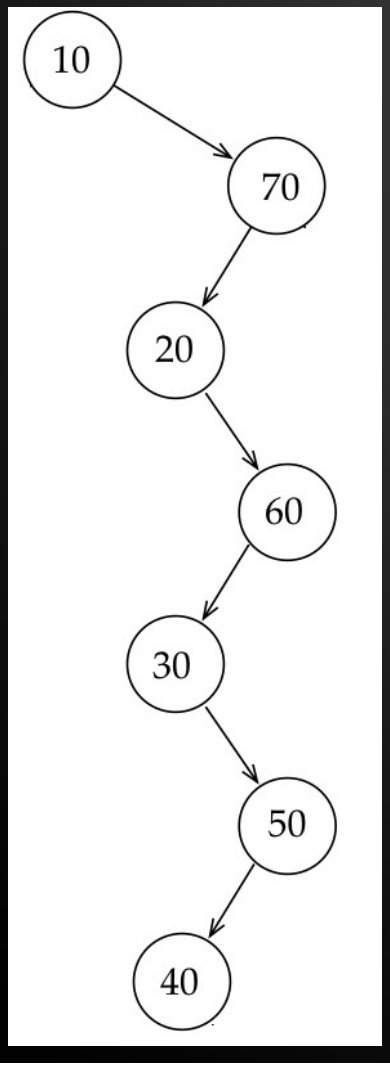
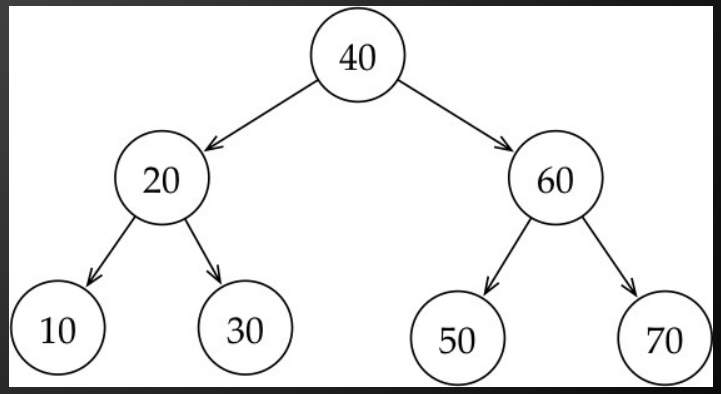

ESTRUTURA
DE DADOS
ÁRVORE
AVL

ÁRVORE BINÁRIA DE BUSCA



40, 20, 10, 60, 30, 50, 70



10, 70, 20, 60, 30, 50, 40

Professor
Douglas Maioli

ÁRVORE AVL

Árvore AVL é uma árvore binária de busca balanceada, que tenta minimizar o número de comparações efetuadas no pior caso para uma busca. Contudo, para garantir essa propriedade é preciso reconstruir a árvore para seu estado ideal a cada operação sobre seus nós (inclusão ou exclusão), o que causa um aumento no tempo computacional.

O nome AVL vem de seus criadores soviéticos Adelson, Velsky e Landis, e sua primeira referência encontra-se no documento *"Algoritmos para organização da informação"* de 1962.

FATOR DE BALANCEAMENTO

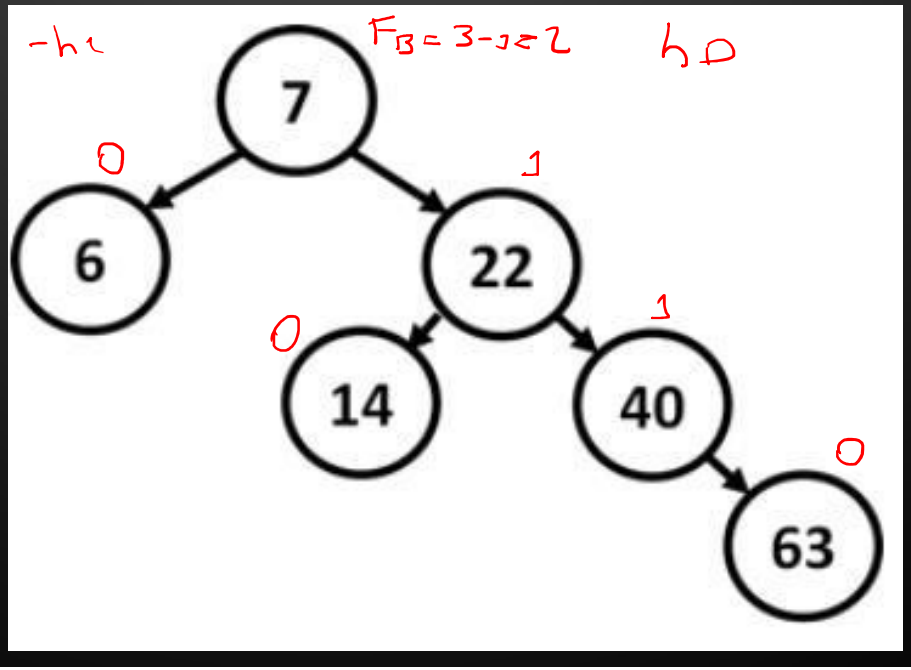
O fator de balanceamento de um nó em uma Árvore Binária é a diferença de altura entre as subárvores da direita e da esquerda, ou seja, é dado por:

$$F_B = h_D - h_E$$

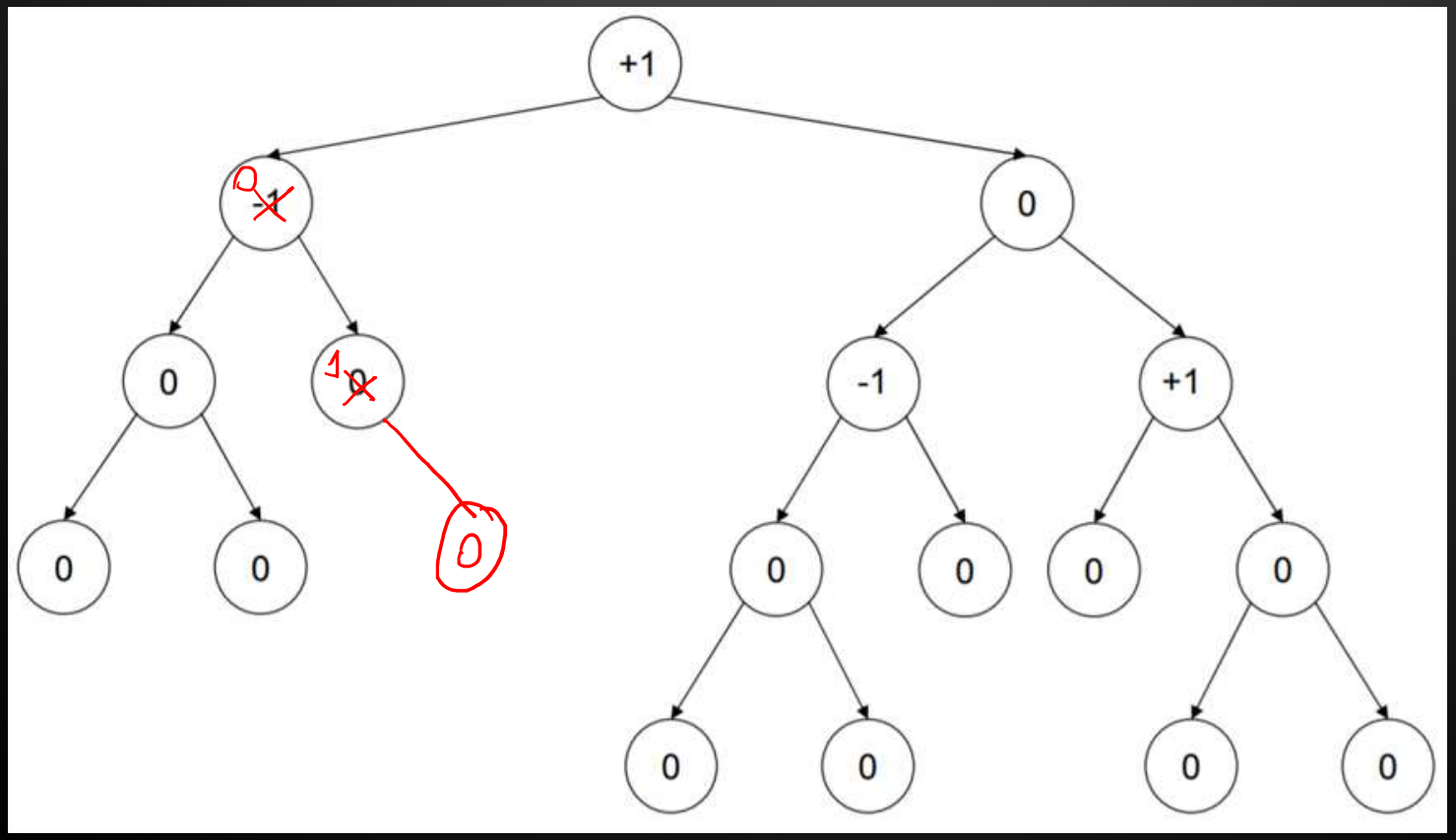
Onde h_D é a altura da subárvore à direita do nó e h_E é a altura da subárvore à esquerda.

