AVL

SCC-502 – Algoritmos e Estruturas de Dados I

Prof. Thiago A. S. Pardo

ABB

- Problemas da ABB
  - O que acontece com a inserção de elementos ordenados?
    - A, B, C, D, E, ..., Z
    - 1000, 999, 998, ..., 1

ABB

- O desbalanceamento da árvore pode tornar a busca tão ineficiente quanto a busca sequencial (no pior caso)
  - O(N)

- Solução?

Balanceamento da árvore!

AVL

- Árvore binária de busca balanceada
  - Para cada nó, as alturas das subárvores diferem em 1, no máximo
  - Proposta em 1962 pelos matemáticos russos G.M. Adelson-Velskki e E.M. Landis
    - Métodos de **inserção** e **remoção** de elementos da árvore de forma que ela fique balanceada

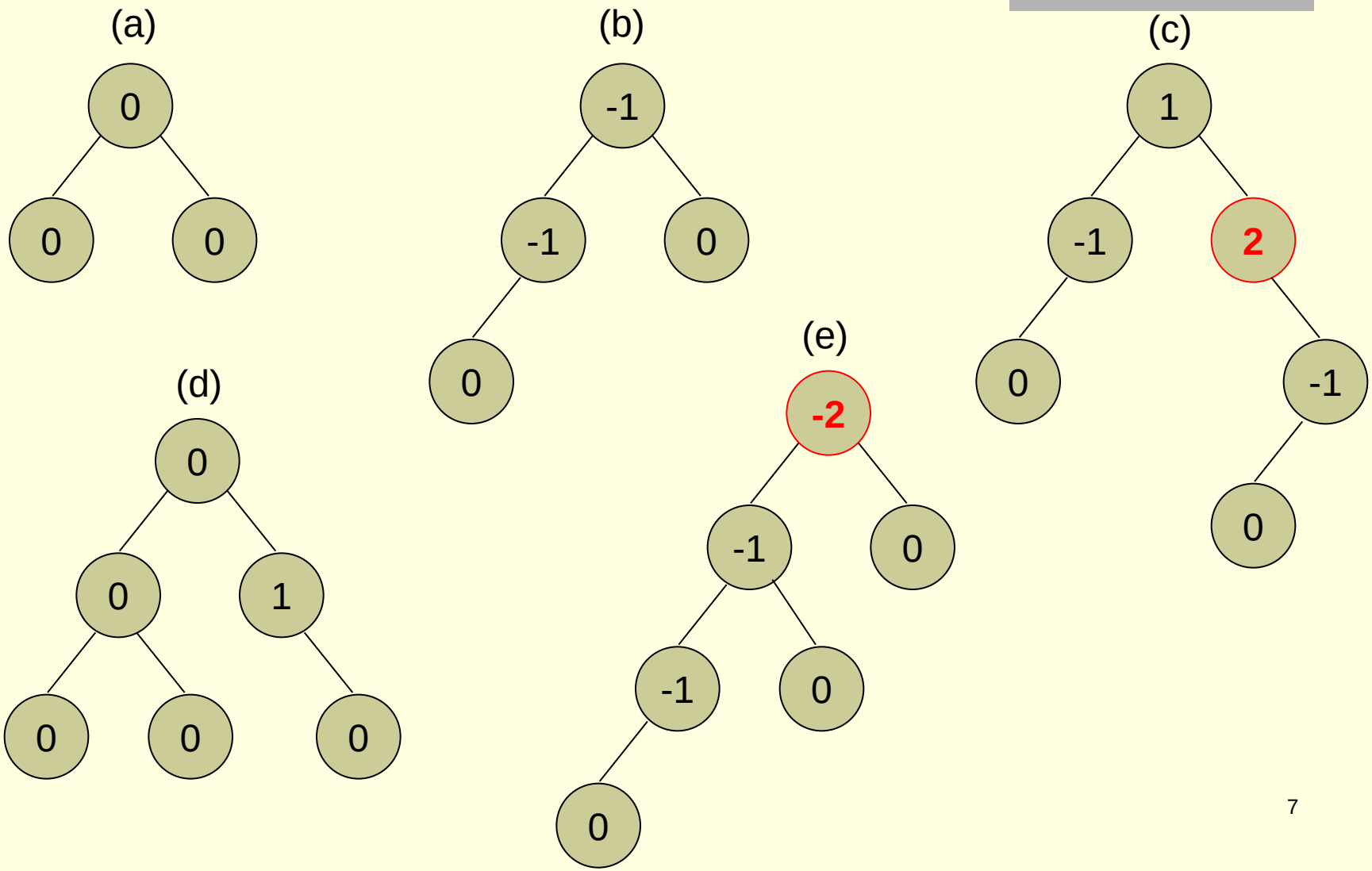
AVL

- Como é que se sabe quando é necessário balancear a árvore?
  - Se a diferença de altura das subárvores deve ser 1, no máximo, então temos que procurar diferenças de altura maior do que isso
  - Possível solução: cada nó pode manter a diferença de altura de suas subárvores
    - Convencionalmente chamada de fator de balanceamento do nó

AVL

- Fatores de balanceamento dos nós
  - Altura da subárvore direita menos altura da subárvore esquerda
    - Hd-He
  - Atualizados sempre que a árvore é alterada (elemento é inserido ou removido)
  - Quando um fator é 0, 1 ou -1, a árvore está balanceada
  - Quando um fator se torna 2 ou -2, a árvore está desbalanceada
    - Operações de balanceamento!

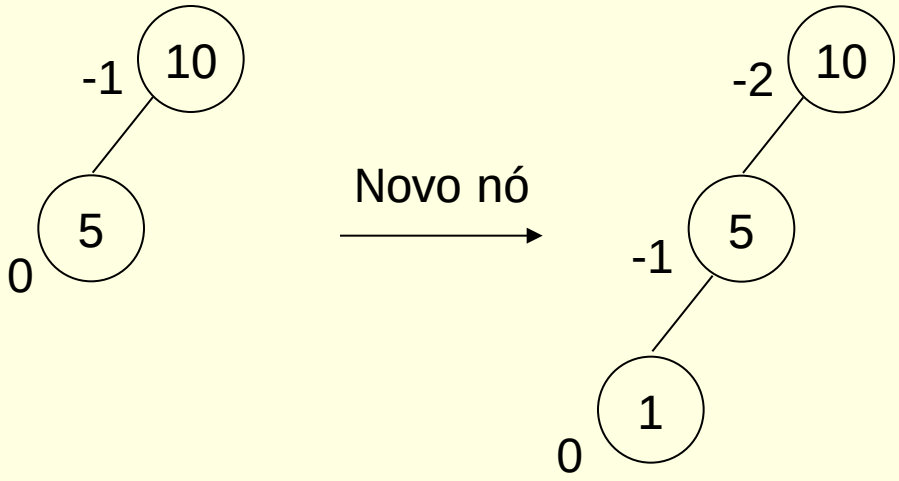
AVL: quem é e quem não é



AVL

- Balanceamento
  - Intuitivamente

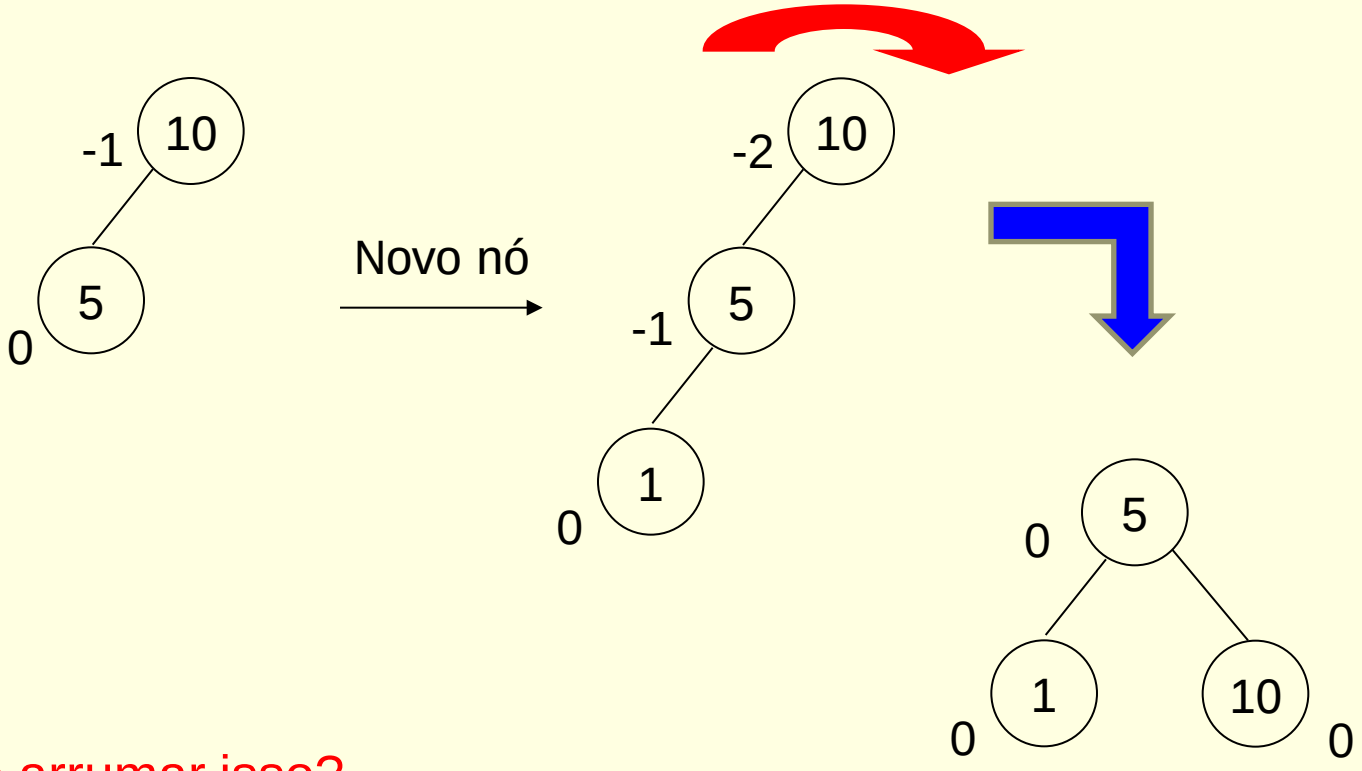
AVL: exemplo de desbalanceamento



Desbalanceou!!!

