

Algoritmos e Estrutura de Dados

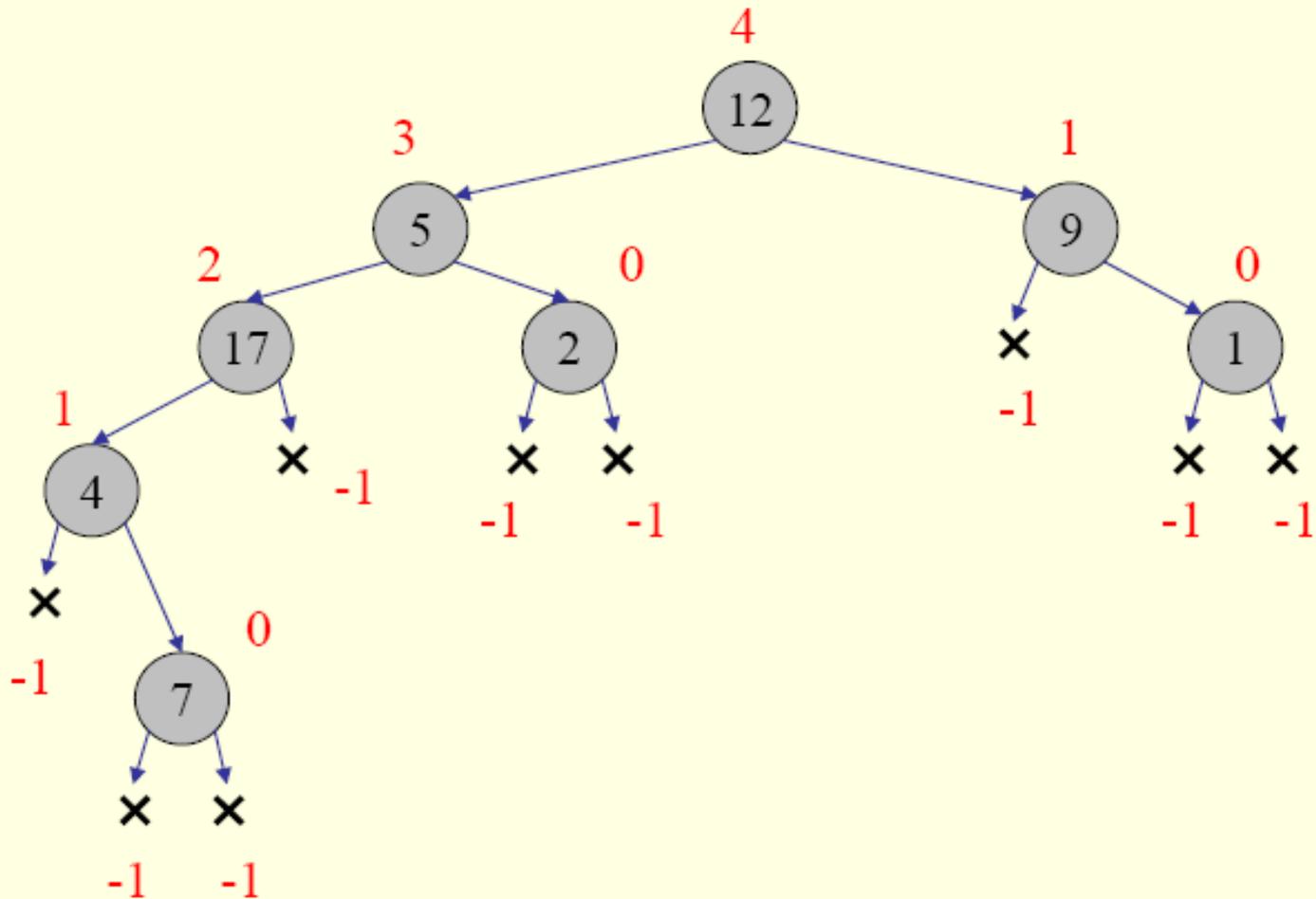


Aula 14 – Estrutura de Dados:
Árvores Balanceadas
Prof. Tiago A. E. Ferreira

Introdução

- Uma árvore balanceada é:
 - É uma árvore de busca binária onde a altura da sub-árvore esquerda e a sub-árvore direita diferem de no máximo uma unidade.
- Estas árvores balanceadas são chamadas de Árvores AVL (ou aproximadamente balanceadas - Árvores Vermelho-Preto)
 - O nome **AVL** vem de seus criadores **Adelson Velskye Landis**, cuja primeira referência encontra-se no documento "*Algoritmos para organização da informação*" de 1962.

Altura de Uma Árvore



Função para Retornar Altura de um Árvore Binária

Altura(x)

SE $x = \text{None}$

retorne -1

$h1 = \text{Altura}(x.\text{esquerda})$

$h2 = \text{Altura}(x.\text{direita})$

retorne $(1 + \text{Max}(h1, h2))$

Balanceamento

- Uma árvore AVL é dita balanceada quando, para cada nodo da árvore, a diferença entre as alturas das suas sub-árvores (direita e esquerda) não é maior do que um.
- Caso a árvore não estiver balanceada é necessário seu balanceamento através da rotação simples ou rotação dupla. O balanceamento é requerido para as operações de adição e exclusão de elementos.

