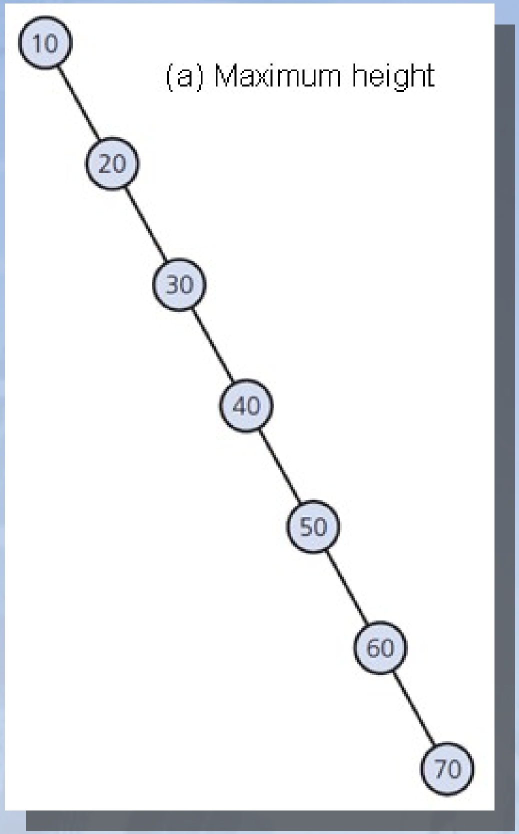
Balanced Search Trees

Chapter 19

Balanced Search Trees

- Height of binary search tree
  - Sensitive to order of additions and removals
- Various search trees can retain balance
  - Despite additions and removals

Balanced Search Trees



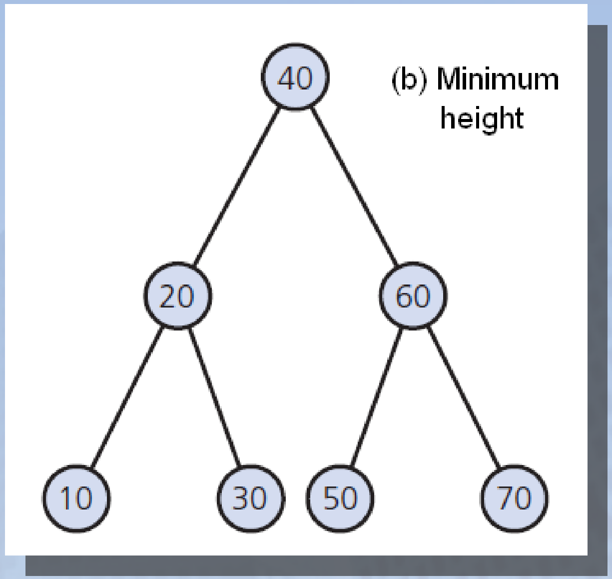


FIGURE 19-1 The tallest and shortest binary search trees containing the same data

AVL Trees

- An AVL tree
  - A balanced binary search tree
- Maintains its height close to the minimum
- Rotations restore the balance

AVL Trees

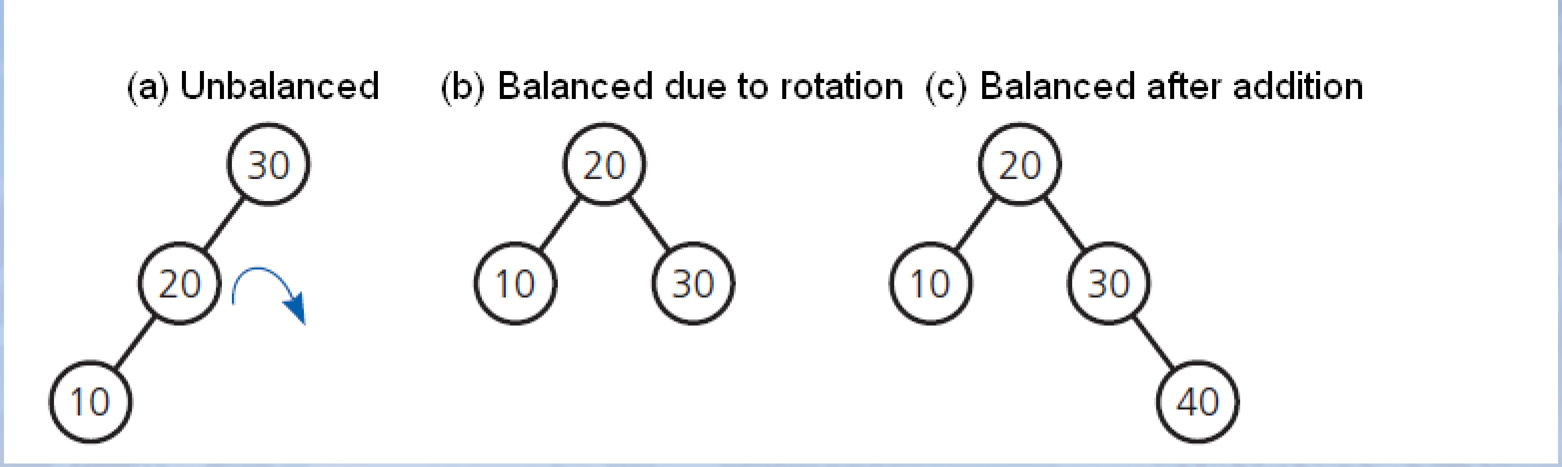


FIGURE 19-2 An unbalanced binary search tree

AVL Trees

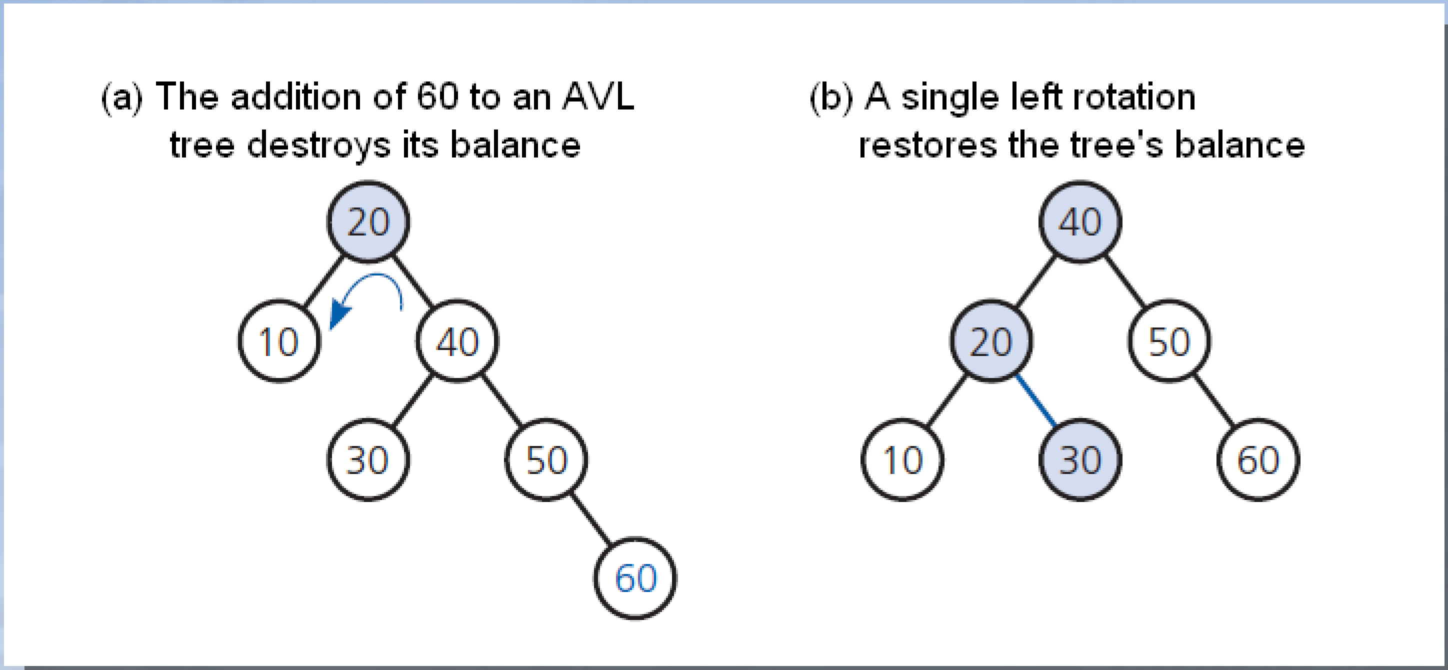


FIGURE 19-3 Correcting an imbalance in an AVL tree due to an addition by using a single rotation to the left

AVL Trees

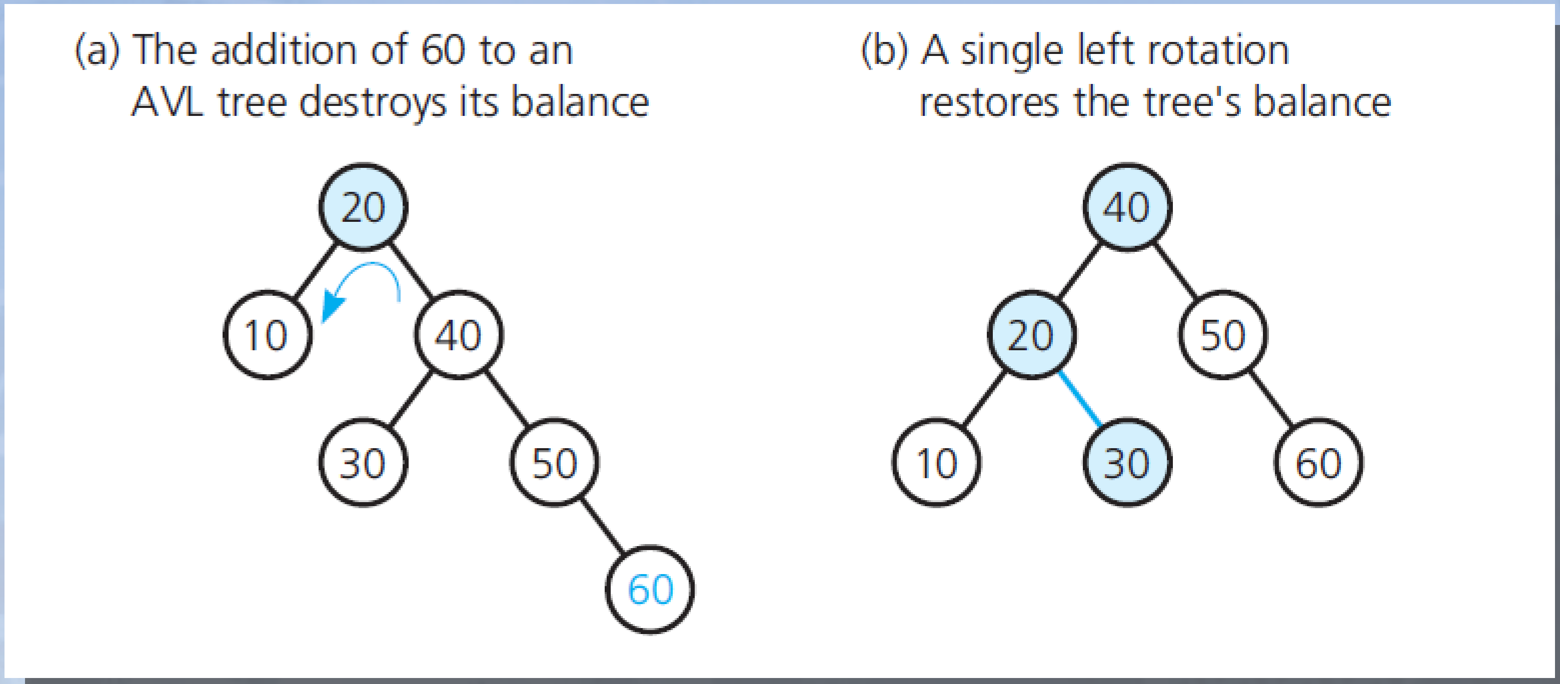


FIGURE 19-3 Correcting an imbalance in an AVL tree due to an addition by using a single rotation to the left

AVL Trees

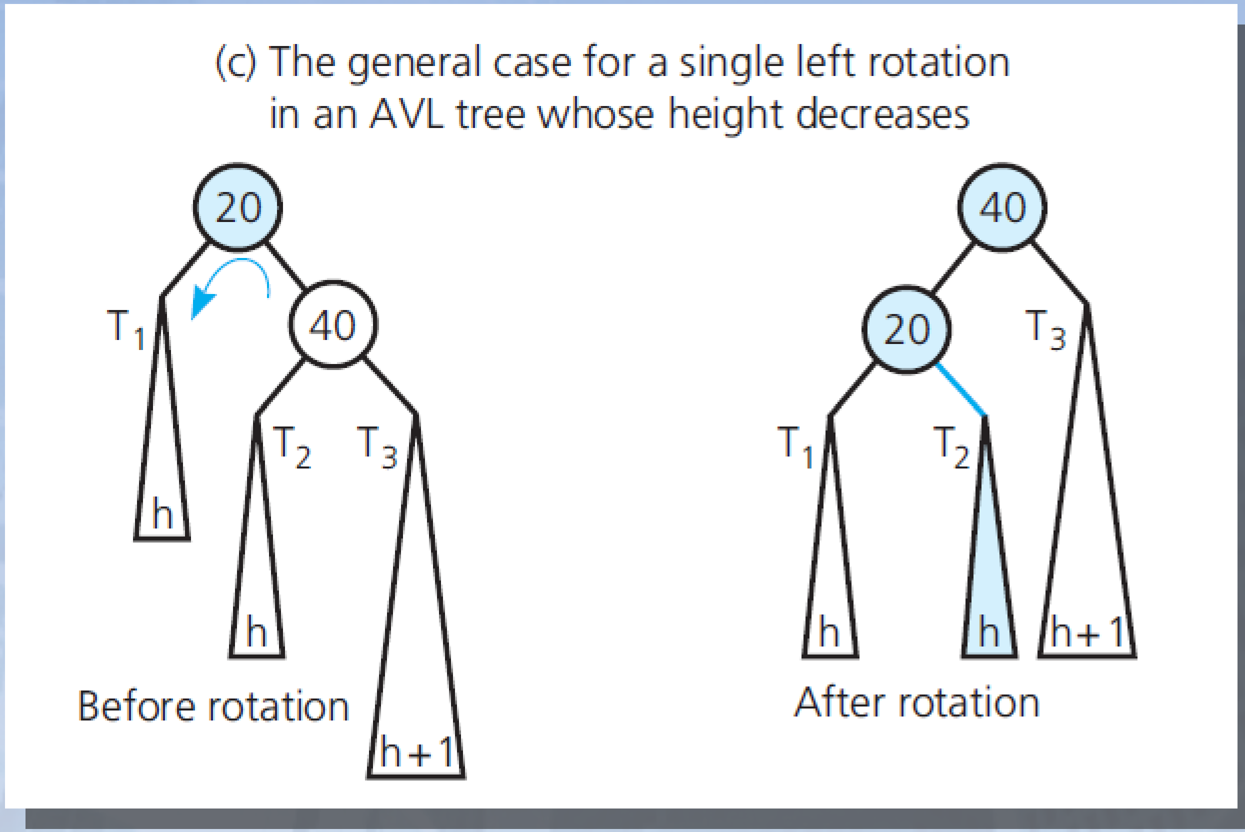


FIGURE 19-3 Correcting an imbalance in an AVL tree due to an addition by using a single rotation to the left

AVL Trees

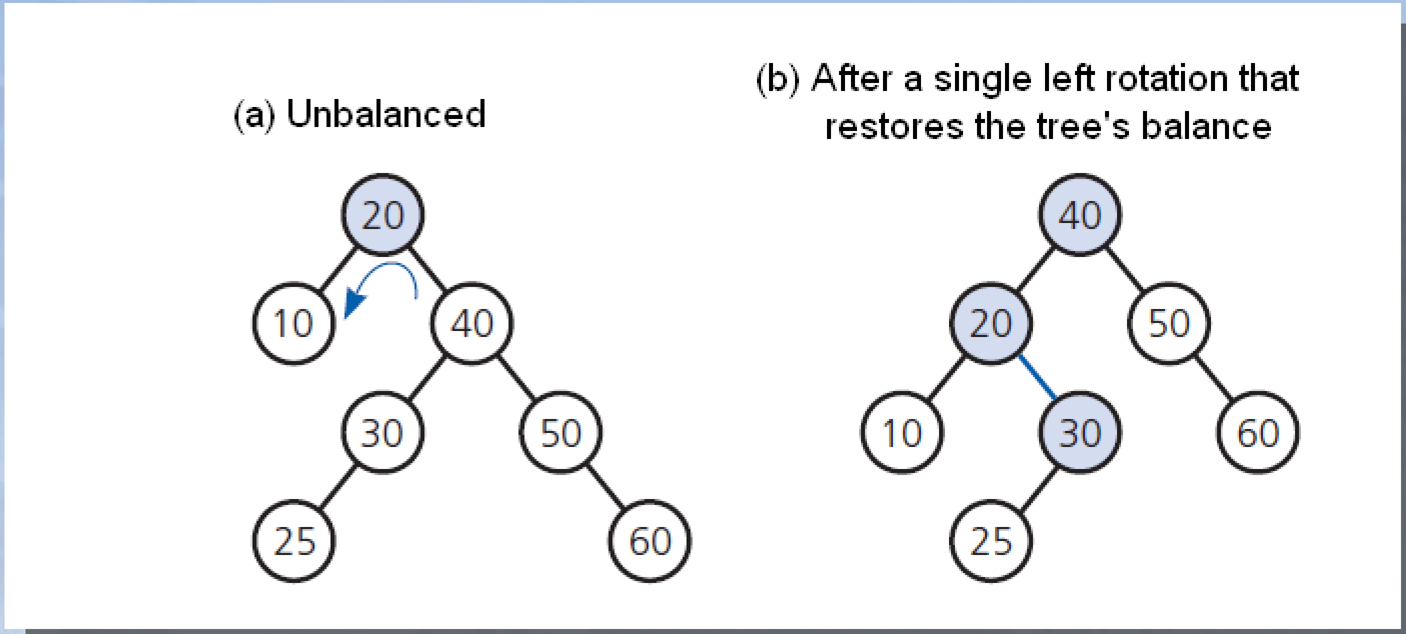


FIGURE 19-4 A single rotation to the left that does not affect the height of an AVL tree

AVL Trees

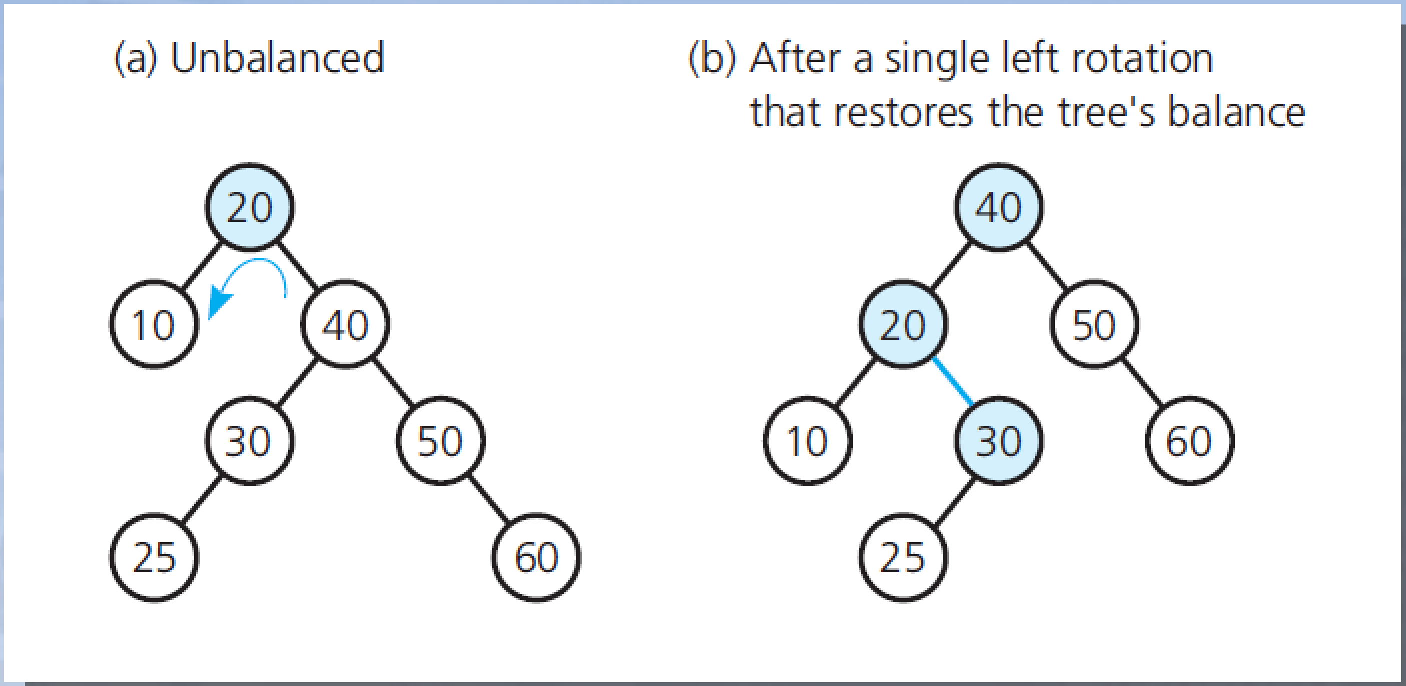


FIGURE 19-4 A single rotation to the left that does not affect the height of an AVL tree