Numa Árvore AVL devemos ter que o fator de balanceamento de cada nó deve estar entre -1 e 1.

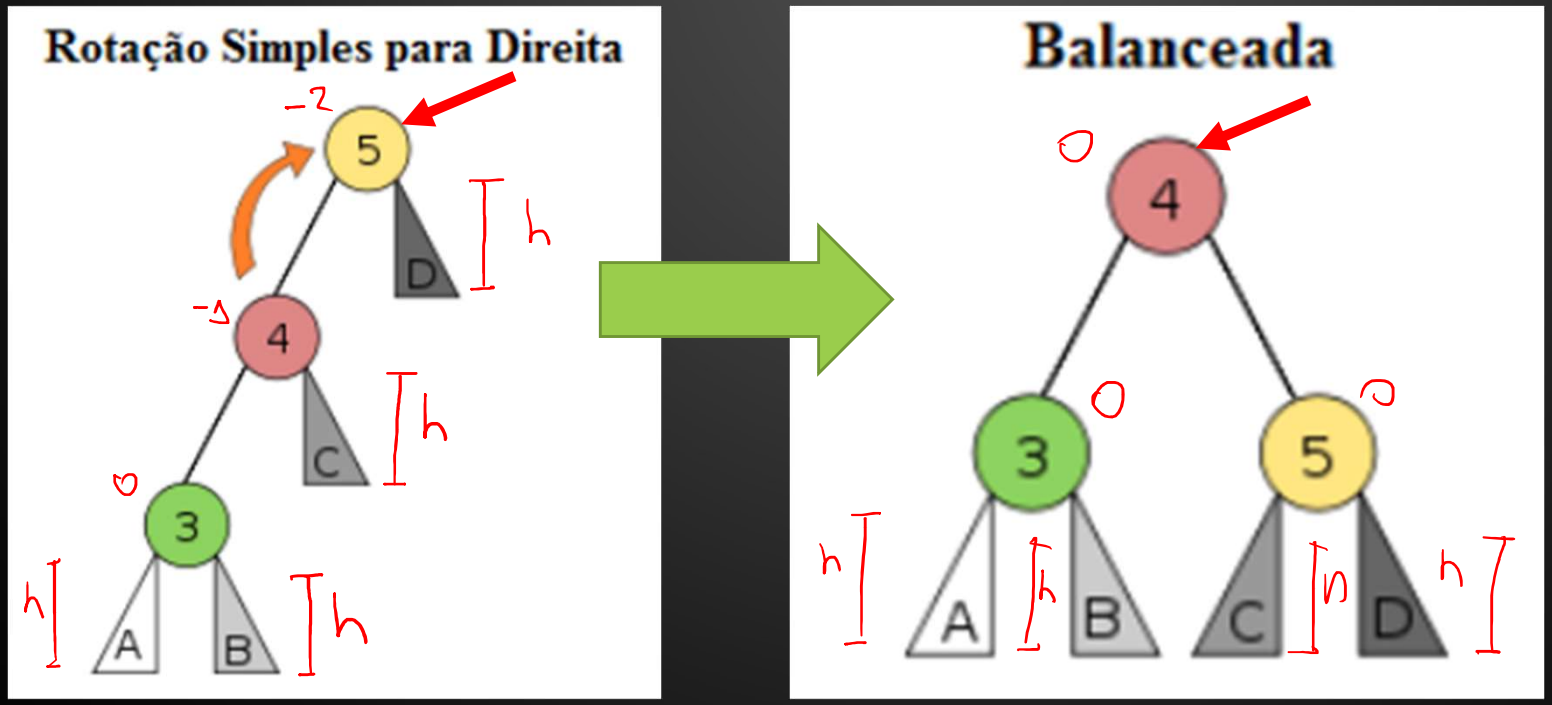
$$|h_D - h_E| \leq 1$$



FATOR DE BALANCEAMENTO