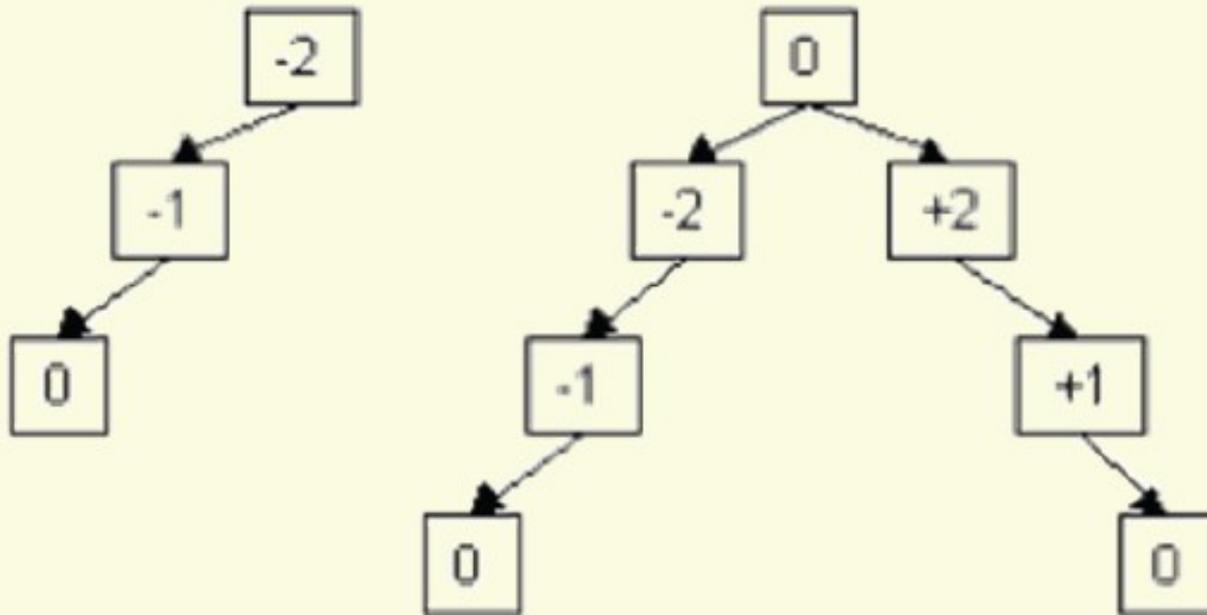
Fator de Balanceamento

- O fator de balanceamento de um nodo é dado pelo seu peso em relação a sua sub-árvore.
 - Um nodo pode ter um fator balanceado de 1, 0, ou -1.
 - Um nodo com fator de balanceamento -2 ou 2 é considerado um árvore não AVL e requer um balanceamento por rotação ou dupla-rotação.

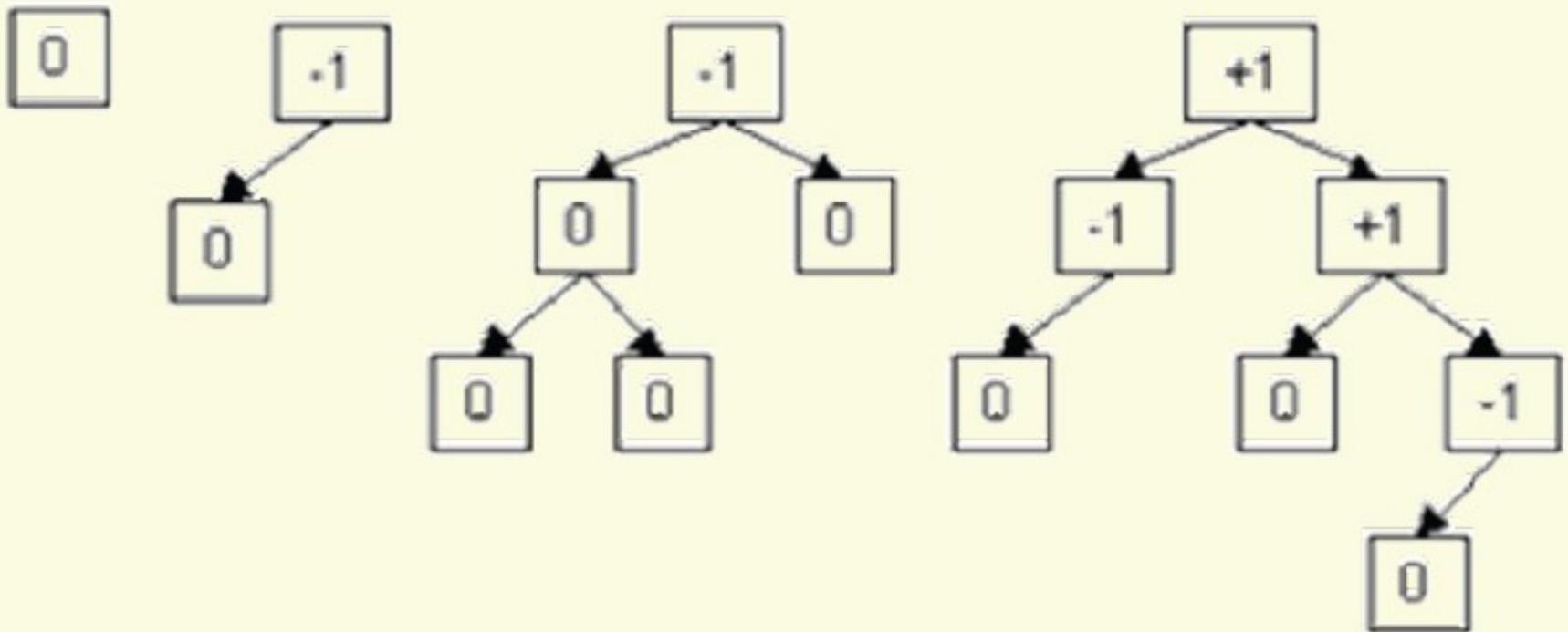
Cálculo do Fator de Balanceamento

- O fator de balanceamento pode ser dado como:
 - $h(\text{sub-árvore esquerda}) - h(\text{sub-árvore direita})$
 - Ou, **$h(\text{sub-árvore direita}) - h(\text{sub-árvore esquerda})$**
 - Onde $h(x)$ é a altura do nodo x .
 - Essa escolha irá influenciar no momento do balanceamento.

Exemplo de Árvores Não AVL



Exemplos da Árvores AVL



Balanceando uma Árvore

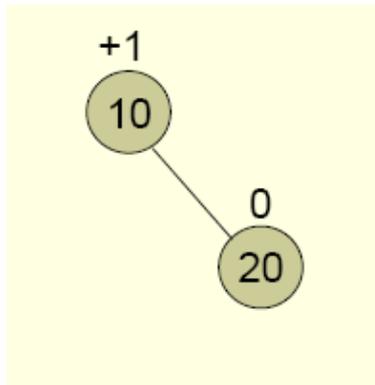
- Dada uma árvore não balanceada, como balanceá-la?
 - Através de operações de **rotações!!!!**
- Existem 4 operações de Rotações:
 - Rotação à Esquerda
 - Rotação à Direita
 - Rotação Dupla à Esquerda
 - Rotação Dupla à Direita