AVL Trees

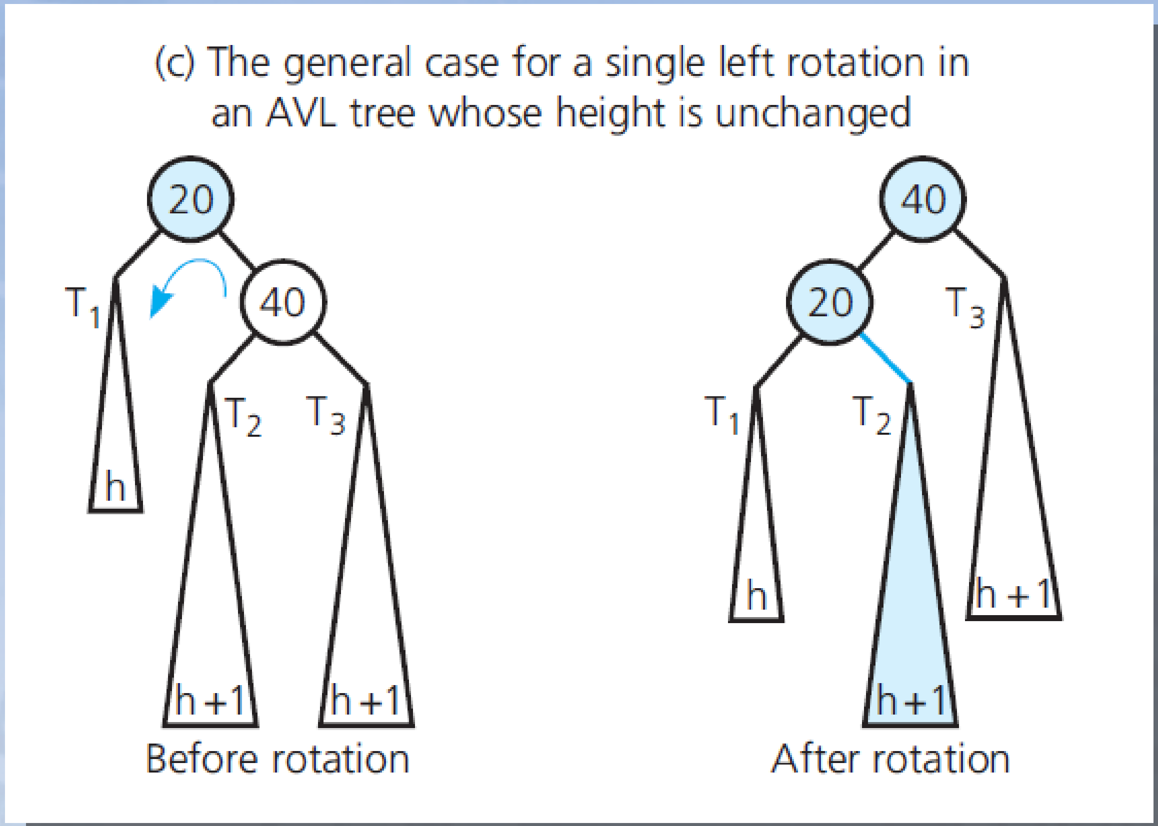


FIGURE 19-4 A single rotation to the left that does not affect the height of an AVL tree

AVL Trees

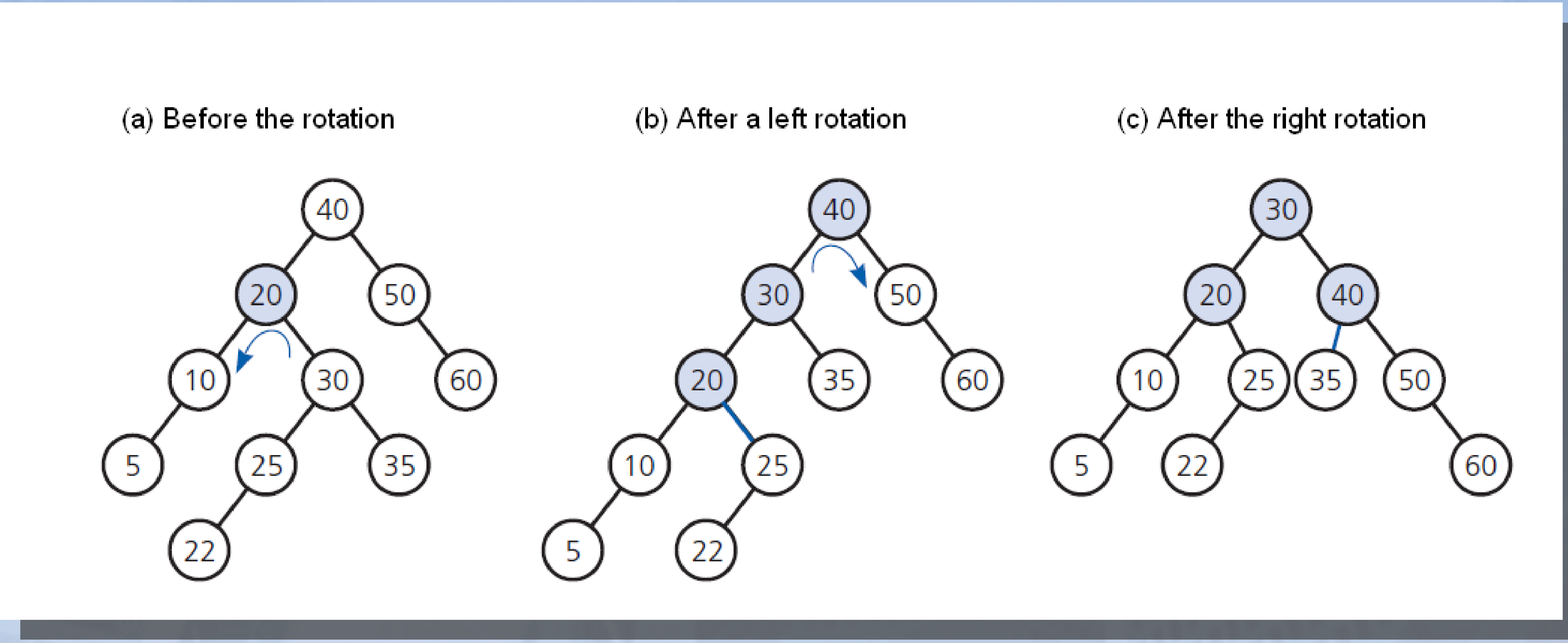


FIGURE 19-5 A double rotation that decreases the height of an AVL tree

AVL Trees

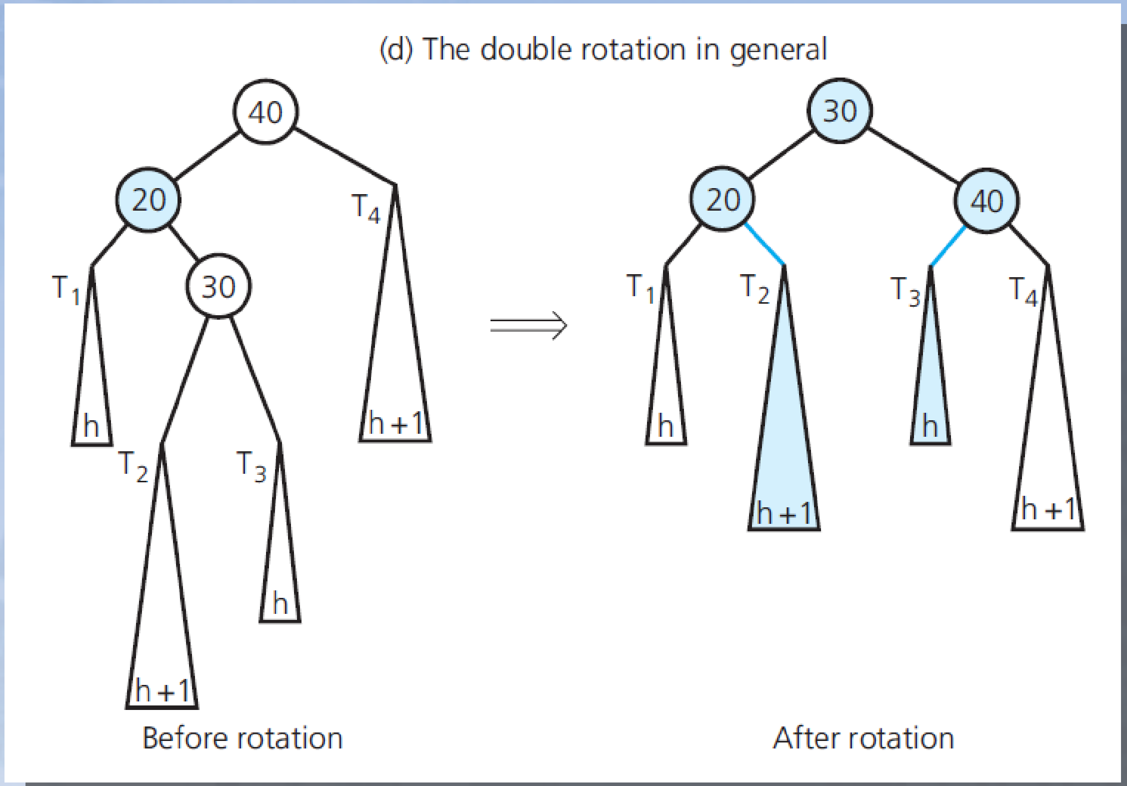


FIGURE 19-5 A double rotation that decreases the height of an AVL tree

2-3 Trees

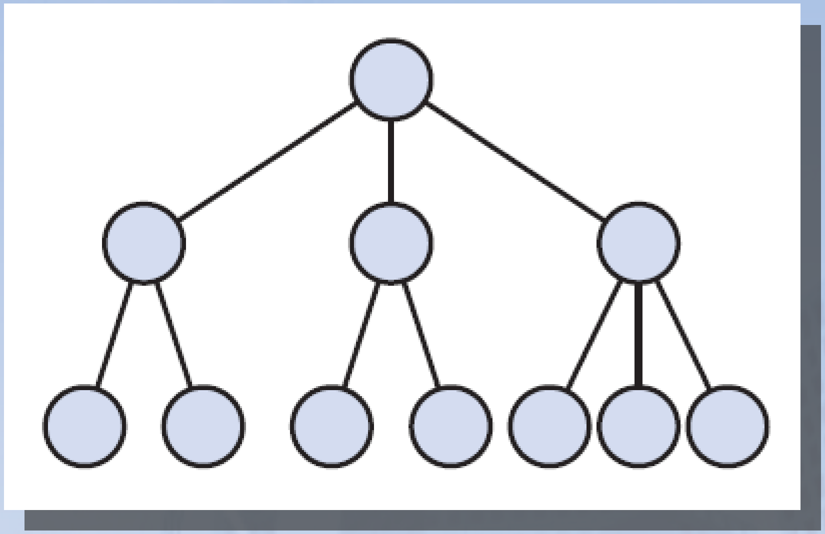


FIGURE 19-6 A 2-3 tree of height 3

2-3 Trees

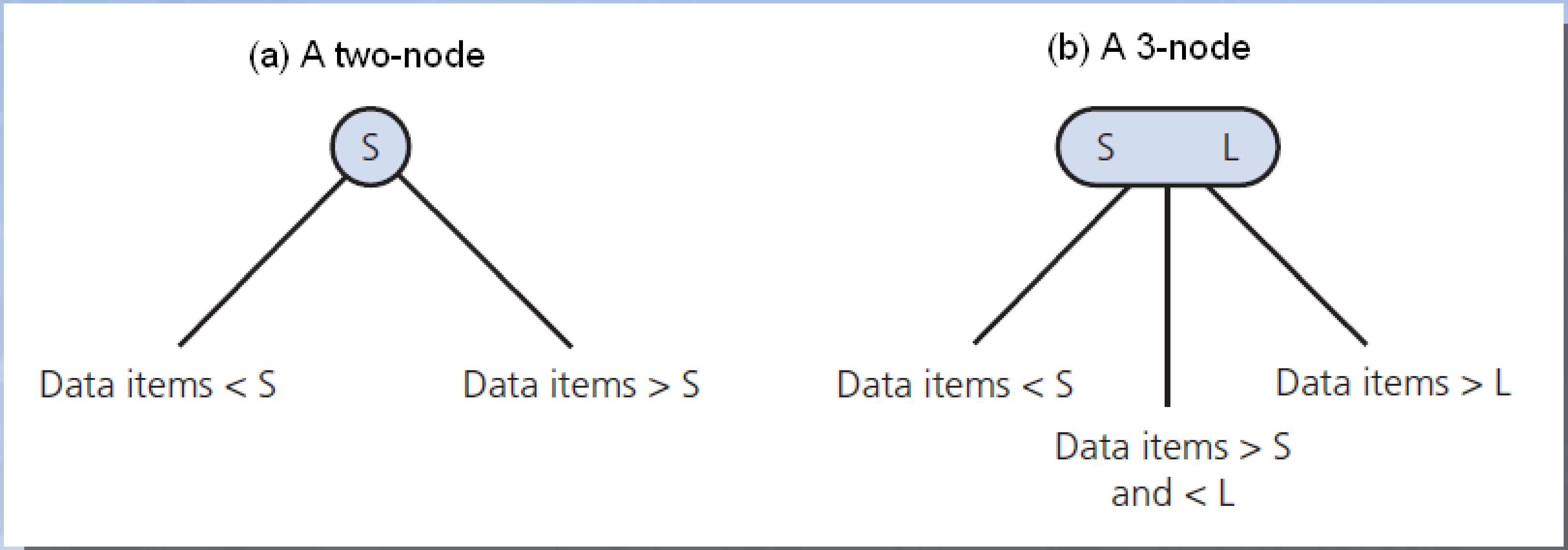


FIGURE 19-7 Nodes in a 2-3 tree

2-3 Trees

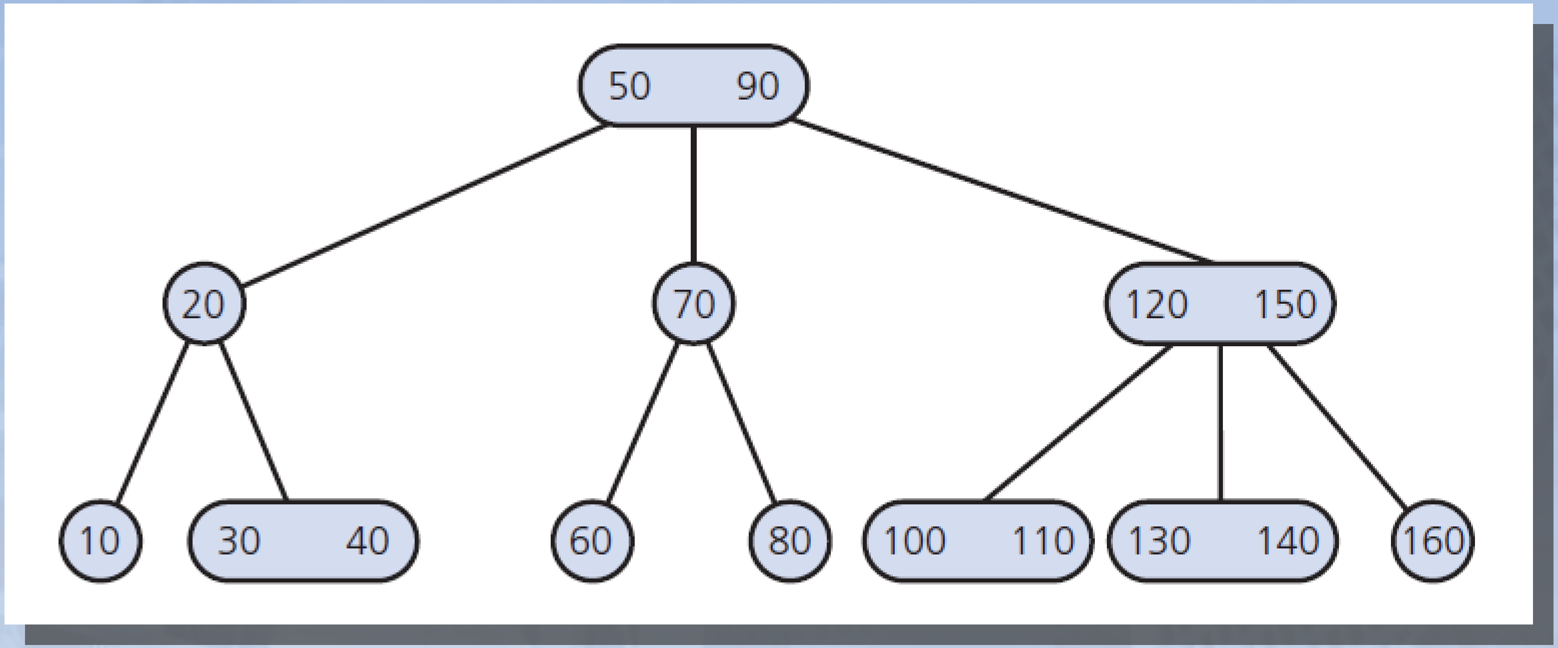


FIGURE 19-8 A 2-3 tree

2-3 Trees

```
/** A class of nodes for a link-based 2-3 tree.
 @file TriNode.h */

#ifndef TRI_NODE_
#define TRI_NODE_

template<class ItemType>
class TriNode
{
private:
   ItemType smallItem;                             // Data portion
   ItemType largeItem;                             // Data portion
   std::shared_ptr<TriNode<ItemType>> leftChildPtr;  // Left-child pointer
   std::shared_ptr<TriNode<ItemType>> midChildPtr;   // Middle-child pointer
   std::shared_ptr<TriNode<ItemType>> rightChildPtr; // Right-child pointer

public:
   TriNode();
```

LISTING 19-1 A header file for a class of nodes for a 2-3 tree

2-3 Trees

```
    bool isLeaf() const;
    bool isTwoNode() const;
    bool isThreeNode() const;

    ItemType getSmallItem() const;
    ItemType getLargeItem() const;

    void setSmallItem(const ItemType& anItem);
    void setLargeItem(const ItemType& anItem);
    auto getLeftChildPtr() const;
    auto getMidChildPtr() const;
    auto getRightChildPtr() const;

    void setLeftChildPtr(std::shared_ptr<TriNode<ItemType>> leftPtr);
    void setMidChildPtr(std::shared_ptr<TriNode<ItemType>> midPtr);
    void setRightChildPtr(std::shared_ptr<TriNode<ItemType>> rightPtr);
}; // end TriNode
#include "TriNode.cpp"
#endif
```

LISTING 19-1 A header file for a class of nodes for a 2-3 tree

Traversing a 2-3 Tree

```
// Traverses a nonempty 2-3 tree in sorted order.
inorder(23Tree: TwoThreeTree): void
{
   if (23Tree's root node r is a leaf)
      Visit the data item(s)
   else if (r has two data items)
   {
      inorder(left subtree of 23Tree's root)
      Visit the first data item
      inorder(middle subtree of 23Tree's root)
      Visit the second data item
      inorder(right subtree of 23Tree's root)
   }
   else // r has one data item
   {
      inorder(left subtree of 23Tree's root)
      Visit the data item
      inorder(right subtree of 23Tree's root)
   }
}
```

Performing the analogue of an inorder traversal on a binary tree:

Searching a 2-3 Tree

```
// Locates the value target in a nonempty 2-3 tree. Returns either the located
// entry or throws an exception if such a node is not found.
findItem(23Tree: TwoThreeTree, target: ItemType): ItemType
{
   if (target is in 23Tree's root node r)
   {  // The data item has been found
      treeItem = the data portion of r
      return treeItem // Success
   }
   else if (r is a leaf)
      throw NotFoundException  // Failure

   // Else search the appropriate subtree
   else if (r has two data items)
   {
      if (target < smaller data item in r)
```

Retrieval operation for a 2-3 tree

Searching a 2-3 Tree

```
    else if (r has two data items)
    {
       if (target < smaller data item in r)
          return findItem(r's left subtree, target)
       else if (target < larger data item in r)
          return findItem(r's middle subtree, target)
       else
          return findItem(r's right subtree, target)
    }
    else // r has one data item
    {
       if (target < r's data item)
          return findItem(r's left subtree, target)
       else
          return findItem(r's right subtree, target)
    }
}
```

Retrieval operation for a 2-3 tree

Searching a 2-3 Tree

- Search of a 2-3 and shortest binary search tree approximately same efficiency
  - A binary search tree with *n* nodes cannot be shorter than $\log_2(n + 1)$
  - A 2-3 tree with n nodes cannot be taller than $\log_2(n + 1)$
  - Node in a 2-3 tree has at most two data items
- Searching 2-3 tree is O(log *n*)