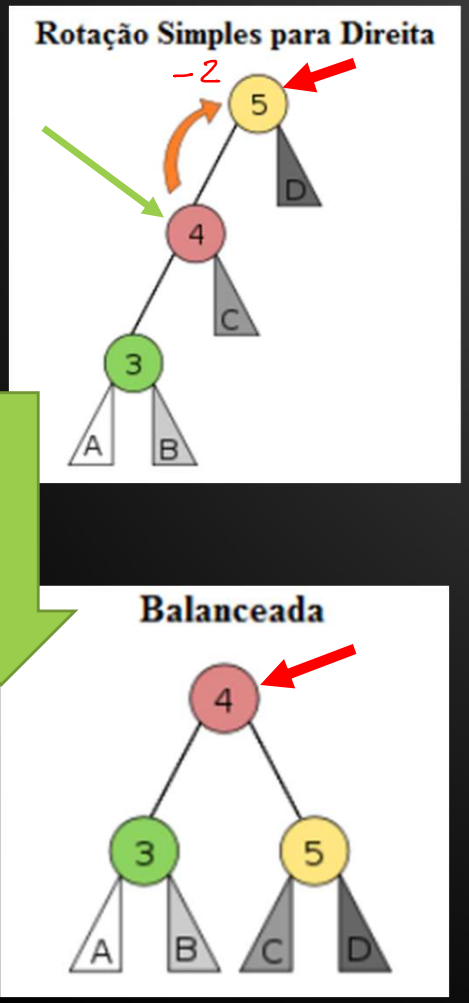


ROTAÇÃO PARA A DIREITA



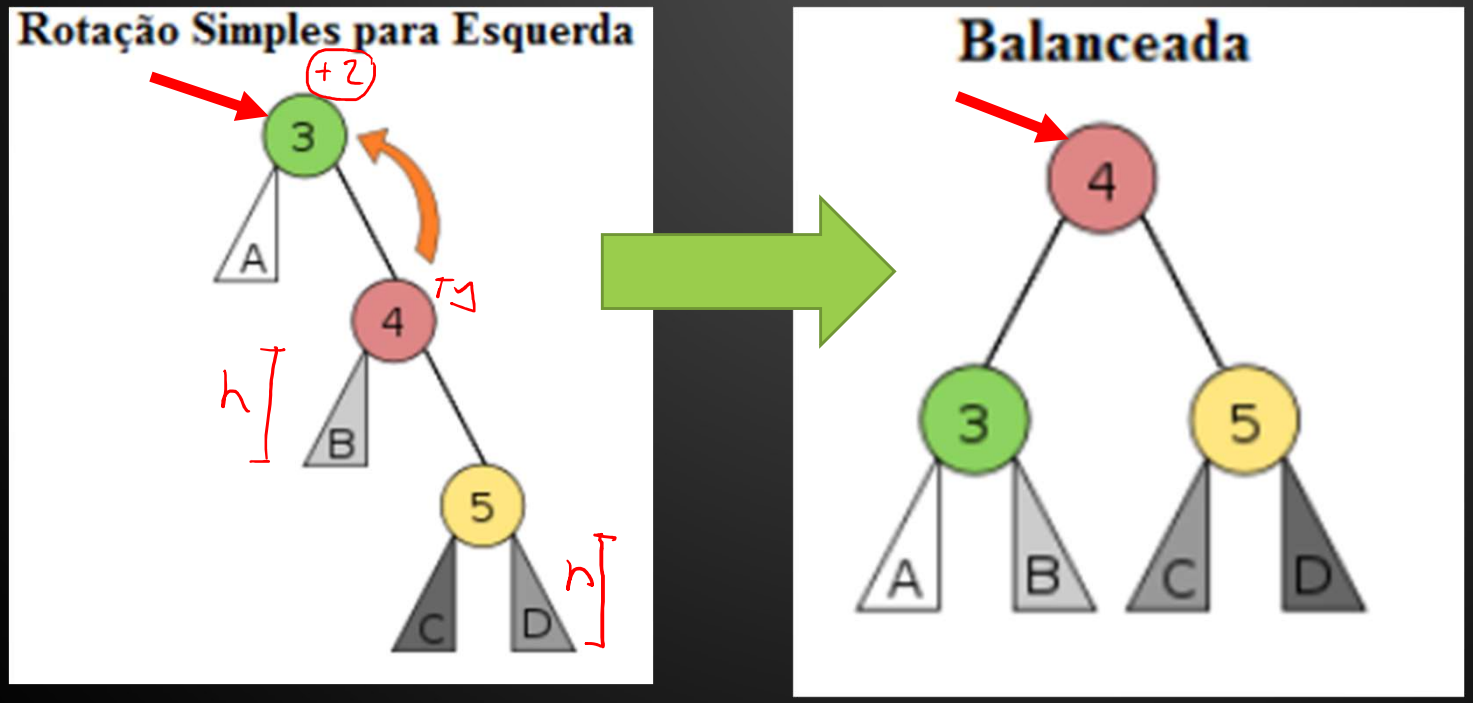
Professor
Douglas Maioli

ROTAÇÃO PARA A DIREITA



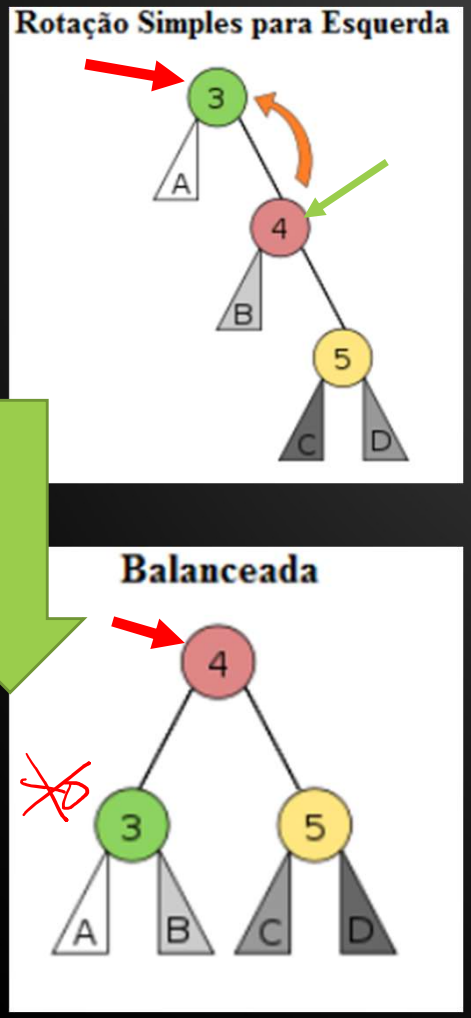

```
void ArvoreAVL::rotacaodireita(No*& pai)
{
    No* novopai = pai->filhoesquerda;
    pai->filhoesquerda = novopai->filhodireita;
    novopai->filhodireita = pai;
    pai = novopai;
}
```