Quando usar as Rotações

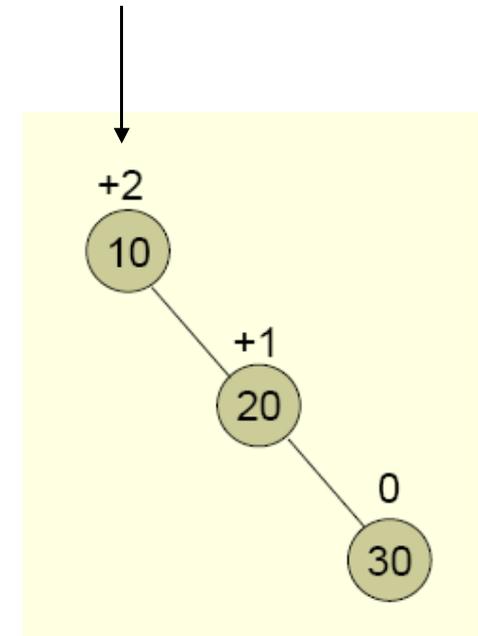
- Na inserção de um elemento
- Na remoção de um elemento
 - É provado que no máximo uma rotação é suficiente para realizar o balanceamento de uma árvore quando é inserido um novo elemento
 - A idéia da operação de balanceamento é equilibrar o fator de balanceamento da árvore

Exemplo: Rotação à Esquerda

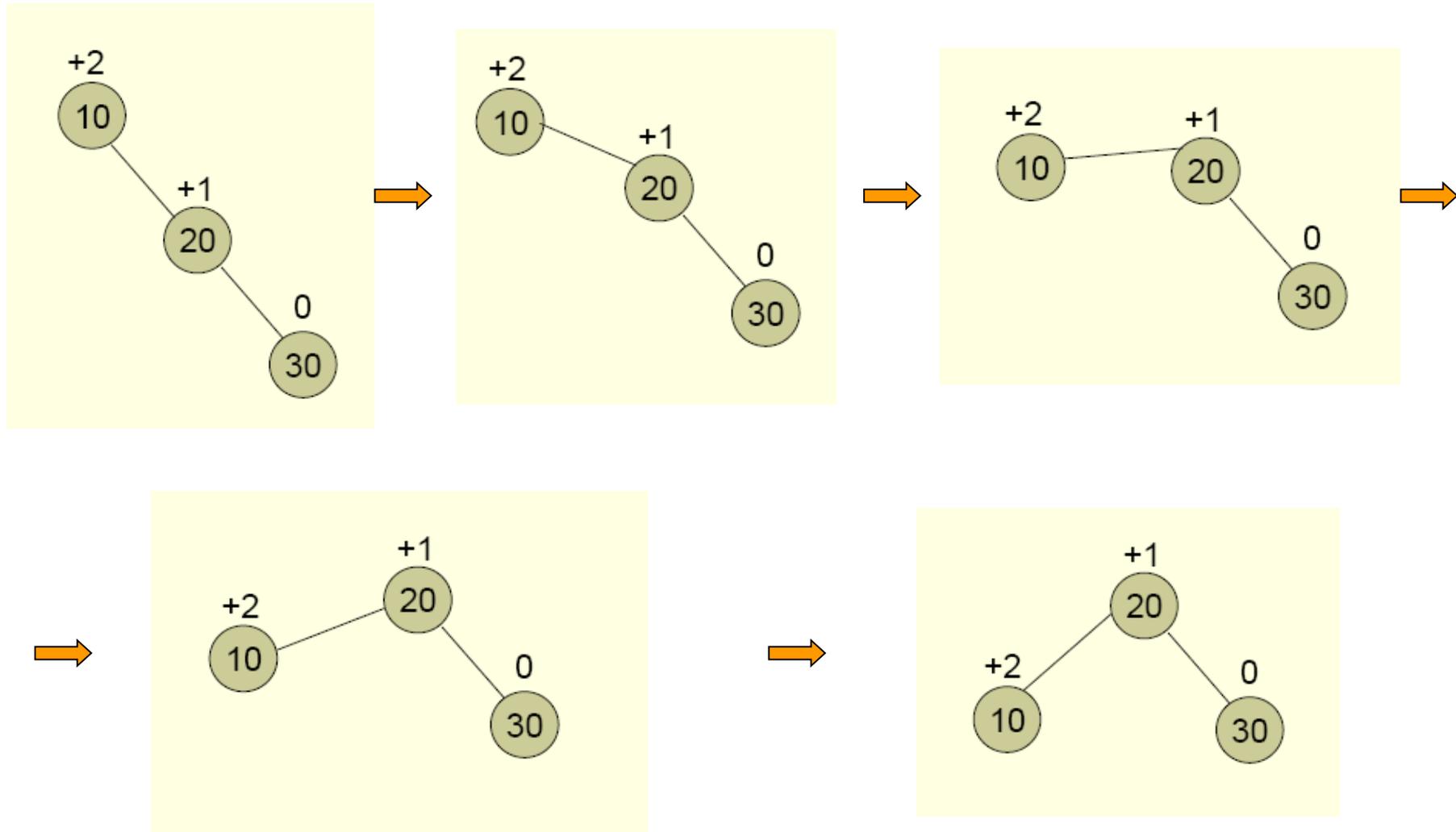
A árvore torna-se desbalanceada no nodo de valor 10



Inserindo um novo nodo

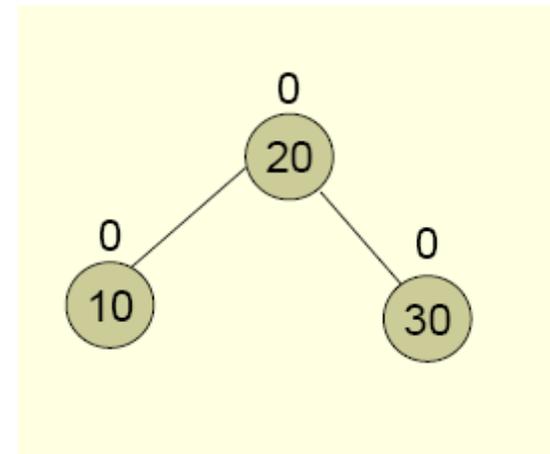
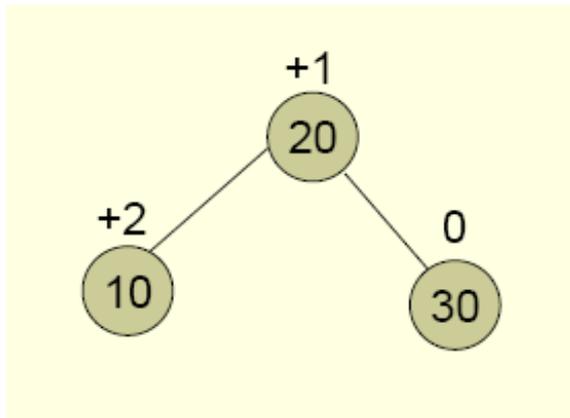


