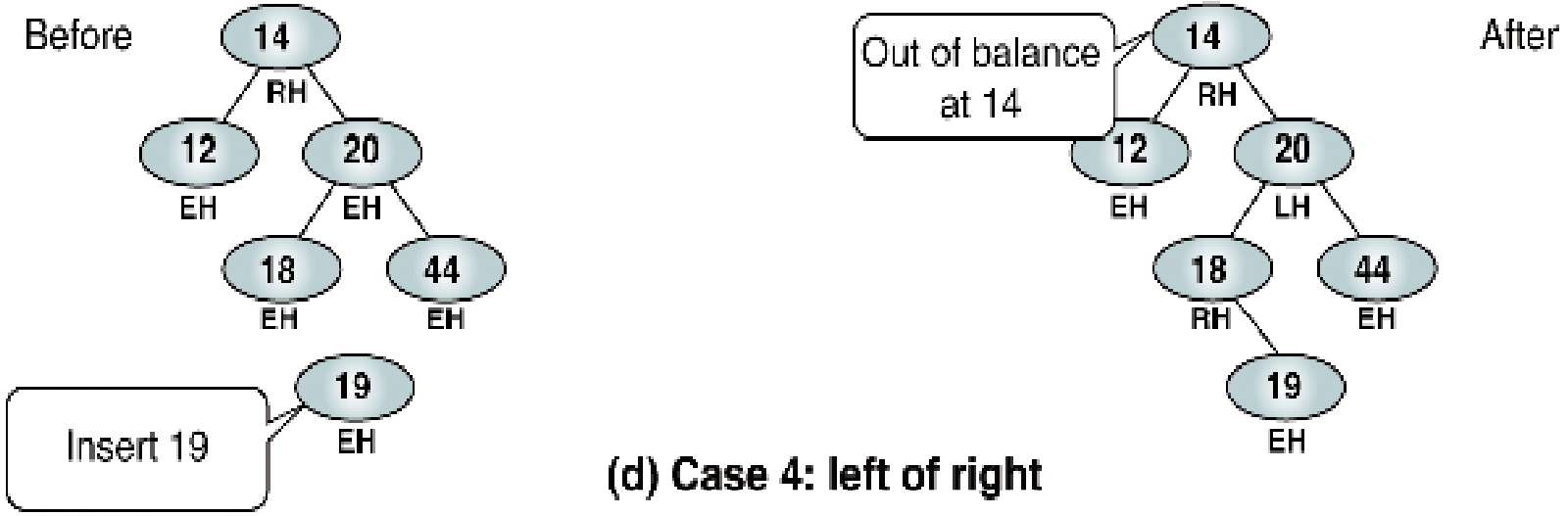


(b) Case 2: right of right

FIGURE 8-3 Out-of-balance AVL Trees (continued)