Searching a 2-3 Tree

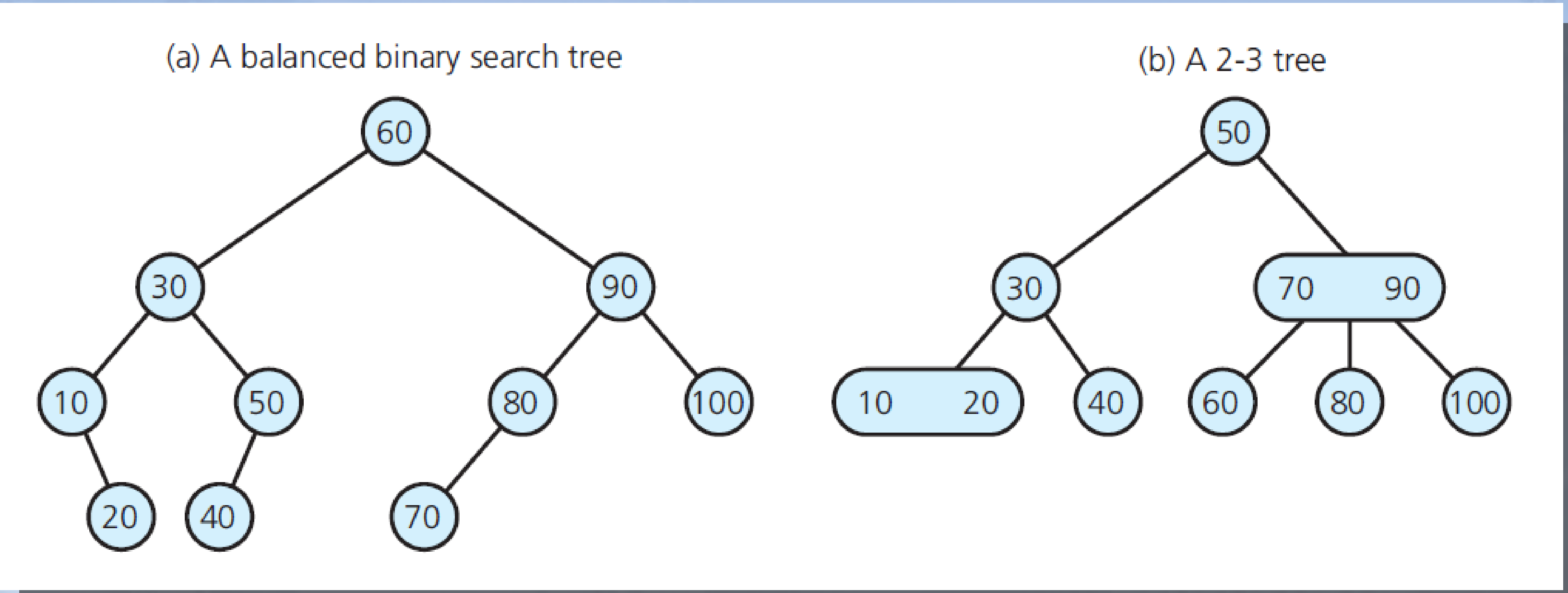


FIGURE 19-9 A balanced binary search tree

Searching a 2-3 Tree

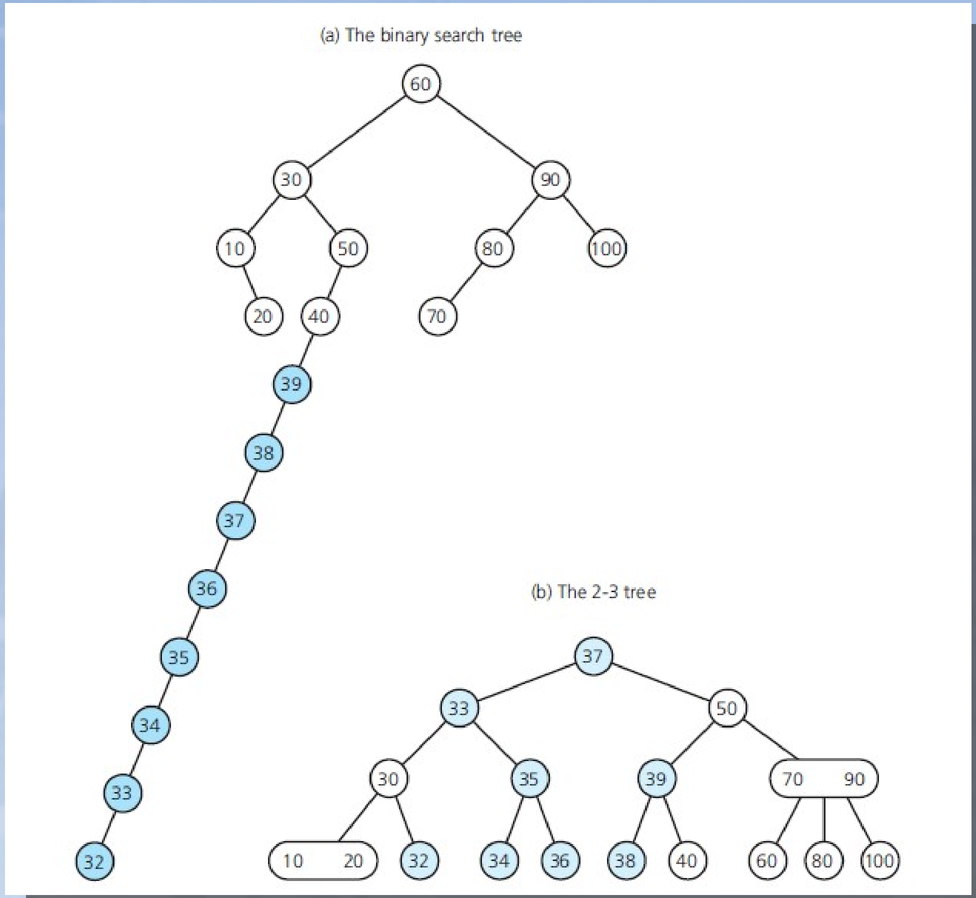


FIGURE 19-10 The trees of Figure 19-9 after adding the values 39 down to 32

Adding Data to a 2-3 Tree

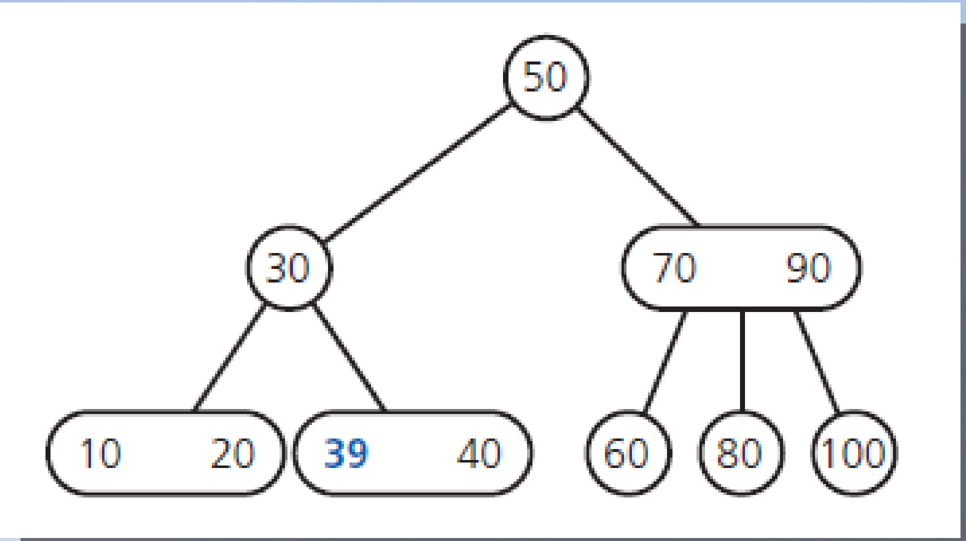


FIGURE 19-11 After inserting 39 into the tree in Figure 19-9b

Adding Data to a 2-3 Tree

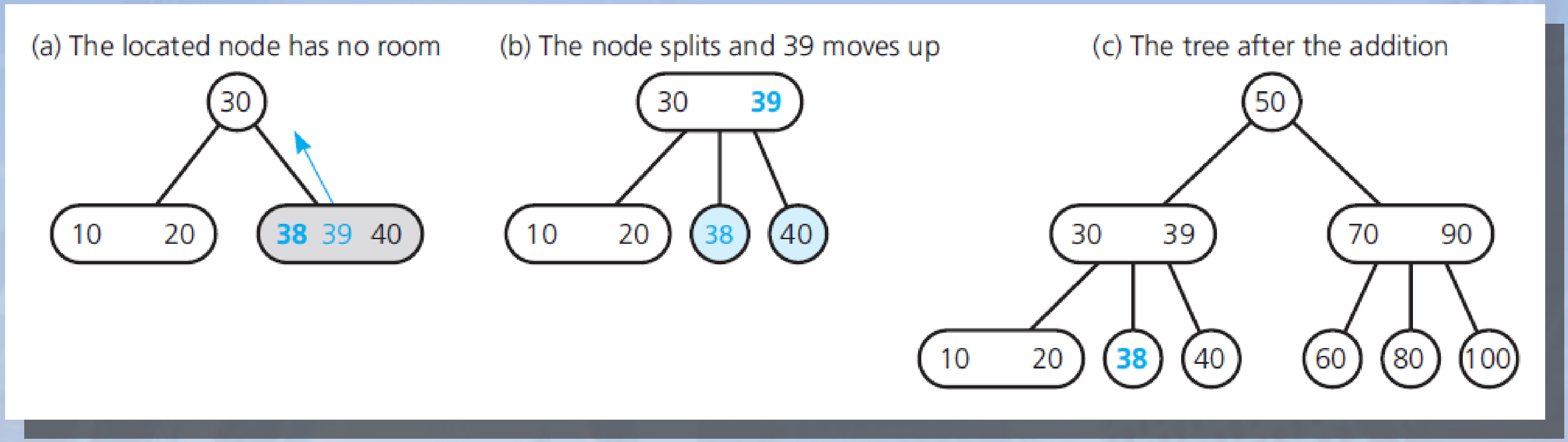


FIGURE 19-12 The steps for adding 38 to the tree in Figure 19-11

Adding Data to a 2-3 Tree

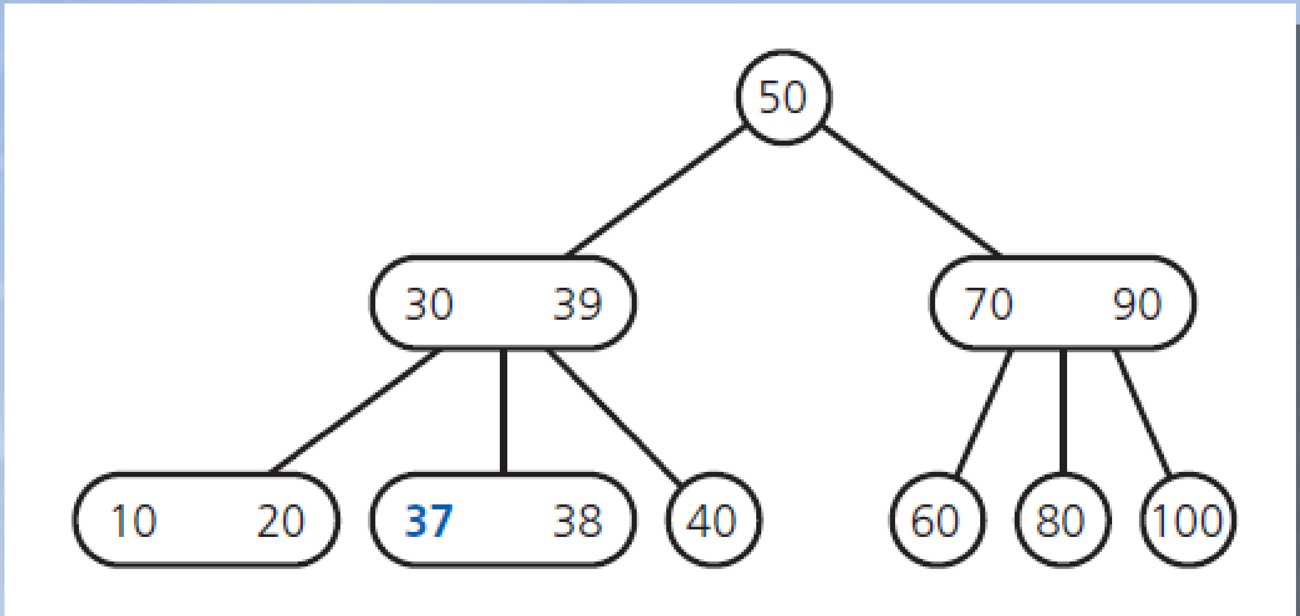


FIGURE 19-13 After adding 37 to the tree in Figure 19-12c

Adding Data to a 2-3 Tree

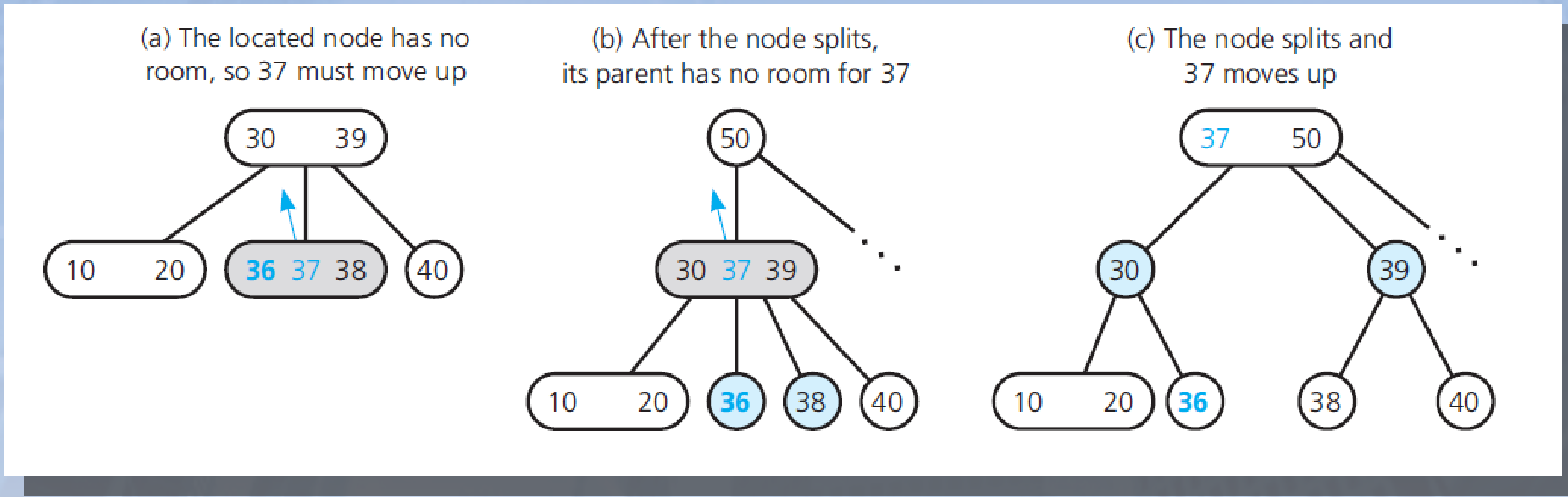


FIGURE 19-14 The steps for adding 36 to the tree in Figure 19-13

Adding Data to a 2-3 Tree

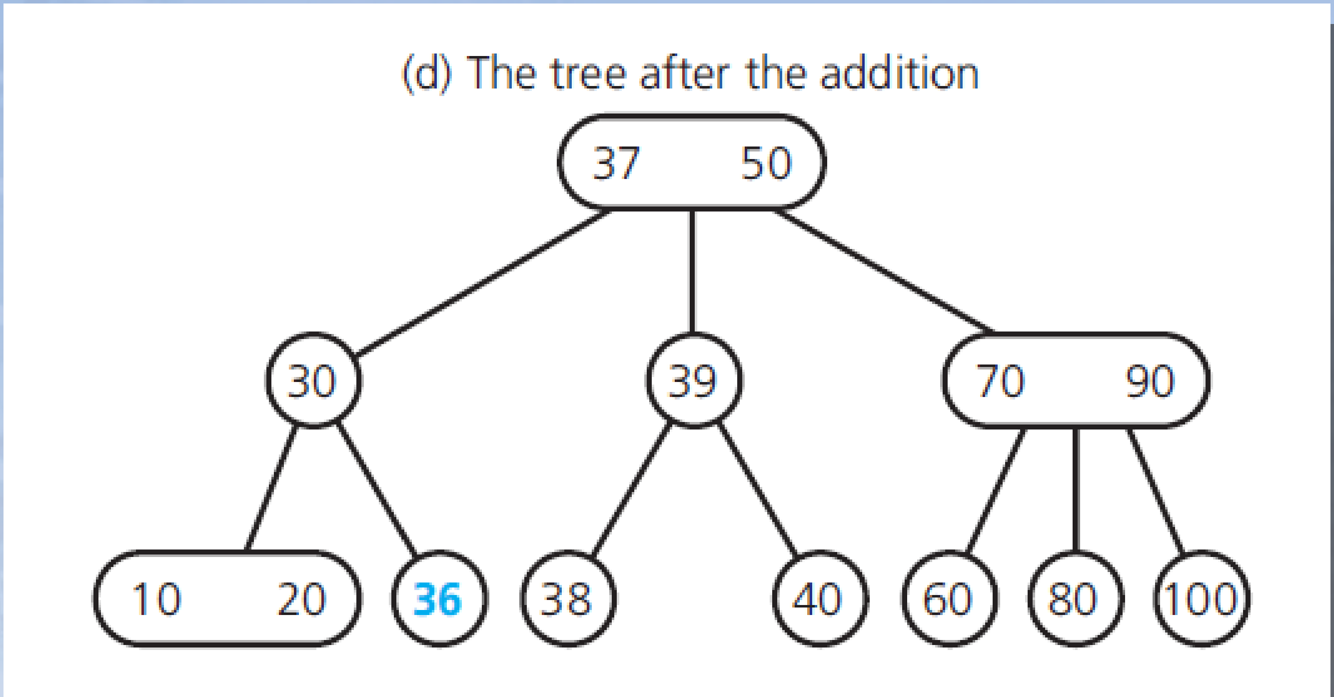


FIGURE 19-14 The steps for adding 36 to the tree in Figure 19-13

Adding Data to a 2-3 Tree

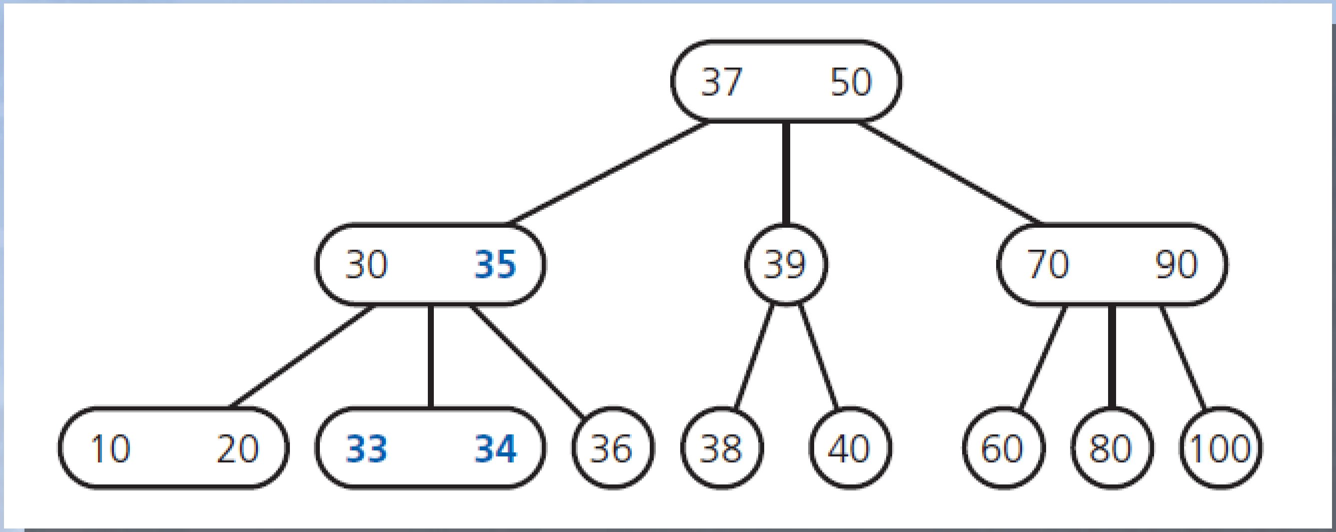


FIGURE 19-15 The tree after the adding 35, 34, and 33 to the tree in Figure 19-14d