Exemplo: Rotação à Esquerda



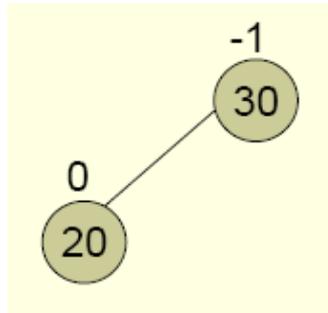
Exemplo: Rotação à Esquerda

Recalculando o Fator de Balanceamento



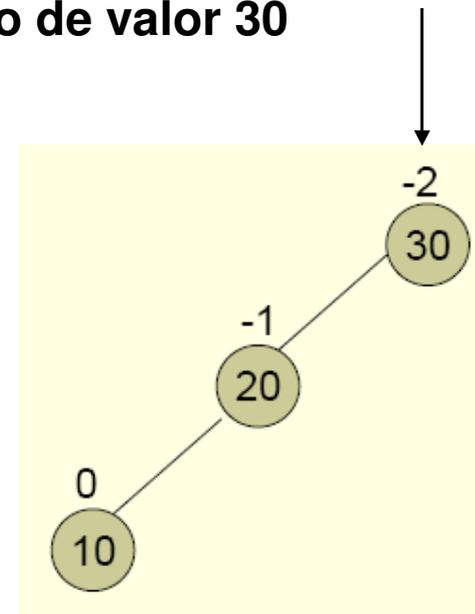
Árvore Balanceada

Exemplo: Rotação à Direita

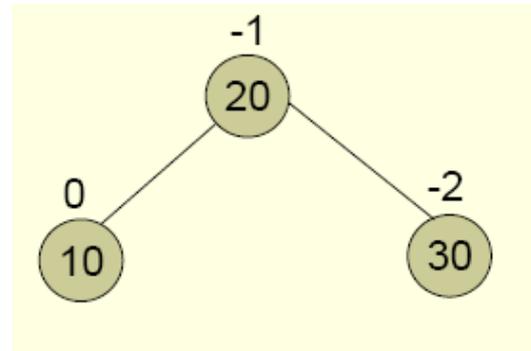
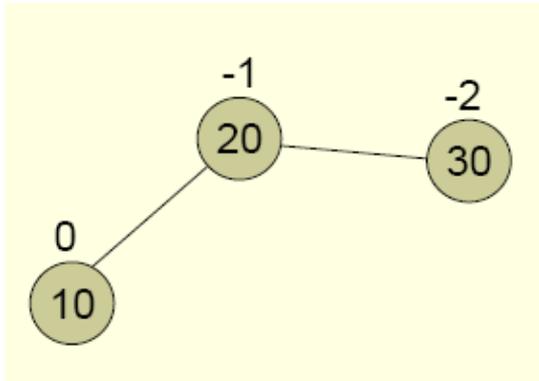
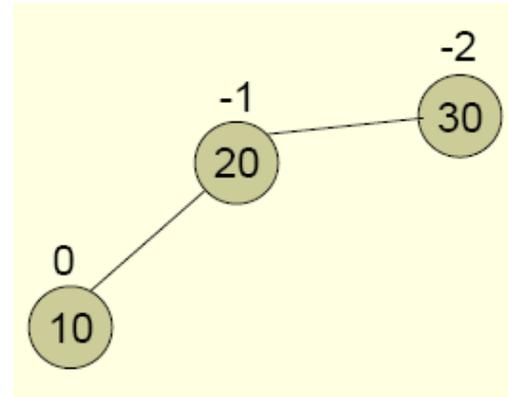
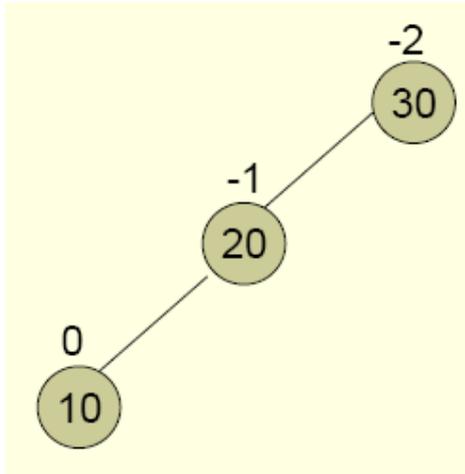


Inserindo um novo nodo

A árvore torna-se desbalanceada no nodo de valor 30

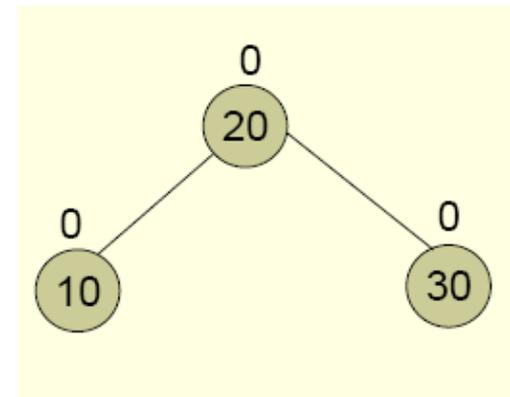
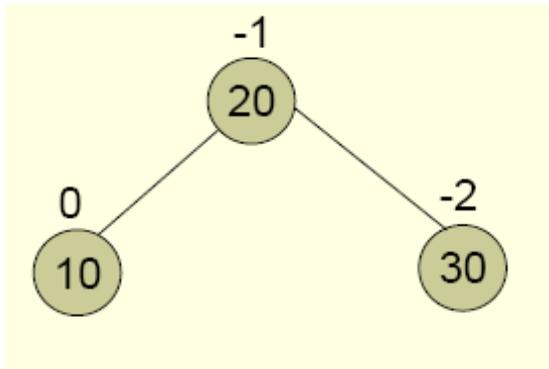


