



Algoritmos e Estrutura de Dados II

Árvore AVL

Prof Márcio Bueno

ed2tarde@marciobueno.com / ed2noite@marciobueno.com

Material da Prof^a Ana Eliza

Introdução

- Árvores Binárias de Busca

- Objetivo da Utilização

- Minimizar o número de comparações efetuadas, no pior caso, para uma busca com chaves de probabilidades de ocorrência idênticas.

- Restrições

- Aplicações estáticas \Rightarrow Ideal
- Aplicações dinâmicas \Rightarrow Desaconselhável

Introdução

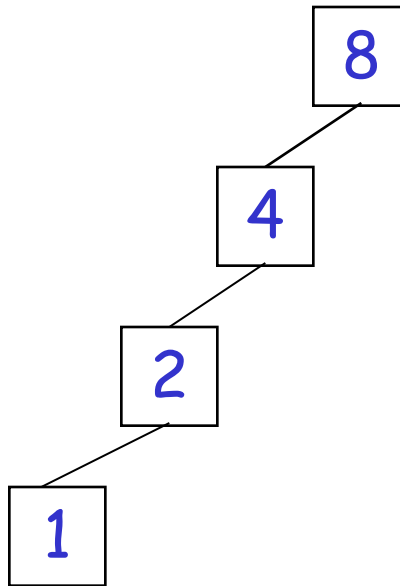
- Árvores Binárias de Busca

- Problema

- Sucessivas inserções e remoções de nós em uma ABB fazem com que existam diferenças sensíveis entre os níveis das suas folhas, o que acarreta grandes diferenças de desempenho no acesso às informações.
- **Pior caso:** Árvores assimétricas.

Introdução

- Árvores Binárias de Busca
 - Problema
 - **Pior caso**: Árvores assimétricas (semelhante a uma lista linear).



Introdução

- Árvores Binárias de Busca

- Alternativa

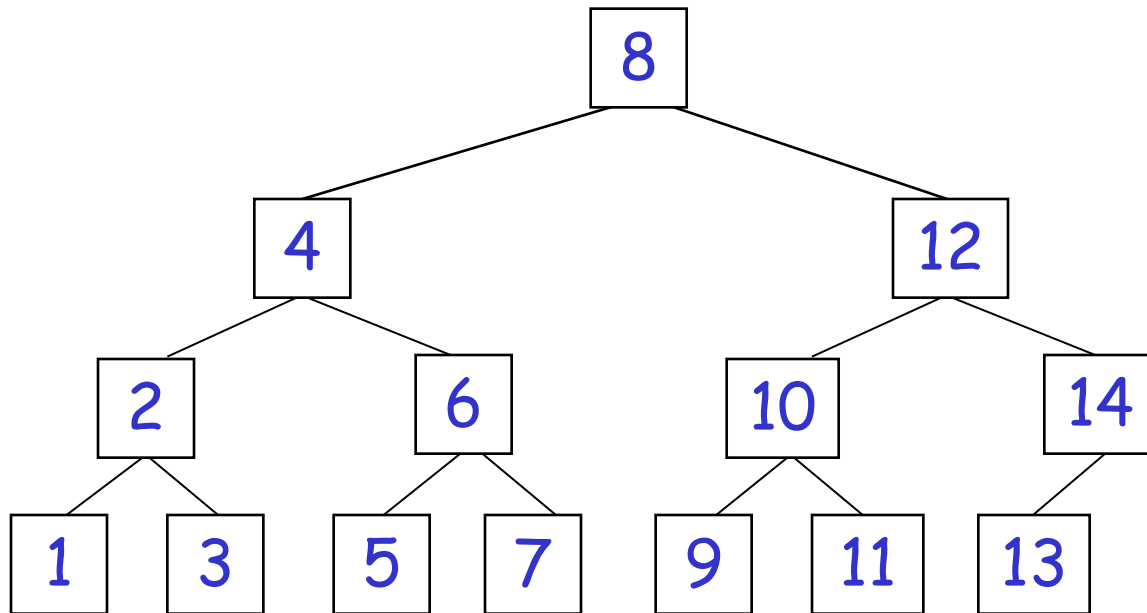
- Tornar a árvore novamente completa, tão logo tal característica seja perdida após uma inserção ou remoção.

- Problema

- Alto custo de reestruturação da árvore.
- Solução pouco eficiente e, às vezes, impossível.

Introdução

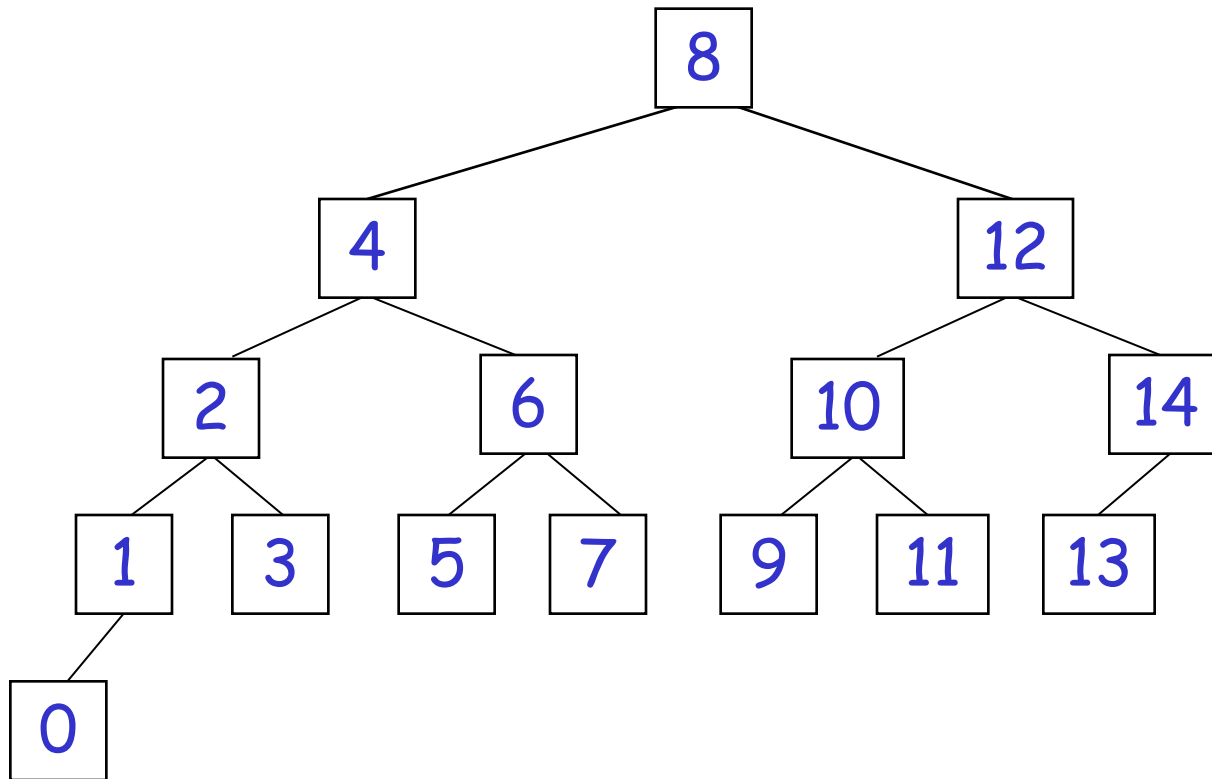
- Árvores Binárias de Busca Completa



- Inserir 0

Introdução

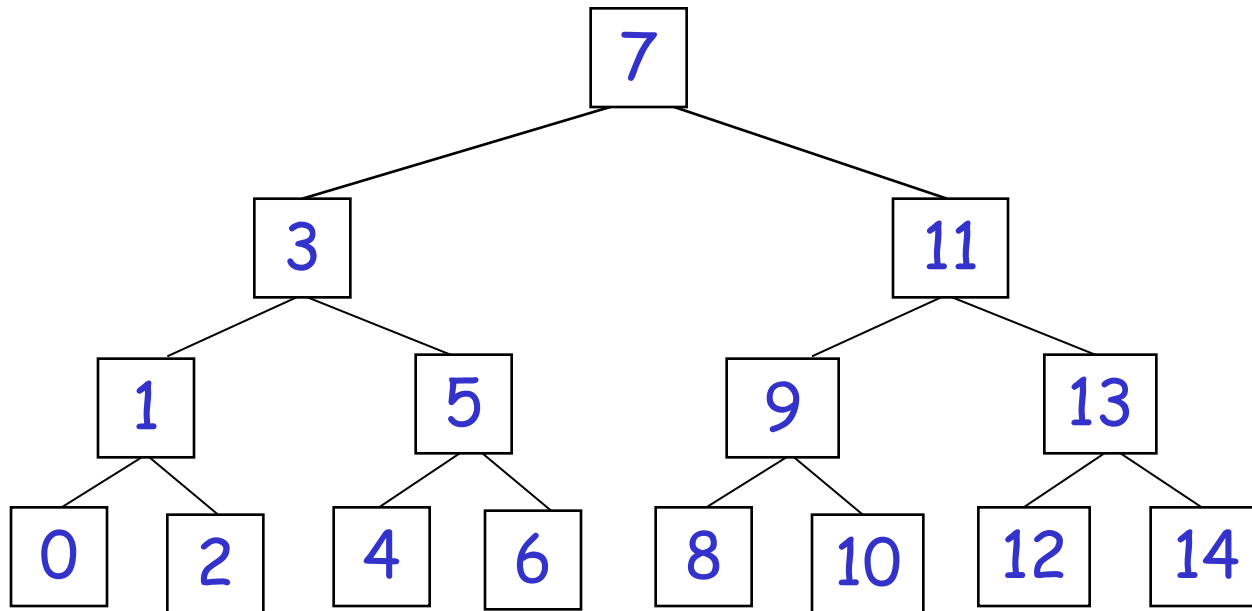
- Árvores Binárias de Busca Completa



- Não é mais completa!

Introdução

- Árvores Binárias de Busca Completa (e Cheia)



Árvores Balanceadas

- **Definição** (Rigorosa)
 - Uma árvore é dita balanceada quando, para qualquer nó, as suas subárvores à esquerda e à direita possuem a mesma altura.
 - Isso equivale a dizer que todas as suas folhas estão no mesmo nível.
 - Ou seja, que a árvore está cheia.

Árvores Balanceadas

- **Definição** (Menos Rigorosa)
 - Uma árvore é considerada balanceada quando, para cada nó n , as alturas das subárvores à esquerda e à direita diferem, no máximo, de um.
 - Essa diferença é chamada de "*fator de balanceamento*" do nó n
 - **FatBal** (n)

Árvores Balanceadas

- **Definição** (Menos Rigorosa)
 - Conceito introduzido em 1962 pelos matemáticos russos Adelson-Velskii e Landis.
 - Conhecidas como **árvores AVL**.

Árvores AVL

- **Características Básicas**

- O fator de balanceamento deve constar em cada nó de uma árvore AVL.

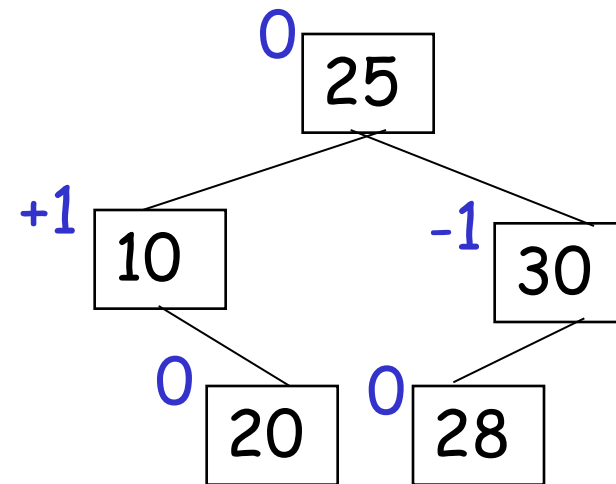
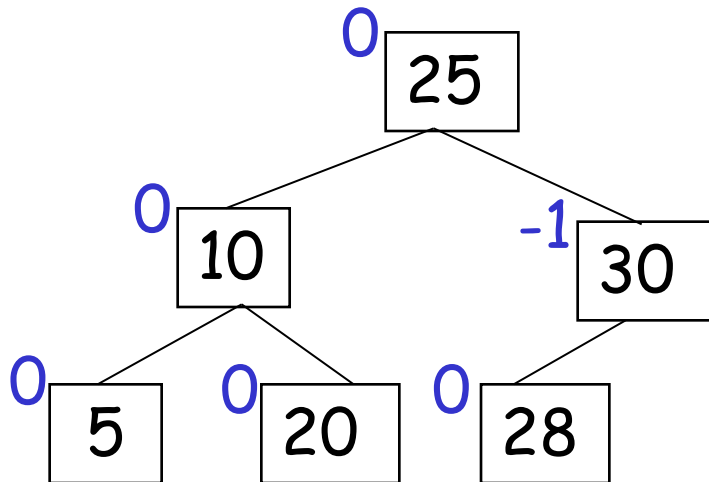
- $\text{FatBal}(n) = \text{Negativo} \Rightarrow$ subárvore à esquerda mais alta.

- $\text{FatBal}(n) = \text{Positivo} \Rightarrow$ subárvore à direita mais alta.