Adding Data to a 2-3 Tree

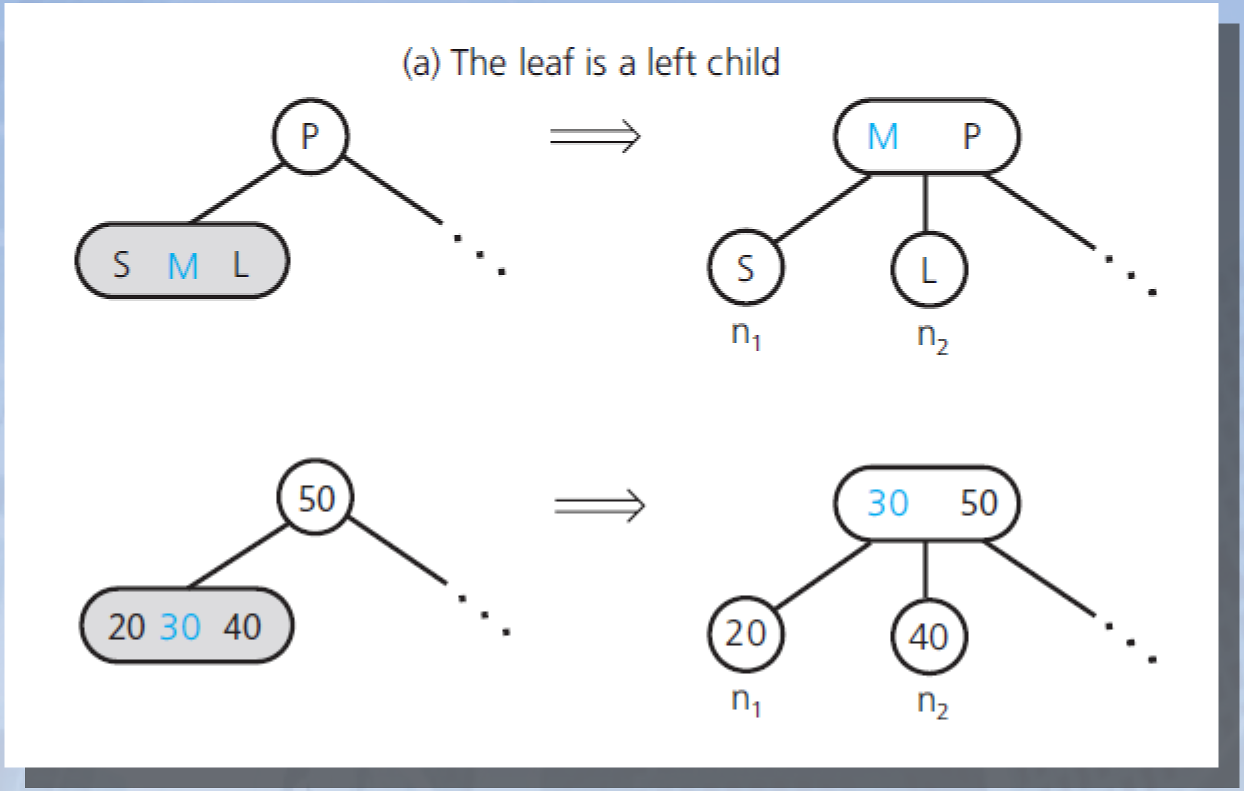


FIGURE 19-16 Splitting a leaf in a 2-3 tree in general and in a specific example

Adding Data to a 2-3 Tree

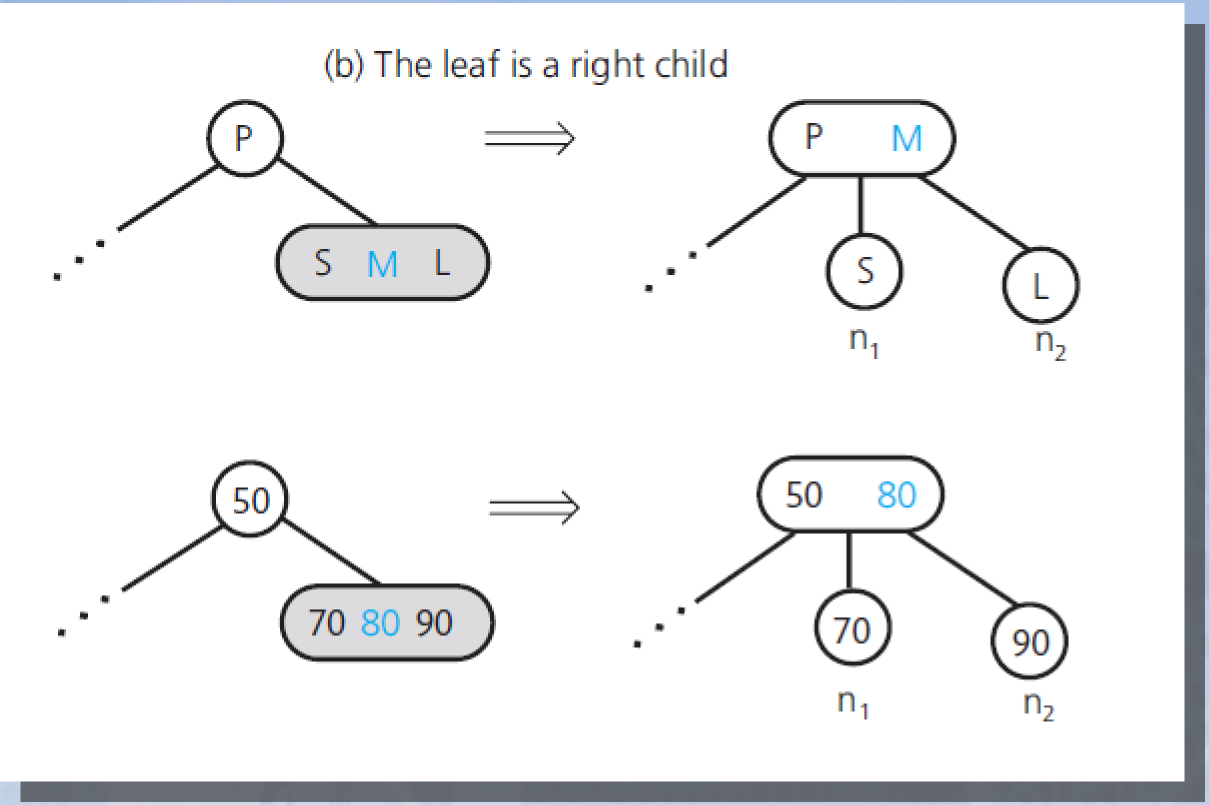


FIGURE 19-16 Splitting a leaf in a 2-3 tree in general and in a specific example

Adding Data to a 2-3 Tree

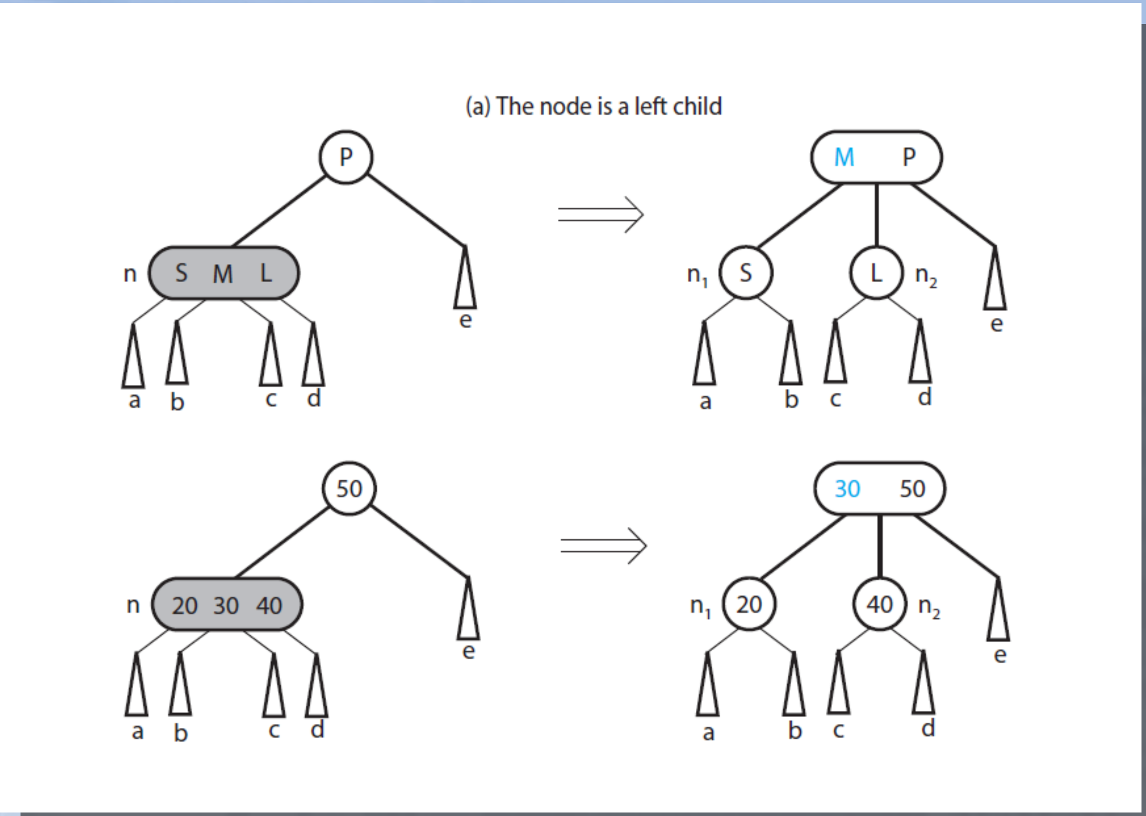


FIGURE 19-17 Splitting an internal node in a 2-3 tree in general and in a specific example

Adding Data to a 2-3 Tree

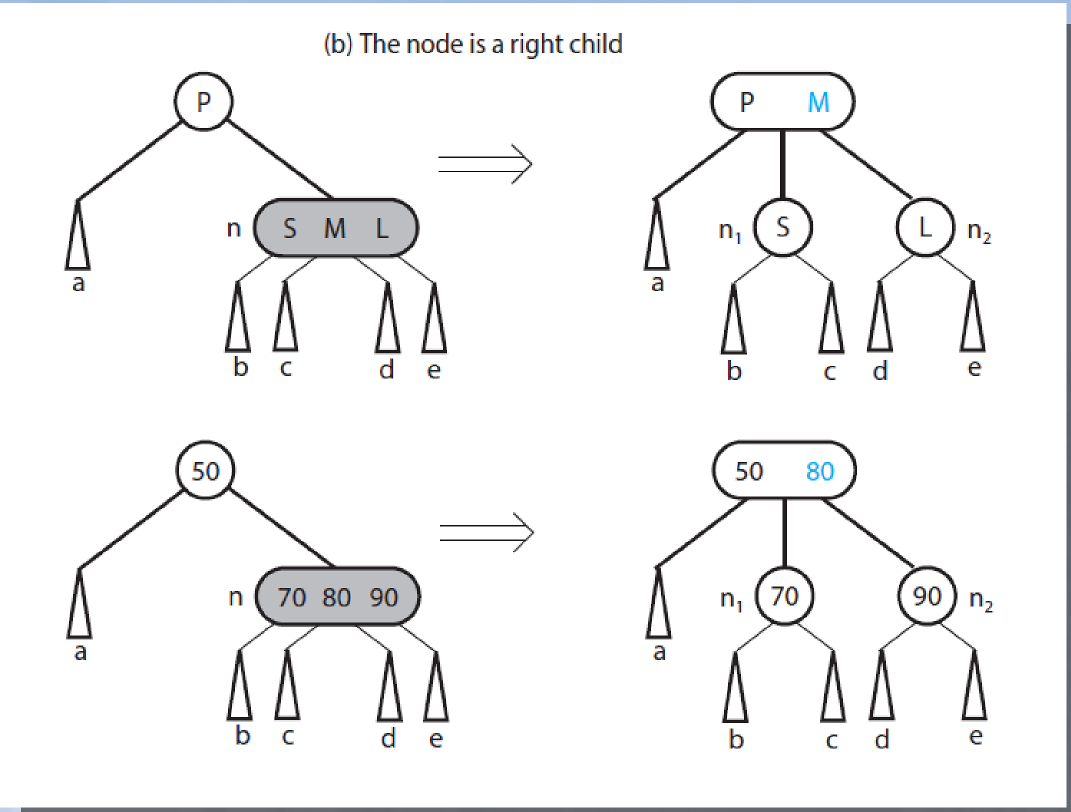


FIGURE 19-17 Splitting an internal node in a 2-3 tree in general and in a specific example

Adding Data to a 2-3 Tree

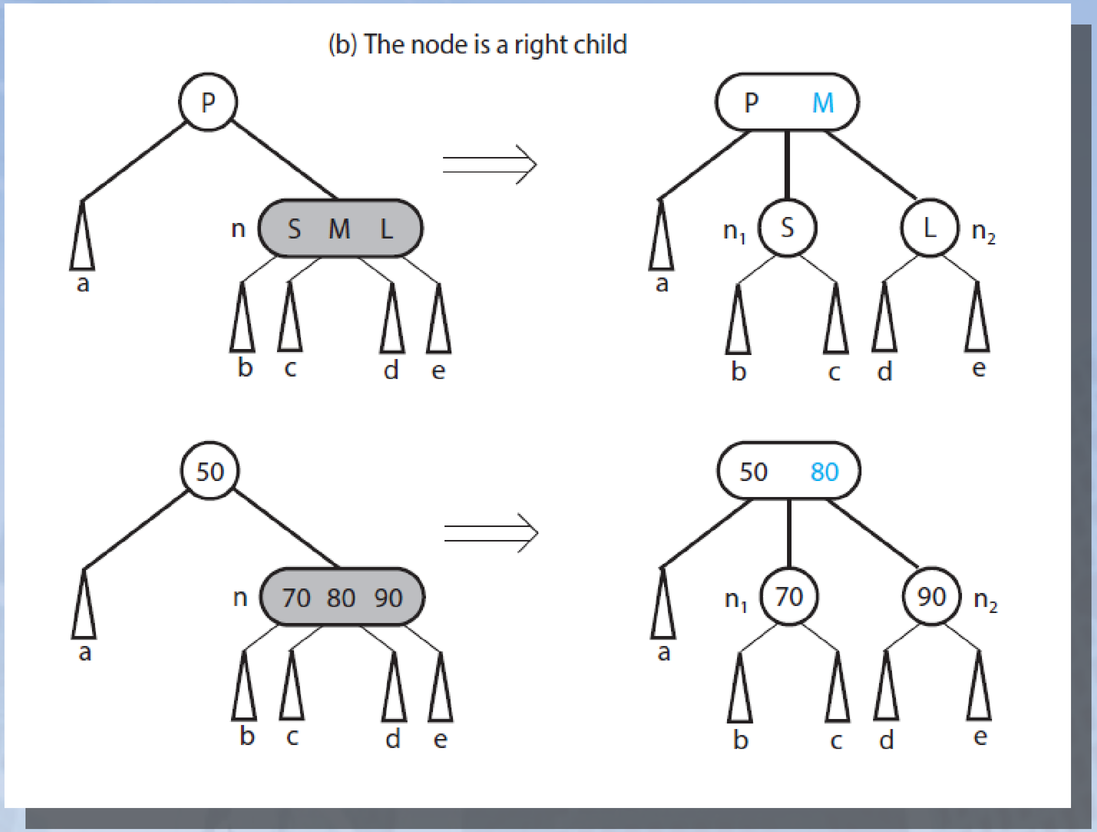


FIGURE 19-17 Splitting an internal node in a 2-3 tree in general and in a specific example