Exemplo: Rotação à Direita



Exemplo: Rotação à Direita

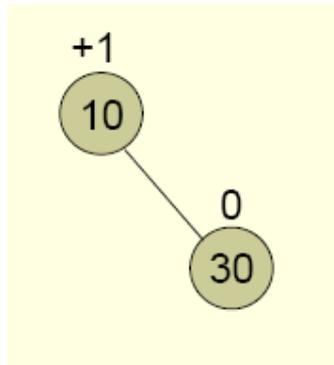
Recalculando o Fator de Balanceamento



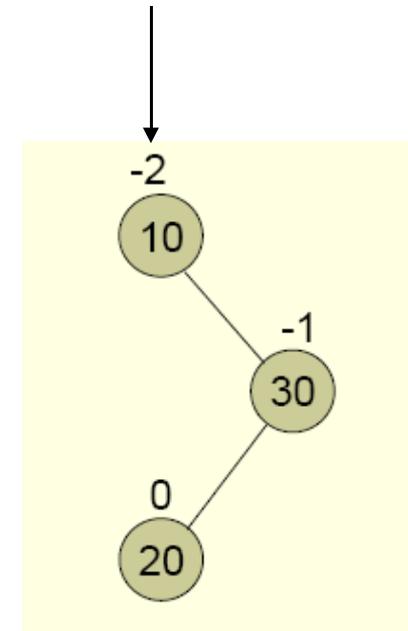
Árvore Balanceada

Exemplo: Rotação Dupla à Esquerda

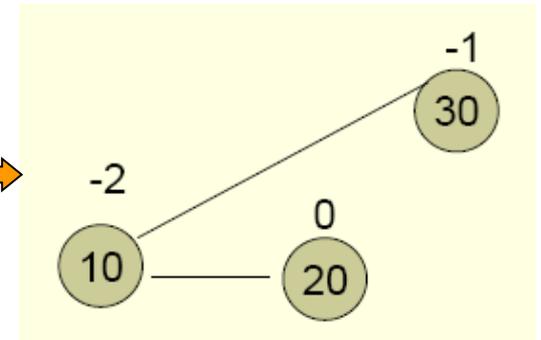
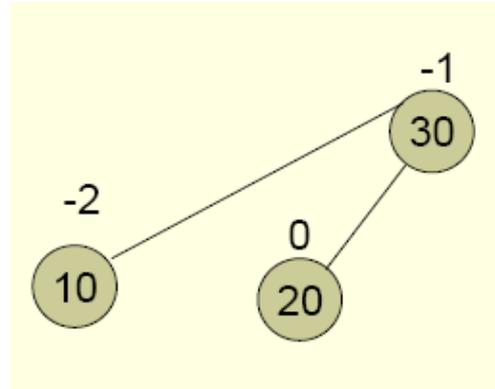
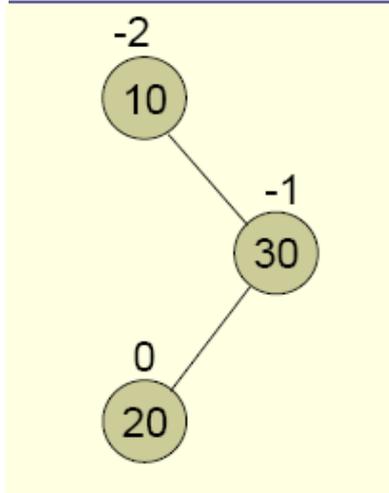
A árvore torna-se desbalanceada no nodo de valor 10



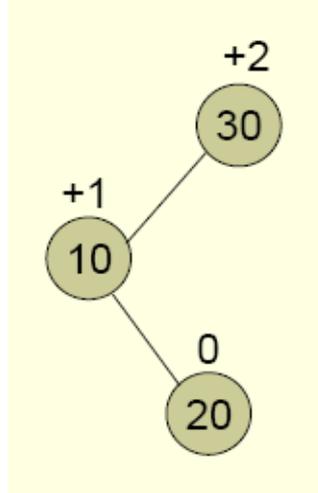
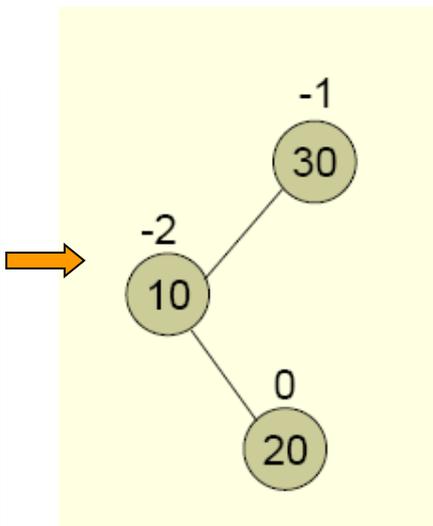
Inserindo um novo nodo



Exemplo: Rotação Dupla à Esquerda



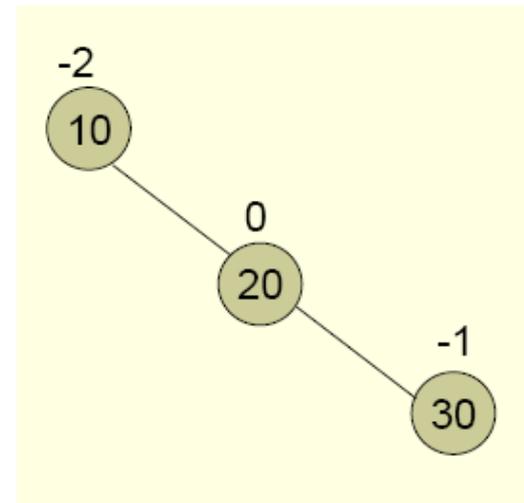
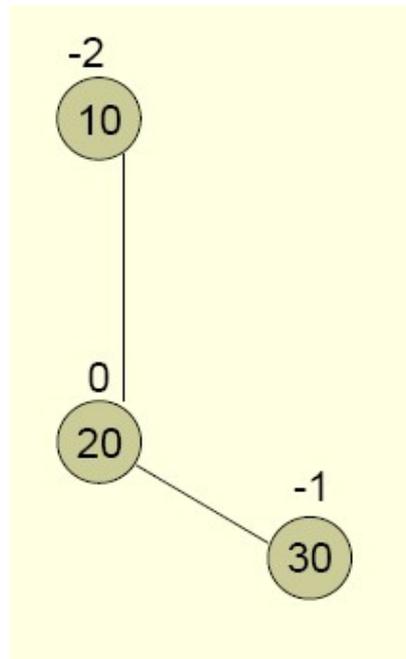
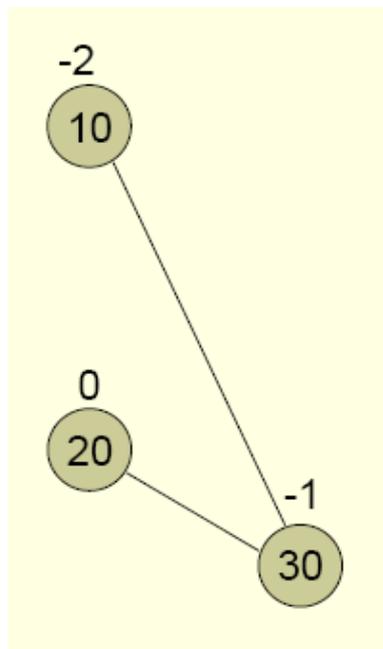
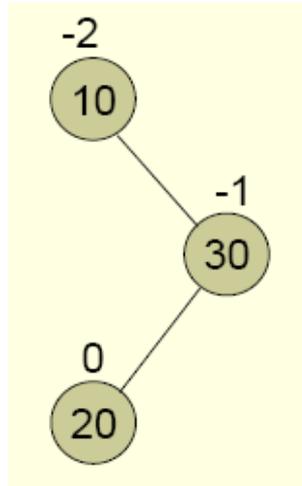
Recalculando o Fator de Balanceamento



NOVO PROBLEMA!!!!
Girando no sentido contrário,
volta-se para a árvore inicial!