ROTAÇÃO PARA A ESQUERDA



Professor
Douglas Maioli

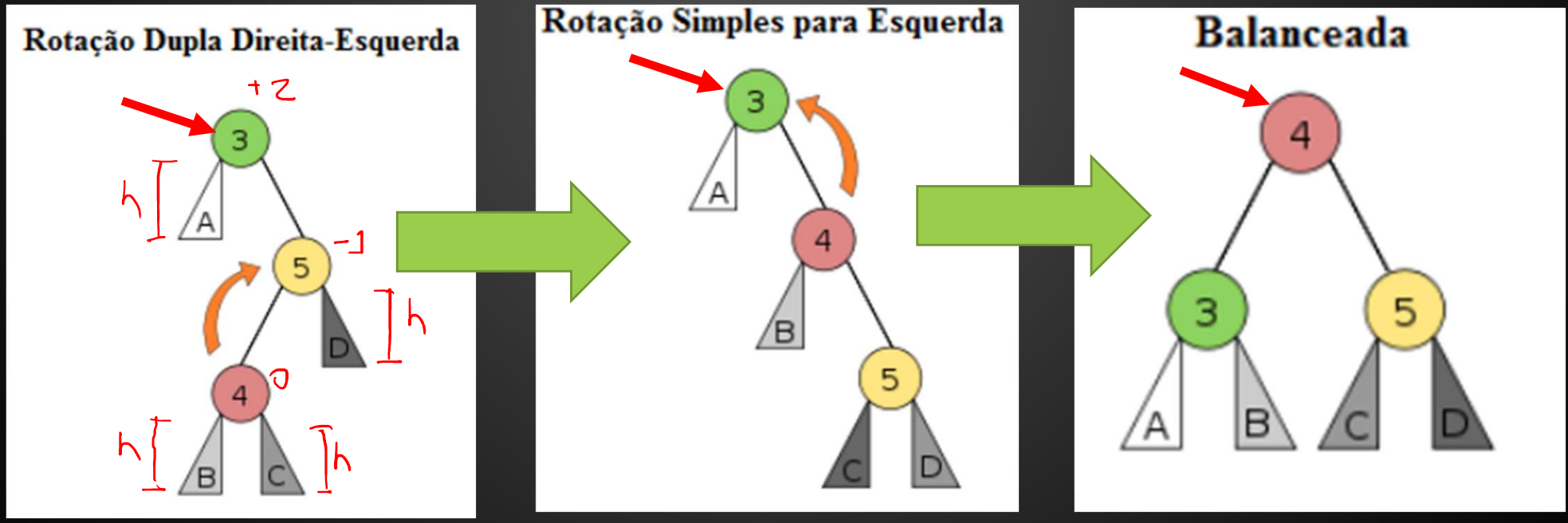
ROTAÇÃO PARA A ESQUERDA



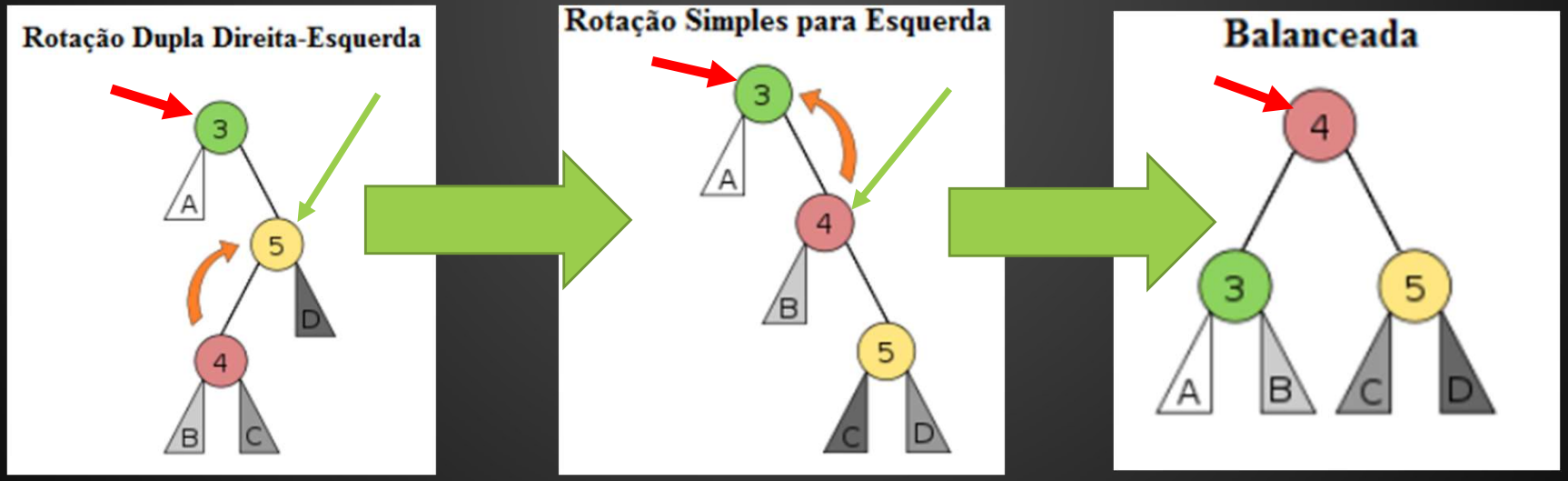

```
void ArvoreAVL::rotacaoesquerda(No*& pai)
{
    No* novopai = pai->filhodireita;
    pai->filhodireita = novopai->filhoesquerda;
    novopai->filhoesquerda = pai;
    pai = novopai;
}
```