- $\text{FatBal}(n) = \text{Zero} \Rightarrow$ subárvores à esquerda e à direita com mesma altura.

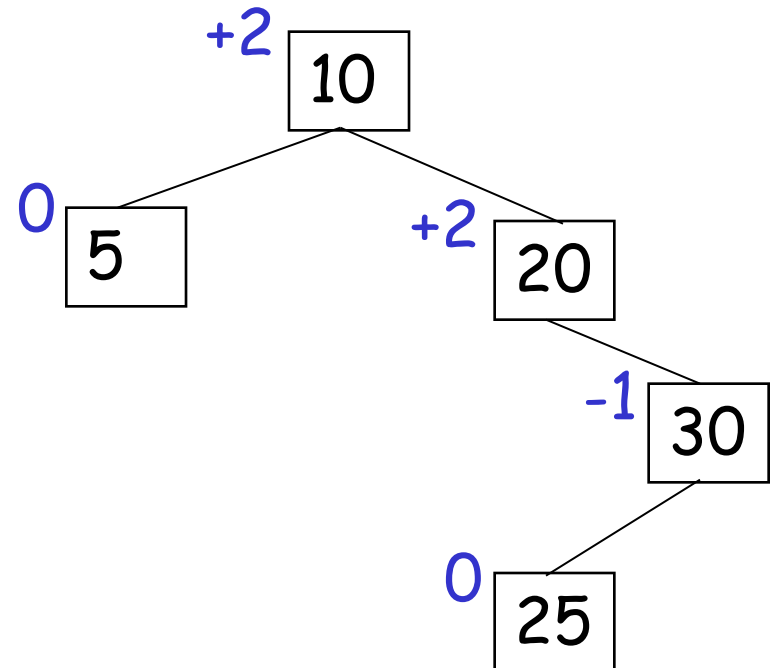
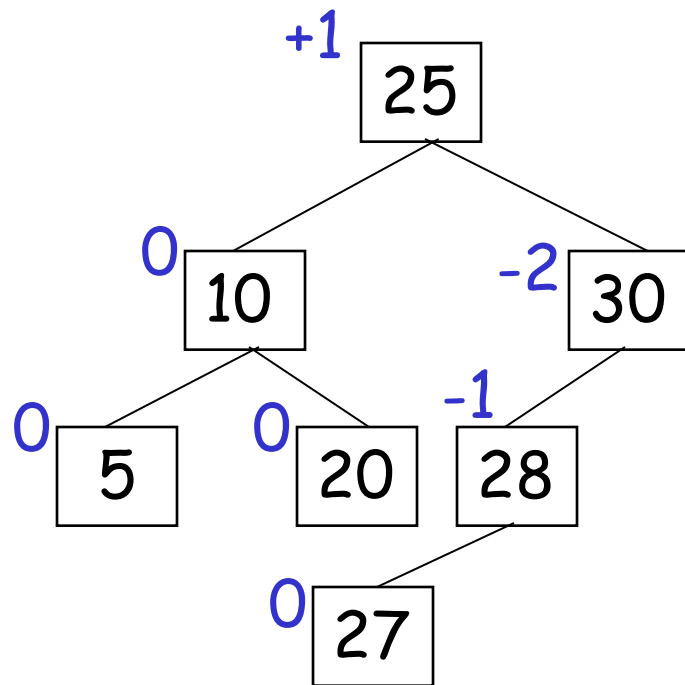
Árvores AVL

- Exemplos de árvores balanceadas



Árvores AVL

- Exemplos de árvores não balanceadas



Árvores AVL

- Operações Críticas
 - Inserção
 - Remoção

Inserção em Árvores AVL

- Sempre ocorre nas folhas;
- **Pode ocasionar:**
 - O aumento da altura da subárvore onde o nó foi inserido;
 - A alteração dos fatores de balanceamento dos nós daquela subárvore.

Inserção em Árvores AVL

- **Algoritmo de Inserção**
 - Efetuar a inserção;
 - Ajustar os fatores de balanceamento;
 - Verificar a quebra do equilíbrio;
 - Se a árvore não estiver balanceada, corrigir a estrutura através de movimentações dos nós (**Rotações**).

Inserção em Árvores AVL

- Rotação

- Tipos de Rotação

- *Simple*

- *Dupla*

- Principais nós envolvidos:

- **Nó A**: Nó ancestral mais próximo do nó inserido que possuía fator de balanceamento diferente de zero antes da inserção.

- **Nó B**: Filho de A na subárvore onde ocorreu a inserção.

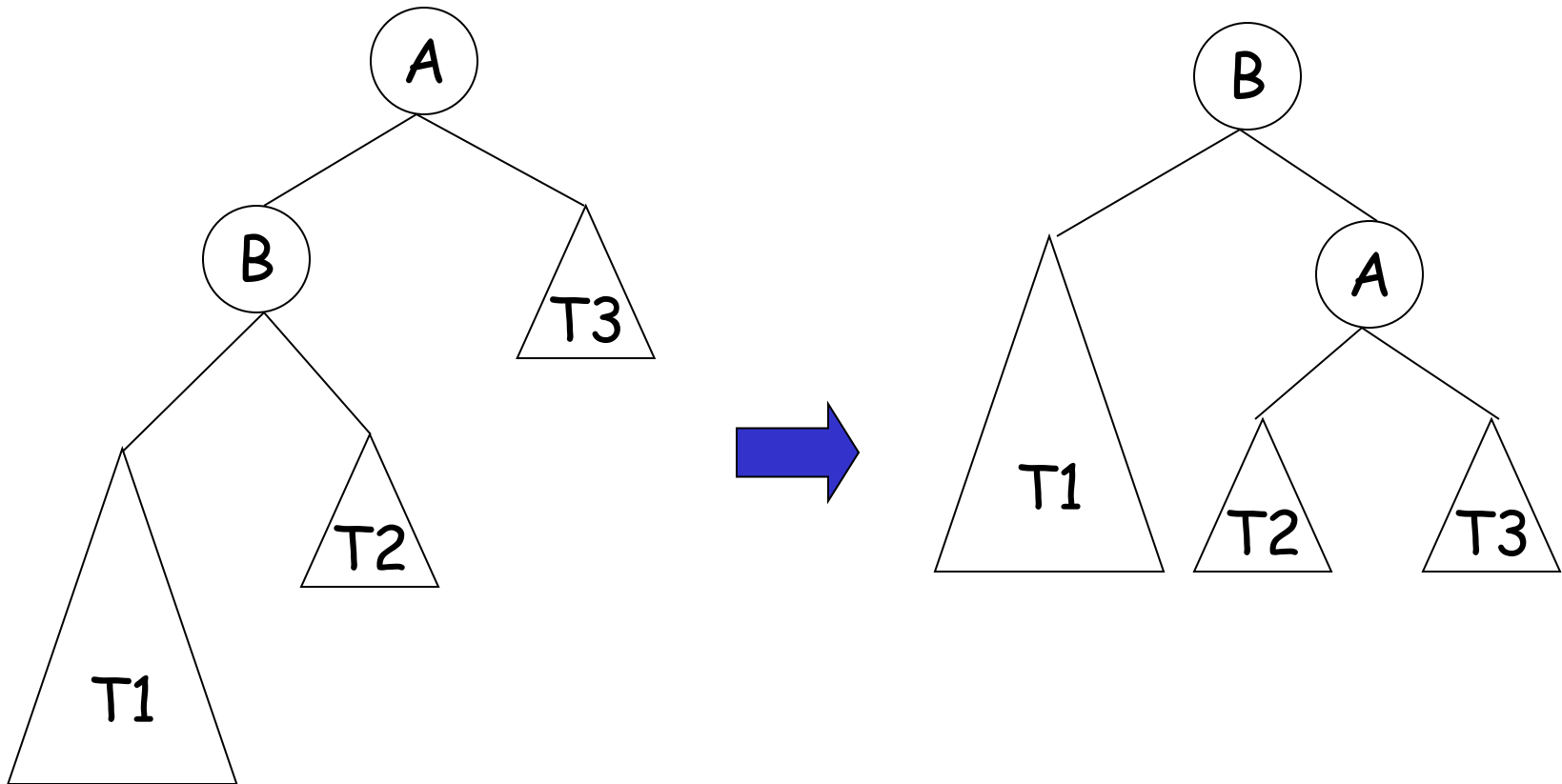
Inserção em Árvores AVL

- **Rotação Simples**

- Utilizada quando a inserção ocorre do "lado de fora".
- Dado um nó A , podemos ter:
 - **Caso 1:** Inserção na subárvore à esquerda do filho à esquerda de $A \Rightarrow$ **Rotação simples à direita.**
 - **Caso 2:** Inserção na subárvore à direita do filho à direita de $A \Rightarrow$ **Rotação simples à esquerda**