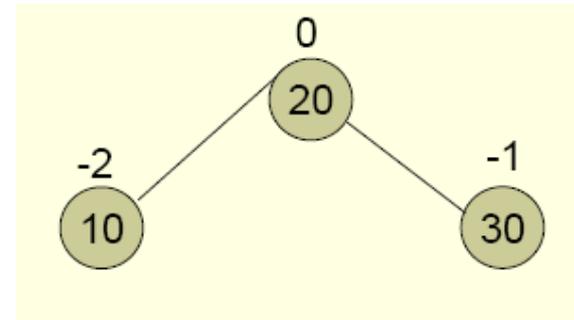
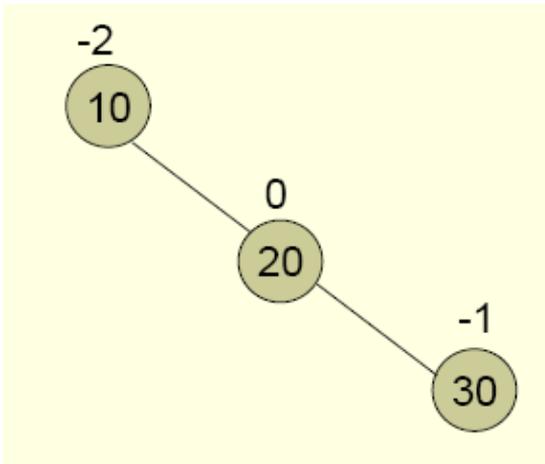
Exemplo: Rotação Dupla à Esquerda

Solução: Girar primeiro a árvore para a direita!