(c) Case 3: right of left



(d) Case 4: left of right

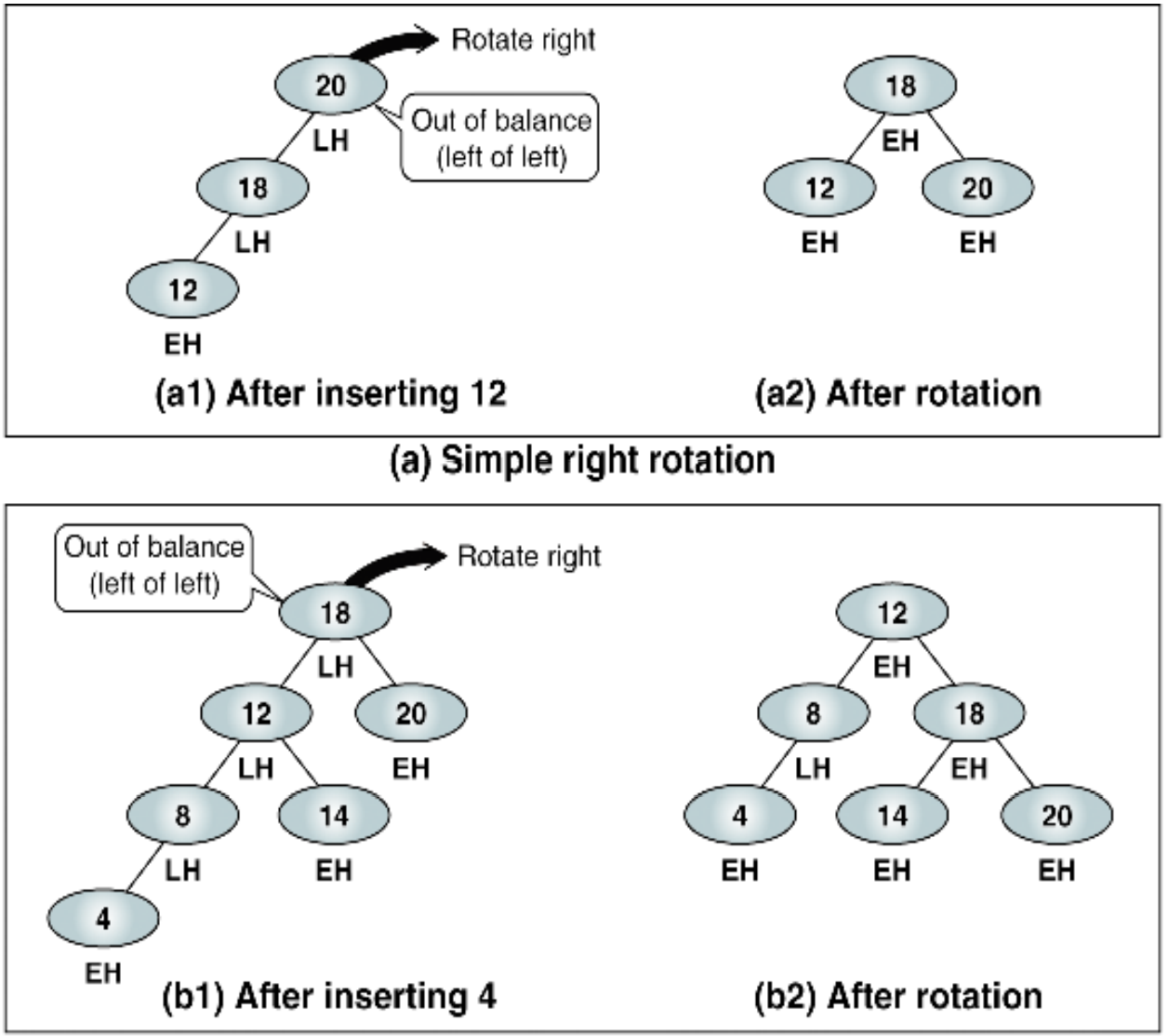