Inserção em Árvores AVL

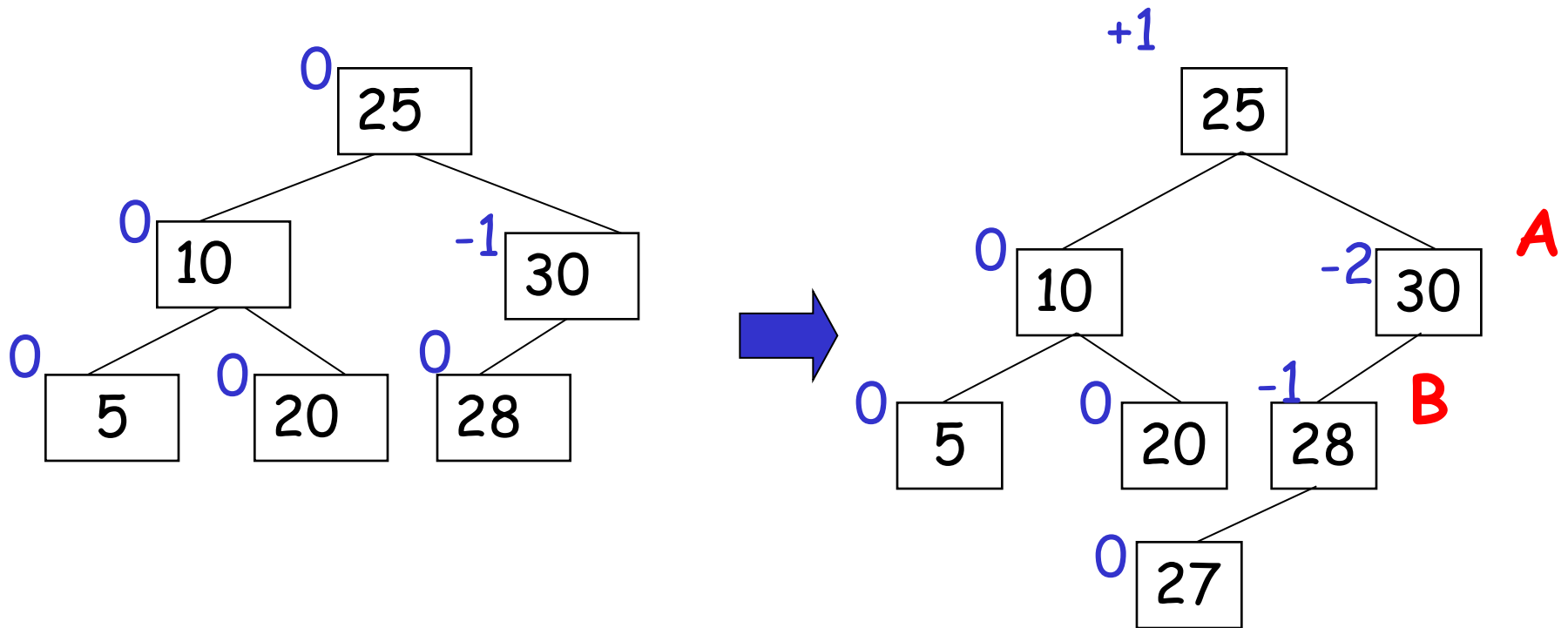
- Rotação Simples à Direita



OBS: O nó foi inserido na subárvore T1

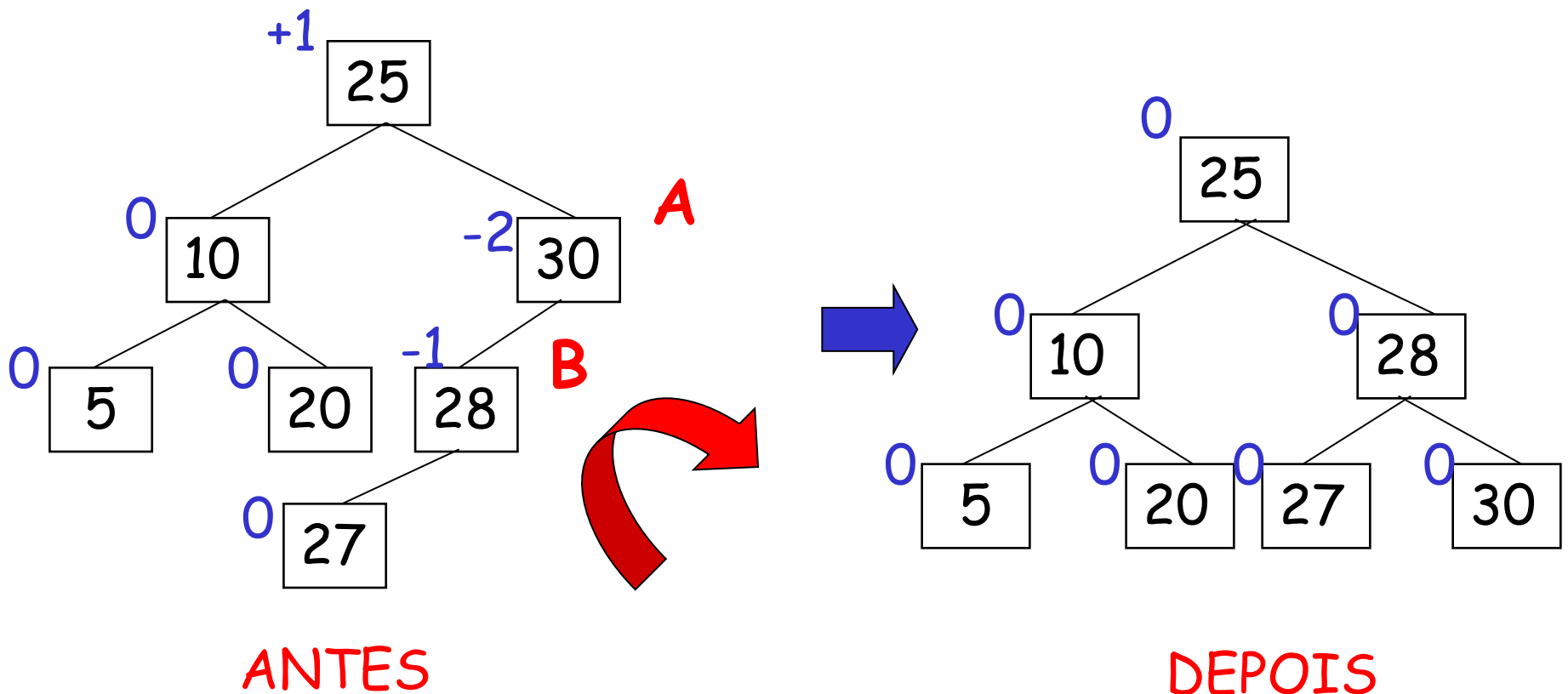
Inserção em Árvores AVL

- Rotação Simples à Direita - Exemplo:



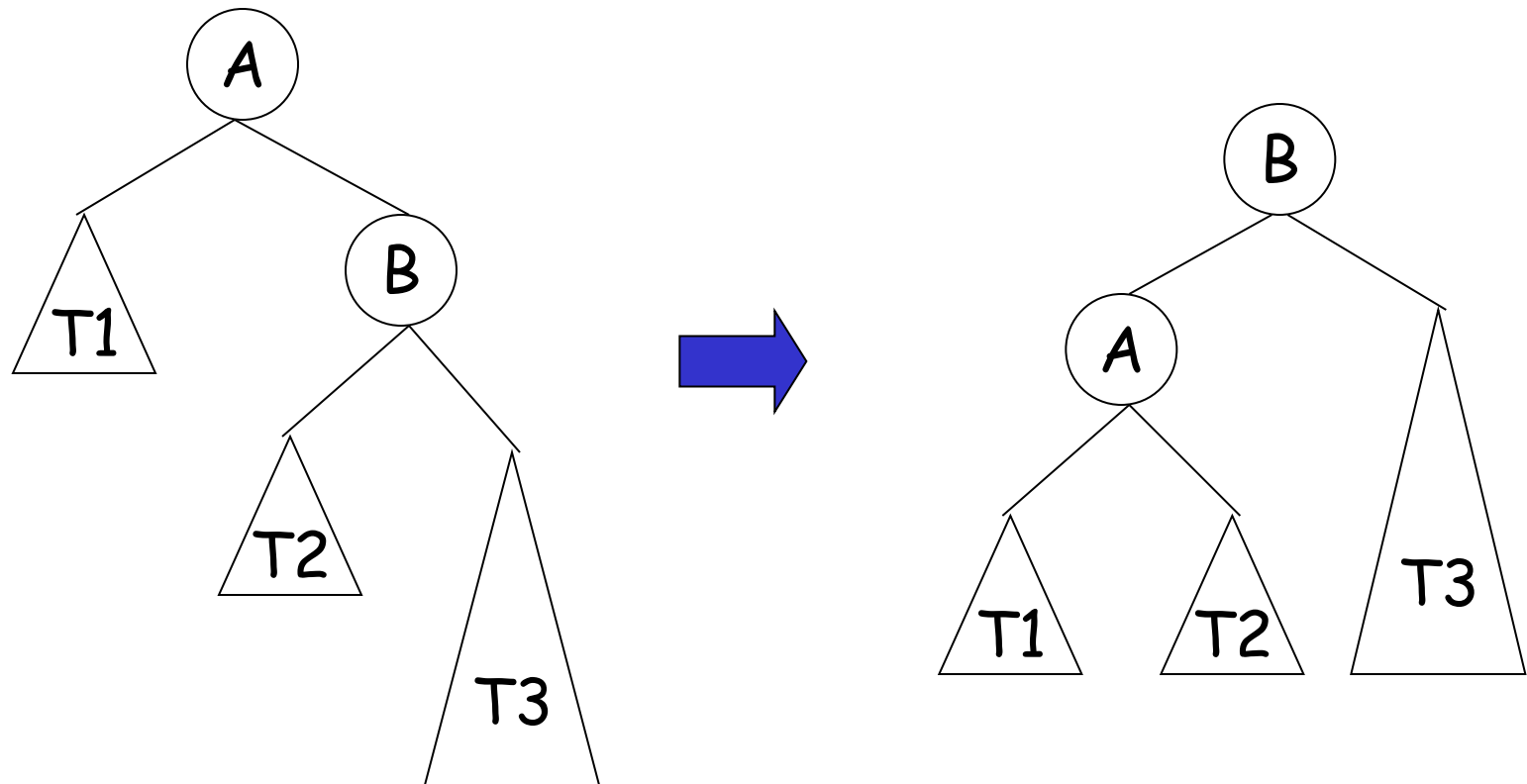
Inserção em Árvores AVL

- Rotação Simples à Direita - Exemplo:



Inserção em Árvores AVL

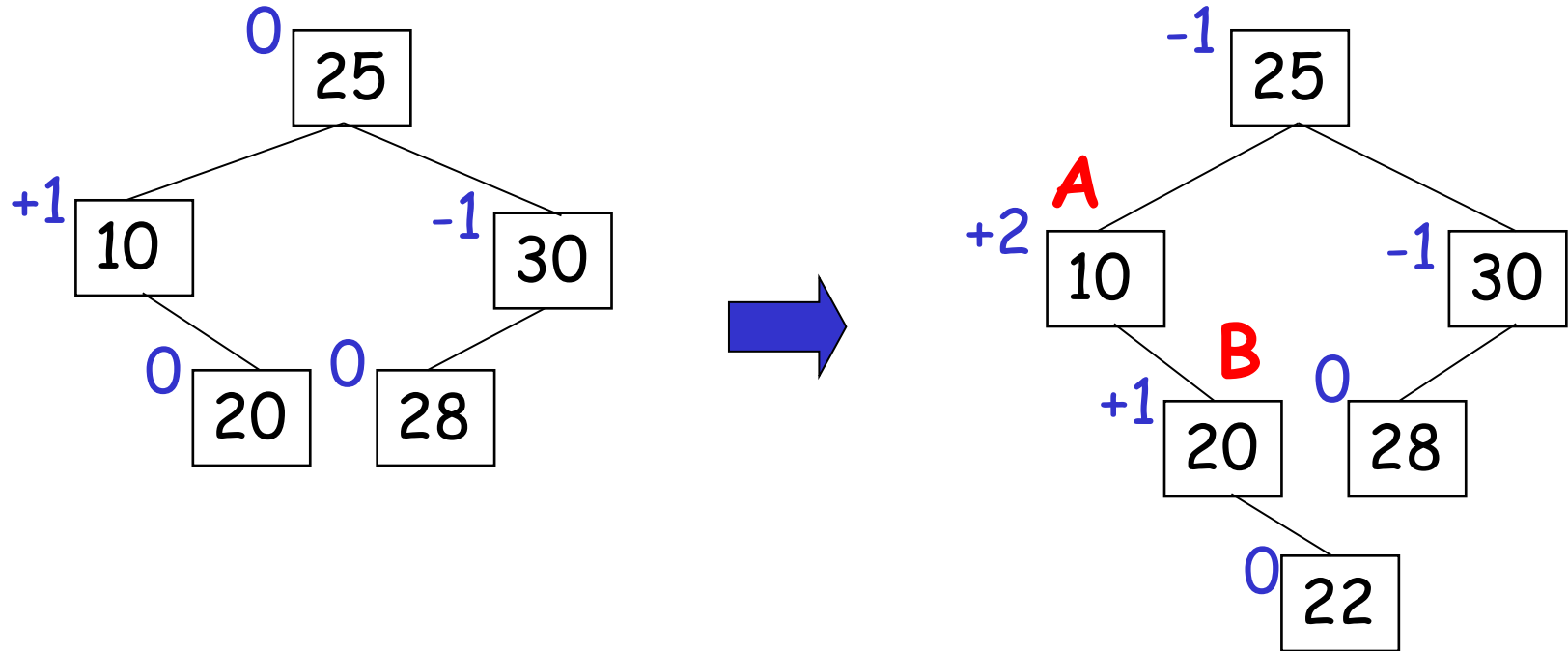
- Rotação Simples à Esquerda



OBS: O nó foi inserido na subárvore T3

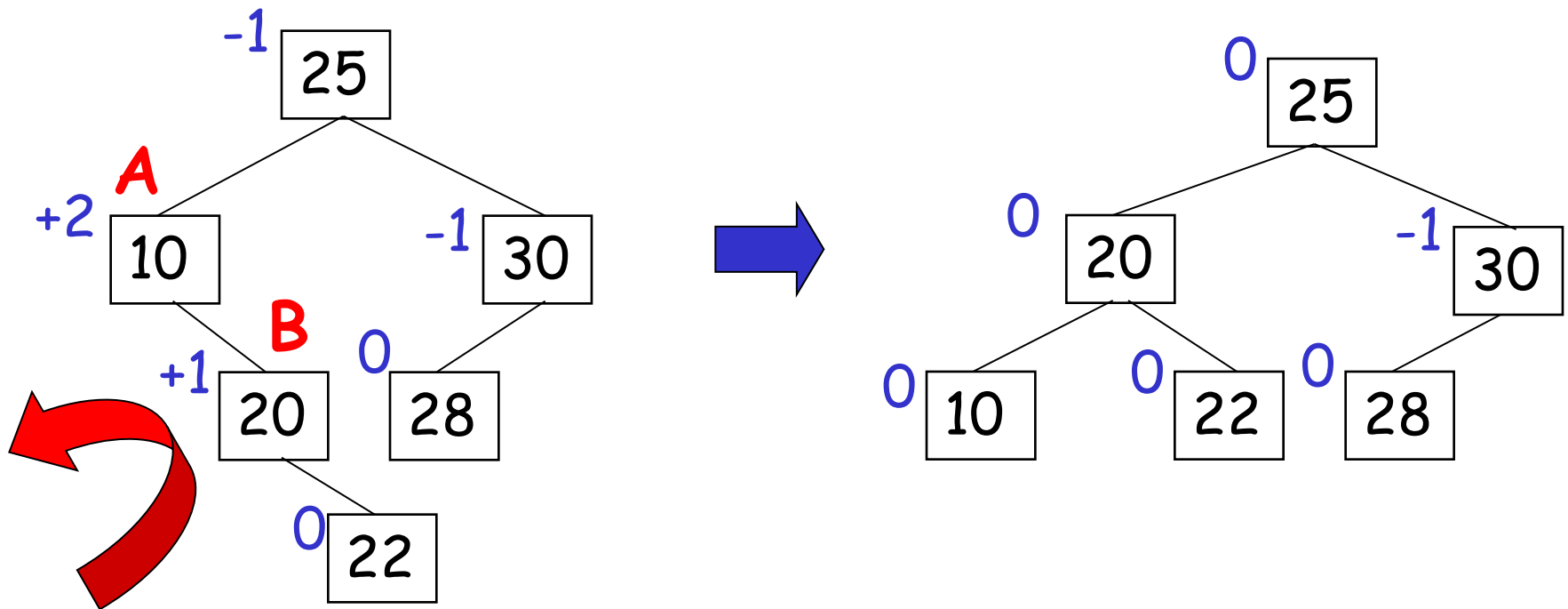
Inserção em Árvores AVL

- Rotação Simples à Esquerda - Exemplo:



Inserção em Árvores AVL

- Rotação Simples à Esquerda - Exemplo:



ANTES

DEPOIS

Inserção em Árvores AVL

- **Rotação Dupla**

- Utilizada quando a inserção ocorre do "*lado de dentro*".
- Dado um nó A , podemos ter:
 - **Caso 3:** Inserção na subárvore à direita do filho à esquerda de $A \Rightarrow$ **Rotação dupla à direita.**
 - **Caso 4:** Inserção na subárvore à esquerda do filho à direita de $A \Rightarrow$ **Rotação dupla à esquerda.**