Adding Data to a 2-3 Tree

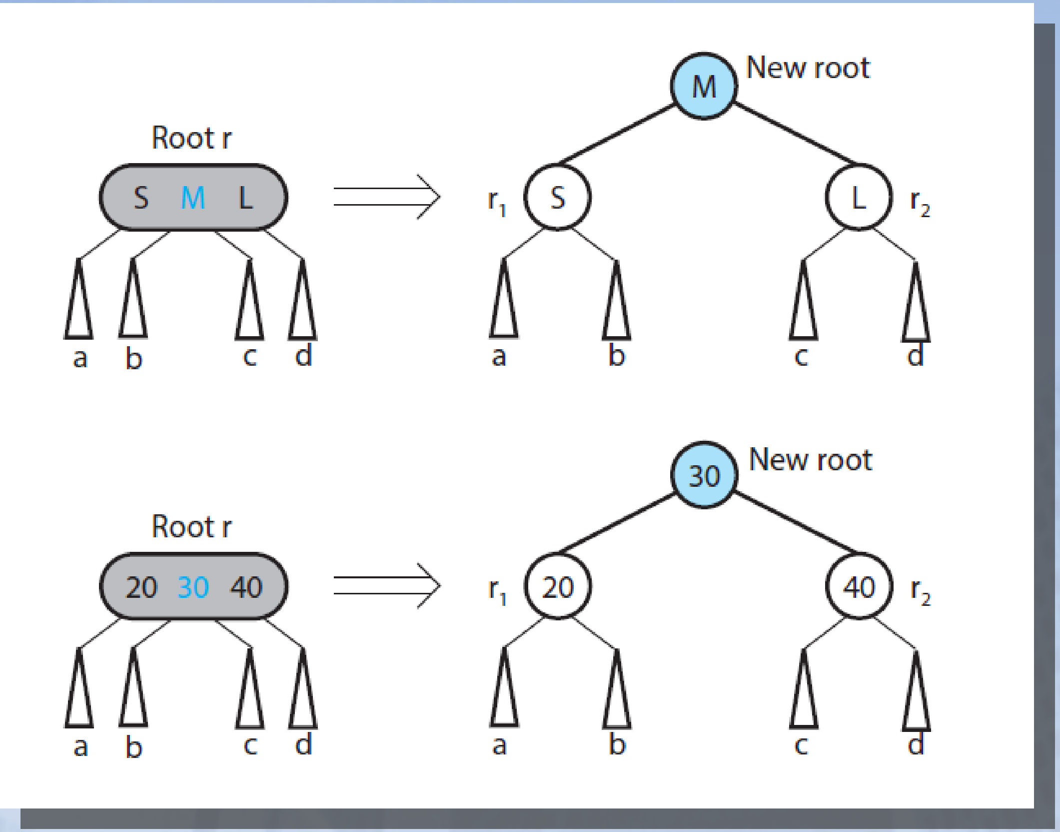


FIGURE 19-18 Splitting the root of a 2-3 tree general and in a specific example

Removing Data from a 2-3 Tree

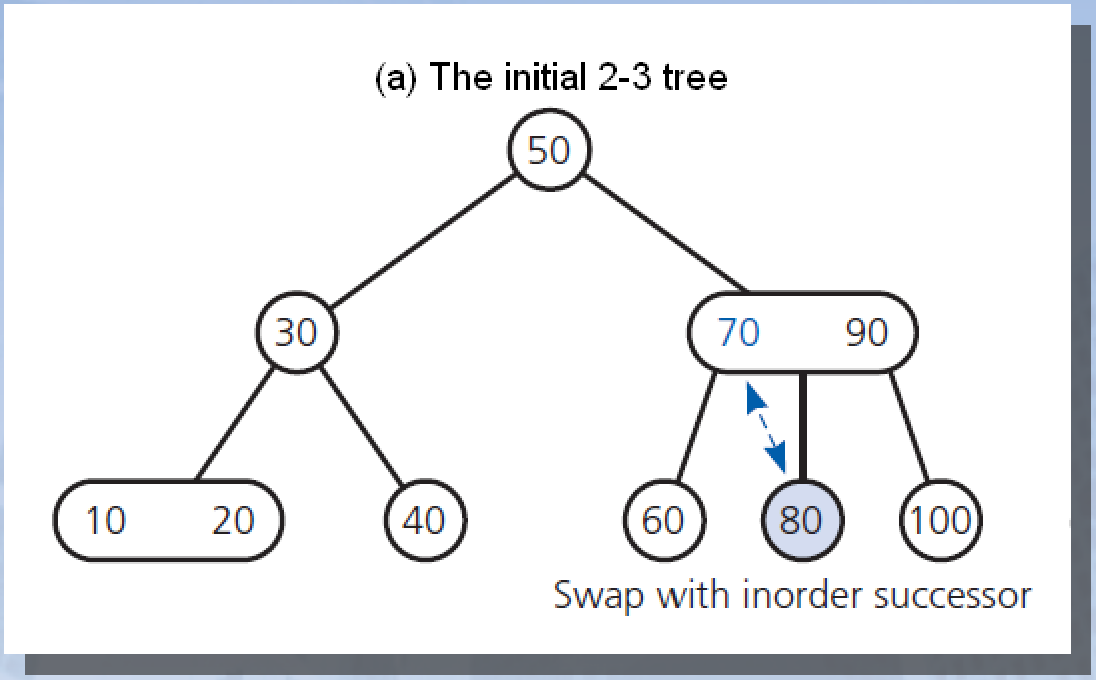


FIGURE 19-19 The steps for removing 70 from the 2-3 tree in Figure 19-9b

Removing Data from a 2-3 Tree

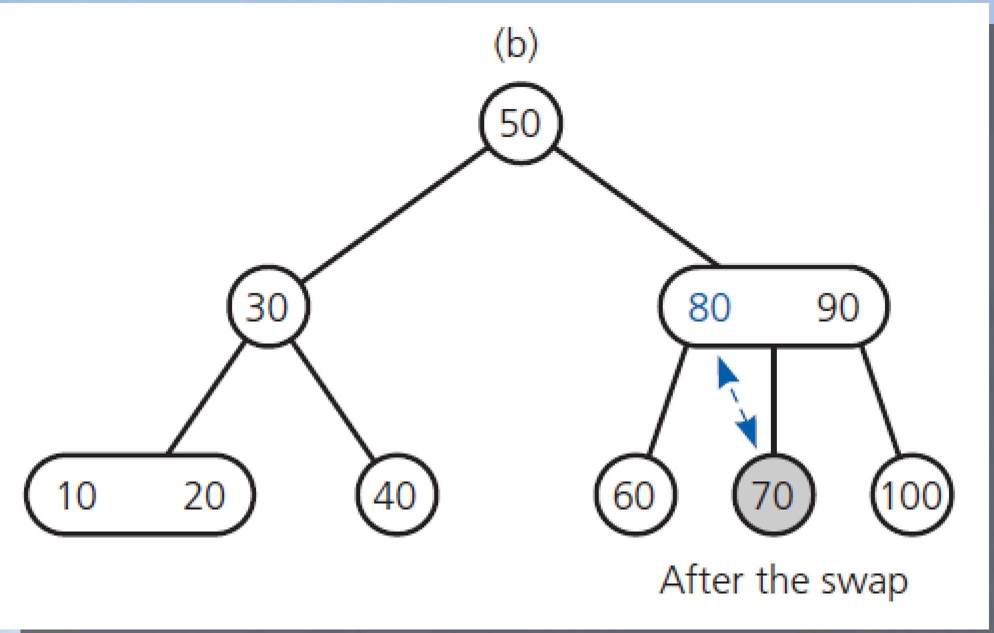


FIGURE 19-19 The steps for removing 70 from the 2-3 tree in Figure 19-9b

Removing Data from a 2-3 Tree

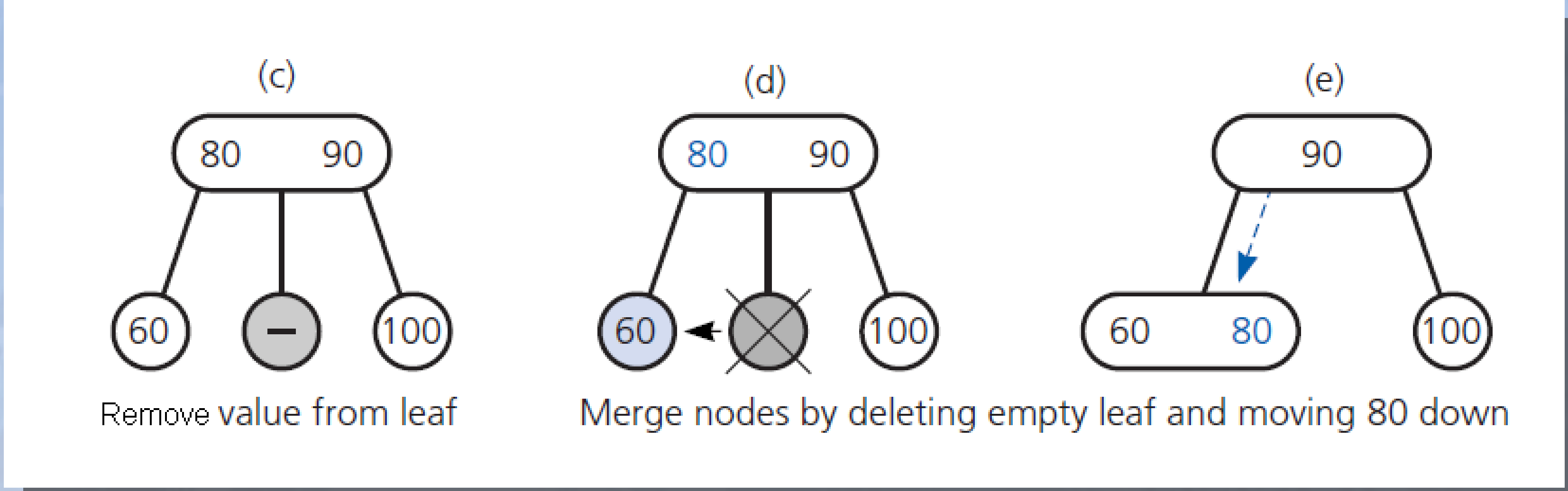


FIGURE 19-19 The steps for removing 70 from the 2-3 tree in Figure 19-9b

Removing Data from a 2-3 Tree

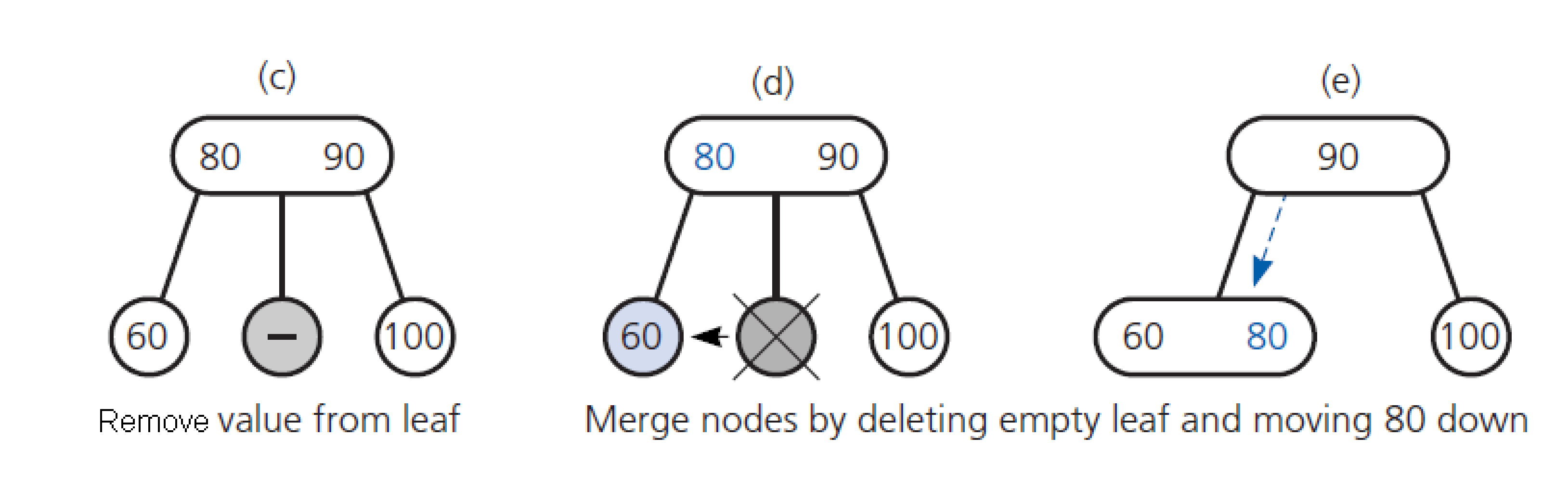


FIGURE 19-19 The steps for removing 70 from the 2-3 tree in Figure 19-9b

Removing Data from a 2-3 Tree

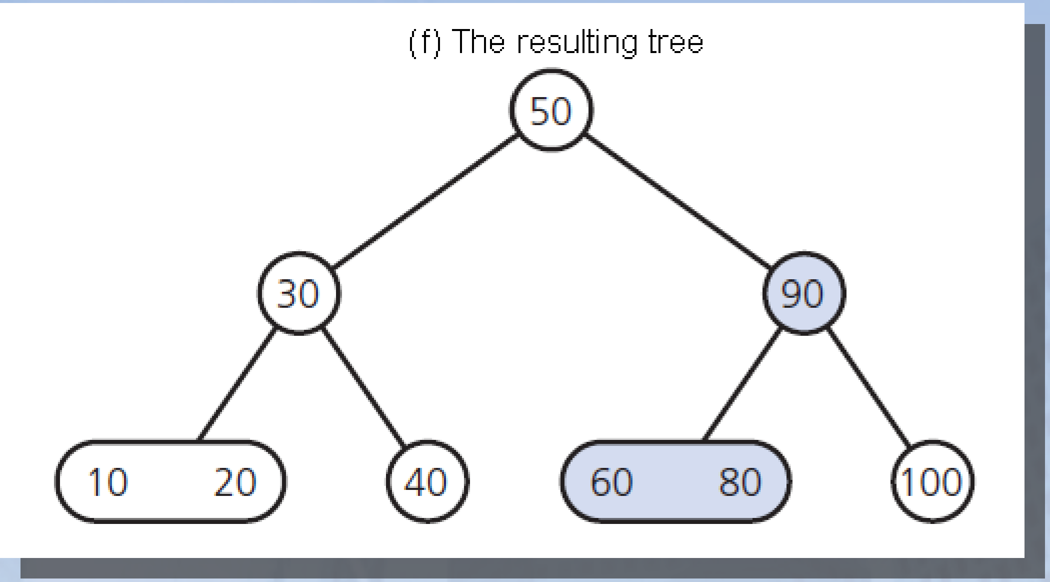


FIGURE 19-19 The steps for removing 70 from the 2-3 tree in Figure 19-9b

Removing Data from a 2-3 Tree

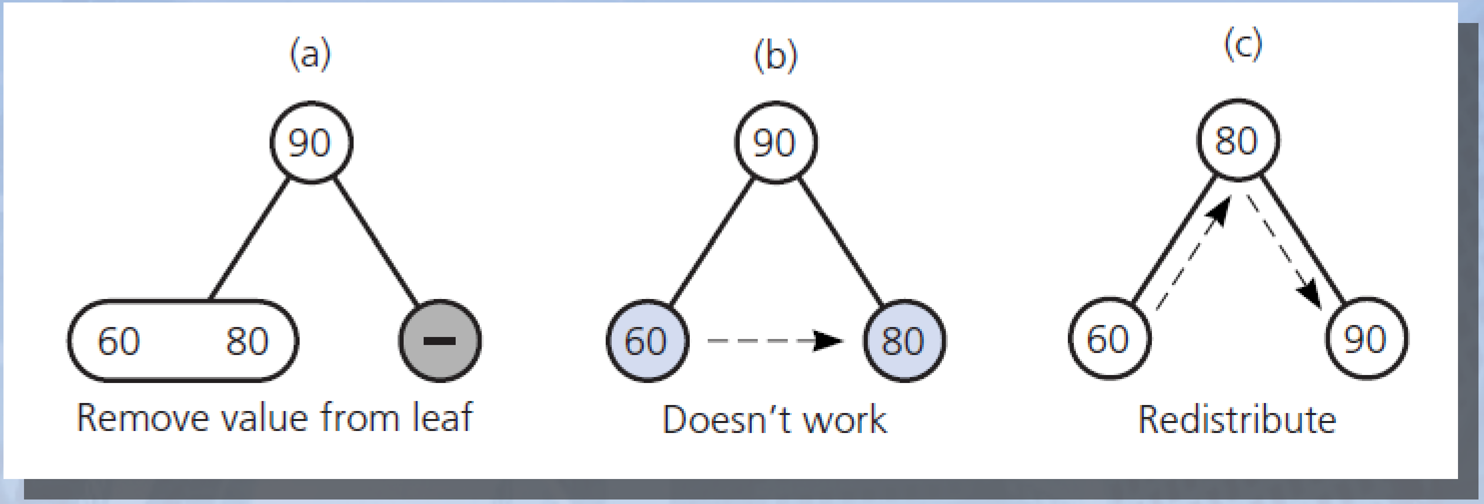


FIGURE 19-20 The steps for removing 100 from the tree in Figure 19-19f;

Removing Data from a 2-3 Tree

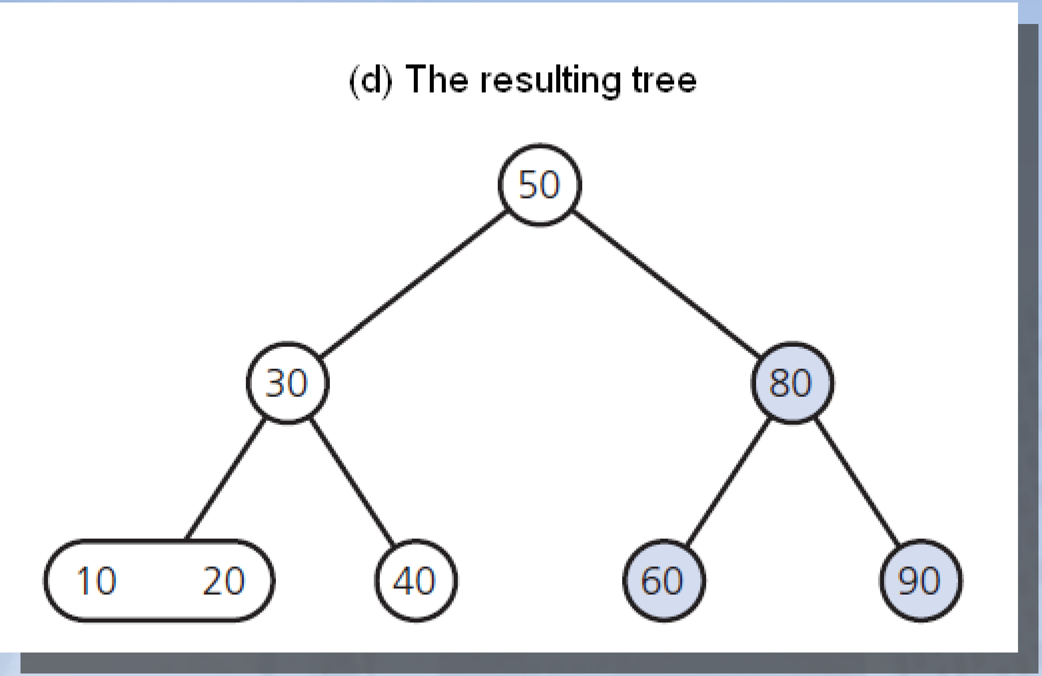


FIGURE 19-20 The steps for removing 100 from the tree in Figure 19-19f;