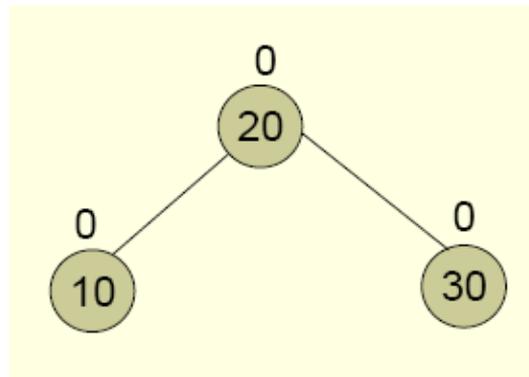


Agora girar para a esquerda

Exemplo: Rotação Dupla à Esquerda

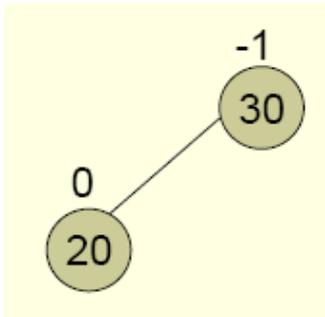


Recalculando o Fator de Balanceamento

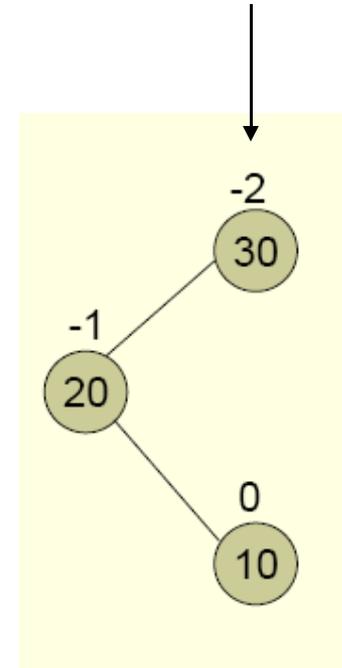


Exemplo: Rotação Dupla à Direita

A árvore torna-se desbalanceada no nodo de valor 30

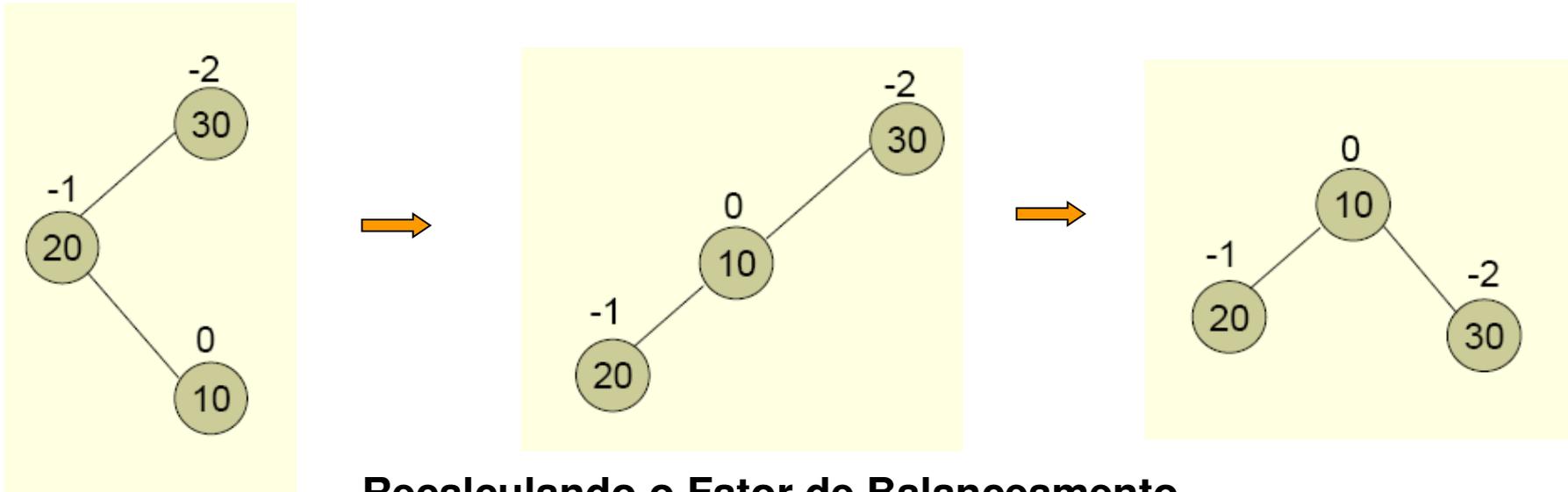


Inserindo um novo nodo

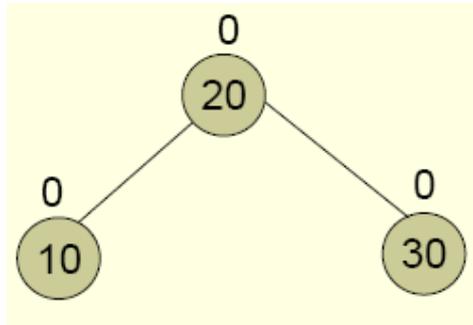


Exemplo: Rotação Dupla à Direita

Primeiro, rotaciona-se à esquerda, e então à direita:



Recalculando o Fator de Balanceamento



Removendo Elemento

- Os exemplos vistos são para o balanceamento de uma árvore binária a partir da inserção de um novo nodo na árvore
- Mas e se for desejado remover um determinado nodo da árvore?
 - Esta também irá, provavelmente, necessitar de um balanceamento!
 - Mas como proceder para a remoção de um nodo?

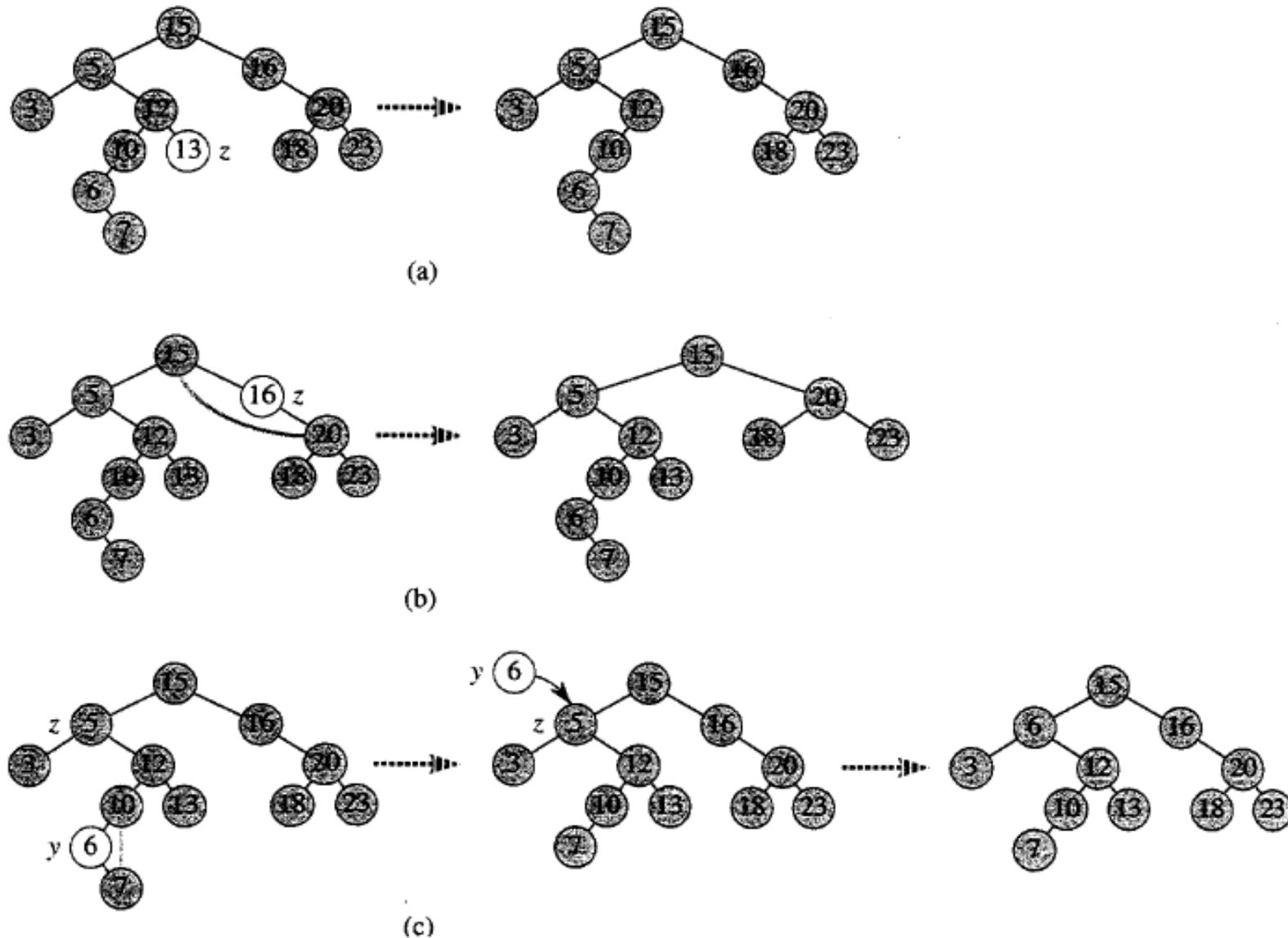


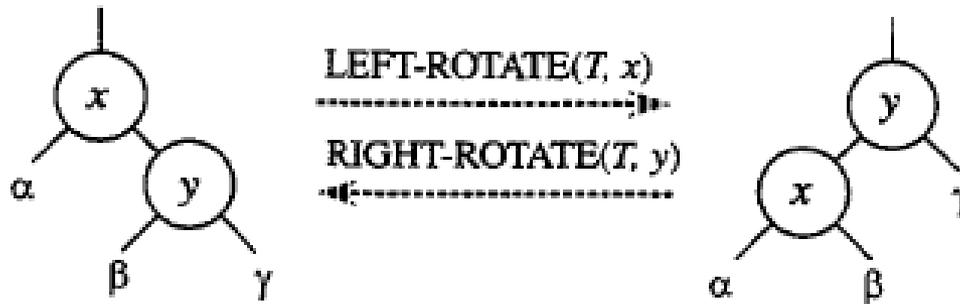
FIGURA 12.4 Eliminação de um nó z de uma árvore de pesquisa binária. O nó realmente removido depende de quantos filhos z tem; esse nó está levemente sombreado. (a) Se z não tem nenhum filho, simplesmente o removemos. (b) Se z tem apenas um filho, extraímos z . (c) Se z tem dois filhos, extraímos seu sucessor y , que tem no máximo um filho, e depois substituímos a chave e os dados satélite de z pela chave e os dados satélite de y