Como arrumar isso?

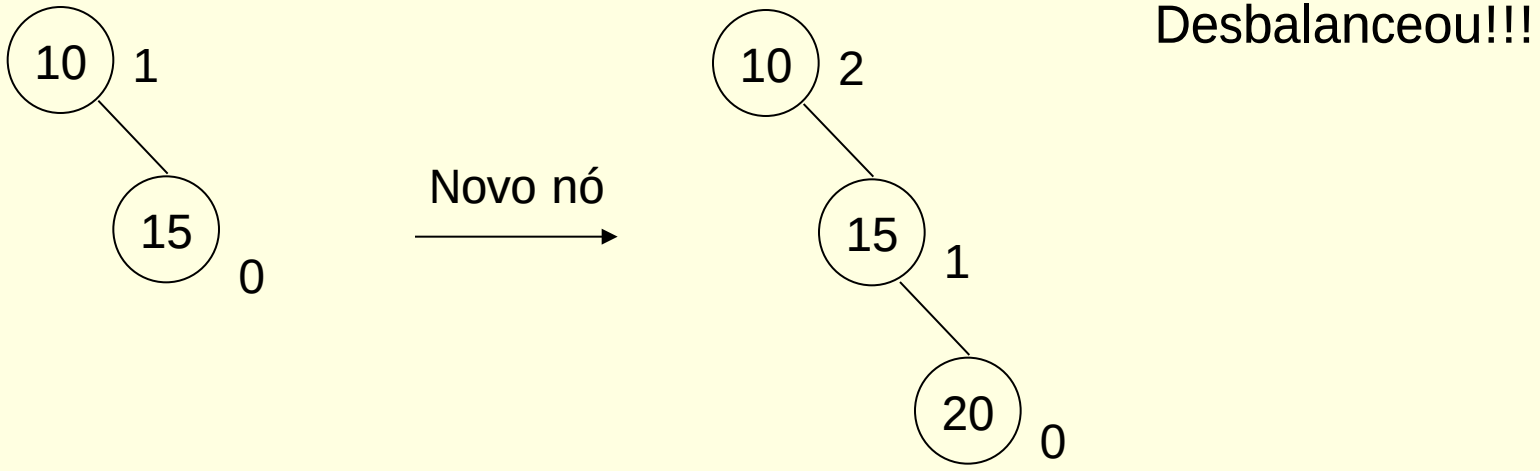
AVL: exemplo de desbalanceamento



Como arrumar isso?

Rotação simples para direita!

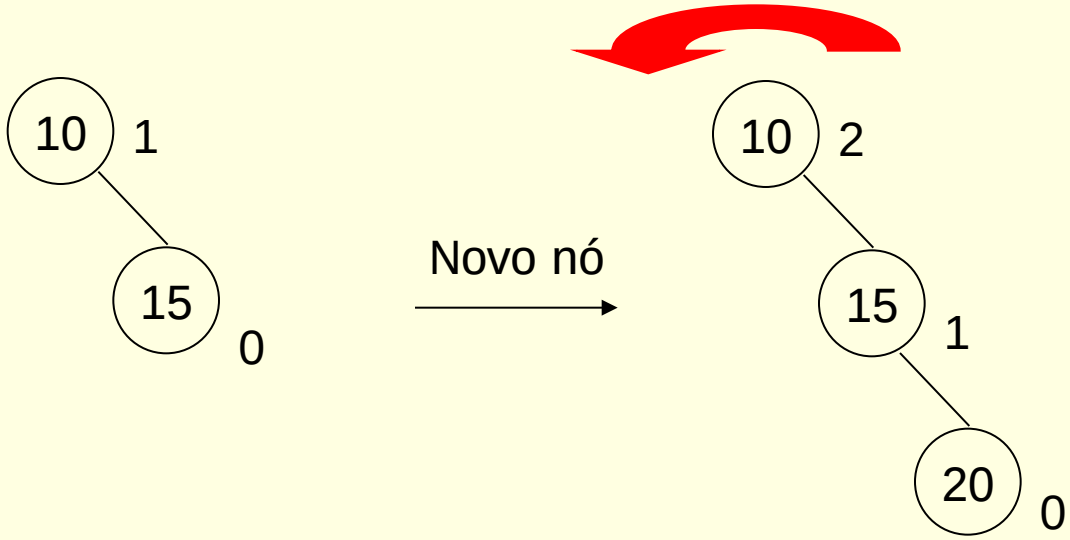
AVL: exemplo de desbalanceamento

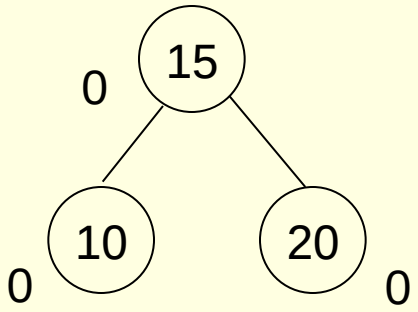


Desbalanceou!!!

Como arrumar isso?

AVL: exemplo de desbalanceamento

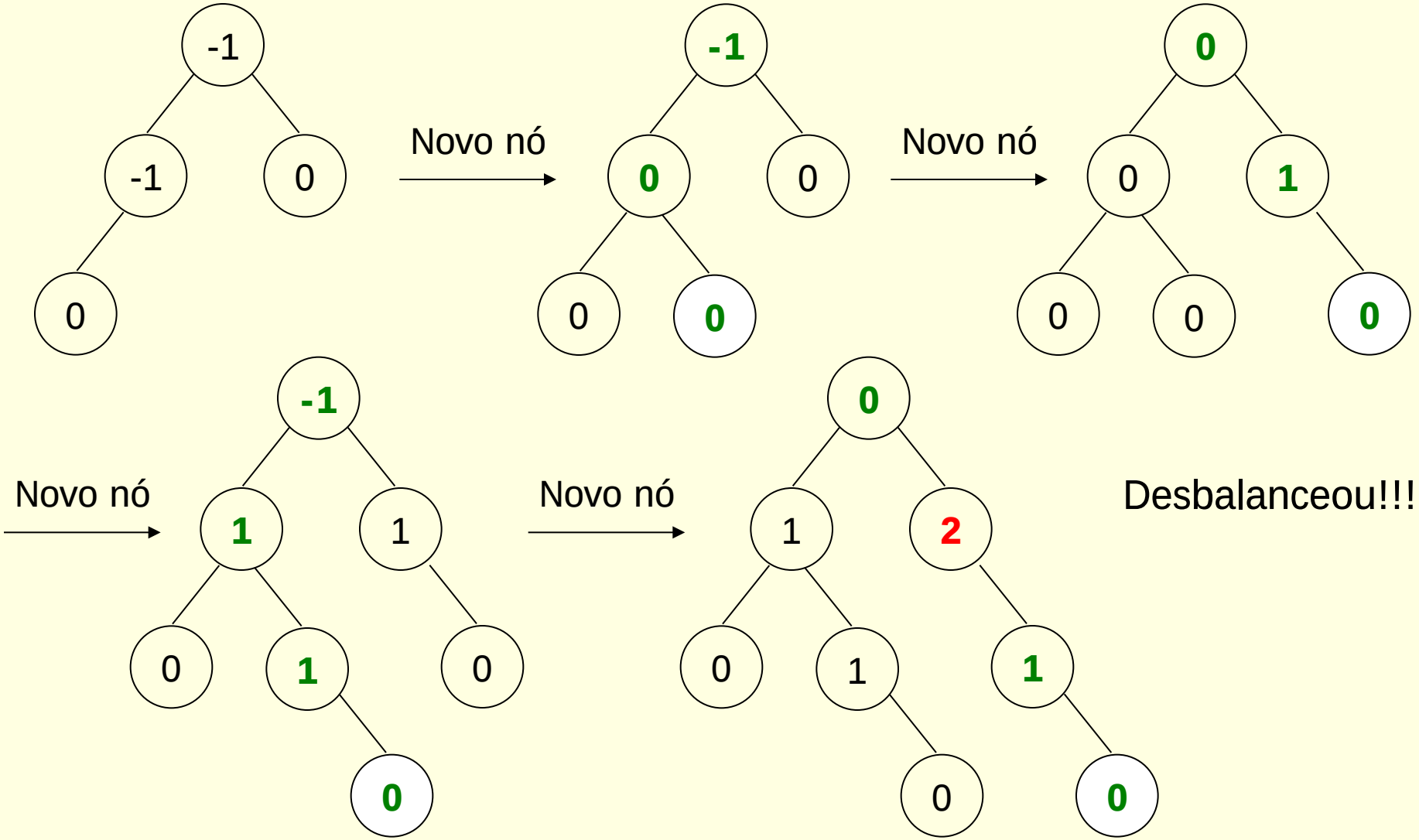




Como arrumar isso?