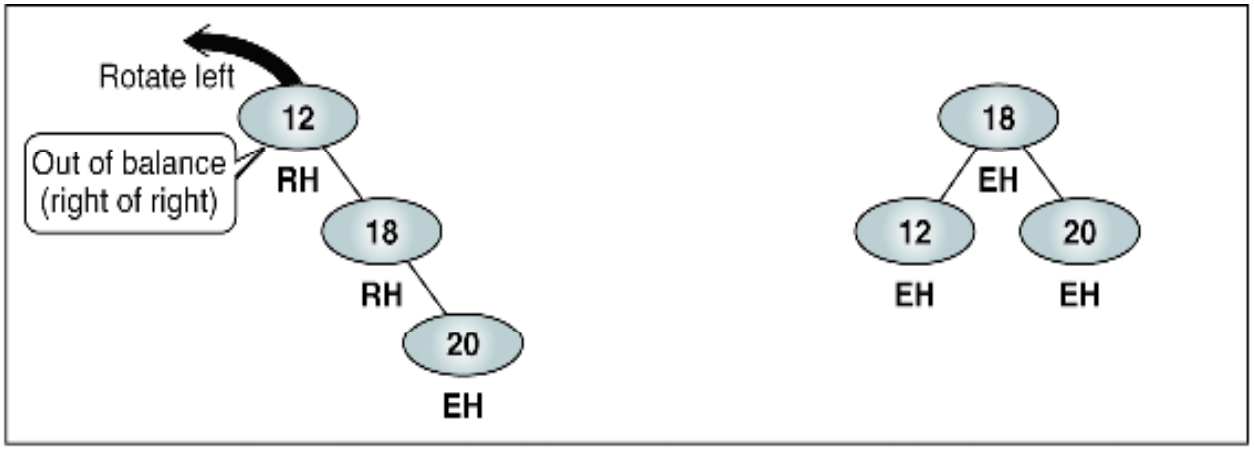
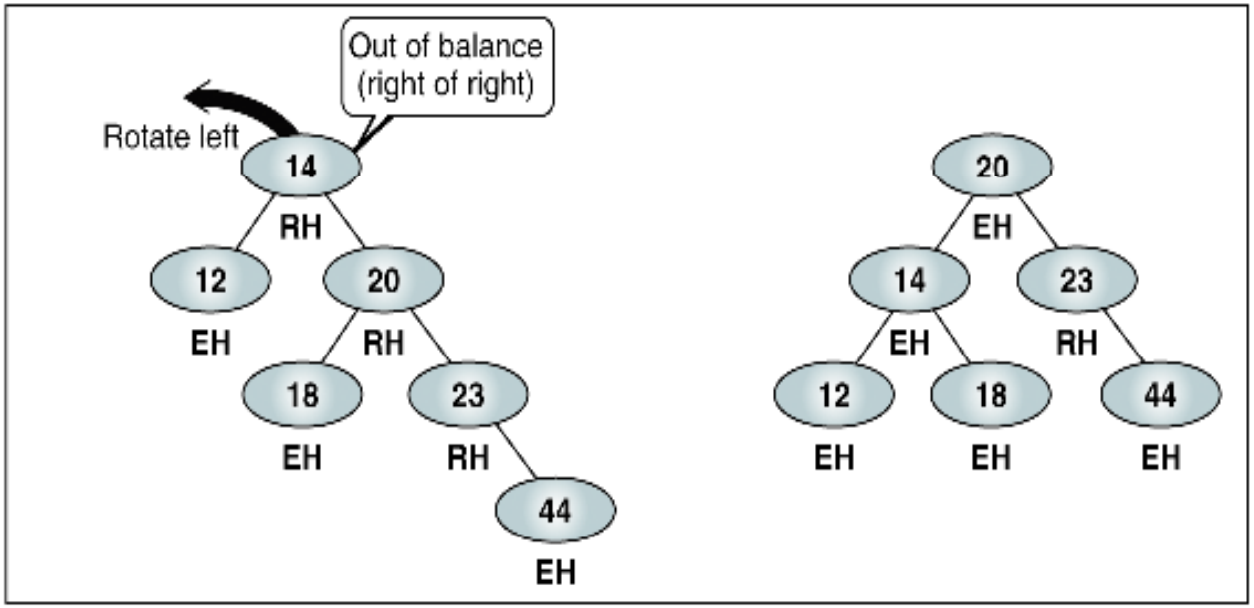