Inserção em Árvores AVL

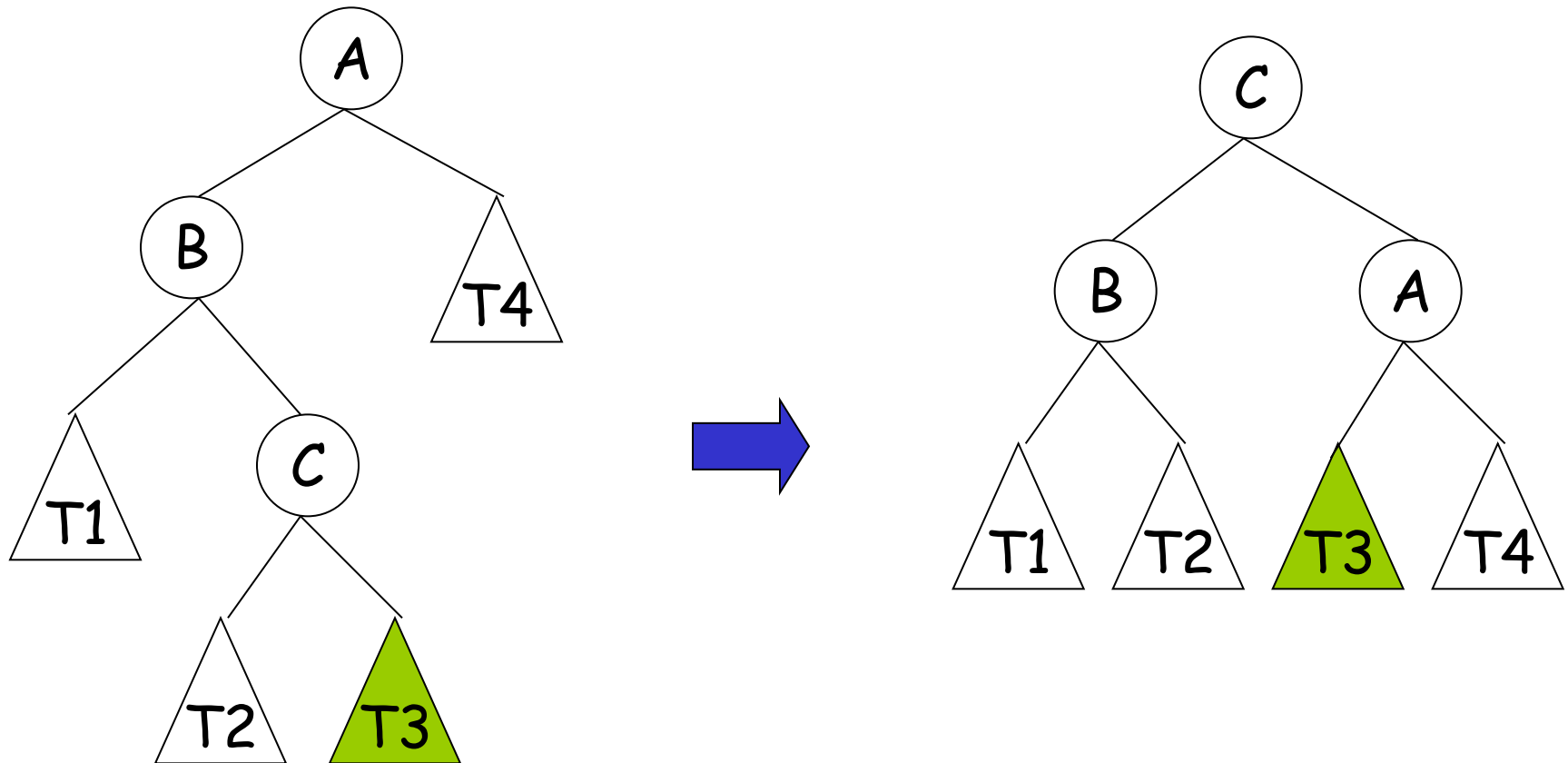
- **Rotação Dupla**

- Principais nós envolvidos:

- **Nó A:** Nó ancestral mais próximo do nó inserido que possuía fator de balanceamento diferente de zero antes da inserção.
- **Nó B:** Filho de A na subárvore onde ocorreu a inserção.
- **Nó C:** Filho de B na subárvore onde ocorreu a inserção.

Inserção em Árvores AVL

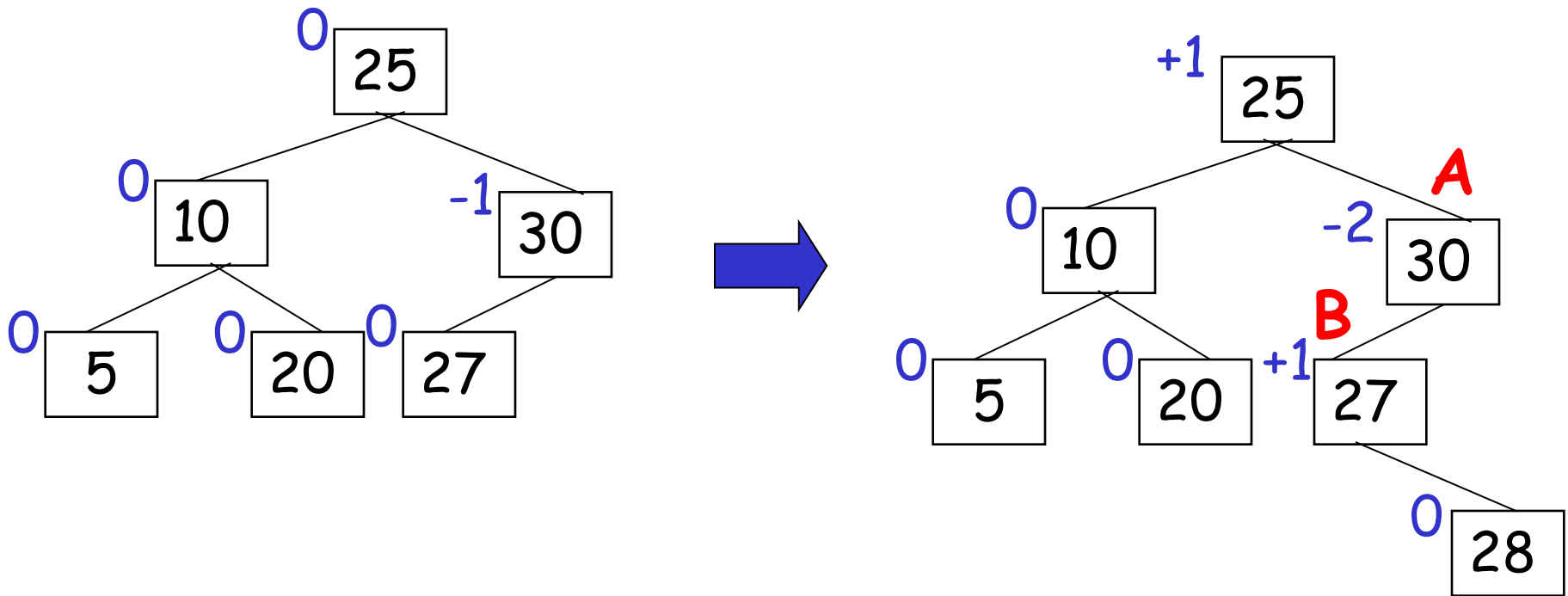
- Rotação Dupla à Direita



OBS: O nó foi inserido na subárvore de raiz C (T2 ou T3)