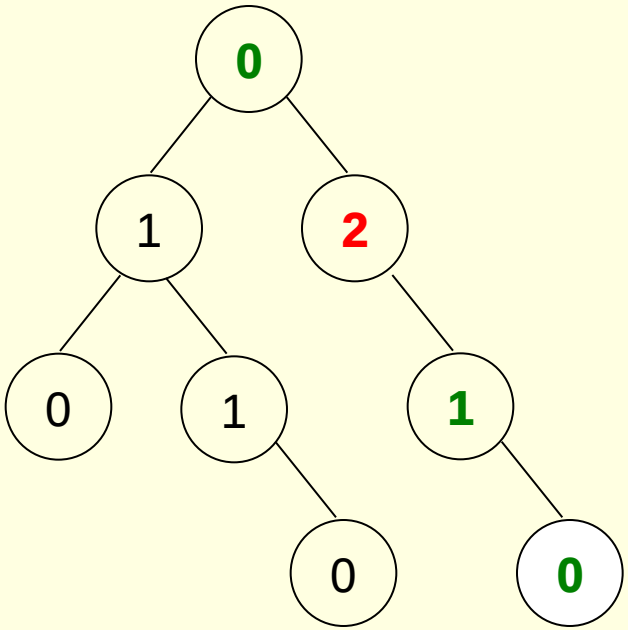
Rotação simples para esquerda!

AVL: exemplo de desbalanceamento



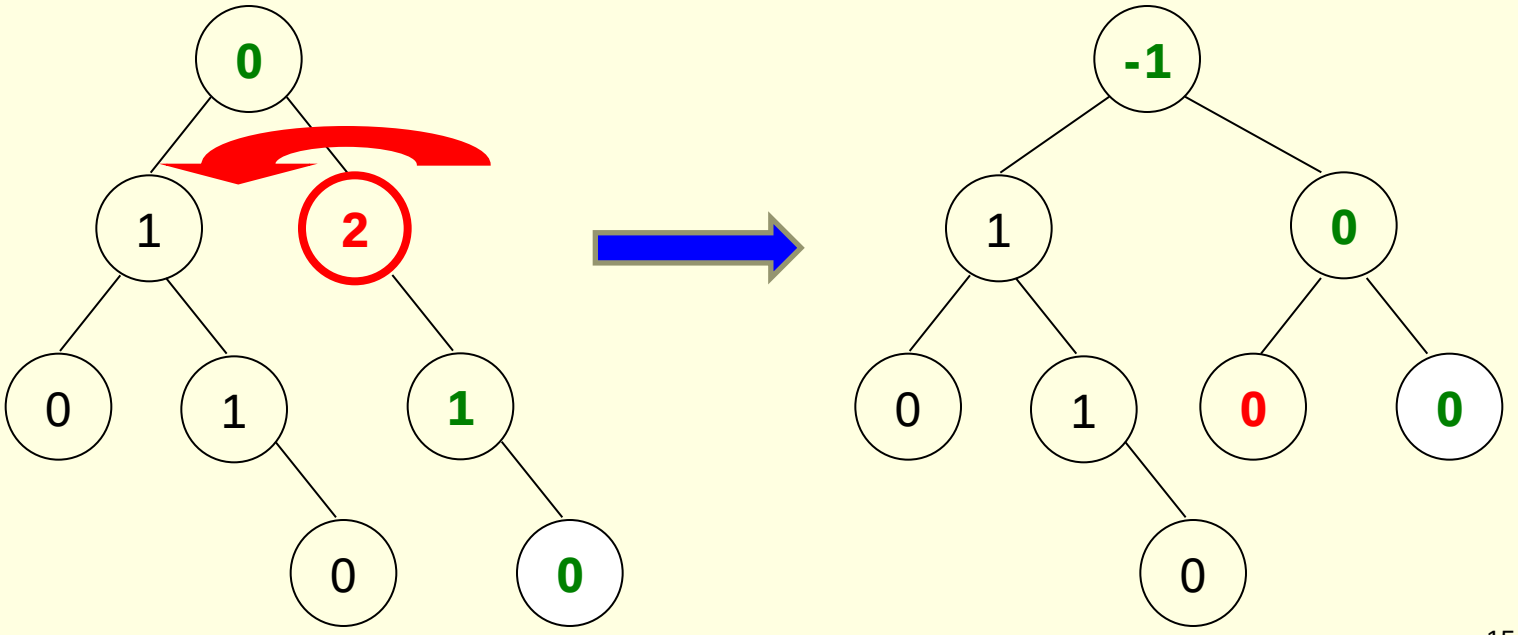
Questão

- Como cuidar disso?



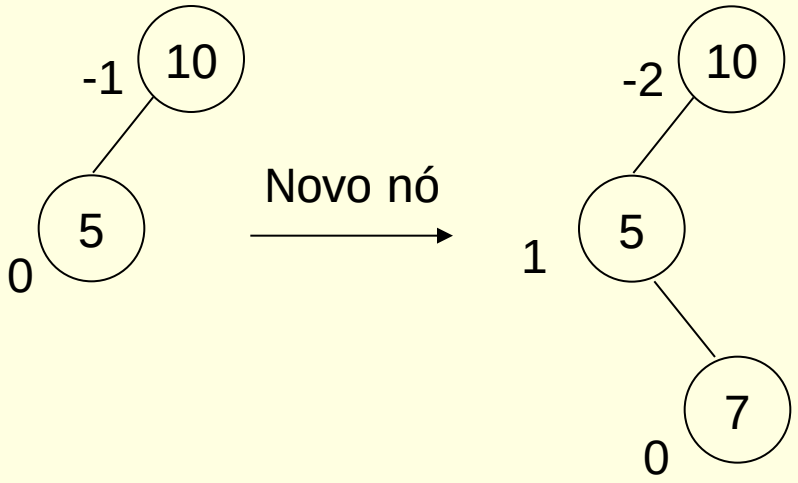
Questão

- Como cuidar disso?
  - Rotação simples para esquerda no local com problema
    - De forma ascendente, procura-se pelo primeiro 2/-2 a partir do local da inserção



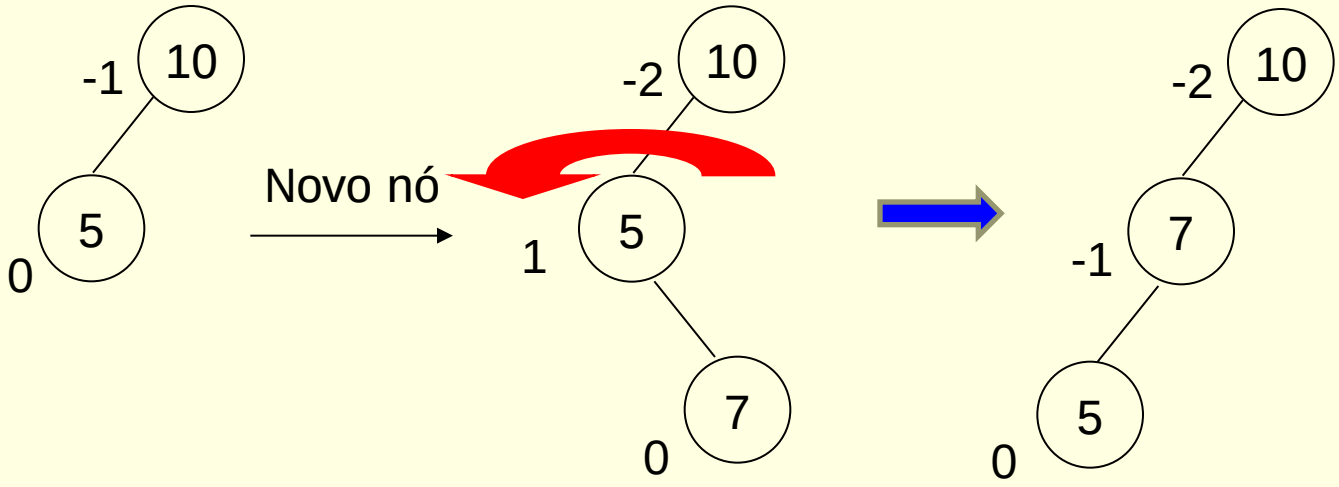
AVL: exemplo de desbalanceamento

Desbalanceou!!!



Como arrumar isso?