FIGURE 8-4 Left of Left—Single Rotation Right

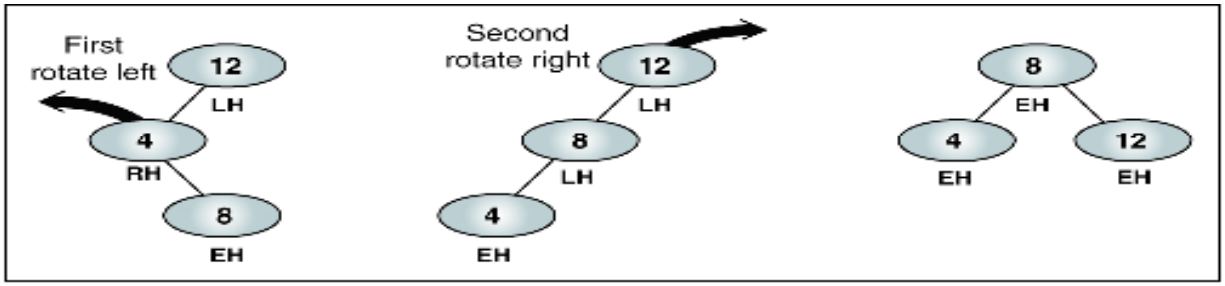


(a) Simple left rotation

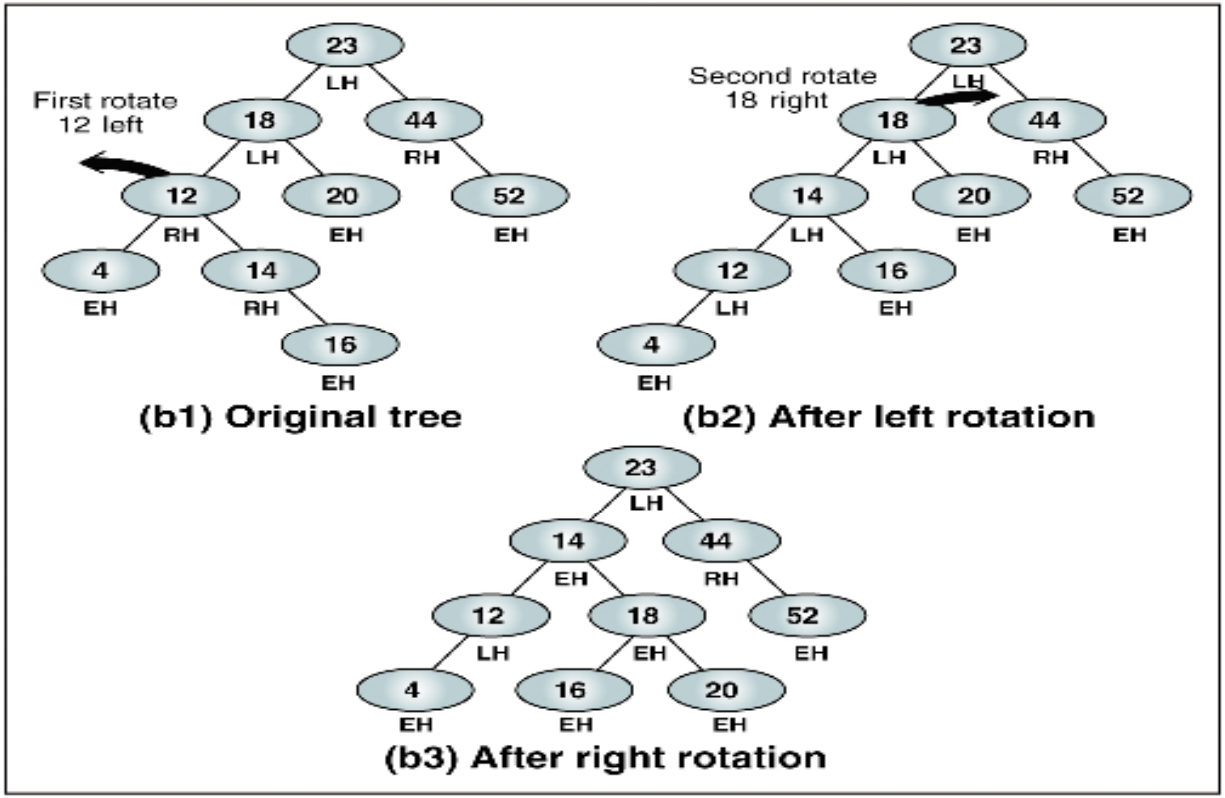


(b) Complex left rotation

FIGURE 8-5 Right of Right—Single Rotation Left



(a) Simple double rotation right



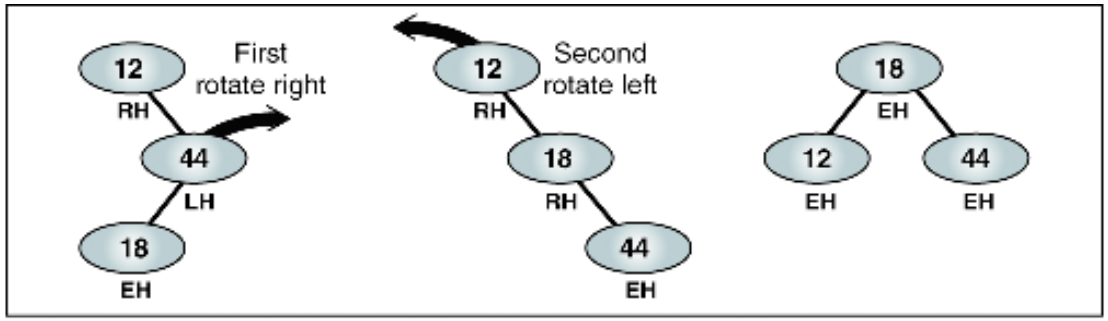