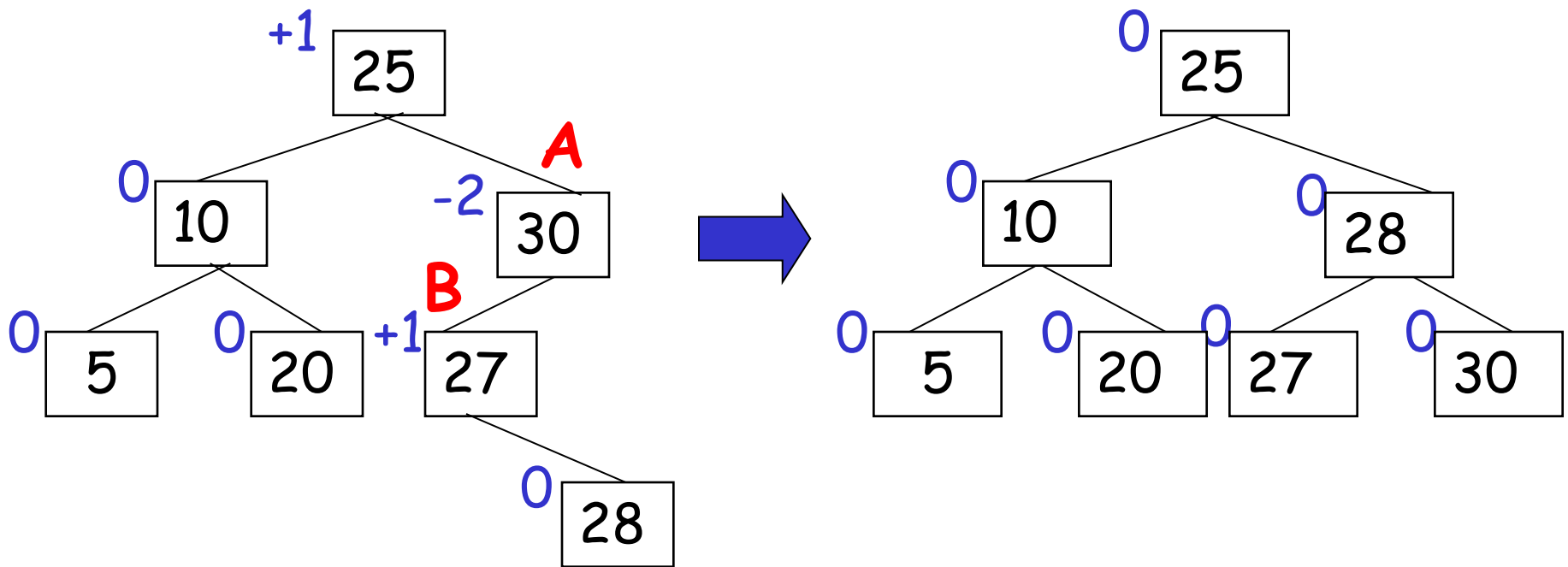
Inserção em Árvores AVL

- Rotação Dupla à Direita - Exemplo



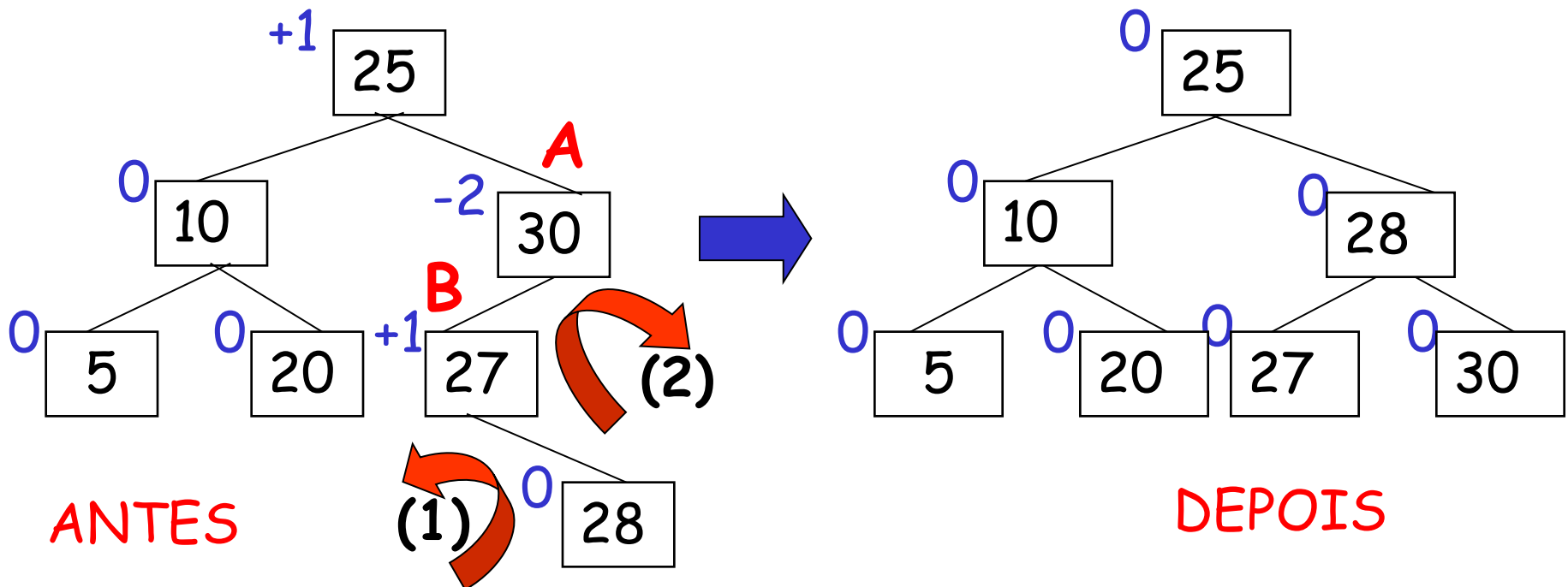
Inserção em Árvores AVL

- Rotação Dupla à Direita - Exemplo



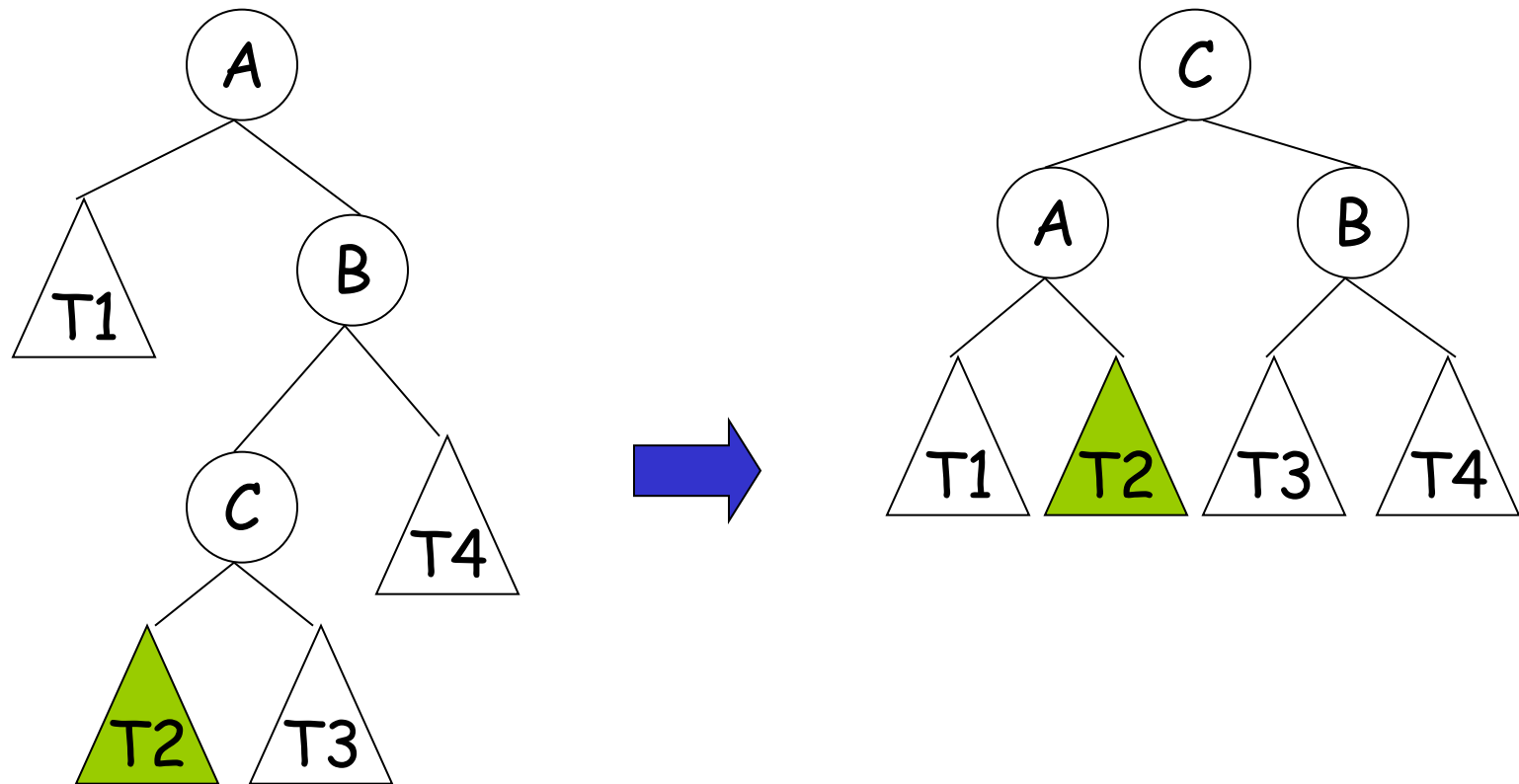
Inserção em Árvores AVL

- **Rotação Dupla à Direita - Observações**
 - Uma rotação dupla à direita é equivalente à:
 - 1) Uma rotação simples à esquerda;
 - 2) Uma rotação simples à direita.



Inserção em Árvores AVL

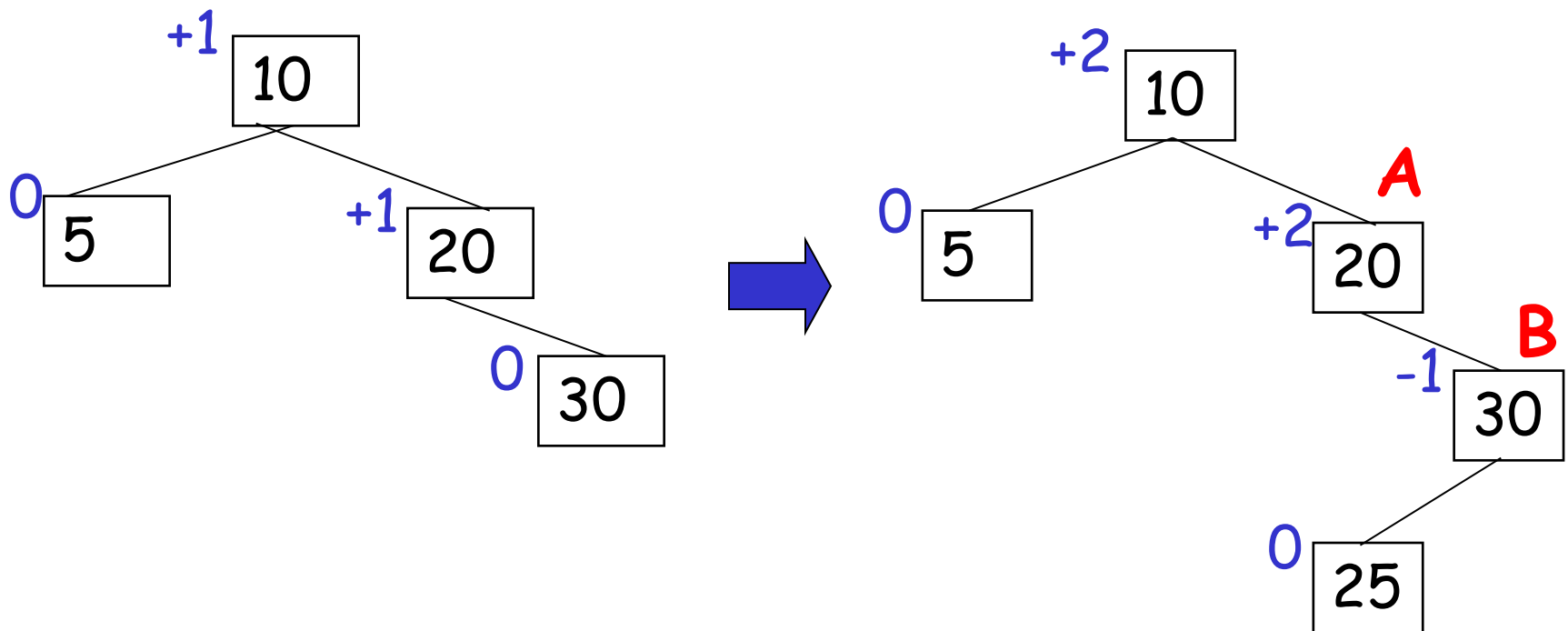
- Rotação Dupla à Esquerda



OBS: O nó foi inserido na subárvore de raiz C (T2 ou T3)

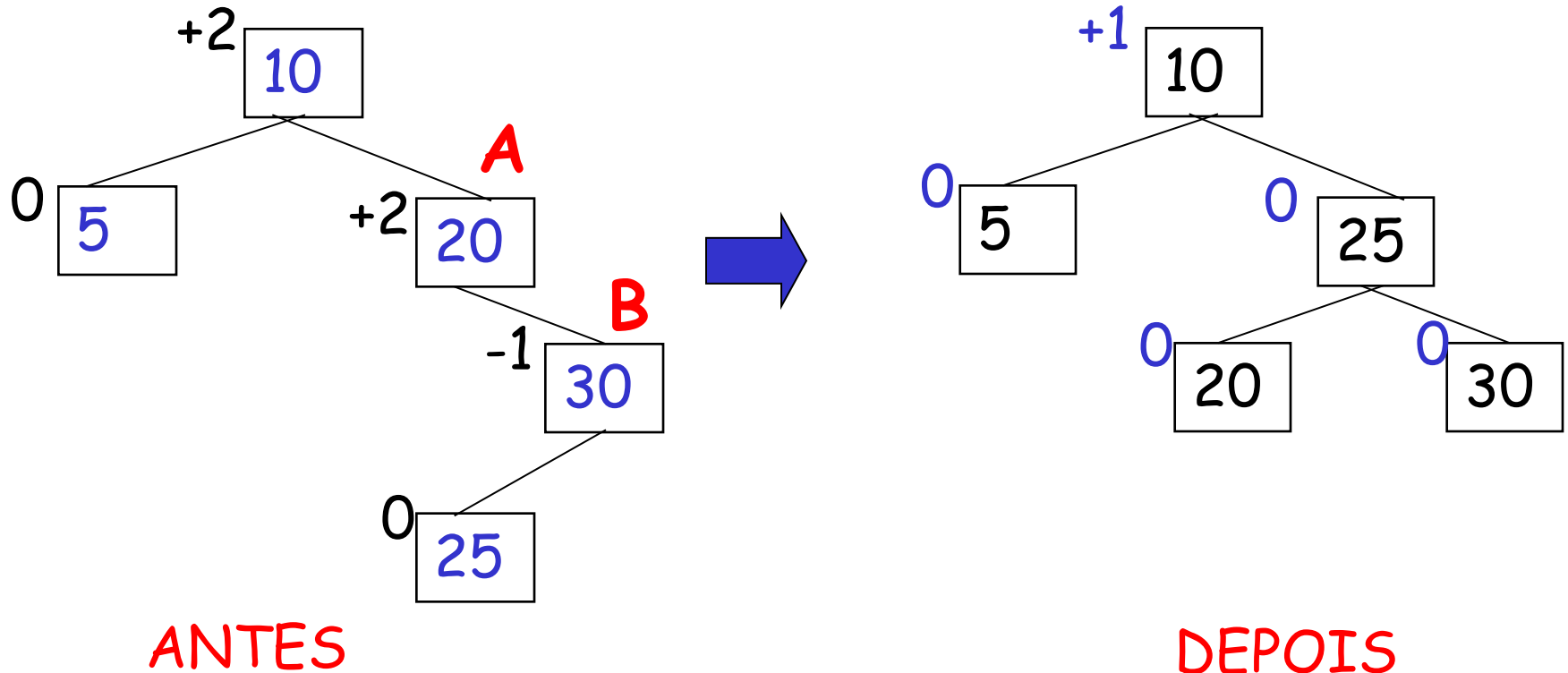
Inserção em Árvores AVL

- Rotação Dupla à Esquerda - Exemplo



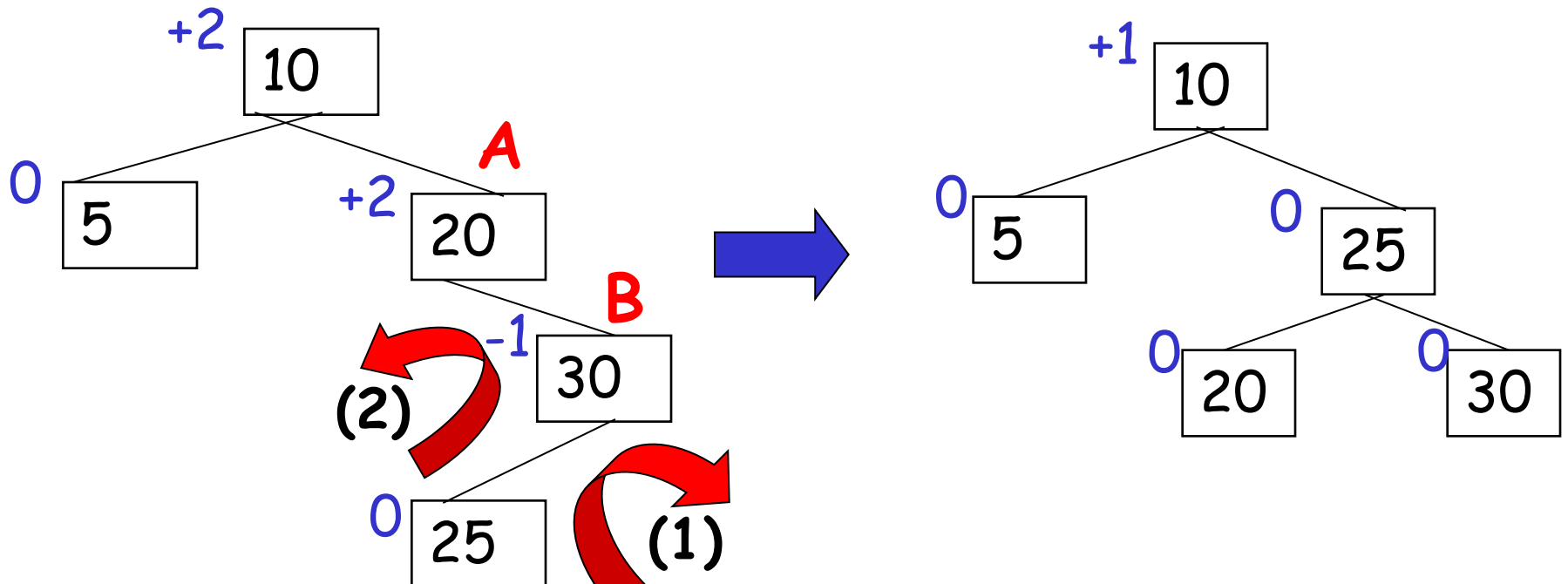
Inserção em Árvores AVL

- Rotação Dupla à Esquerda - Exemplo



Inserção em Árvores AVL

- **Rotação Dupla à Esquerda - Observações**
 - Uma rotação dupla à esquerda é equivalente à:
 - 1) Uma rotação simples à direita;
 - 2) Uma rotação simples à esquerda.