Removing Data from a 2-3 Tree

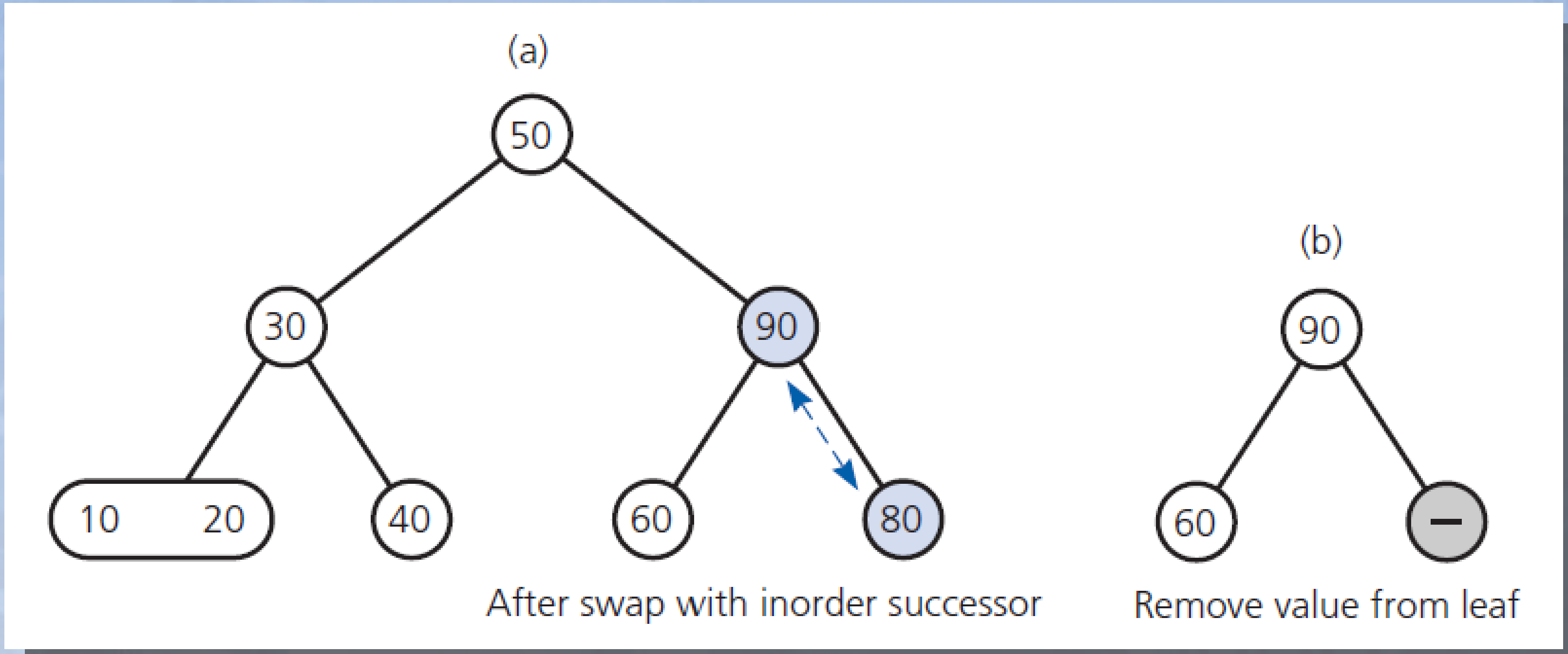


FIGURE 19-21 The steps for removing 80 from the 2-3 tree in Figure 19-20d

Removing Data from a 2-3 Tree

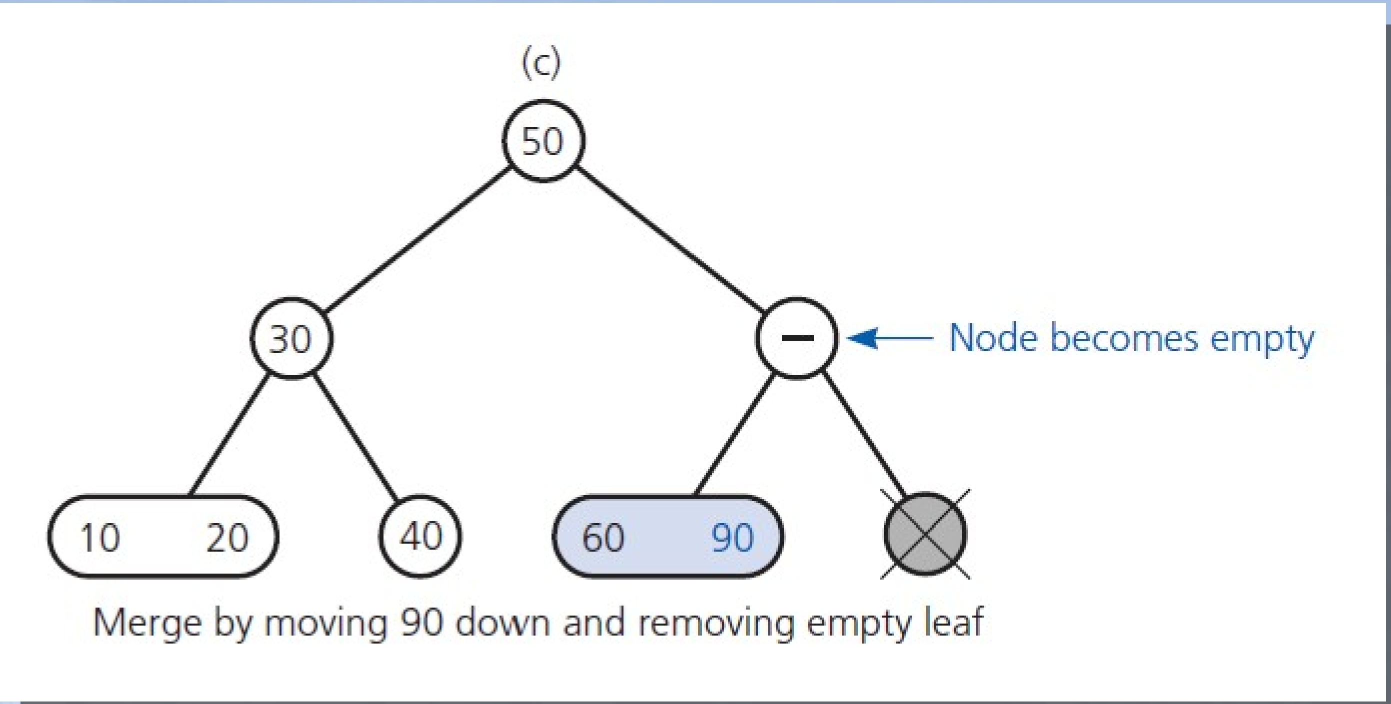


FIGURE 19-21 The steps for removing 80 from the 2-3 tree in Figure 19-20d

Removing Data from a 2-3 Tree

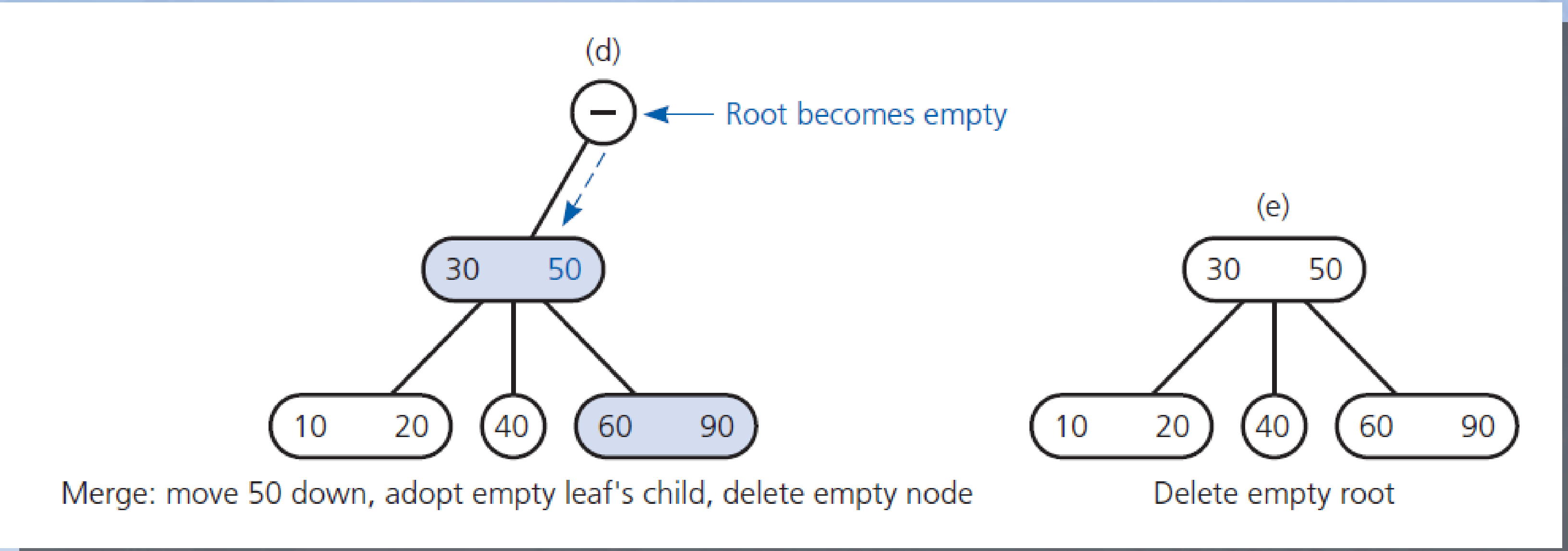


FIGURE 19-21 The steps for removing 80 from the 2-3 tree in Figure 19-20d

Removing Data from a 2-3 Tree

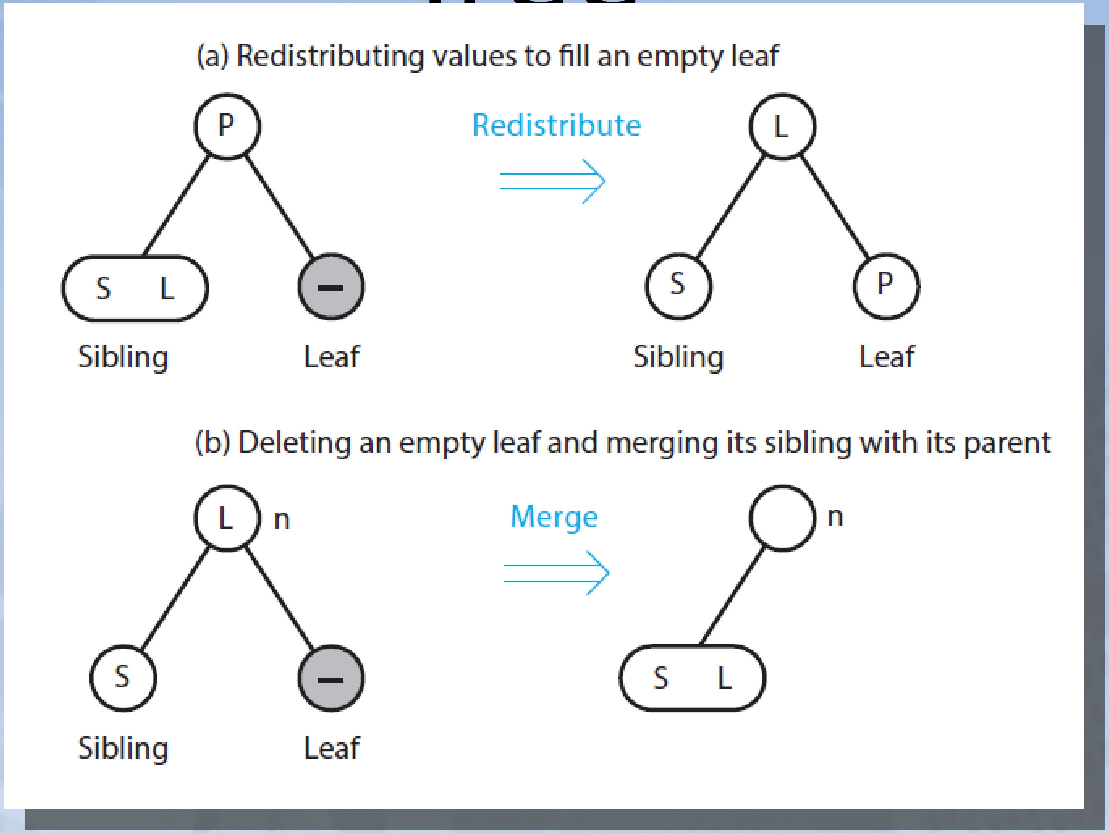


FIGURE 19-23 Possible situations during the removal of a data item from a 2-3 tree

Removing Data from a 2-3 Tree

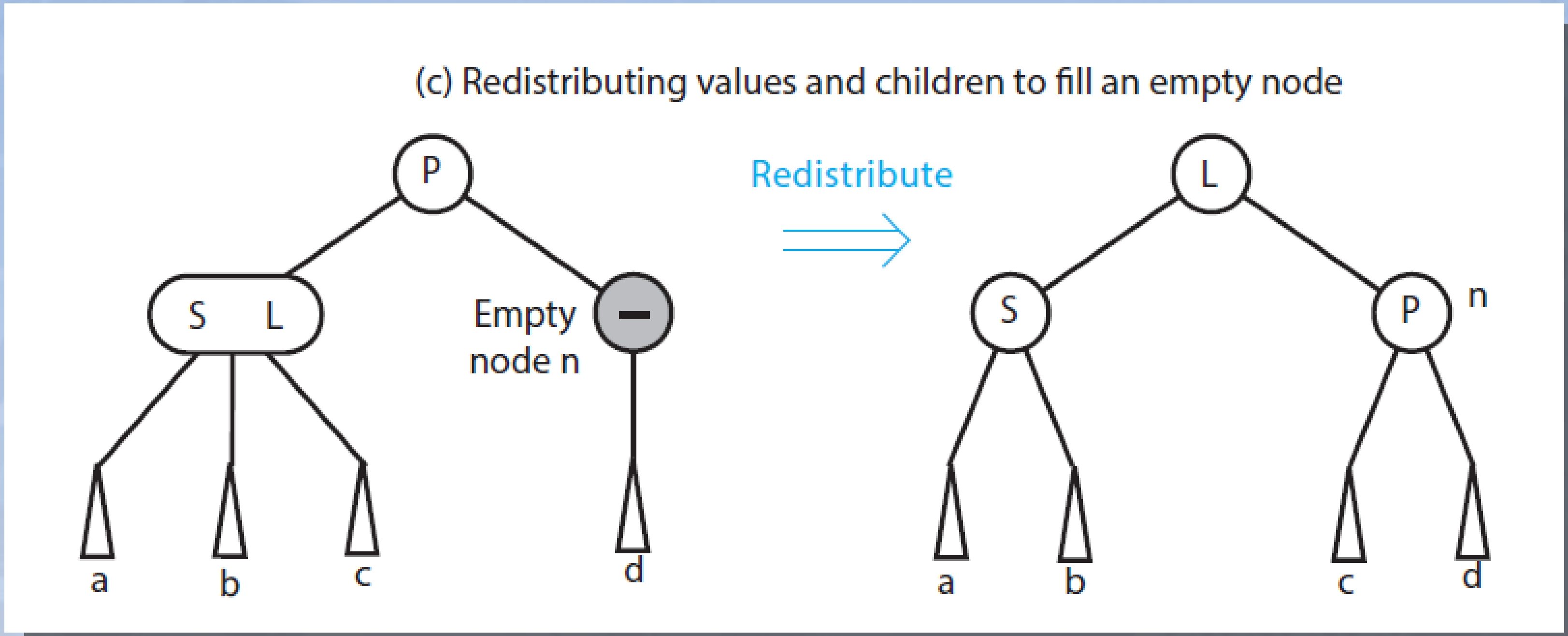


FIGURE 19-23 Possible situations during the removal of a data item from a 2-3 tree

Removing Data from a 2-3 Tree

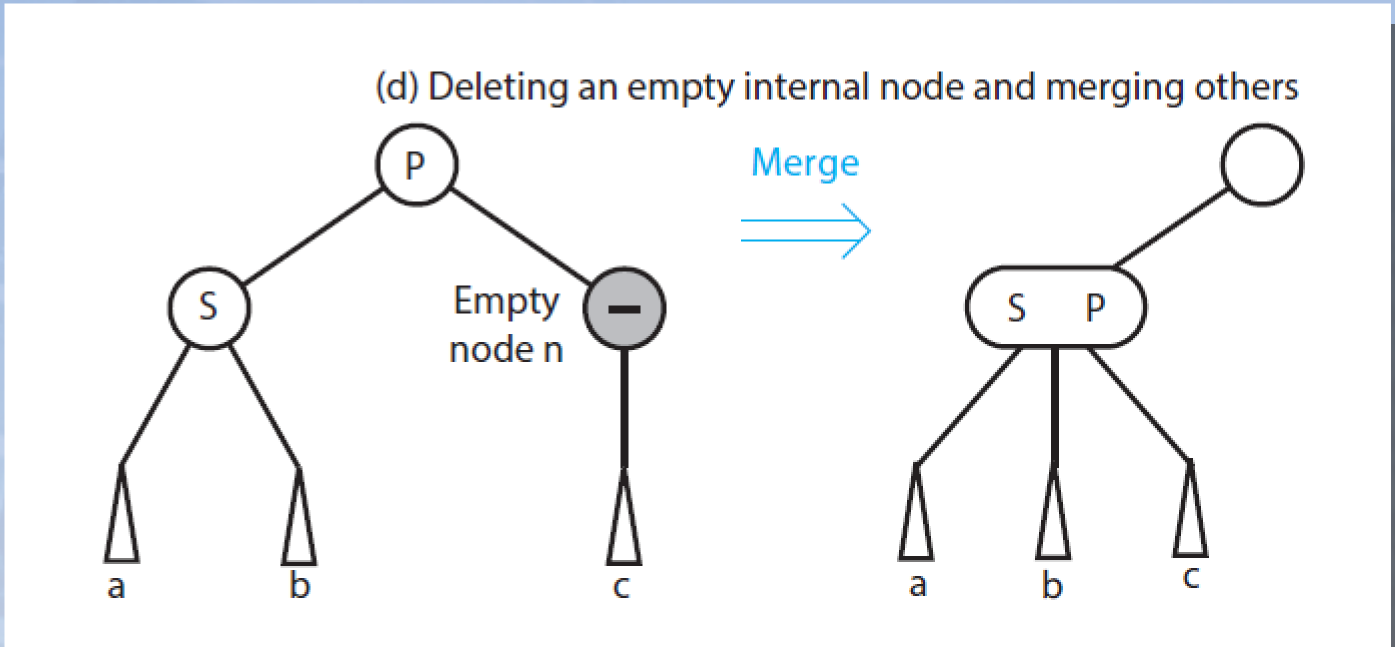


FIGURE 19-23 Possible situations during the removal of a data item from a 2-3 tree

Removing Data from a 2-3 Tree

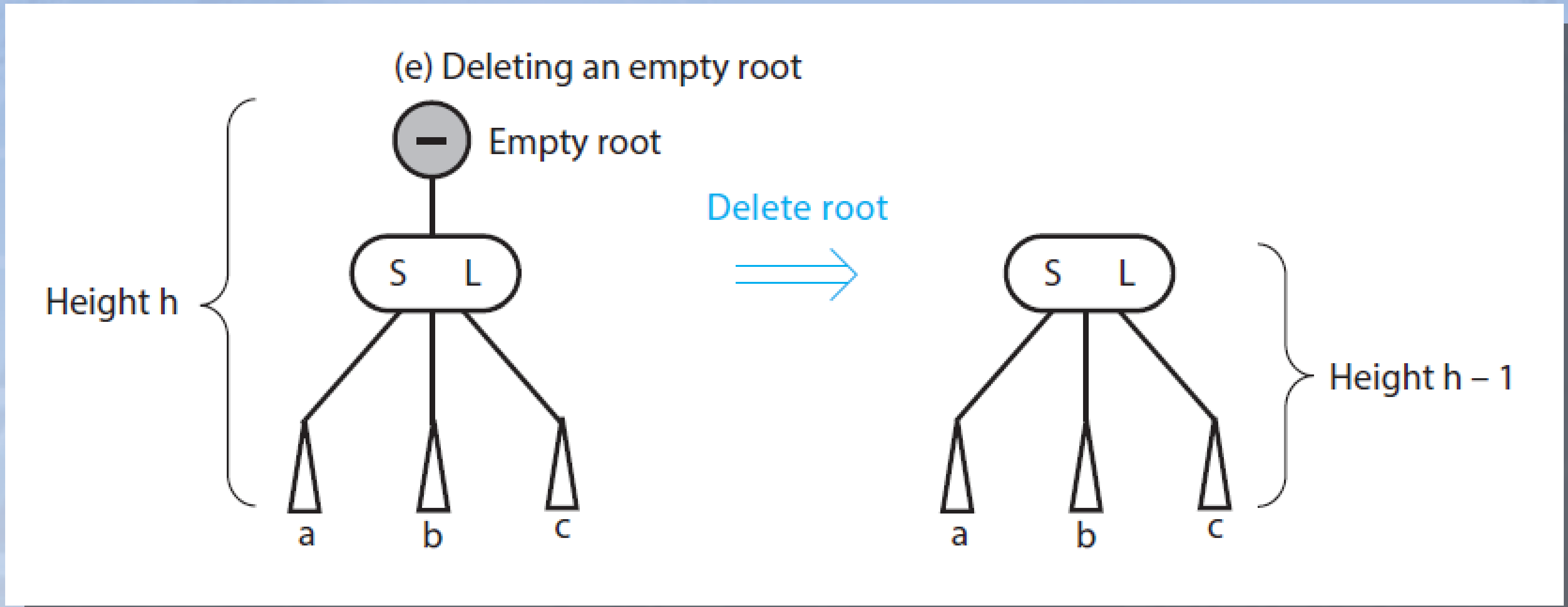


FIGURE 19-23 Possible situations during the removal of a data item from a 2-3 tree

2-3-4 Trees

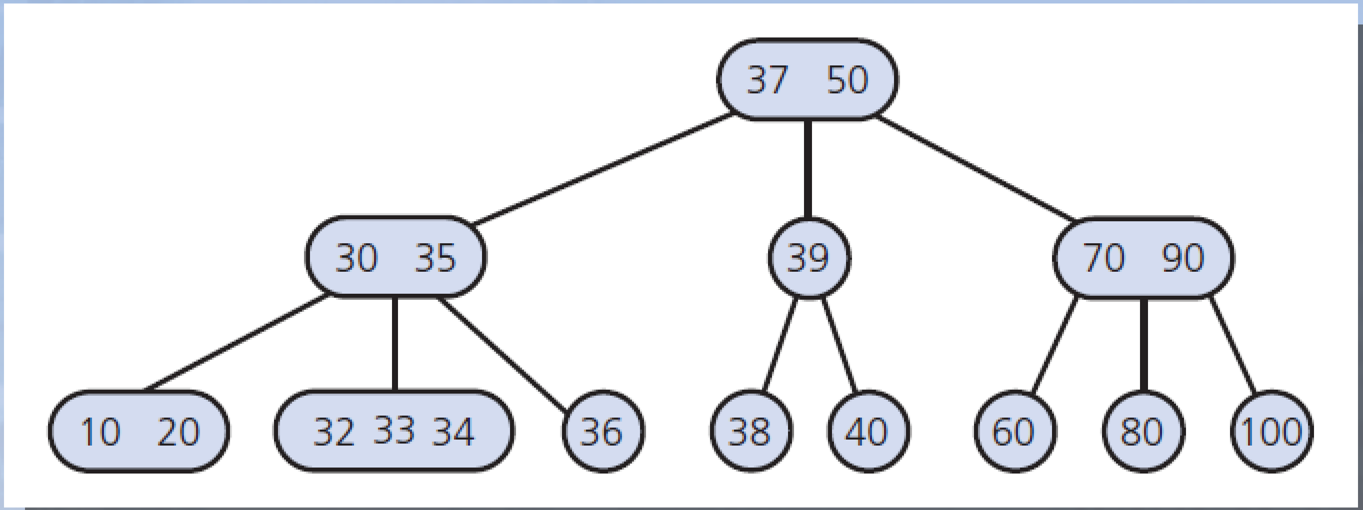


FIGURE 19-24 A 2-3-4 tree with the same data items as the 2-3 tree in Figure 19-10b

2-3-4 Trees

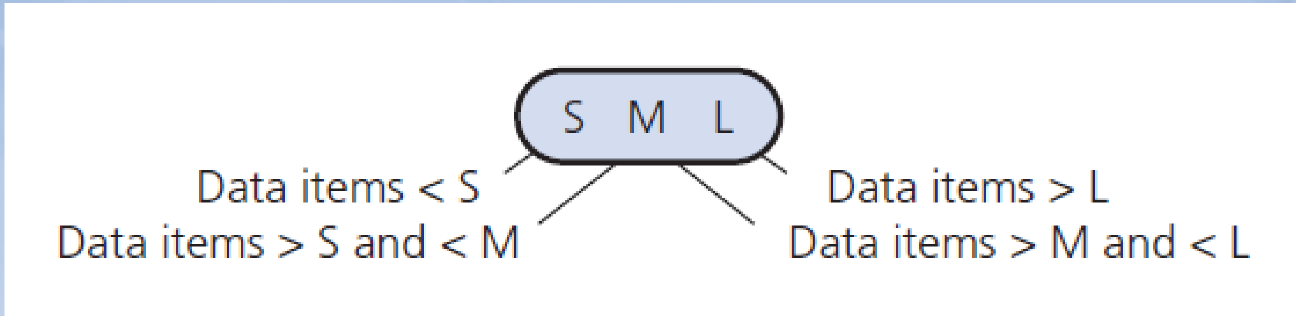


FIGURE 19-25 A 4-node in a 2-3-4 tree

2-3-4 Trees

- Searching and traversing
  - Simple extensions of corresponding algorithms for a 2-3 tree
- Adding data
  - Like addition algorithm for 2-3 tree
  - Splits node by moving one data item up to parent node