Removendo Elemento

```
TREE-DELETE( $T, z$ )
1  if esquerda[ $z$ ] = NIL or direita[ $z$ ] = NIL
2    then  $y \leftarrow z$ 
3    else  $y \leftarrow$  TREE-SUCCESSOR( $z$ )
4  if esquerda[ $y$ ]  $\neq$  NIL
5    then  $x \leftarrow$  esquerda[ $y$ ]
6    else  $x \leftarrow$  direita[ $y$ ]
7  if  $x \neq$  NIL
8  then  $p[x] \leftarrow p[y]$ 
9  if  $p[y] =$  NIL
10   then  $raiz[T] \leftarrow x$ 
11   else if  $y =$  esquerda[ $p[y]$ ]
12         then esquerda[ $p[y]$ ]  $\leftarrow x$ 
13         else direita[ $p[y]$ ]  $\leftarrow x$ 
14  if  $y \neq z$ 
15   then chave[ $z$ ]  $\leftarrow$  chave[ $y$ ]
16         copiar dados satélite de  $y$  em  $z$ 
17  return  $y$ 
```

Removendo Elemento

- Uma Vez removido o elemento, o fator de balanceamento deve ser recalculado e a árvore balanceada, se necessário, como as operações de rotação!
 - Mas como implementar as operações de rotação?



Rotação à Esquerda

LEFT-ROTATE(T, x)

- 1 $y \leftarrow direita[x]$ ▷ Define y .
- 2 $direita[x] \leftarrow esquerda[y]$ ▷ Faz da subárvore esquerda de y a subárvore direita de x .
- 3 $p[esquerda[y]] \leftarrow x$
- 4 $p[y] \leftarrow p[x]$ ▷ Liga o pai de x a y .
- 5 **if** $p[x] = nil[T]$
- 6 **then** $raiz[T] \leftarrow y$
- 7 **else if** $x = esquerda[p[x]]$
- 8 **then** $esquerda[p[x]] \leftarrow y$
- 9 **else** $direita[p[x]] \leftarrow y$
- 10 $esquerda[y] \leftarrow x$ ▷ Coloca x à esquerda de y .
- 11 $p[x] \leftarrow y$

Rotação à Esquerda

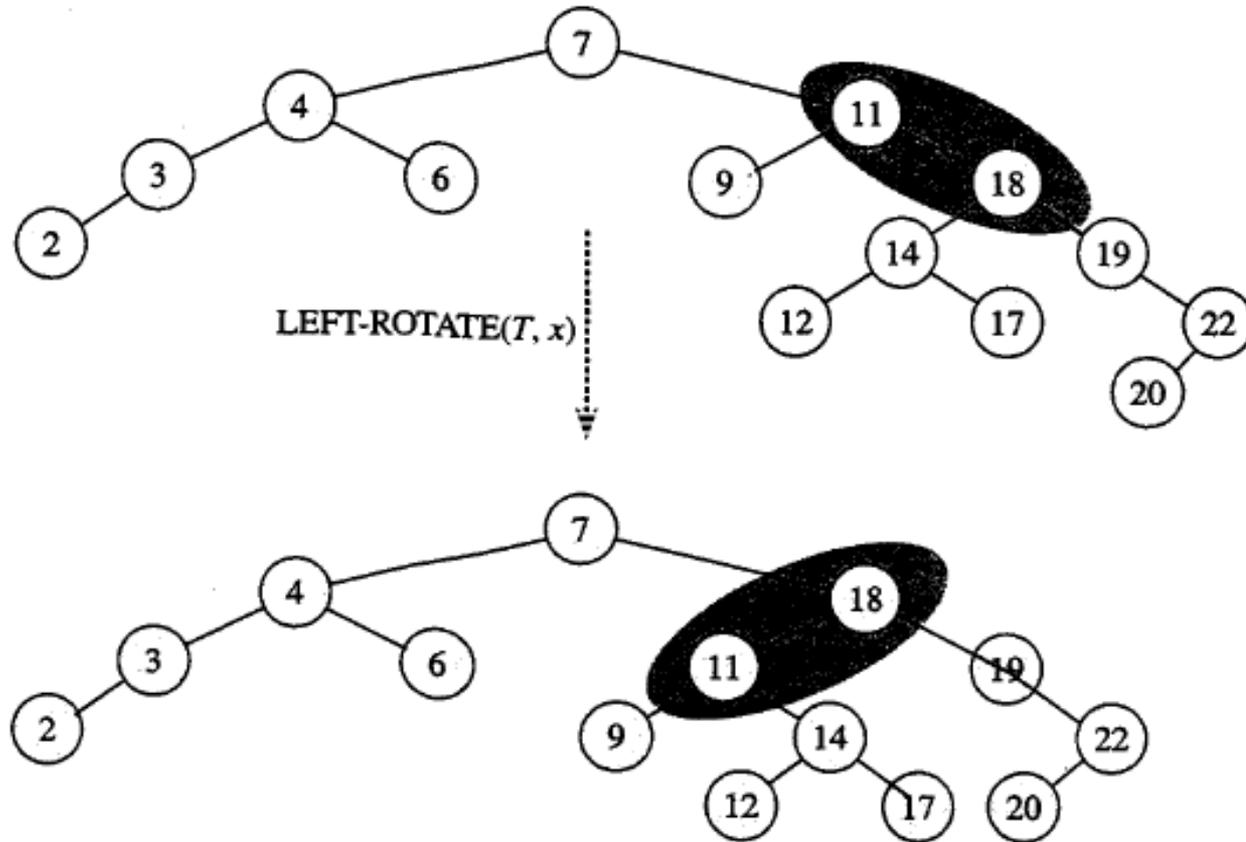


FIGURA 13.3 Um exemplo de como o procedimento LEFT-ROTATE(T, x) modifica uma árvore de pesquisa binária. Os percursos de árvore em ordem da árvore de entrada e a árvore modificada produzem a mesma listagem de valores de chaves

Rotação à Direita

- Esta rotação é o processo inverso da rotação à esquerda...
- Como seria o seu pseudo-código?