(b1) Original tree

(b2) After left rotation

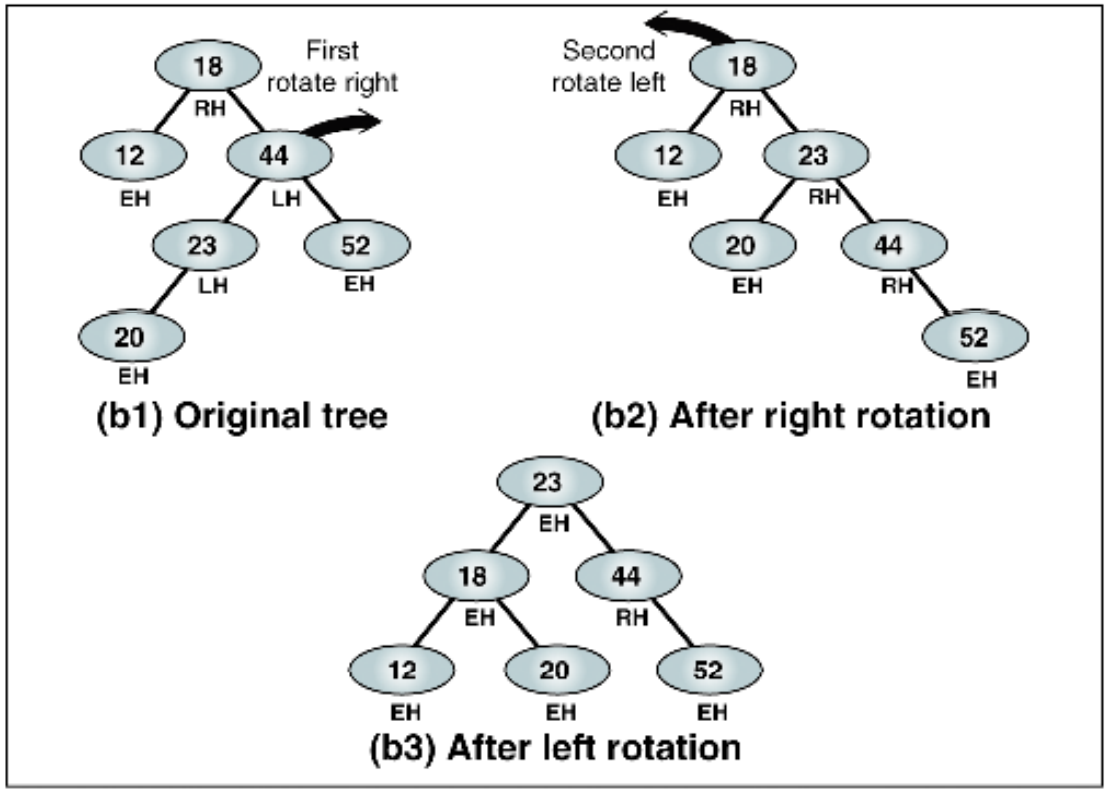
(b3) After right rotation

(b) Complex double rotation right

FIGURE 8-6 Right of Left—Double Rotation Right



(a) Simple double rotation right



(b) Complex double rotation right

FIGURE 8-7 Left of Right—Double Rotation Right