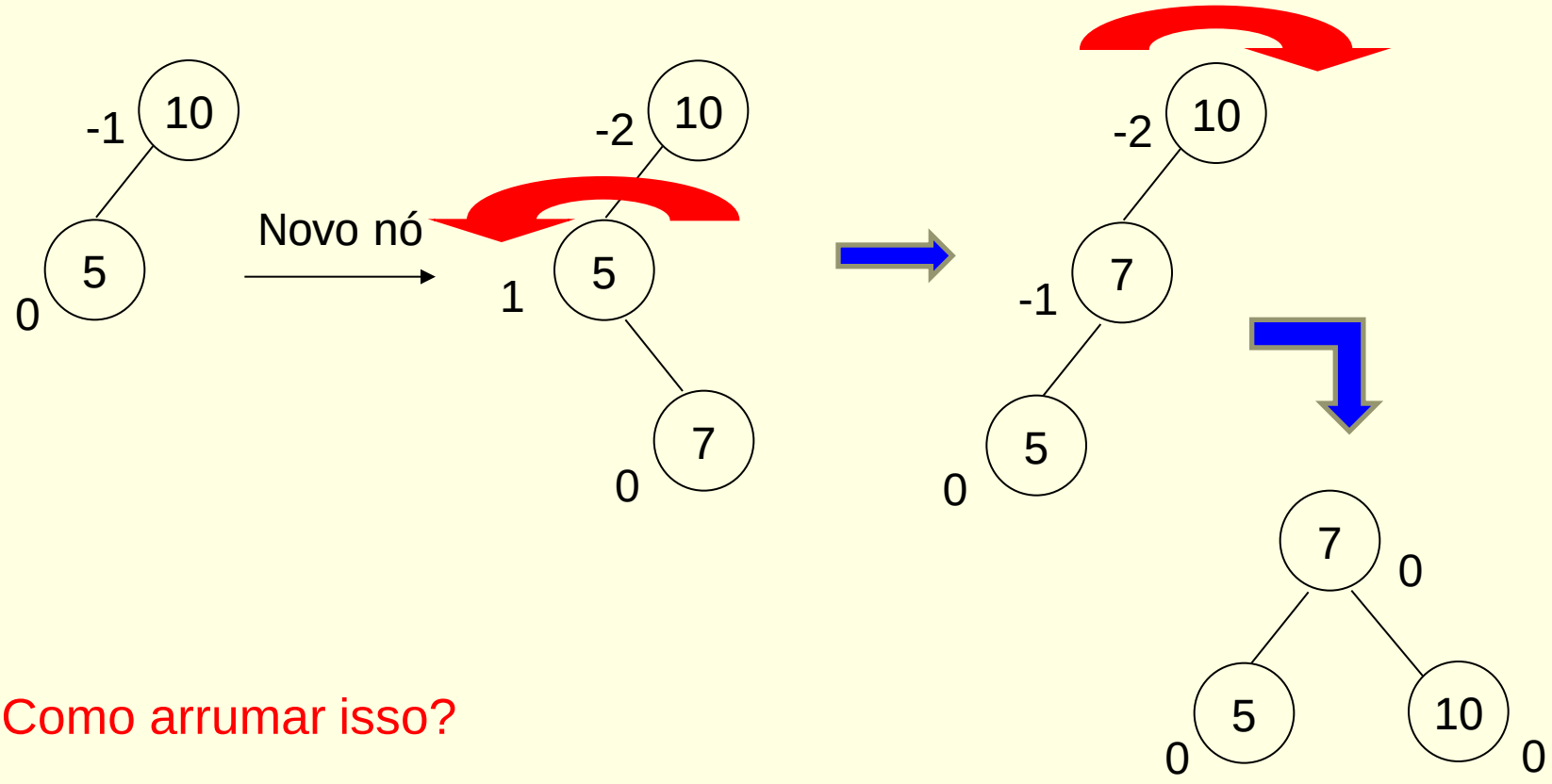
AVL: exemplo de desbalanceamento



Como arrumar isso?

Rotação dupla: esquerda

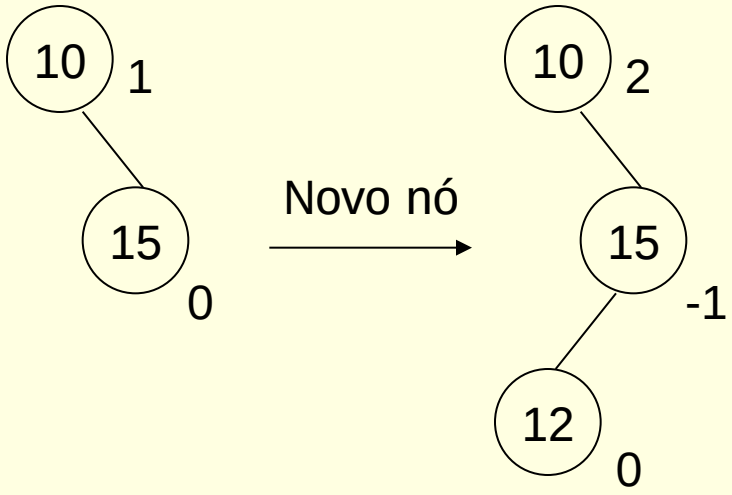
AVL: exemplo de desbalanceamento



Como arrumar isso?

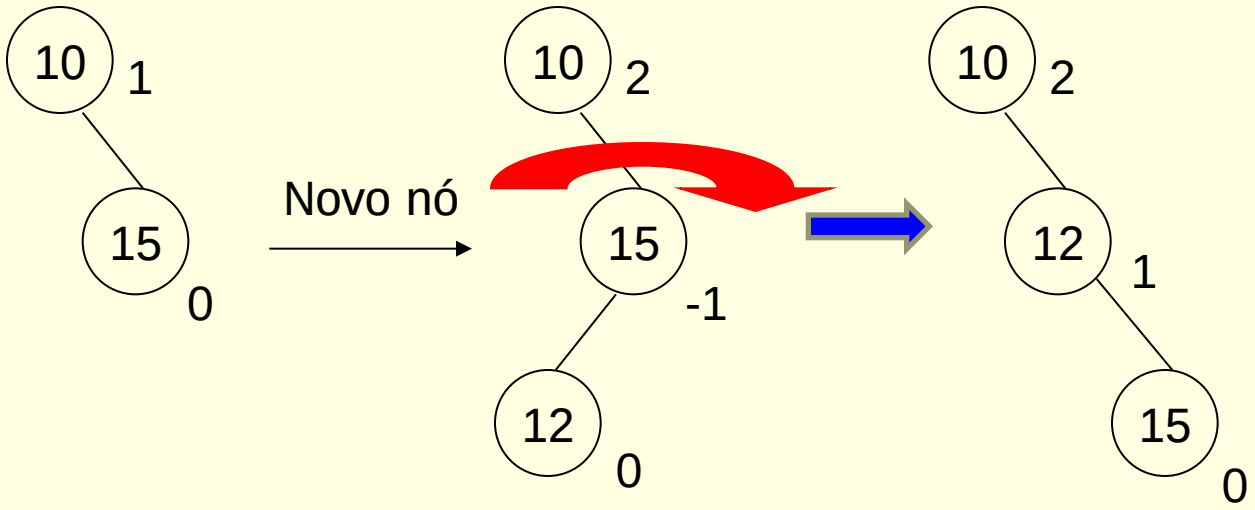
Rotação dupla: esquerda + direita

AVL: exemplo de desbalanceamento



Como arrumar isso?

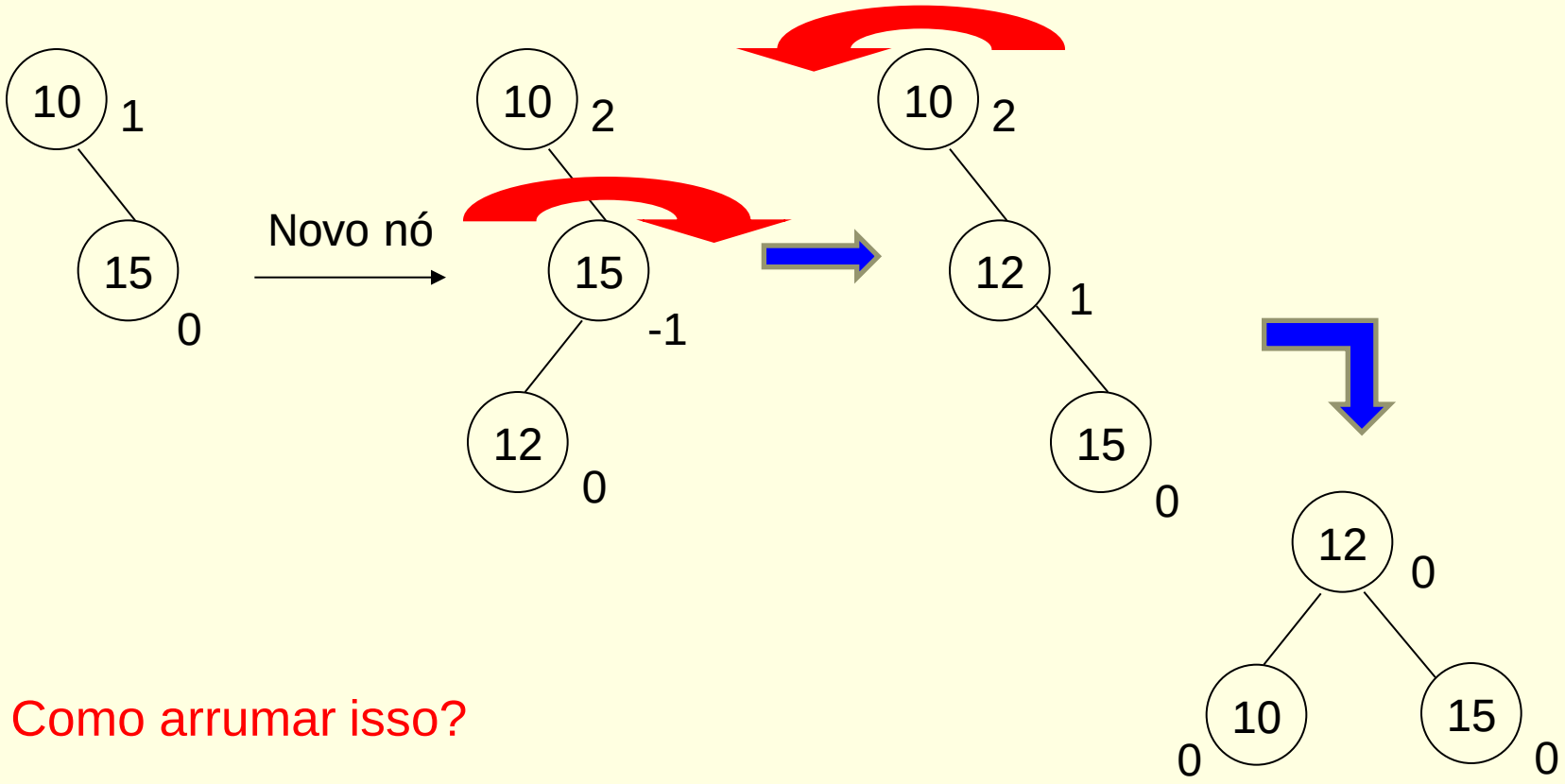
AVL: exemplo de desbalanceamento



Como arrumar isso?