Adding Data to 2-3-4 Trees

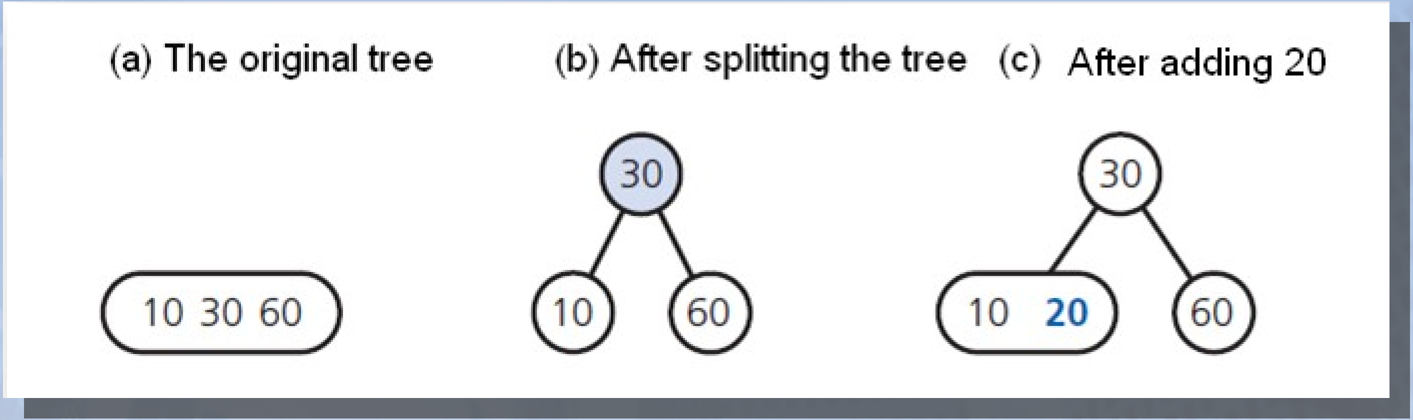


FIGURE 19-26 Adding 20 to a one-node 2-3-4 tree

Adding Data to 2-3-4 Trees

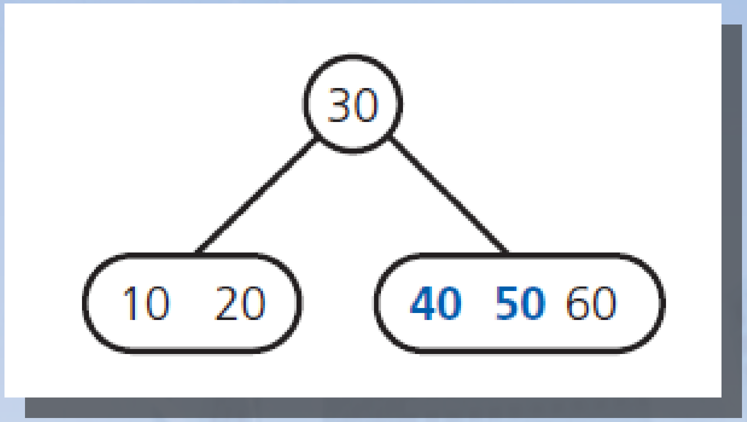


FIGURE 19-27 After adding 50 and 40 to the tree in Figure 19-26c

Adding Data to 2-3-4 Trees

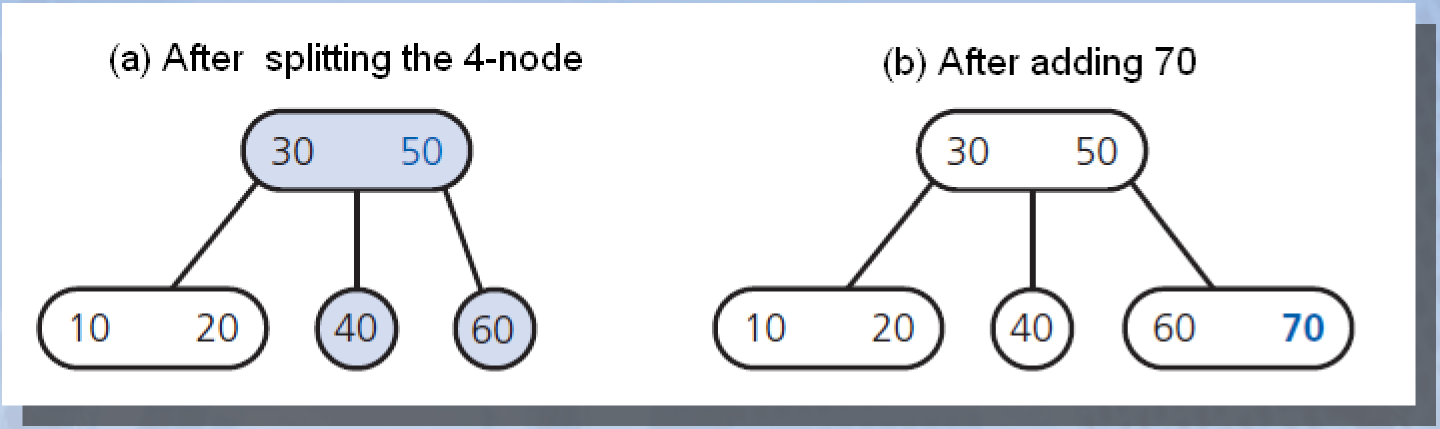


FIGURE 19-28 The steps for adding 70 to the tree in Figure 19-27

Adding Data to 2-3-4 Trees

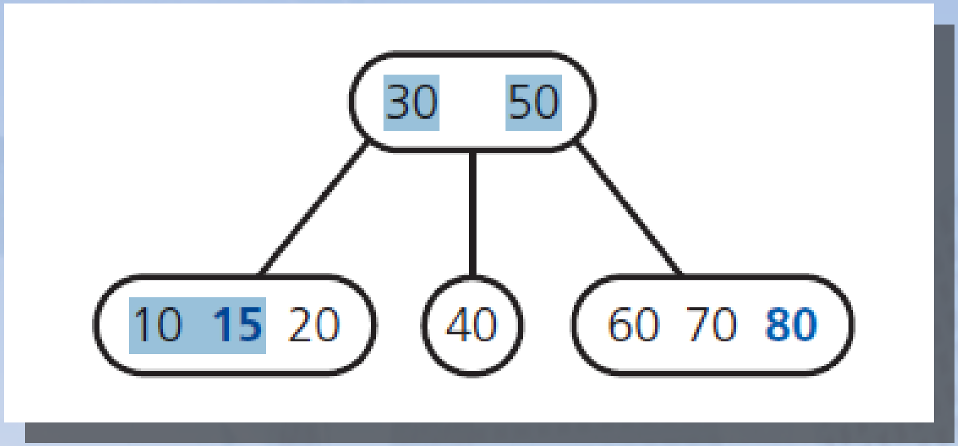


FIGURE 19-29 After adding 80 and 15 to the tree in Figure 19-28b

Adding Data to 2-3-4 Trees

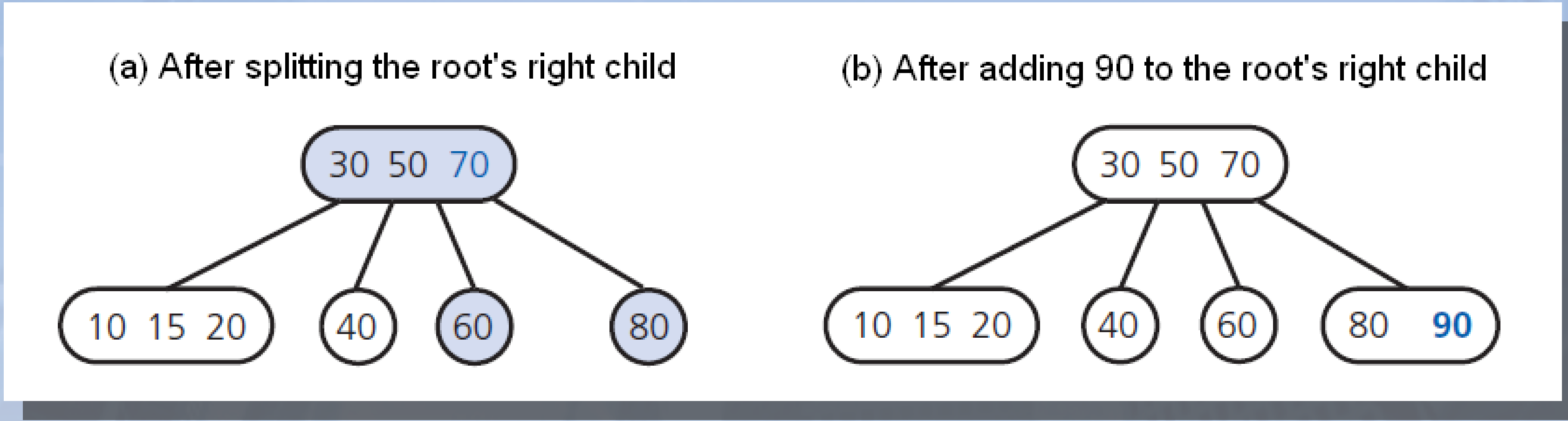


FIGURE 19-30 The steps for adding 90 to the tree in Figure 19-29

Adding Data to 2-3-4 Trees

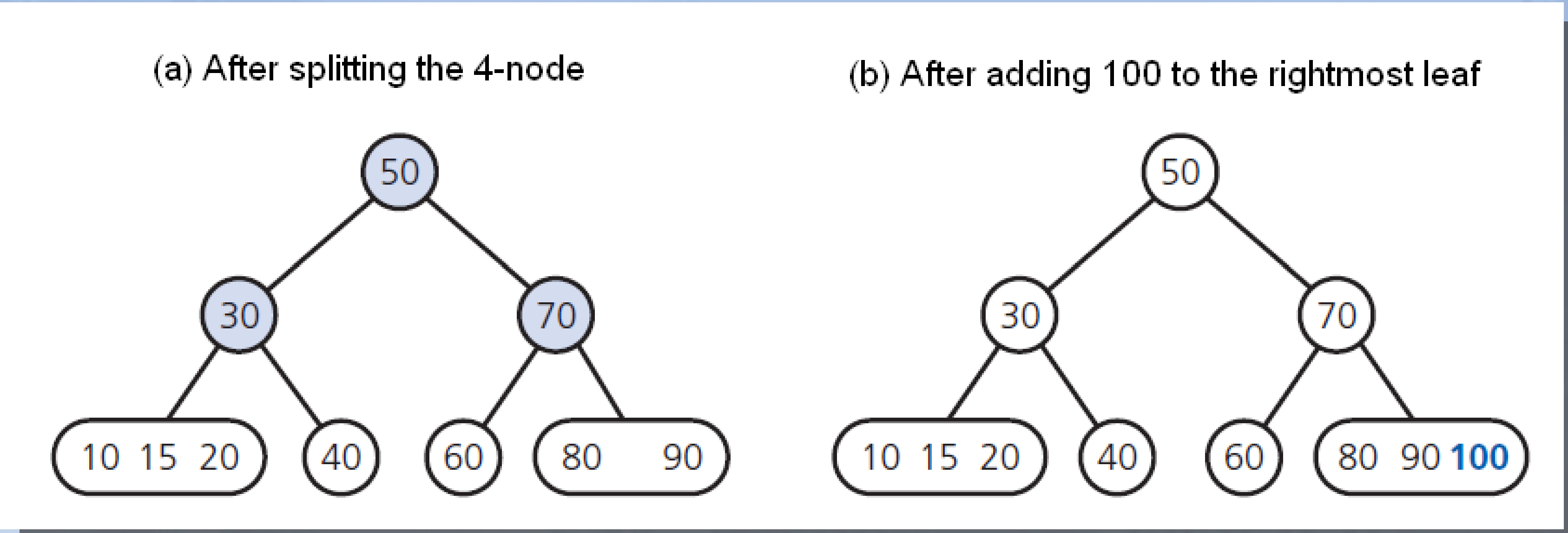


FIGURE 19-31 The steps for adding 100 to the tree in Figure 30b

Adding Data to 2-3-4 Trees

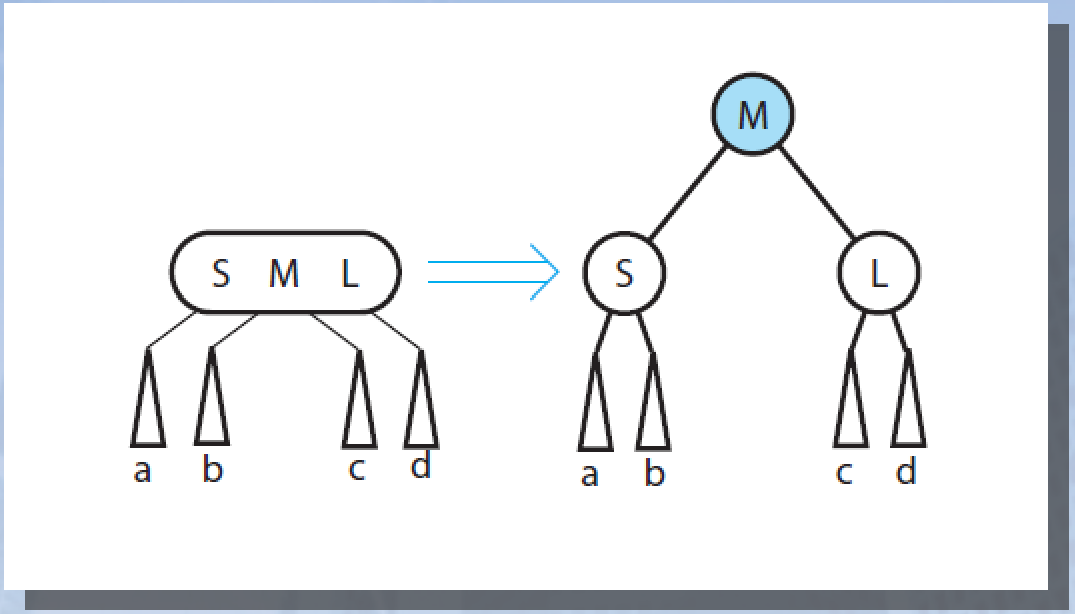


FIGURE 19-32 Splitting a 4-node root when adding data to a 2-3-4 tree

Adding Data to 2-3-4 Trees

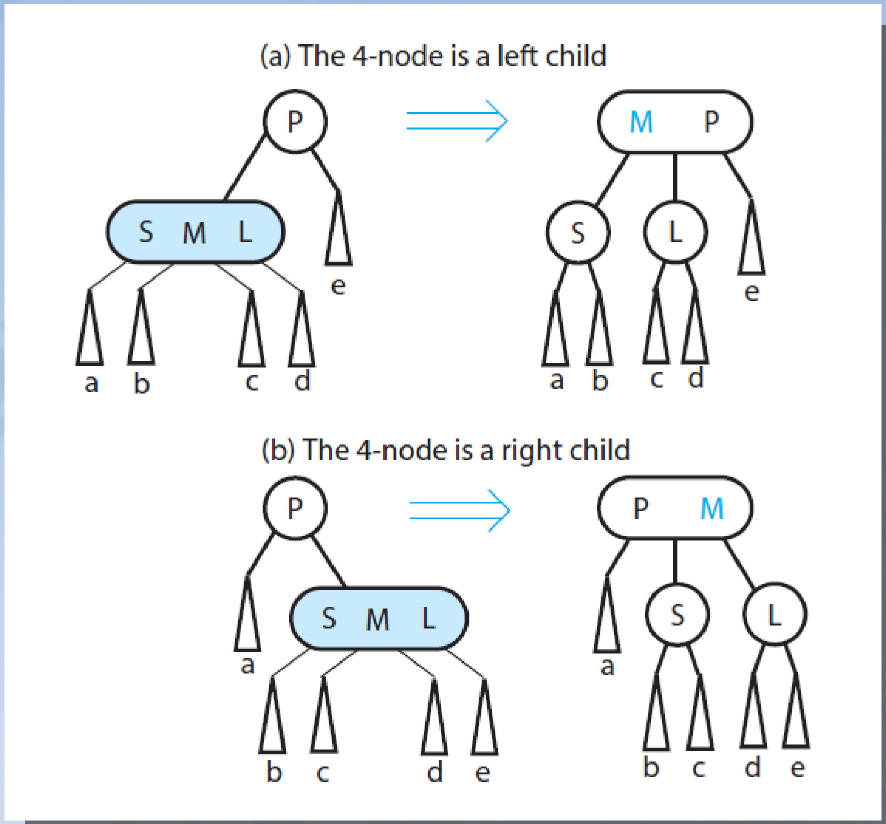


FIGURE 19-33 Splitting a 4-node whose parent is a 2-node when adding data to a 2-3-4 tree

Adding Data to 2-3-4 Trees

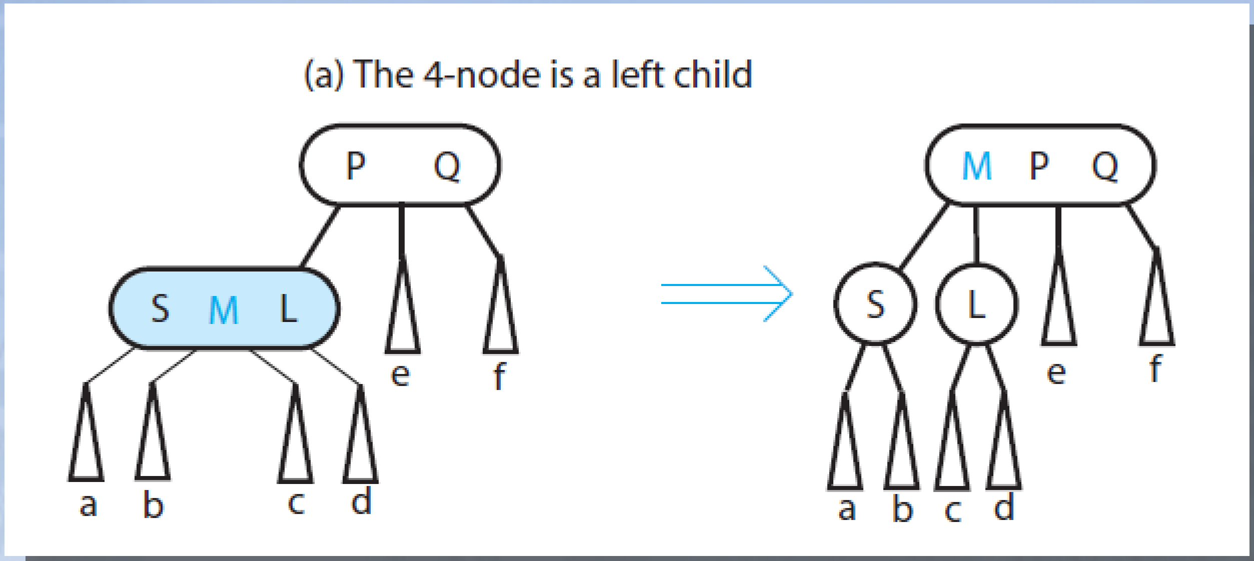


FIGURE 19-34 Splitting a 4-node whose parent is a 3-node when adding data to a 2-3-4 tree

Adding Data to 2-3-4 Trees

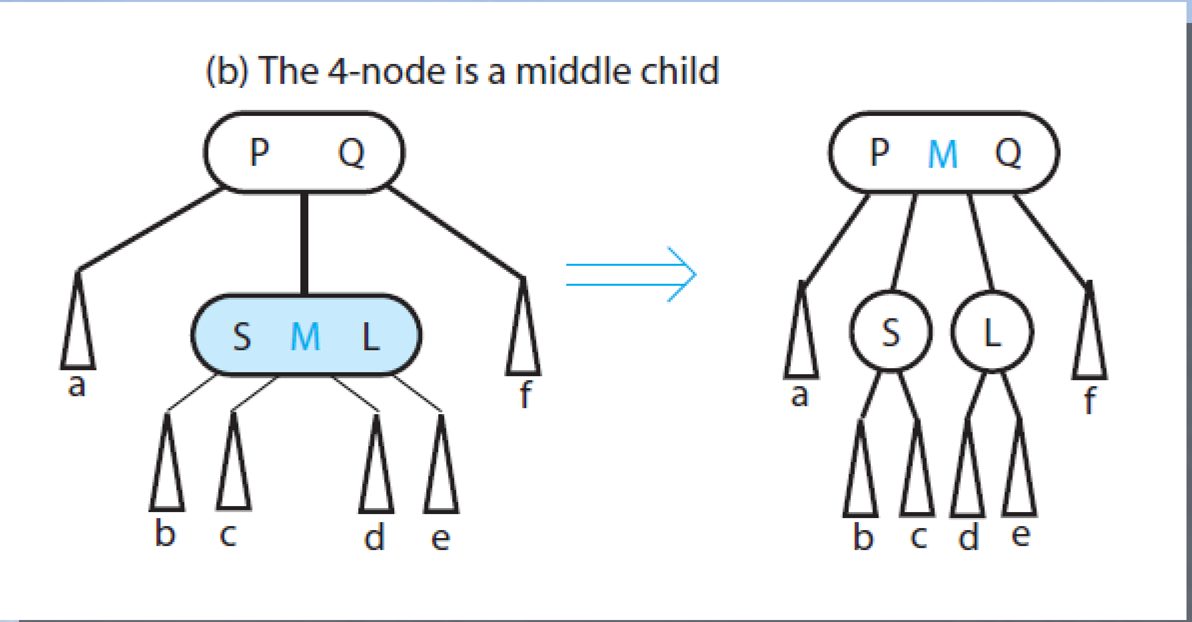


FIGURE 19-34 Splitting a 4-node whose parent is a 3-node when adding data to a 2-3-4 tree

Adding Data to 2-3-4 Trees

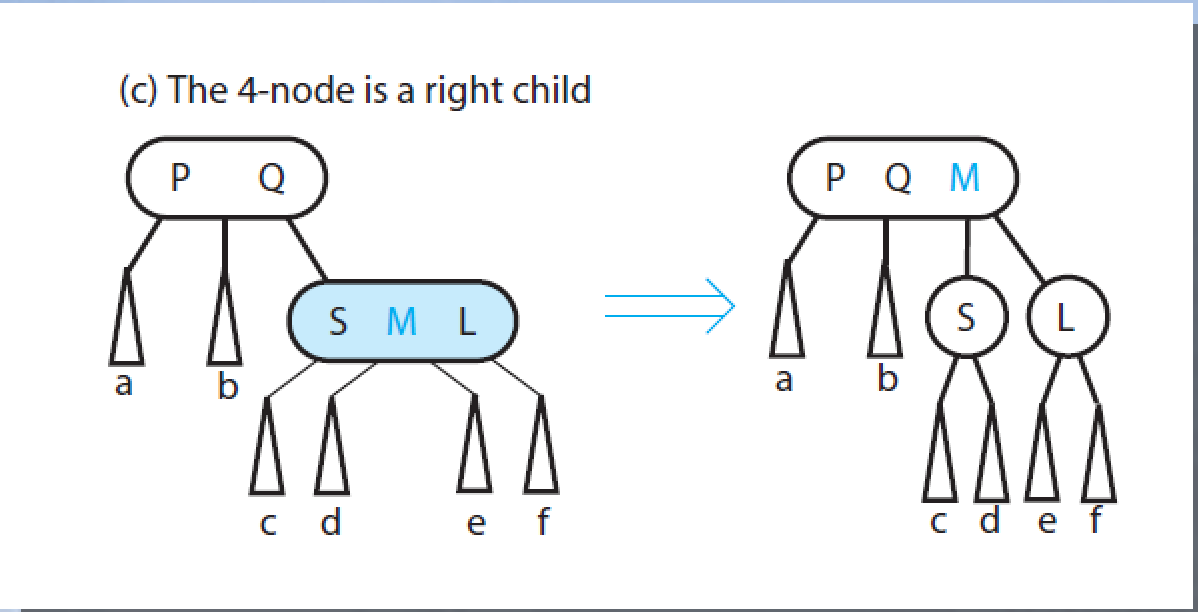


FIGURE 19-34 Splitting a 4-node whose parent is a 3-node when adding data to a 2-3-4 tree

Removing Data from a 2-3-4 Tree

- Has same beginning as removal algorithm for a 2-3 tree
- Transform each 2-node into a 3-node or a 4-node
- Insertion and removal algorithms for 2-3-4 tree require fewer steps than for 2-3 tree

Red-Black Trees

- A 2-3-4 tree requires more storage than binary search tree
- Red-black tree has advantages of a 2-3-4 tree but requires less storage
- In a red-black tree,
  - Red pointers link 2-nodes that now contain values that were in a 3-node or a 4-node.