Professor
Douglas Maioli

ROTAÇÃO DUPLA DIREITA-ESQUERDA

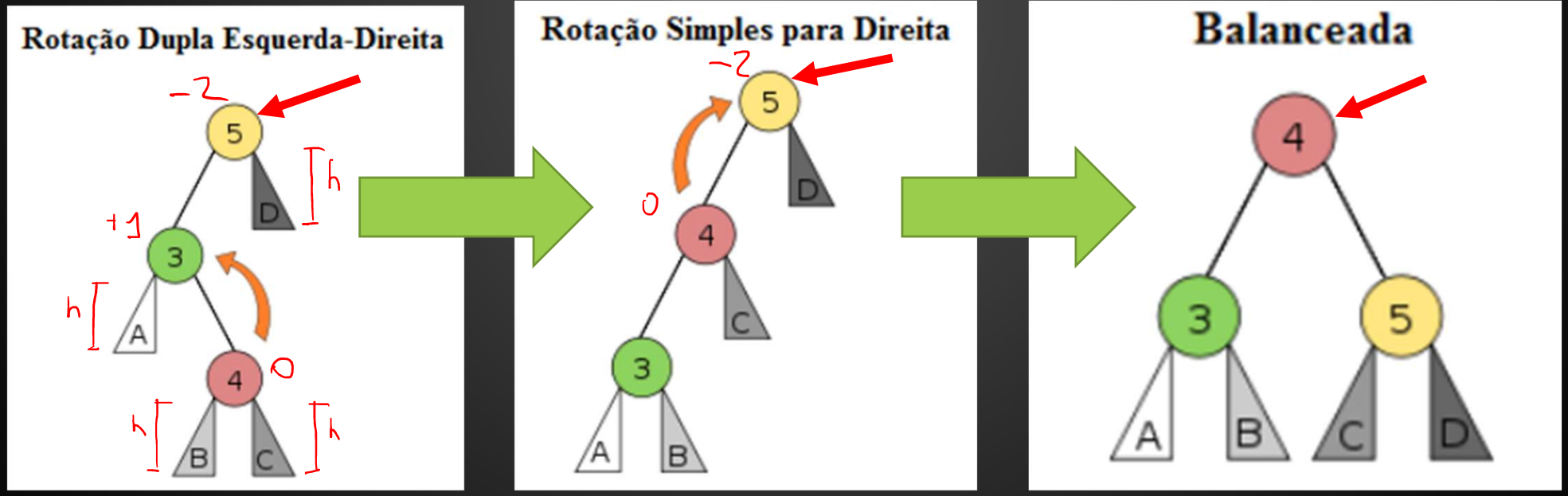


ROTAÇÃO DUPLA DIREITA-ESQUERDA



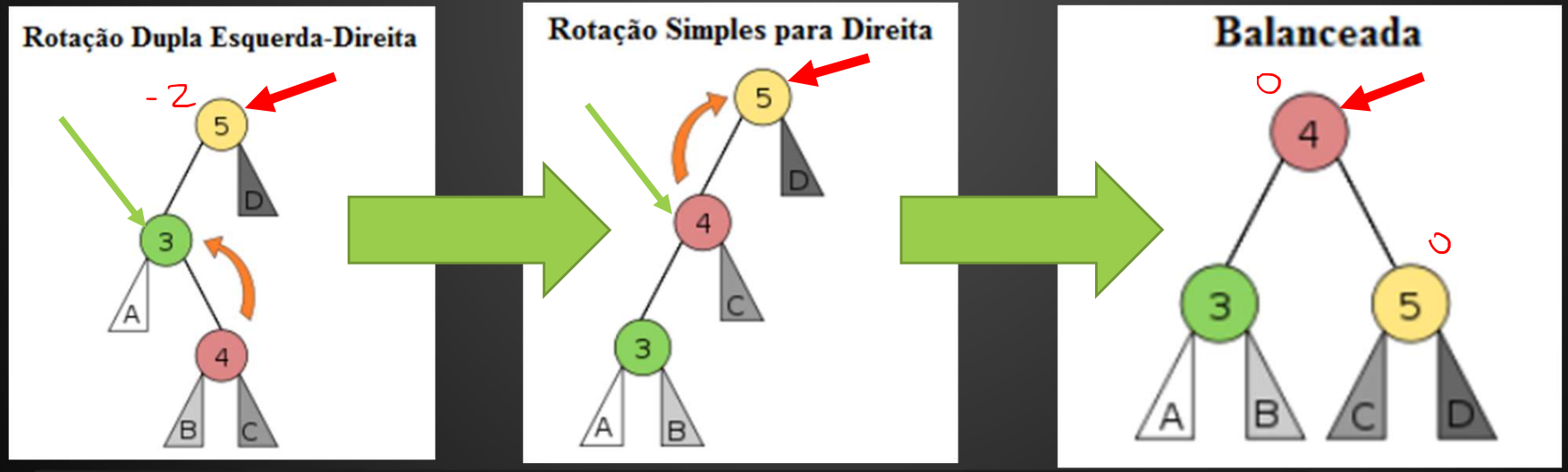

```
void ArvoreAVL::rotacaodireitaesquerda(No*& pai)
{
    No* filho = pai->filhodireita;
    rotacaodireita(filho);
    pai->filhodireita = filho;
    rotacaoesquerda(pai);
}
```