Rotação dupla: direita

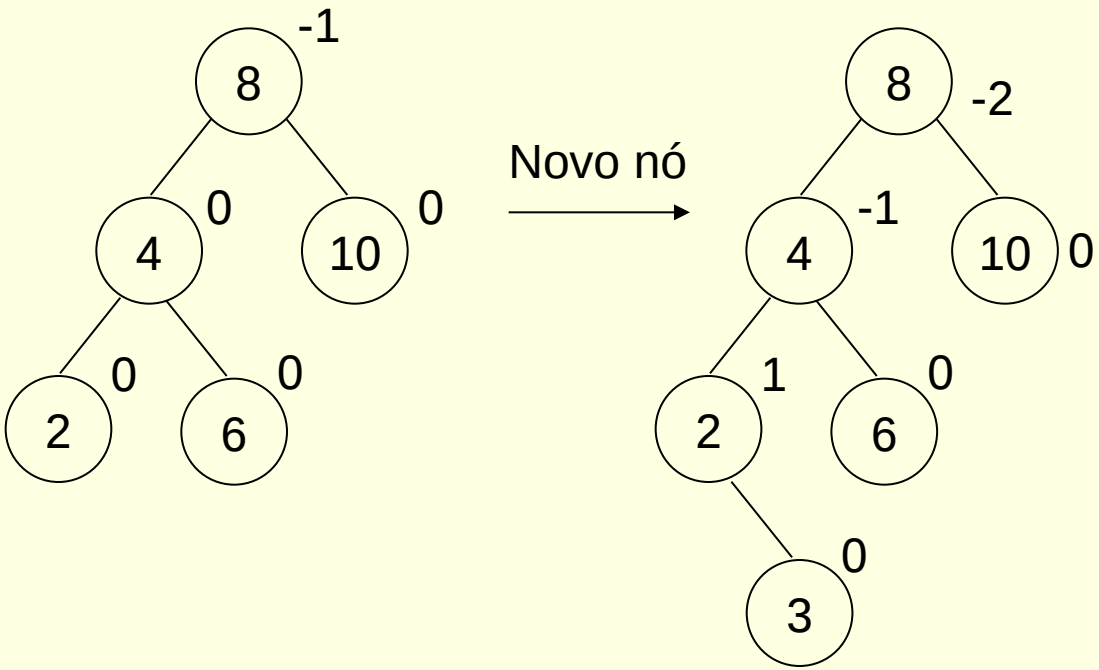
AVL: exemplo de desbalanceamento



Como arrumar isso?

Rotação dupla: direita + esquerda

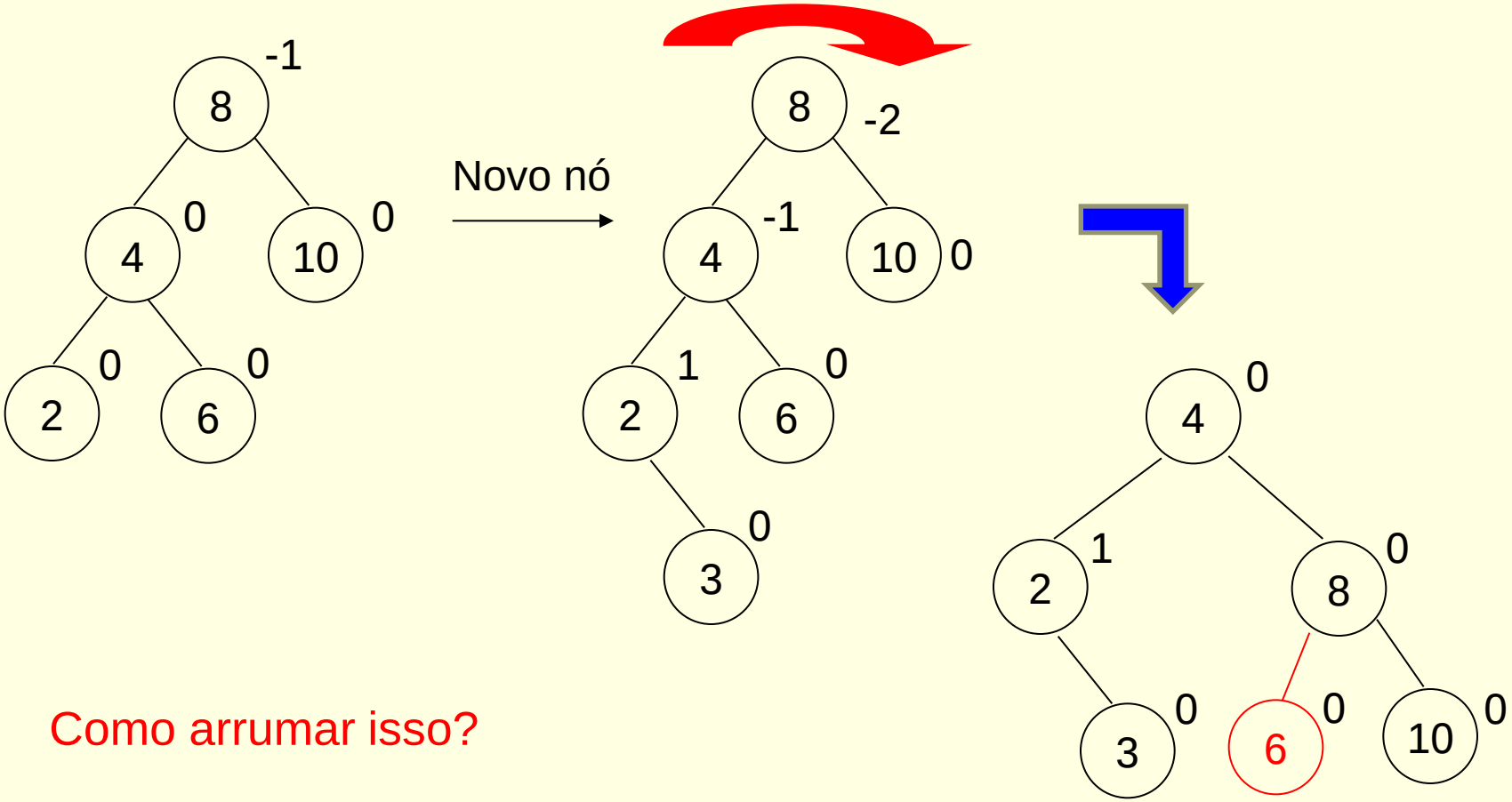
AVL: exemplo de desbalanceamento



Como arrumar isso?

EXERCÍCIO

AVL: exemplo de desbalanceamento



Como arrumar isso?

Rotação simples para direita

Cuidado: realocação!

AVL

- Exercício
  - Inserir os elementos 10, 3, 2, 5, 9, 7, 15, 12 e 13, nesta ordem, em uma árvore e balancear quando necessário

AVL

- Exercício
  - Inserir os elementos A, B, C, ..., J em uma árvore e balancear quando necessário

AVL

- Os percursos em-ordem da árvore original e da balanceada permanecem iguais
  - Exercício: prove para um dos exemplos anteriores!

AVL

- Balanceamento
  - Formalmente

AVL

- Novo algoritmo de inserção
  - A cada inserção, verifica-se o balanceamento da árvore
    - Se necessário, fazem-se as rotações de acordo com o caso (sinais iguais ou não)
  - Em geral, armazena-se uma variável de balanceamento em cada nó para indicar o FB

AVL

- Declaração

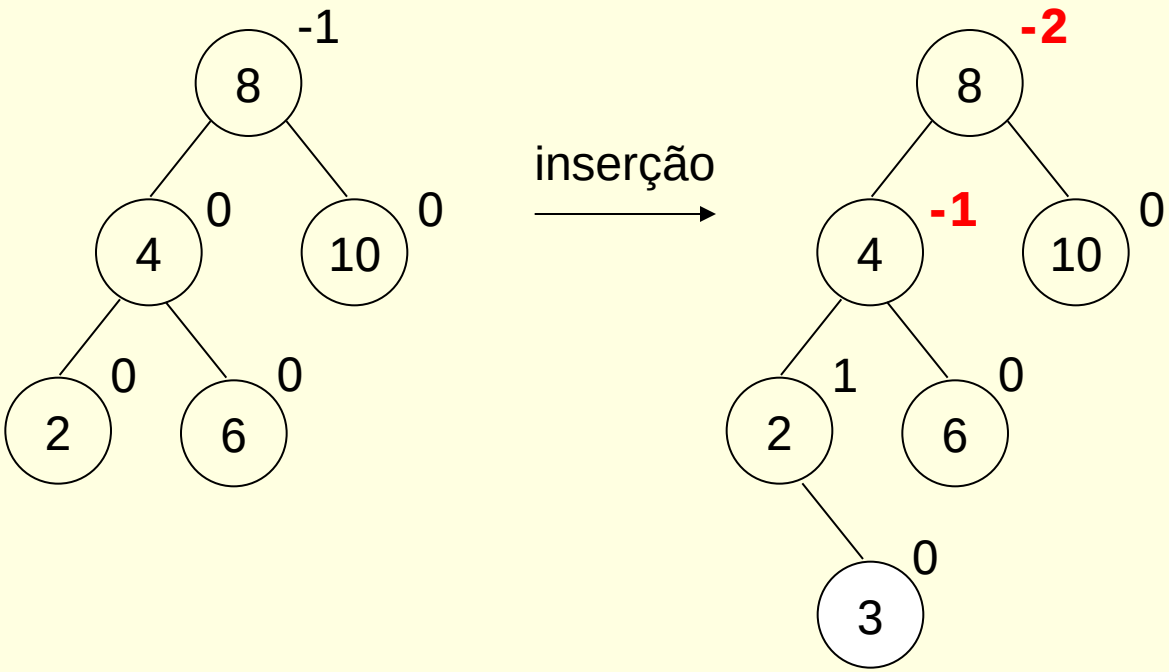
```
typedef int elem;

typedef struct bloco {
        elem info;
        struct bloco *esq, *dir;
        int FB;
} no;

typedef struct {
      no *raiz;
} AVL;
```