Red-Black Trees

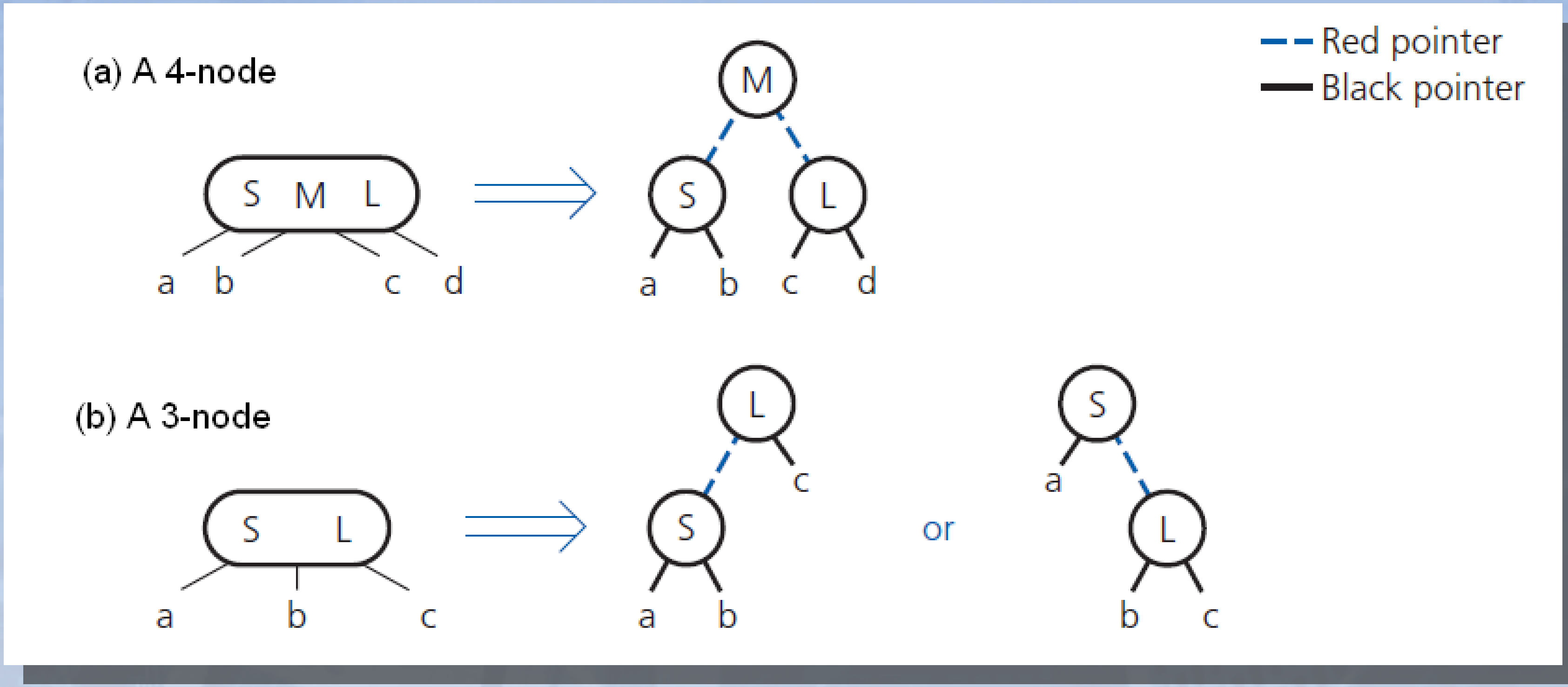


FIGURE 19-35 Red-black representation s of a 4-node and a 3-node

Red-Black Trees

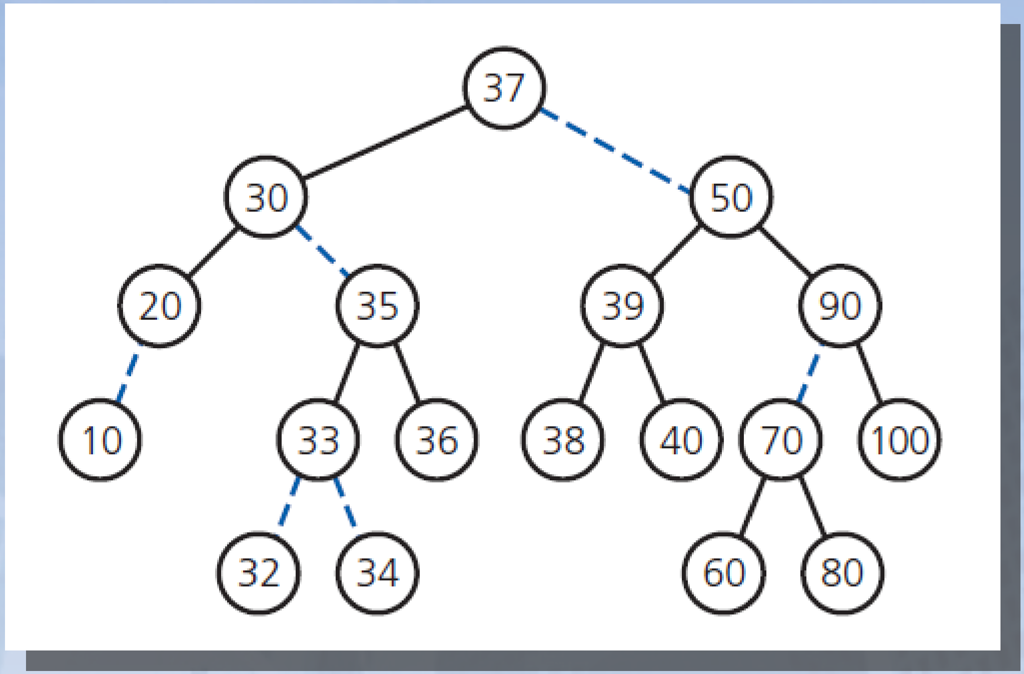


FIGURE 19-36 A red-black tree that represents the 2-3-4 tree in Figure 19-24

Searching and Traversing a Red-Black Tree

- A red-black tree is a binary search tree
- Thus, search and traversal
  - Use algorithms for binary search tree
  - Simply ignore color of pointers

Adding to and Removing from a Red-Black Tree

- Red-black tree represents a 2-3-4 tree
  - Simply adjust 2-3-4 addition algorithms
  - Accommodate red-black representation
- Splitting equivalent of a 4-node requires simple color changes
  - Pointer changes called rotations result in a shorter tree

Adding to and Removing from a Red-Black Tree

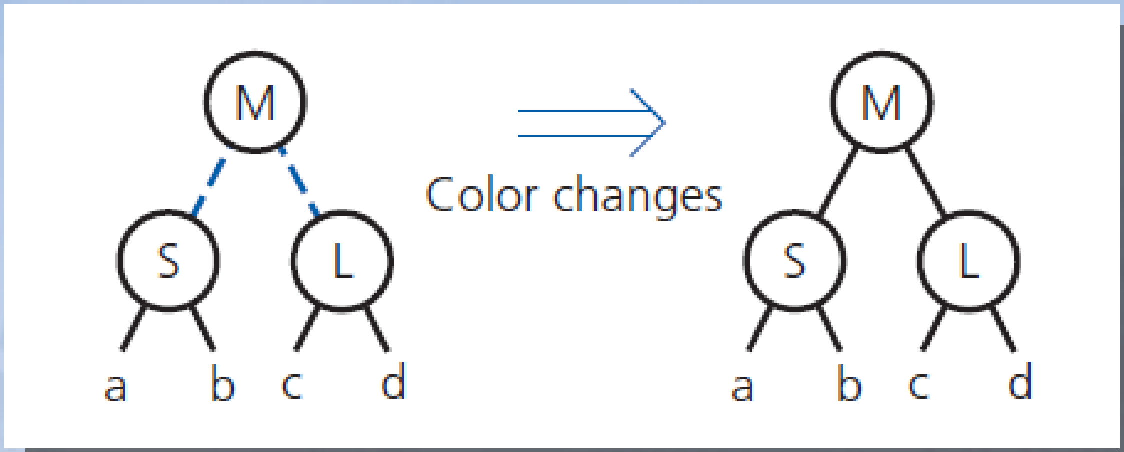


FIGURE 19-37 Splitting a red-black representation of a 4-node root

Adding to and Removing from a Red-Black Tree

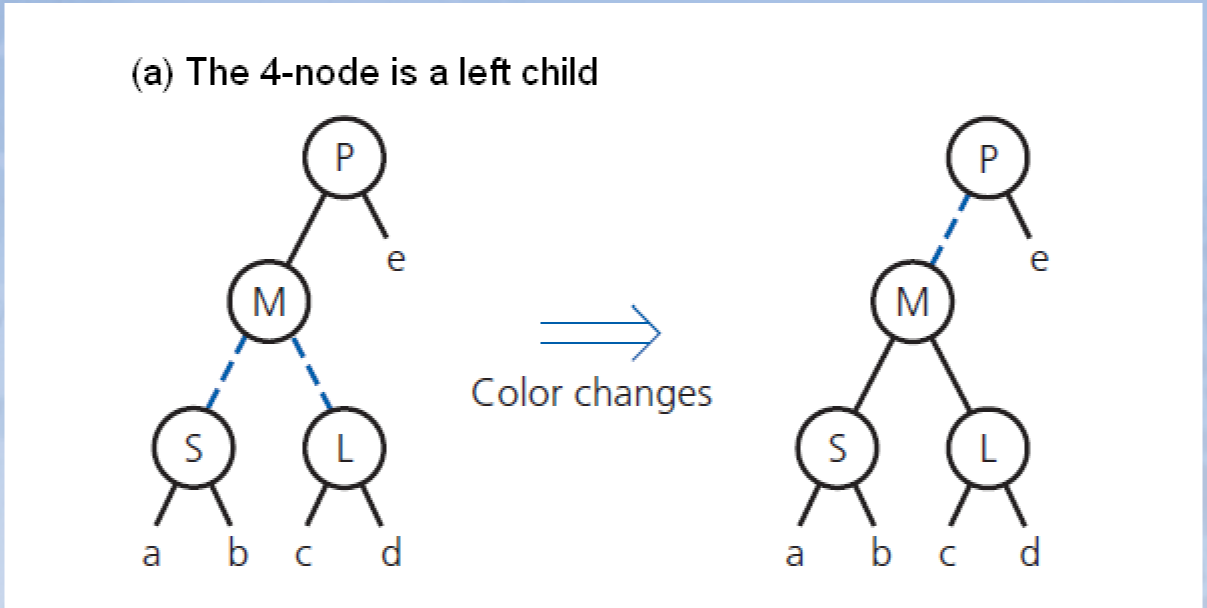


FIGURE 19-38 Splitting a red-black representation of a 4-node whose parent is a 2-node

Adding to and Removing from a Red-Black Tree

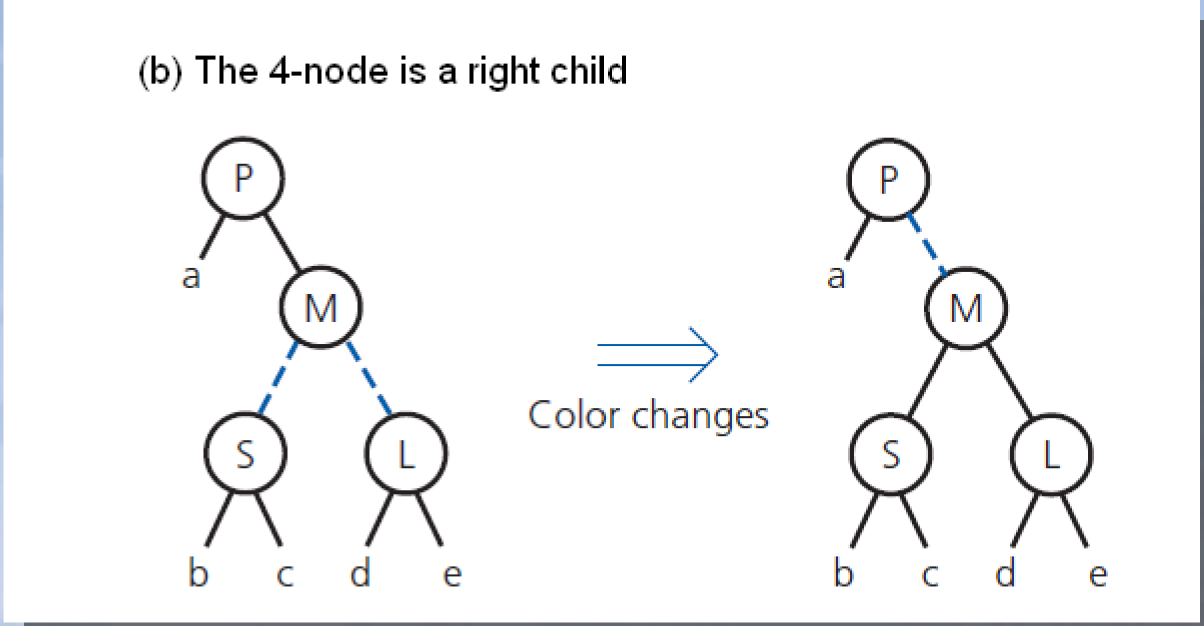


FIGURE 19-38 Splitting a red-black representation of a 4-node whose parent is a 2-node

Adding to and Removing from a Red-Black Tree

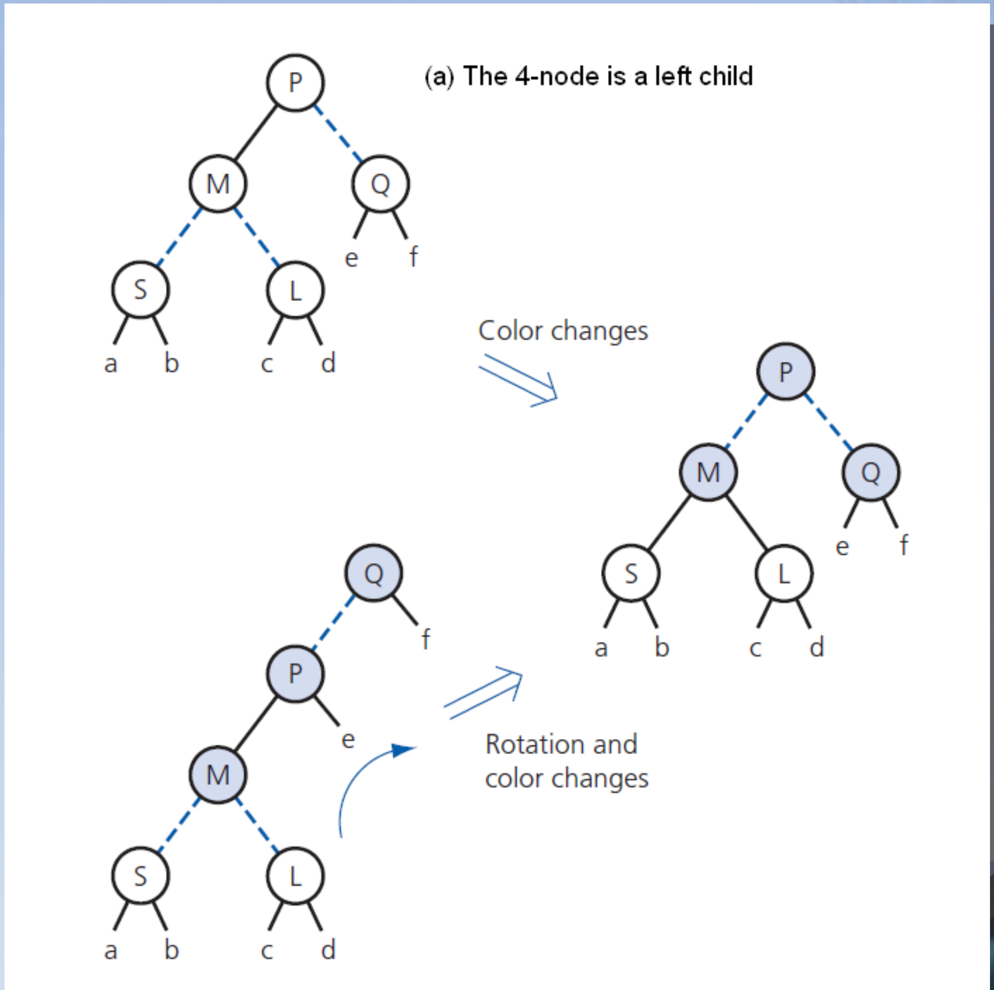
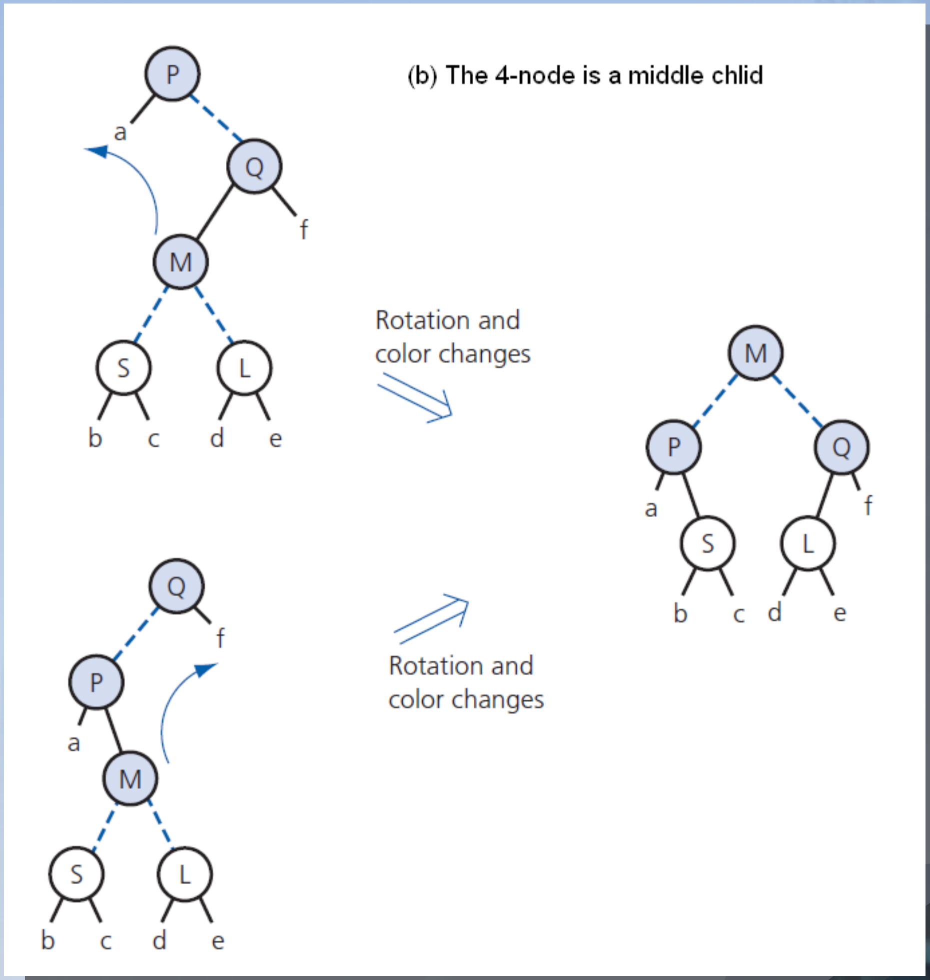


FIGURE 19-39 Splitting a red-black representation of a 4-node whose parent is a 3-node

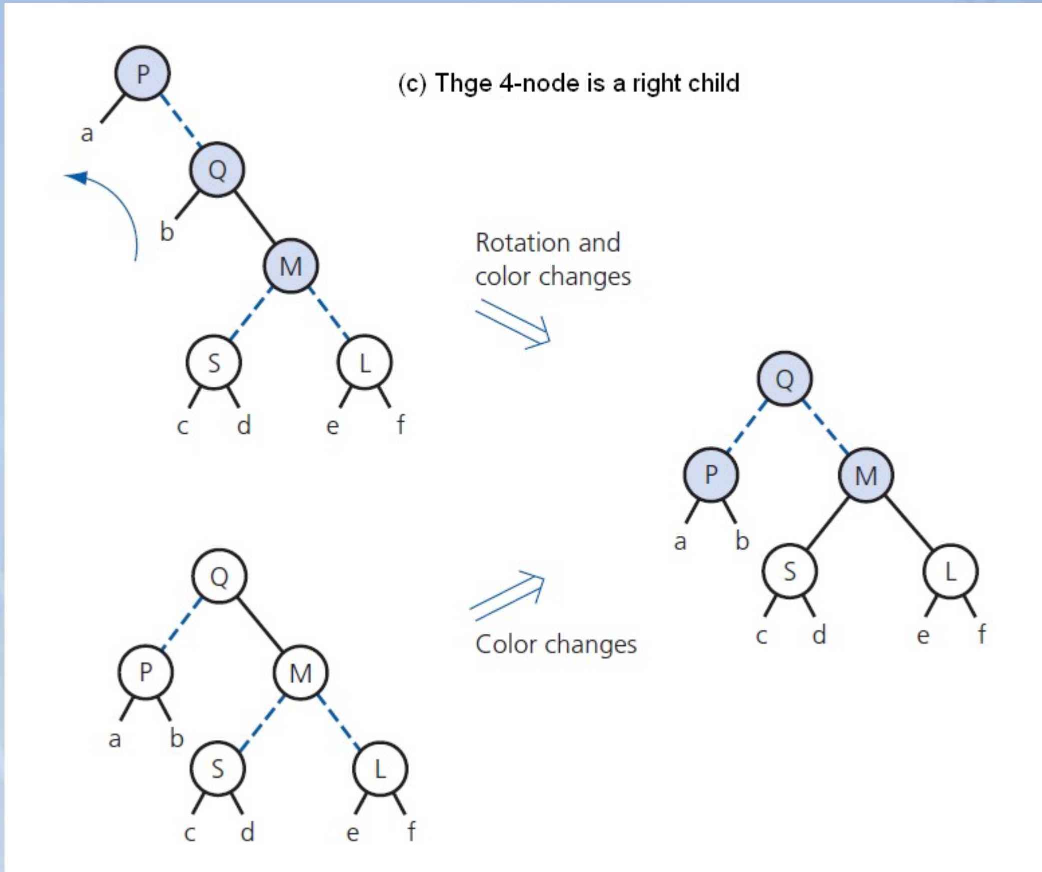
Adding to and Removing from a Red-Black Tree

FIGURE 19-39 Splitting a red-black representation of a 4-node whose parent is a 3-node



Adding to and Removing from a Red-Black Tree

FIGURE 19-39 Splitting a red-black representation of a 4-node whose parent is a 3-node



End

Chapter 19