ROTAÇÃO DUPLA ESQUERDA-DIREITA



Professor
Douglas Maioli

ROTAÇÃO DUPLA ESQUERDA-DIREITA

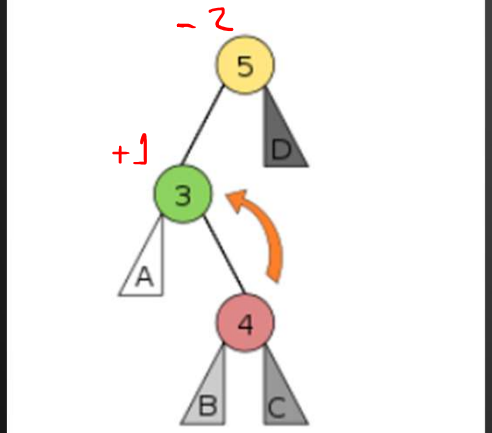



```
void ArvoreAVL::rotacaoesquerdadireita(No*& pai)
{
    No* filho = pai->filhoesquerda;
    rotacaoesquerda(filho);
    pai->filhoesquerda = filho;
    rotacaodireita(pai);
}
```

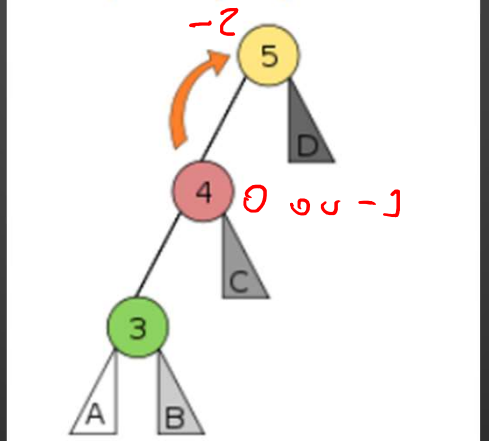
Professor
Douglas Maioli

ROTAÇÕES

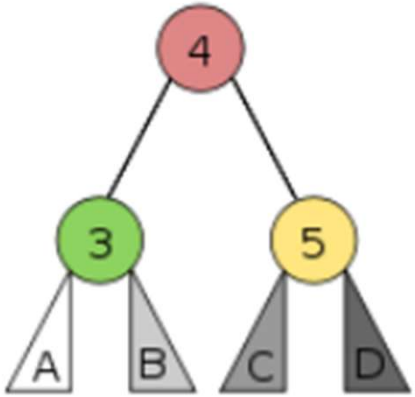
Rotação Dupla Esquerda-Direita



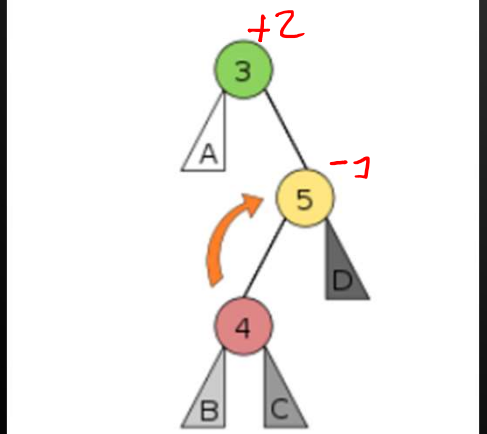
Rotação Simples para Direita



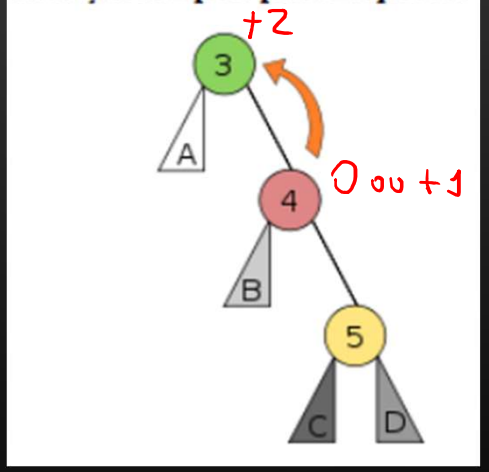
Balanceada



Rotação Dupla Direita-Esquerda



Rotação Simples para Esquerda



ÁRVORE AVL - EXEMPLO

5, 10, 15, 20, 30, 35