AVL

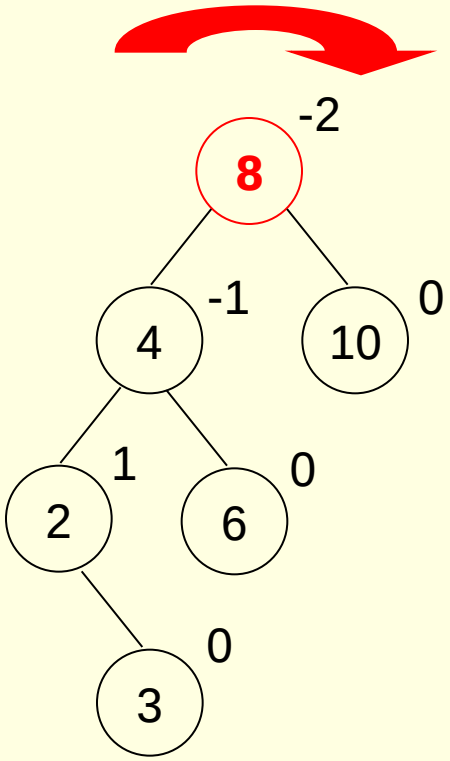
- Controle do balanceamento
  - Altera-se o algoritmo de inserção para balancear a árvore quando ela se tornar desbalanceada após uma inserção (nó com FB 2 ou -2)
    - Rotações
      - Se árvore pende para esquerda (FB negativo), rotaciona-se para a direita
      - Se árvore pende para direita (FB positivo), rotaciona-se para a esquerda
    - 2 casos podem acontecer

AVL: primeiro caso

- Raiz de uma subárvore com FB -2 (ou 2) e um nó filho com FB -1 (ou 1)
  - Os fatores de balanceamento têm sinais iguais: subárvores de nó raiz e filho pendem para o mesmo lado



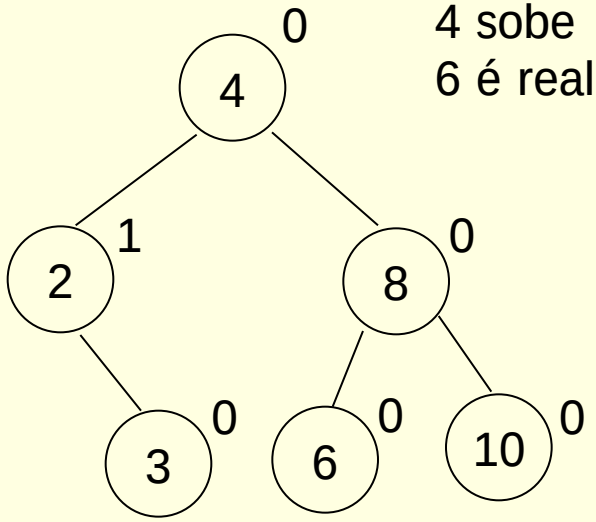
AVL: primeiro caso



Pendendo para
a esquerda

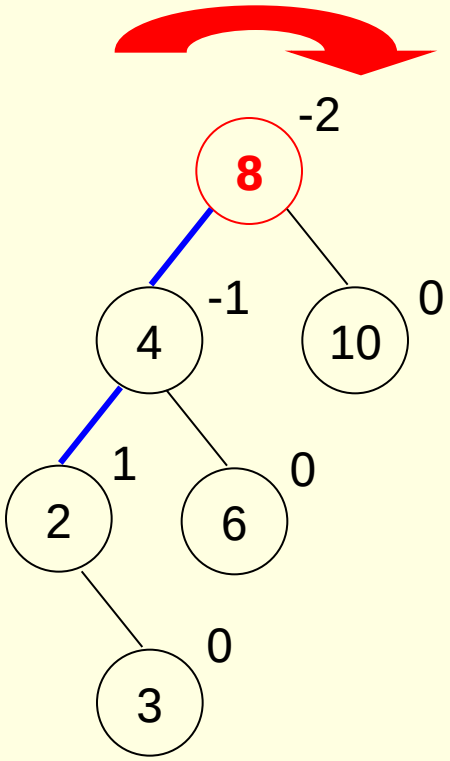
Rotação do nó pai
para a direita
(nó com FB=-2 ou 2)

8 desce
4 sobe
6 é realocado



Árvore balanceada!!!

AVL: primeiro caso

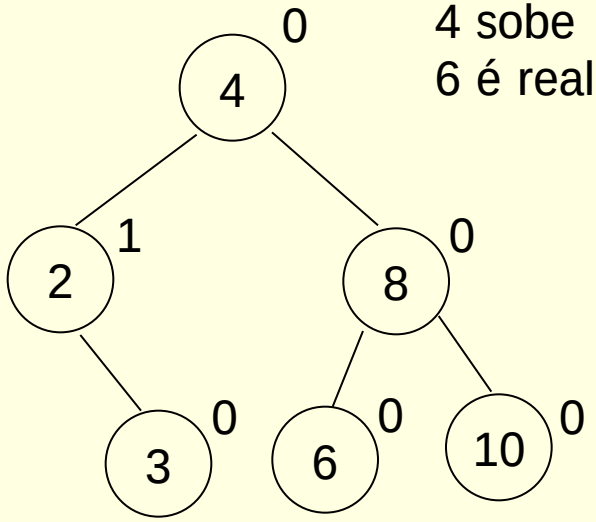


Pendendo para
a esquerda

Rotação do nó pai
para a direita
(nó com FB=-2 ou 2)

Rotação simples EE
(Esquerda-Esquerda)

8 desce
4 sobe
6 é realocado



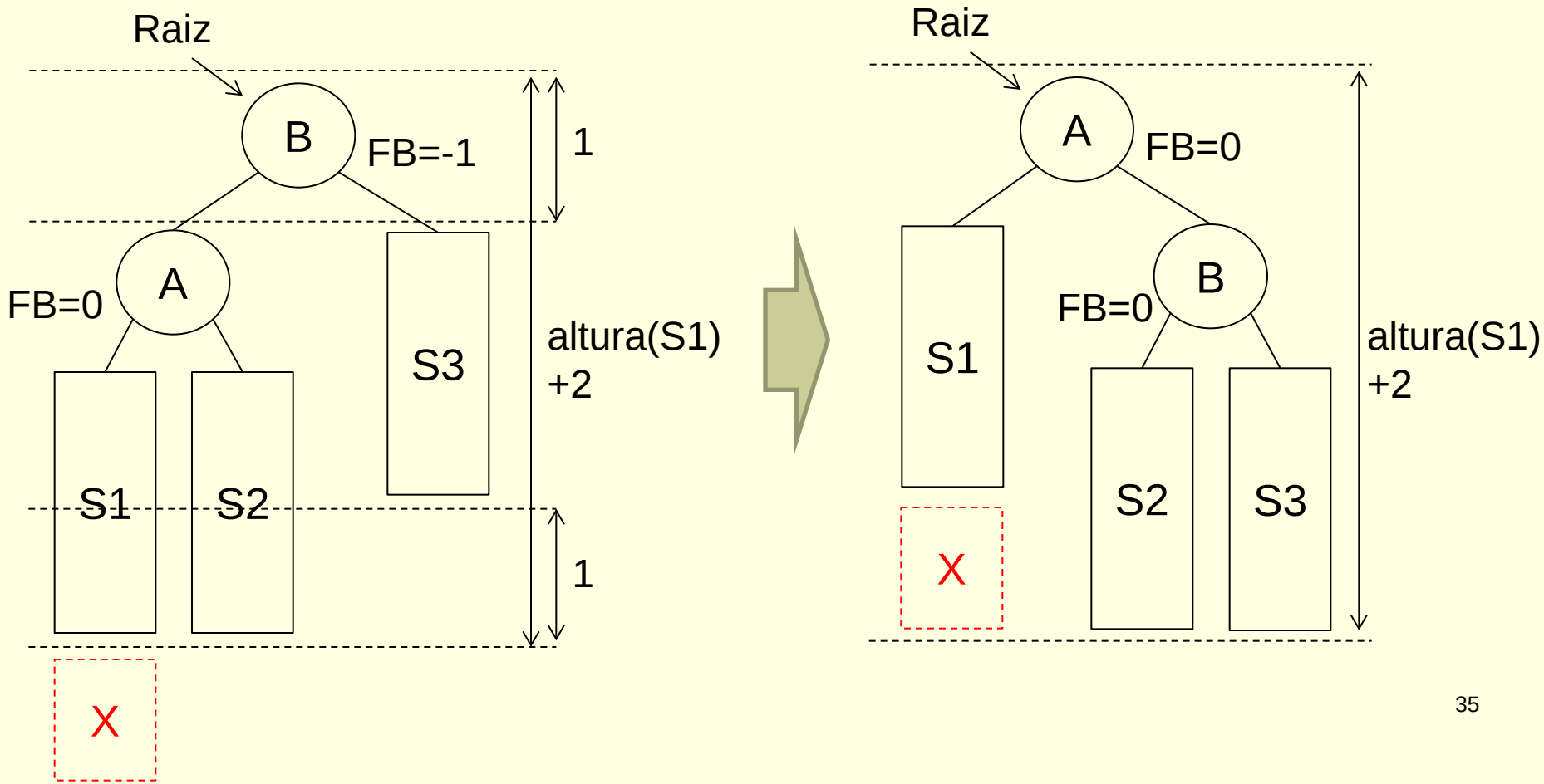
Árvore balanceada!!!