Implementação de Árvores AVL

- Estrutura do nó de uma árvore AVL

```
struct avlNo{  
    int info;  
    avlNo *esq;  
    avlNo *dir;  
    int fatBal;  
};
```

```
//Declaradas dentro do main  
avlNo * raiz = NULL;  
bool status = false;
```

Implementação de Árvores AVL

- Criação de um nó

```
void createNode (avlNo * &raiz, int elem)
{
    raiz = new avlNo;
    raiz->info = elem;
    raiz->esq = NULL;
    raiz->dir = NULL;
    raiz->fatBal = 0;
}
```

```

void insereAvl (avlNo * &raiz, int elem, bool &status) {
    if (raiz == NULL) {
        createNode(raiz,elem);
        status = true;
    } else if (elem == raiz->info){
        printf ("Elemento repetido. \n");
        return;
    } else if (elem < raiz->info){
        insereAvl(raiz->esq,elem,status);
        if (status == true)
            switch (raiz->fatBal) {
                case 1 : raiz->fatBal = 0; status = false; break;
                case 0 : raiz->fatBal = -1; break;
                case -1 : rotacionarDir(raiz,status); break;
            }
    } else {
        insereAvl(raiz->dir,elem,status);
        if (status == true)
            switch (raiz->fatBal) {
                case -1 : raiz->fatBal = 0; status = false; break;
                case 0 : raiz->fatBal = 1; break;
                case 1 : rotacionarEsq(raiz,status); break;
            }
    }
}

```

```

void rotacionarDir (avlNo * &a, bool &status) {
    avlNo *b, *c;
    b = a->esq;
    if (b->fatBal == -1) {                                // rotação simples
        a->esq = b->dir;
        b->dir = a;
        a->fatBal = 0;
        a = b;
    }
    else {                                              // rotação dupla
        c = b->dir;
        b->dir = c->esq;
        c->esq = b;
        a->esq = c->dir;
        c->dir = a;
        if (c->fatBal == -1)
            a->fatBal = 1;
        else a->fatBal = 0;
        if (c->fatBal == 1)
            b->fatBal = -1;
        else b->fatBal = 0;
        a = c;
    }
    a->fatBal = 0;
    status = false;
}

```

```

void rotacionarEsq (avlNo * &a, bool &status) {
    avlNo *b, *c;
    b = a->dir;
    if (b->fatBal == 1) { // rotação simples
        a->dir = b->esq;
        b->esq = a;
        a->fatBal = 0;
        a = b;
    }
    else { // rotação dupla
        c = b->esq;
        b->esq = c->dir;
        c->dir = b;
        a->dir = c->esq;
        c->esq = a;
        if (c->fatBal == 1)
            a->fatBal = -1;
        else a->fatBal = 0;
        if (c->fatBal == -1)
            b->fatBal = 1;
        else b->fatBal = 0;
        a = c;
    }
    a->fatBal = 0;
    status = false;
}

```




Remoção em Árvore AVL

Parte II

Remoção em Árvores AVL

- Considere que o elemento a ser removido encontra-se na raiz de uma árvore T:
 - 1) A raiz não possui filhos: remover a raiz e anular T;
 - 2) A raiz possui um único filho: remover a raiz e substituí-la por seu filho;
 - 3) A raiz possui dois filhos: escolher o nó que armazena o menor elemento na subárvore direita e substituir a raiz por ele.

Remoção em Árvores AVL

- **Pode ocasionar:**
 - A diminuição da altura da árvore, e/ou;
 - A alteração dos fatores de balanceamento de seus nós.

Remoção em Árvores AVL

- **Algoritmo de Remoção**
 - Efetuar a remoção;
 - Ajustar os fatores de balanceamento;
 - Verificar a quebra do equilíbrio;
 - Se a árvore não estiver balanceada, corrigir a estrutura através de movimentações dos nós (**Rotações**).

Remoção em Árvores AVL

- Rotação

- Tipos de Rotação

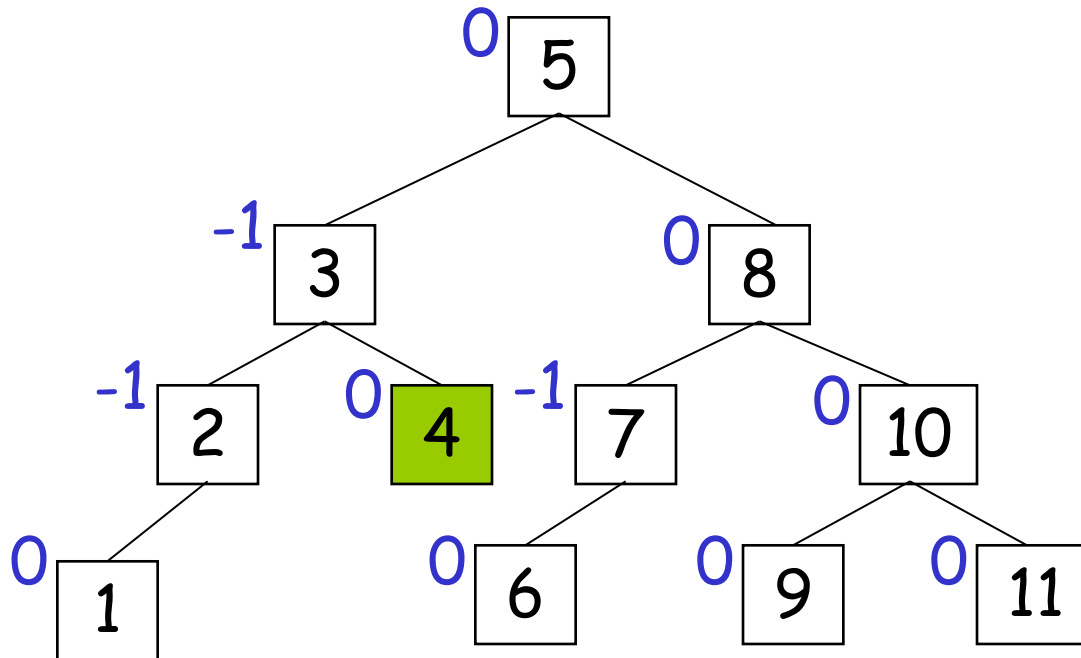
- Simples
- Dupla

- Principais nós envolvidos:

- **Nó A**: Nó ancestral mais próximo do ponto de remoção que possuía fator de balanceamento diferente de zero antes da remoção ser efetuada.
- **Nó B**: Filho de A na subárvore de maior altura.
- **Nó C**: Filho de B na subárvore de maior altura (utilizado apenas em rotações duplas).

Remoção em Árvores AVL

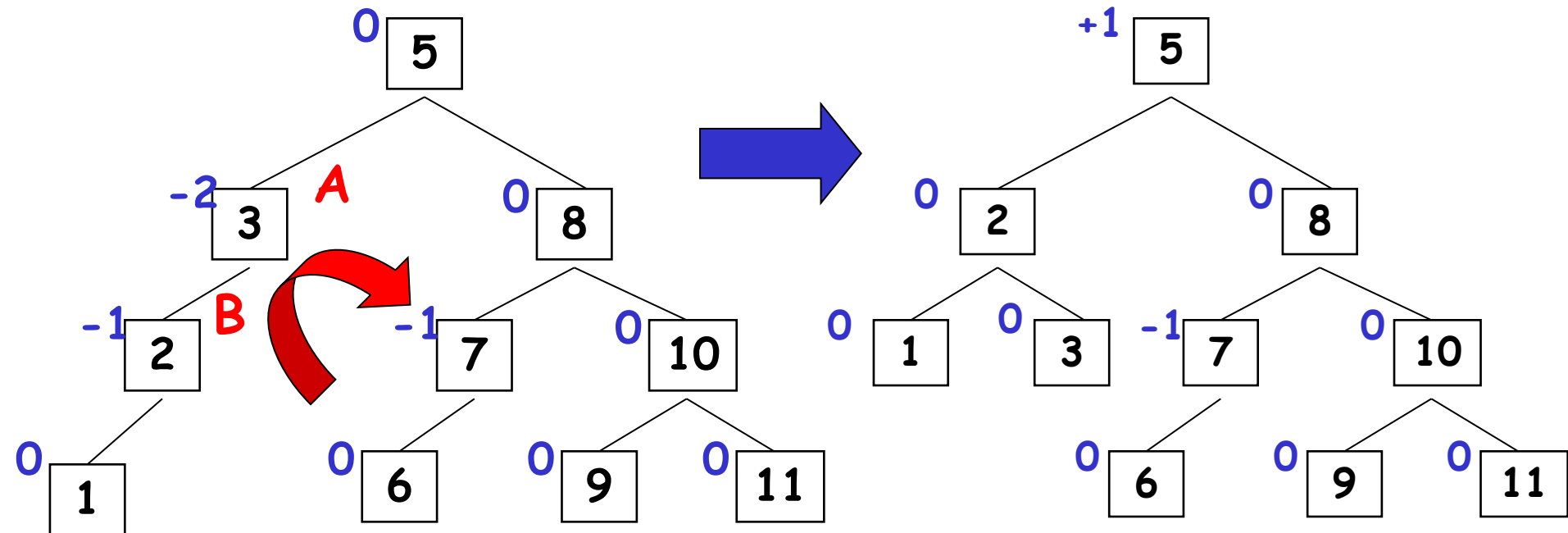
- Exemplo:



Remove 4

Remoção em Árvores AVL

- Exemplo:

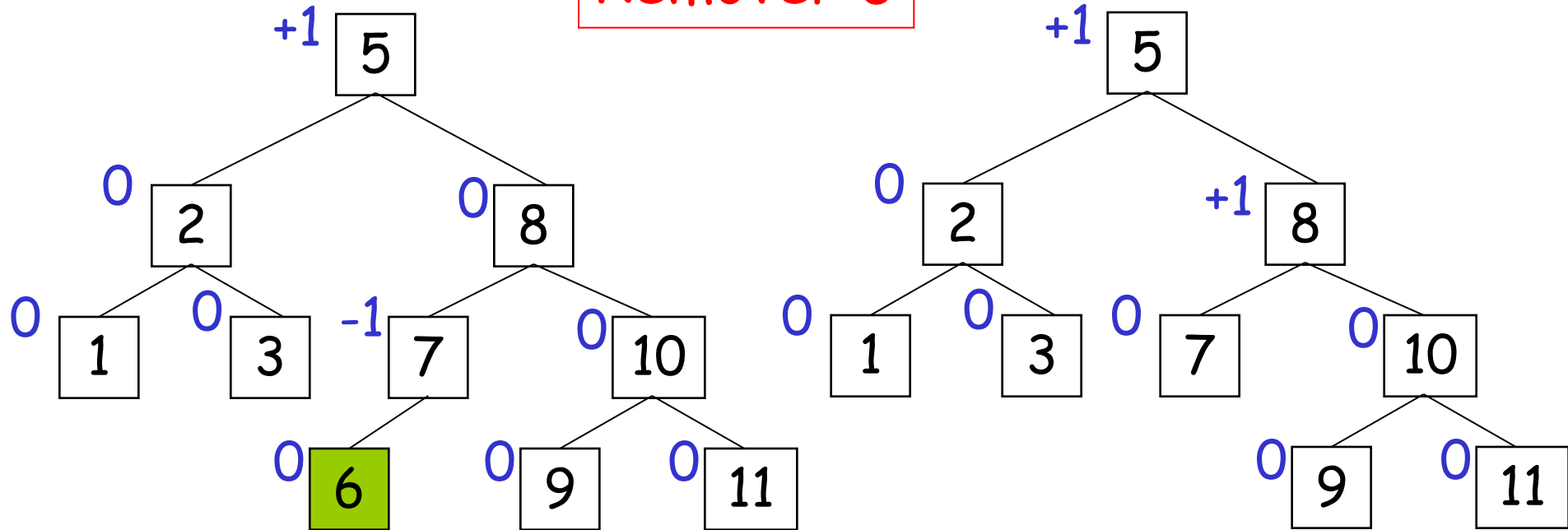


Rebalancear \Rightarrow Rotação Simples à Direita

Remoção em Árvores AVL

- Exemplo:

Remove 6



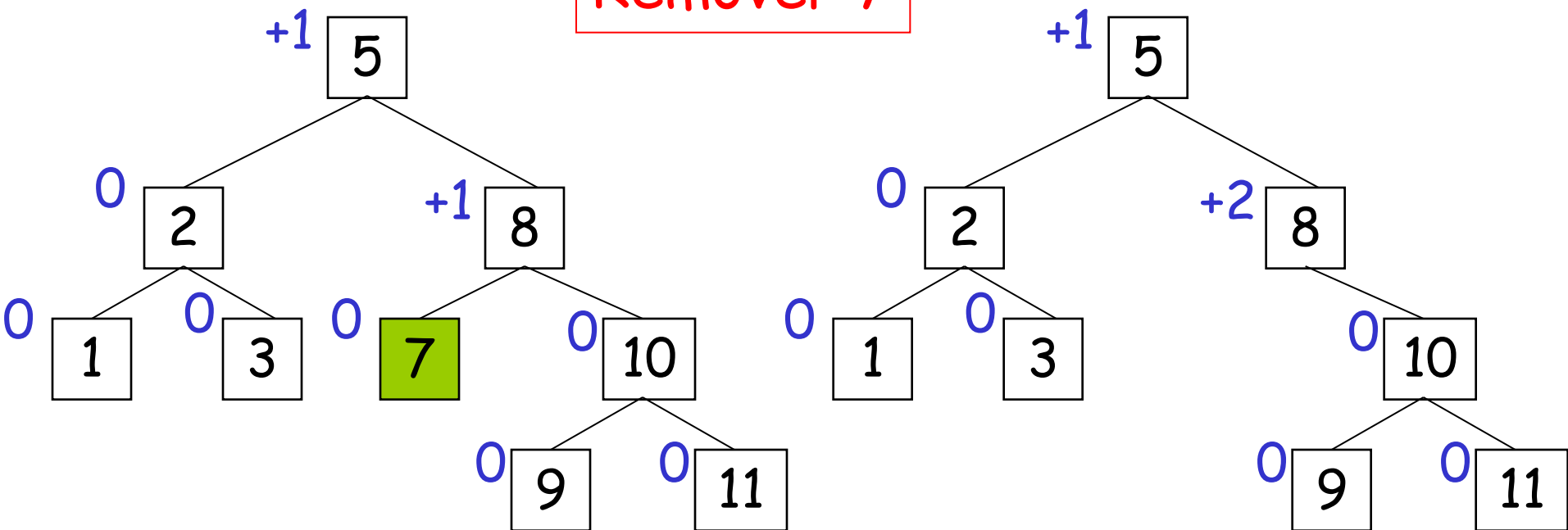
ANTES

DEPOIS

Remoção em Árvores AVL

- Exemplo:

Remove 7

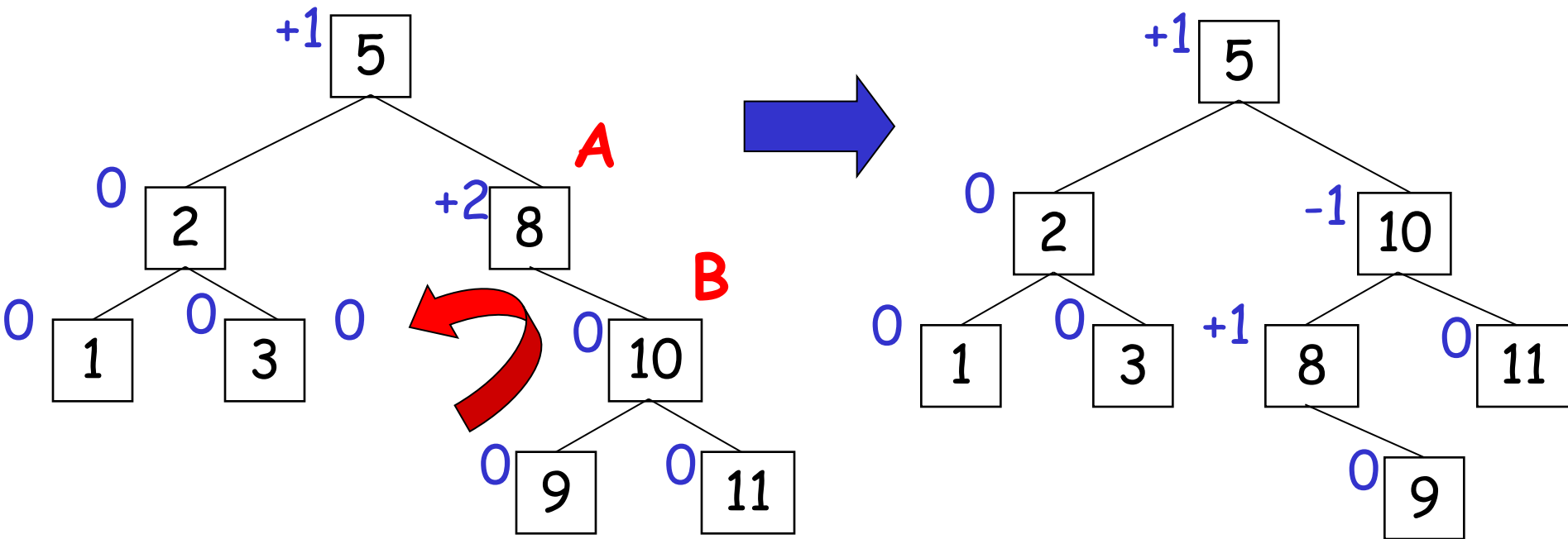


ANTES

DEPOIS

Remoção em Árvores AVL

- Exemplo:

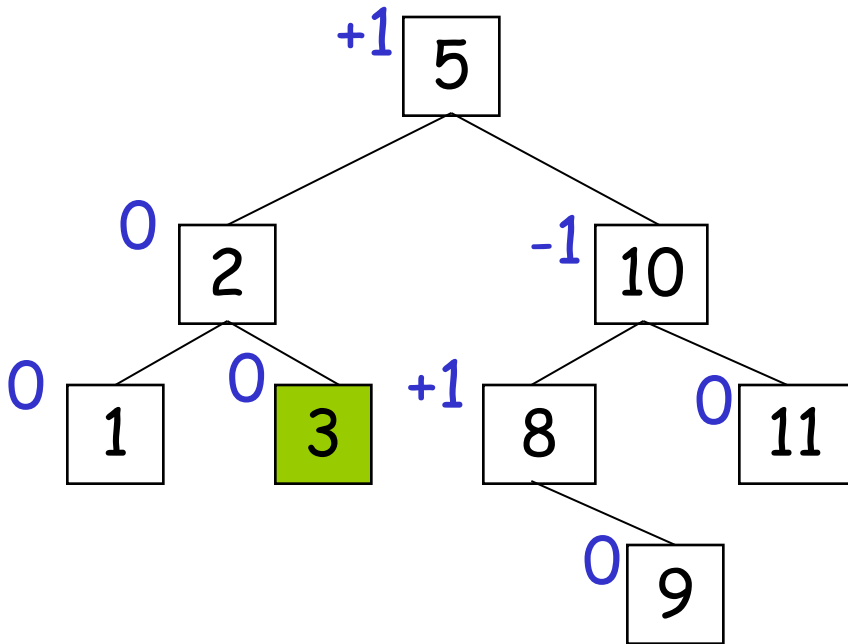


Rebalancear \Rightarrow Rotação Simples à Esquerda

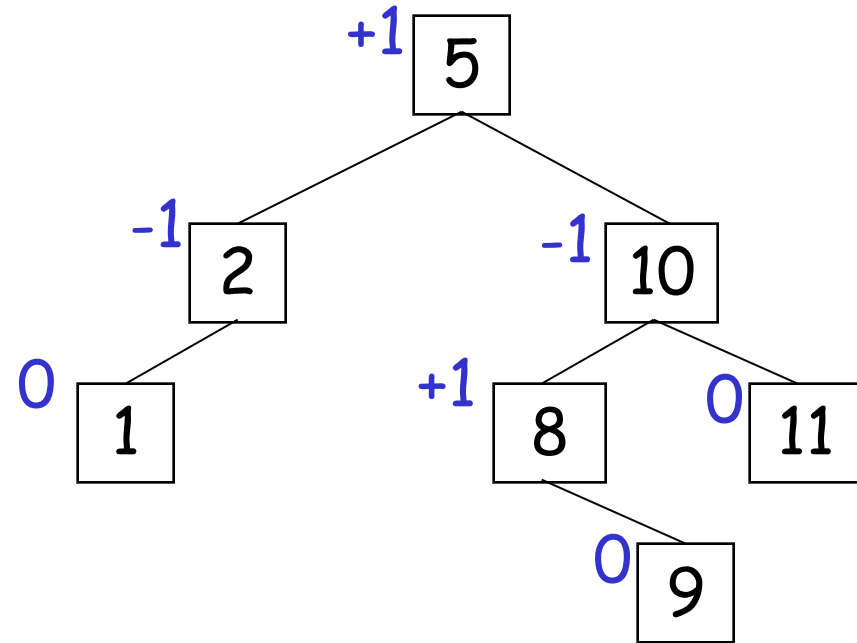
Remoção em Árvores AVL

- Exemplo:

Remove 3



ANTES

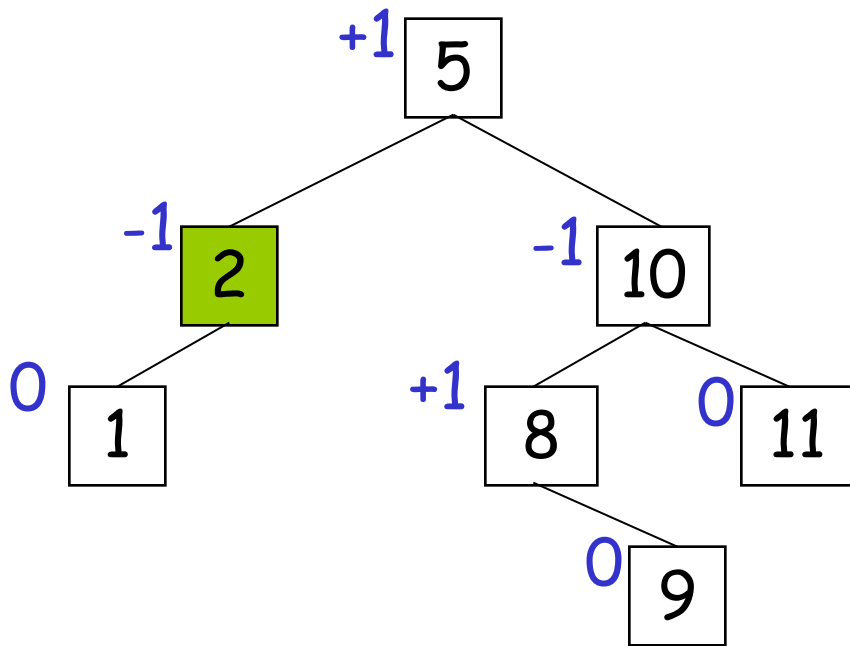


DEPOIS

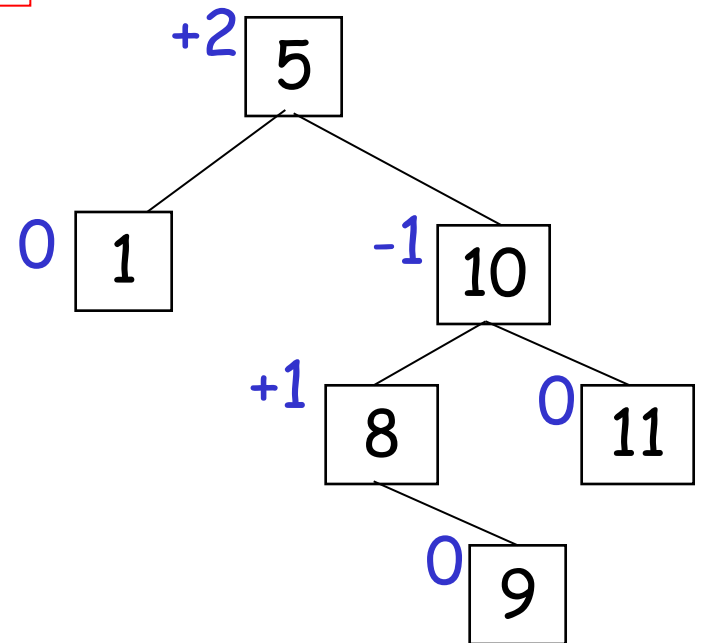
Remoção em Árvores AVL

- Exemplo:

Remove 2



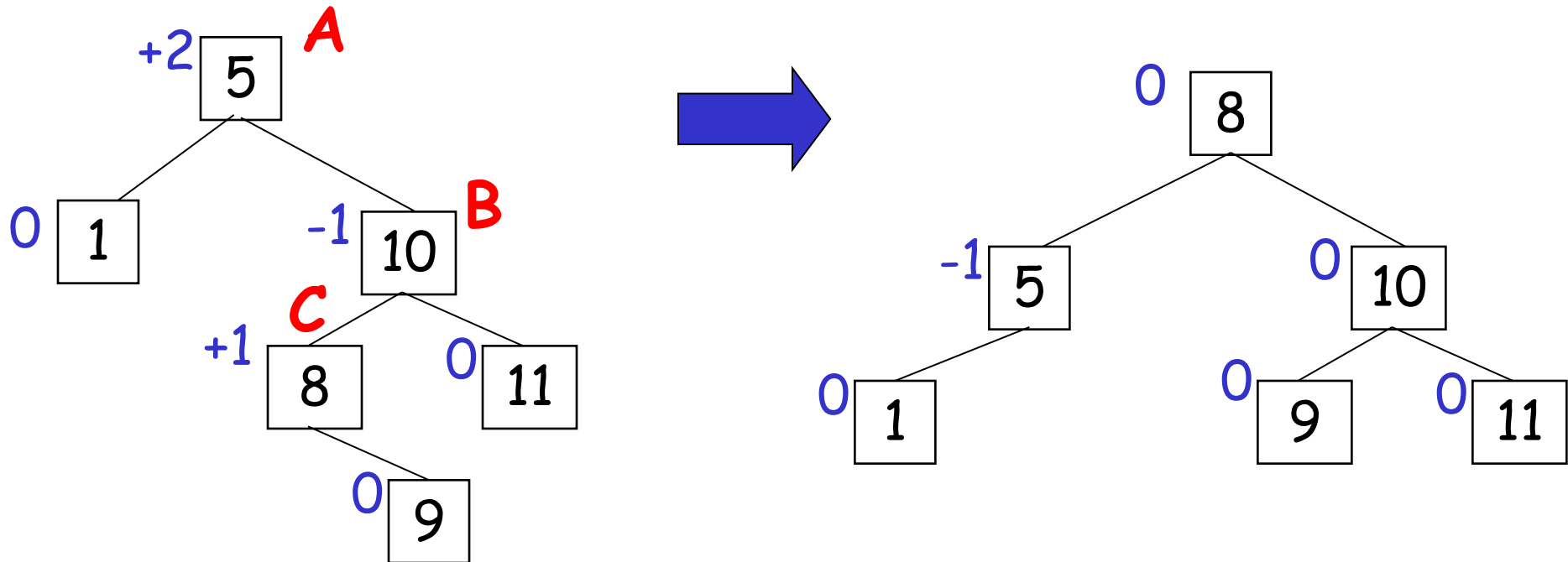
ANTES



DEPOIS

Remoção em Árvores AVL

- Exemplo:

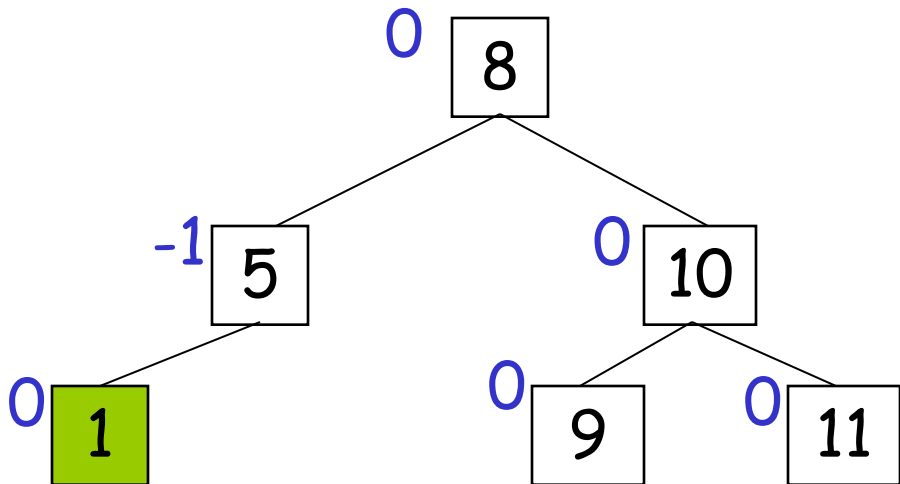


Rebalancear \Rightarrow Rotação Dupla à Esquerda

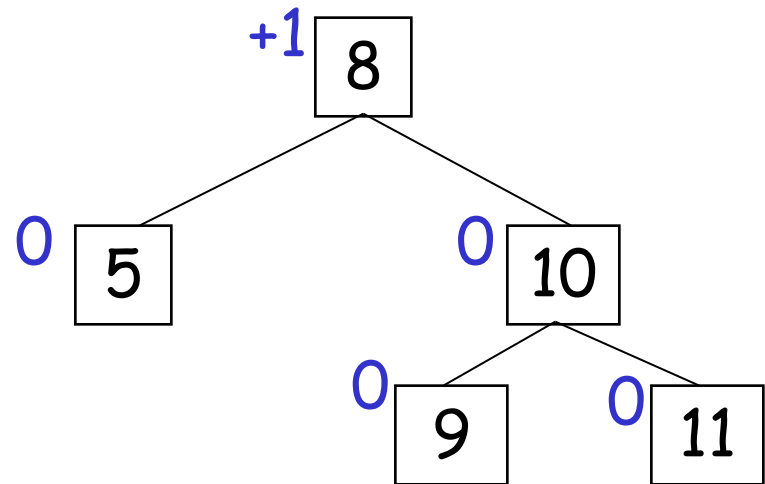
Remoção em Árvores AVL

- Exemplo:

Remover 1



ANTES

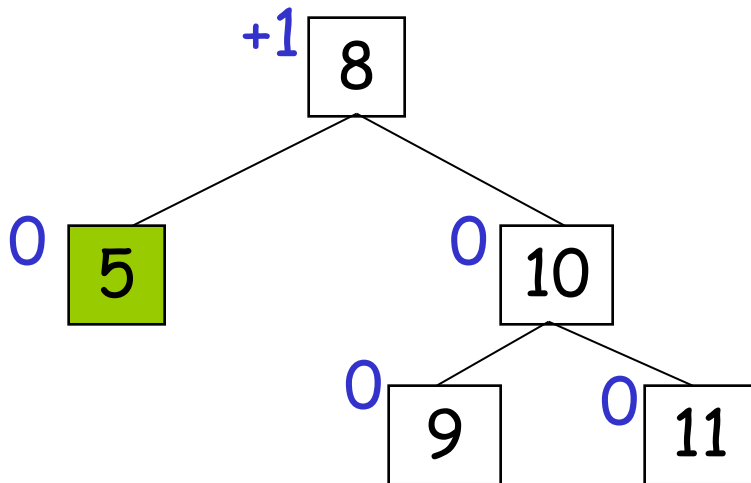


DEPOIS

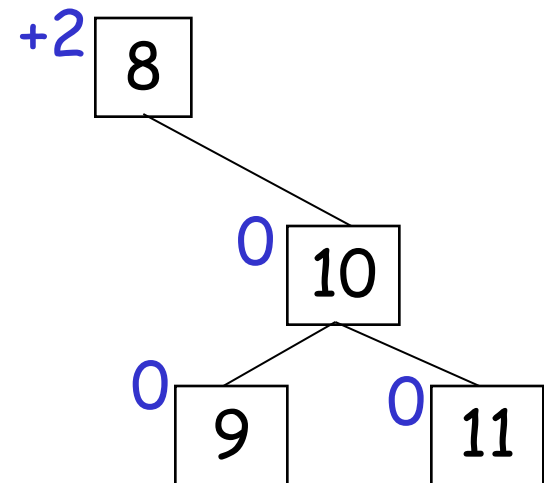
Remoção em Árvores AVL

- Exemplo:

Remove 5



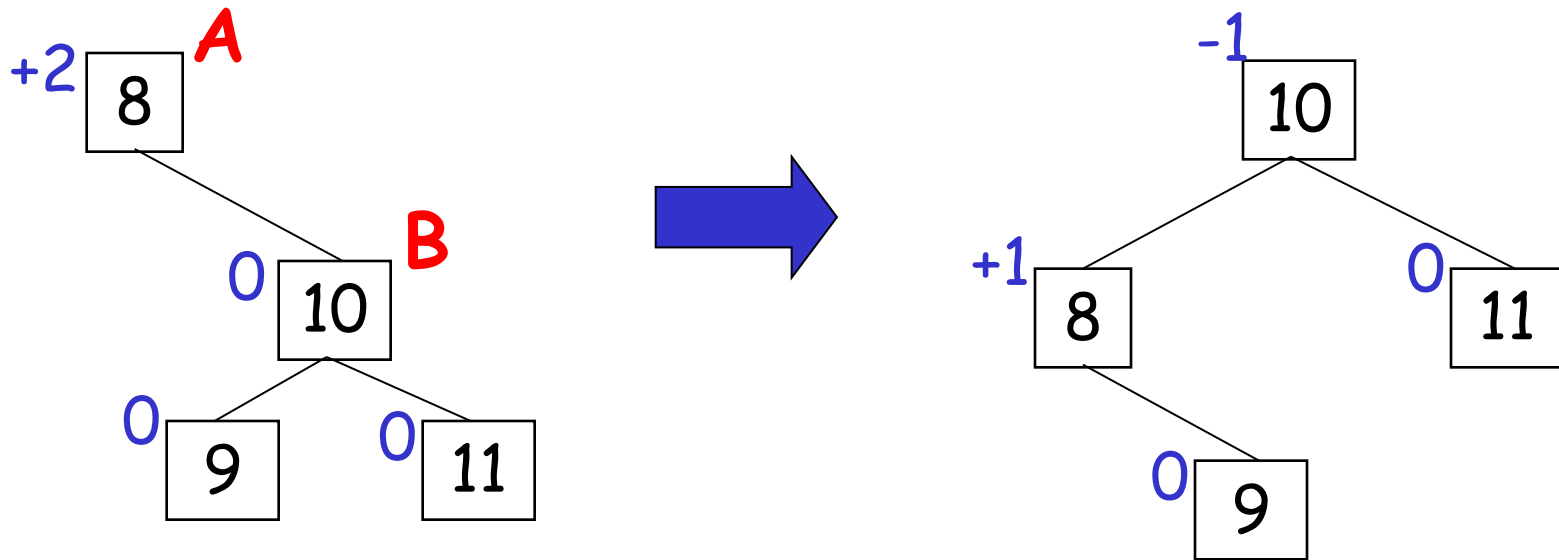
ANTES



DEPOIS

Remoção em Árvores AVL

- Exemplo:



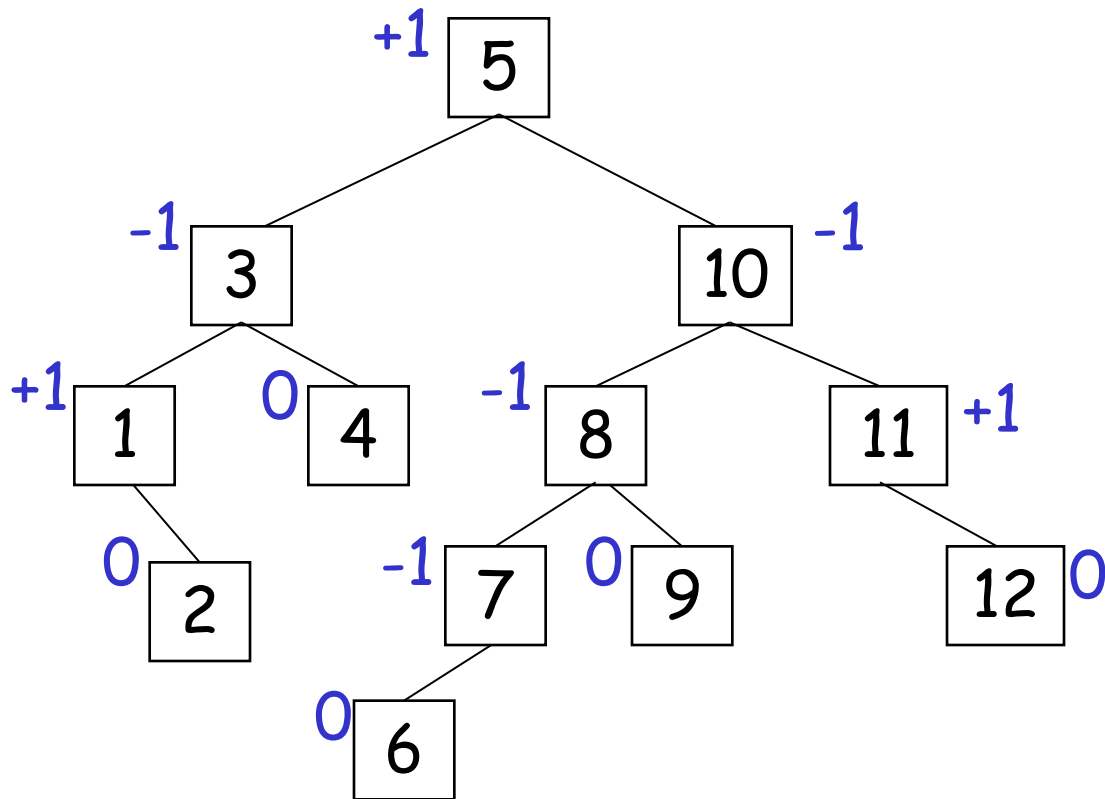
Rebalancear \Rightarrow Rotação Simples à Esquerda

Remoção em Árvores AVL

- **Observação:**
 - Enquanto a **inserção** de uma única chave pode resultar em uma rotação de **dois ou três nós no máximo**, a **remoção** pode exigir uma rotação em cada um dos nós **ao longo de toda a trajetória de busca**.
 - Ver exercício a seguir.

Remoção em Árvores AVL

- **Exercício:** Remover 4



Exercício

- Dada uma árvore AVL inicialmente vazia:
 - Insira: 50, 70, 90, 30, 10, 40, 80, 60, 75, 35, 100, 73
 - Remova: 10, 70, e depois as raízes das árvores resultantes até obter novamente uma árvore vazia

Bibliografia Recomendada

- SZWARCFITER, Jayme Luiz; MARKENZON, Lilian. **Estruturas de Dados e seus Algoritmos**. 2. ed. rev. Rio de Janeiro: LTC, 1994.
- WIRTH, Niklaus. **Algoritmos e estruturas de dados**. 1. ed. Rio de Janeiro: LTC, 1999.