AVL: primeiro caso

- Quando subárvores do pai e filho pendem para um mesmo lado
  - Rotação simples para o lado oposto
    - EE ou DD (Direita-Direita, com raciocínio inverso)
  - Às vezes, é necessário realocar algum elemento, pois ele perde seu lugar na árvore

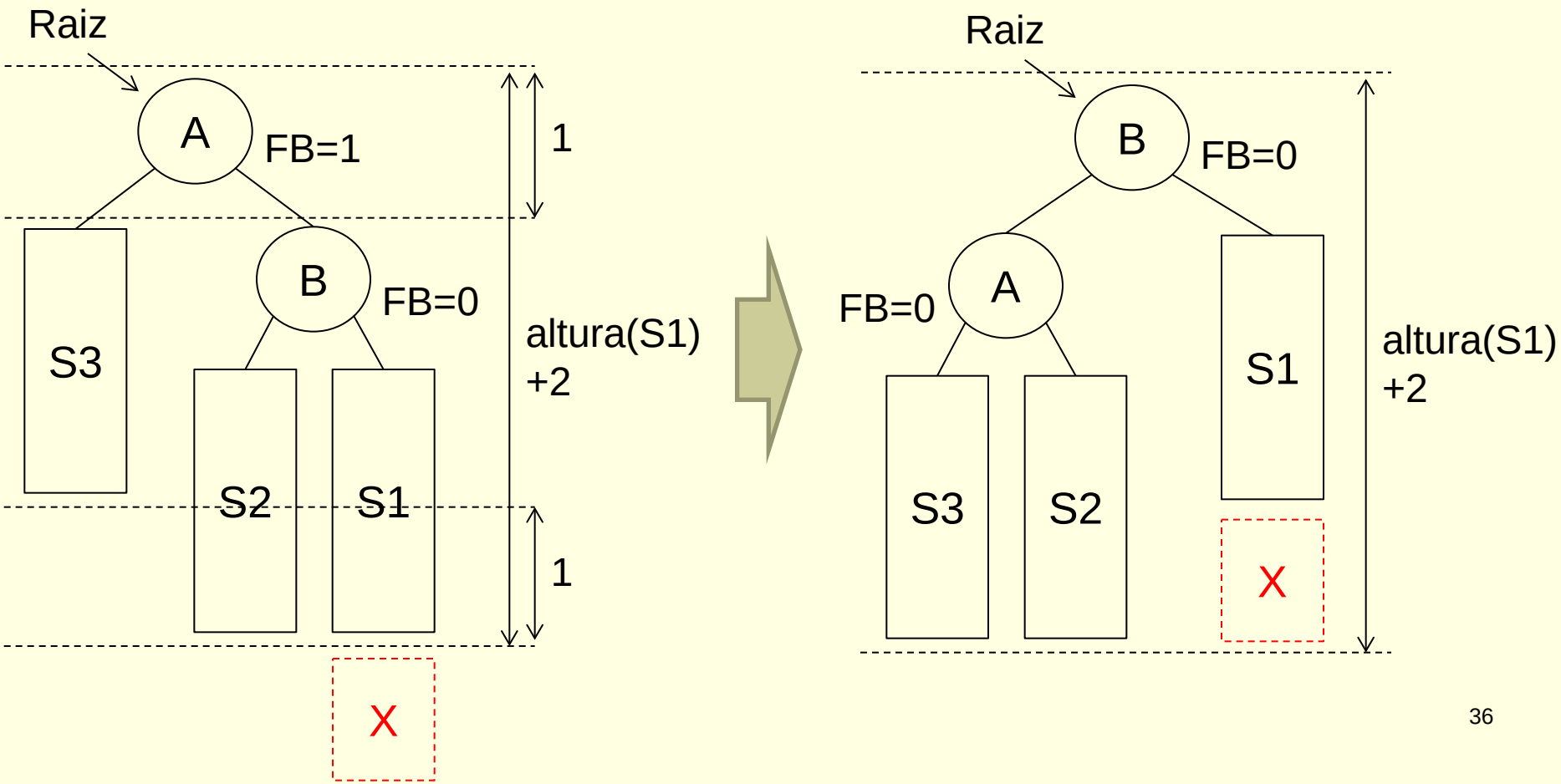
Formalmente: EE

Generalizando



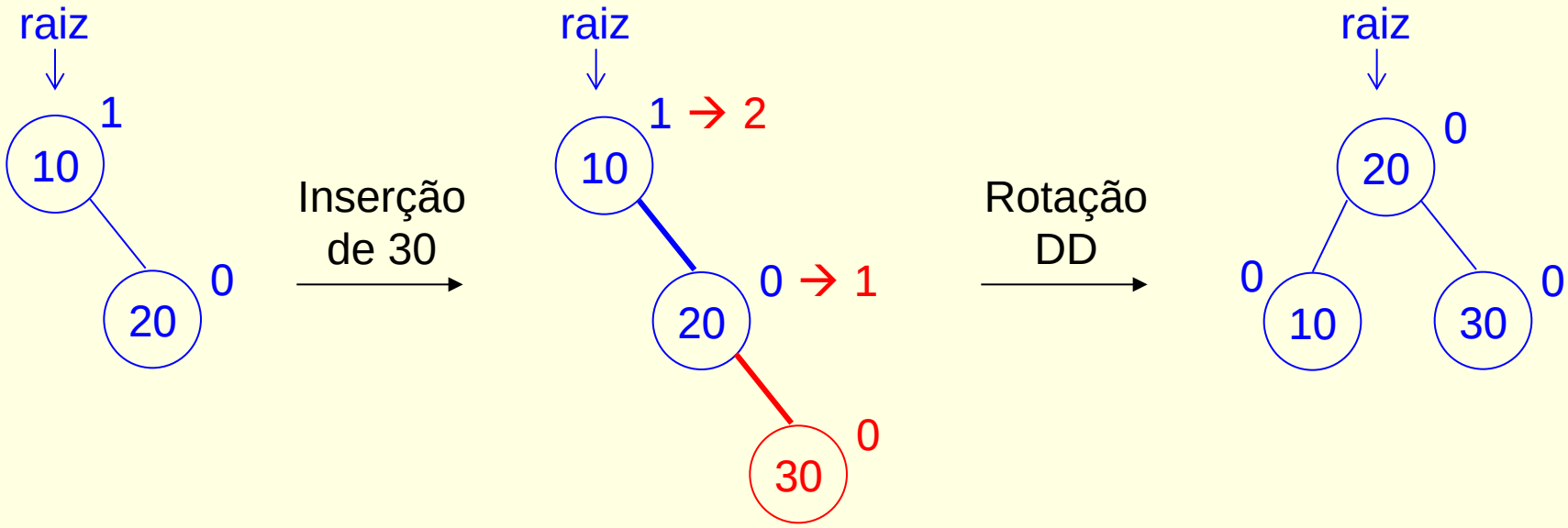
Formalmente: DD (similar)

Generalizando



Exercício

- Passo a passo: implementar rotação DD

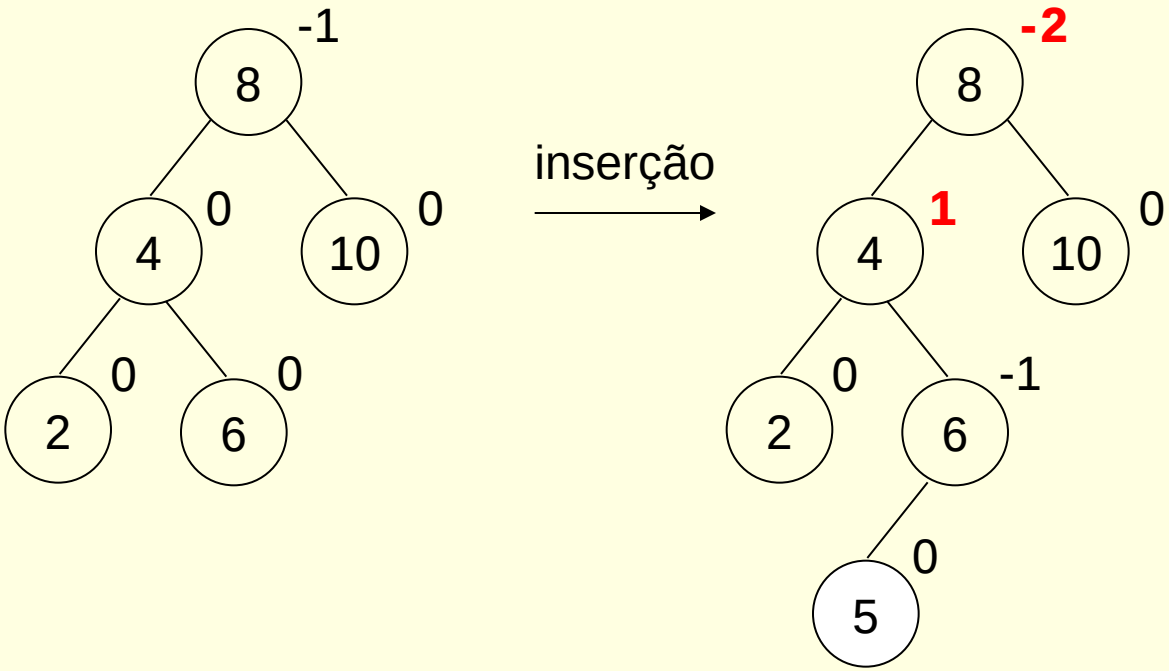


Exercício

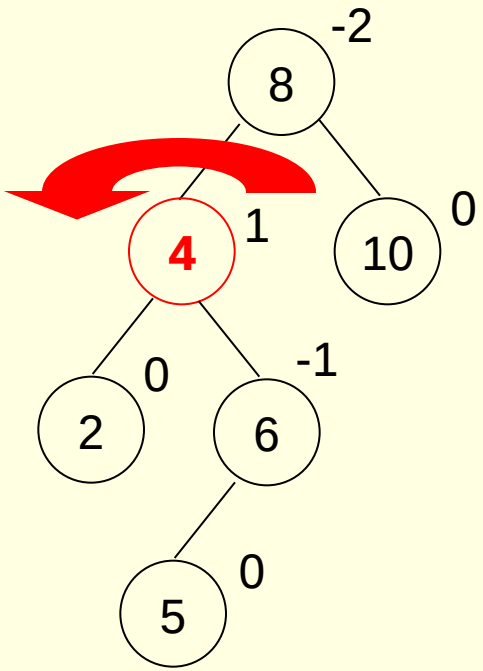
- Passo a passo: implementar rotação EE
  - Raciocínio similar

AVL: segundo caso

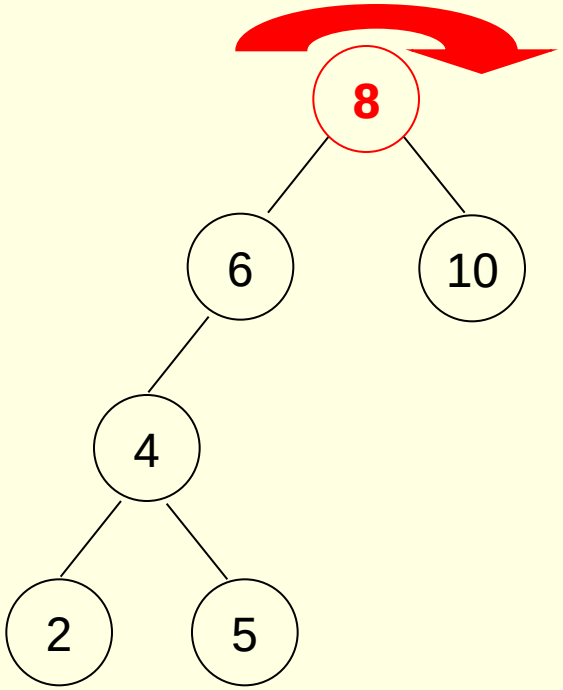
- Raiz de uma subárvore com FB -2 (ou 2) e um nó filho com FB 1 (ou -1)
  - Os fatores de balanceamento têm sinais opostos: subárvore de nó raiz pende para um lado e subárvore de nó filho pende para o outro (ou o contrário)



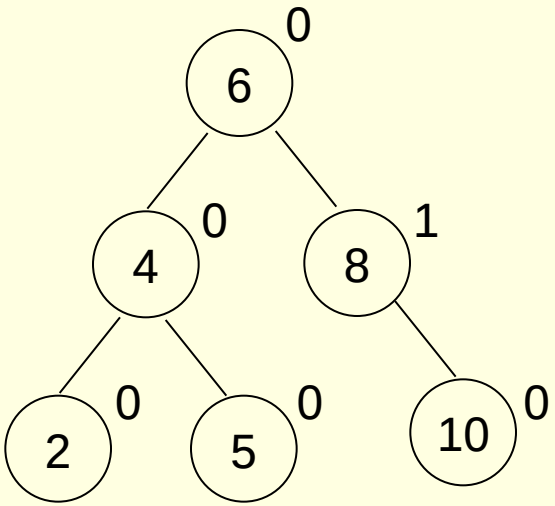
AVL: segundo caso



Rotação do nó filho
para a esquerda
(nó com FB=1 ou -1)

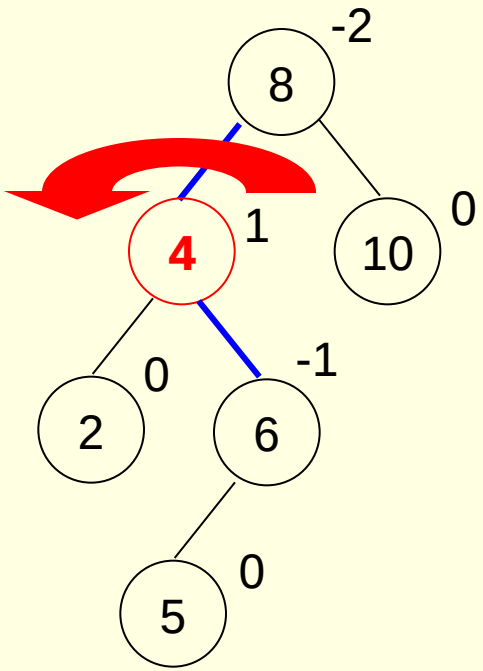


Rotação do nó pai
para a direita
(nó com FB=-2 ou 2)



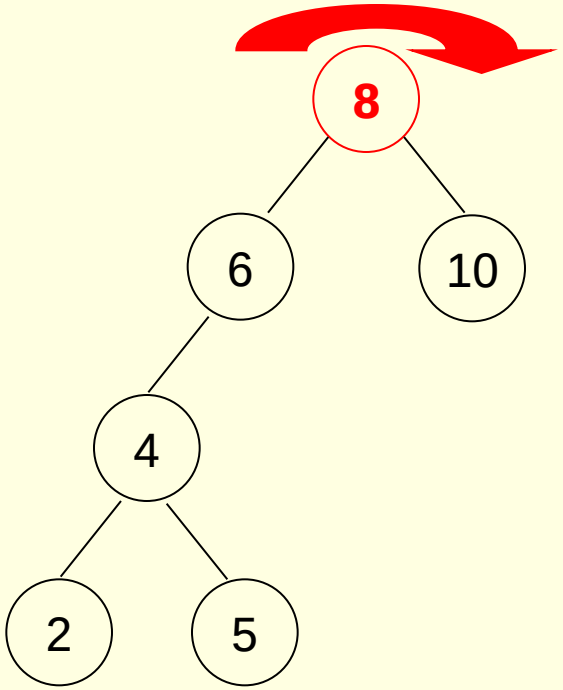
Árvore balanceada!!!

AVL: segundo caso

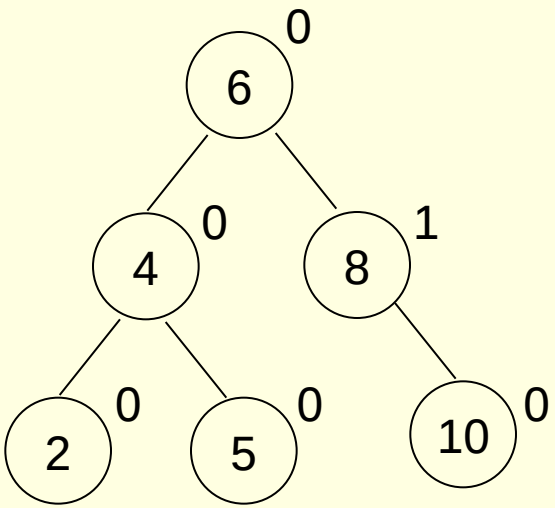


Rotação do nó filho
para a esquerda
(nó com FB=1 ou -1)

Rotação dupla ED
(Esquerda-Direita)



Rotação do nó pai
para a direita
(nó com FB=-2 ou 2)



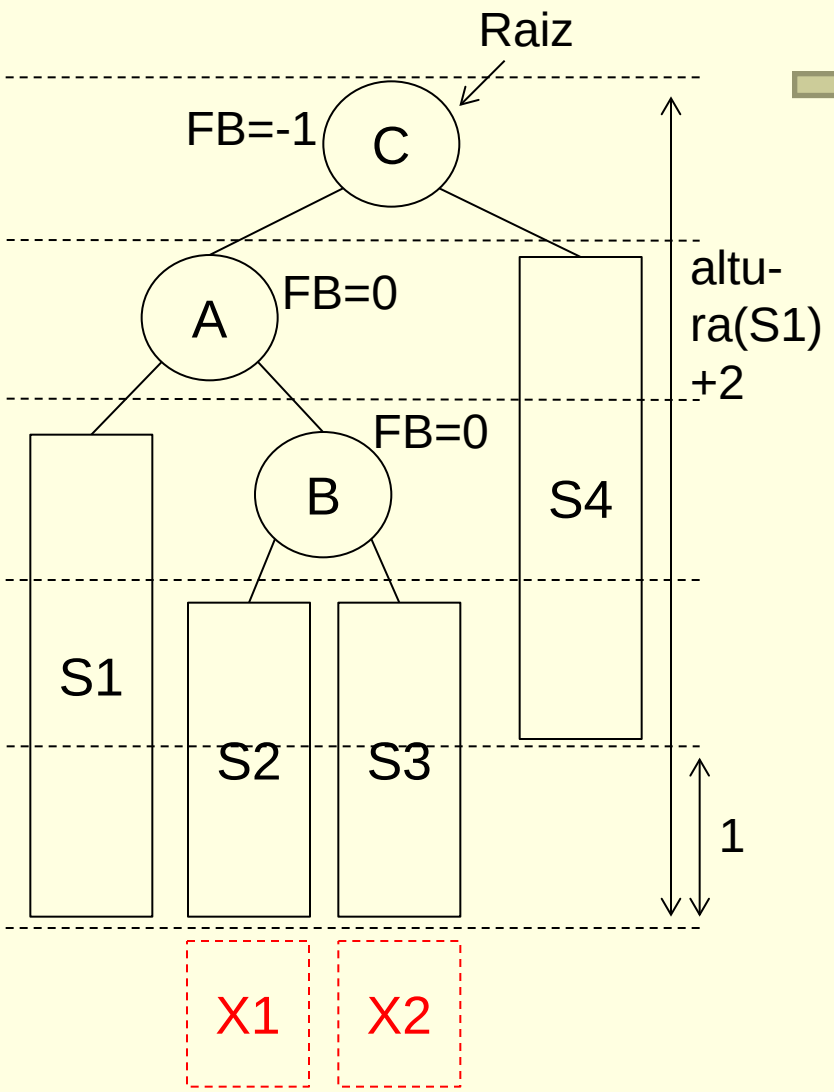
Árvore balanceada!!!

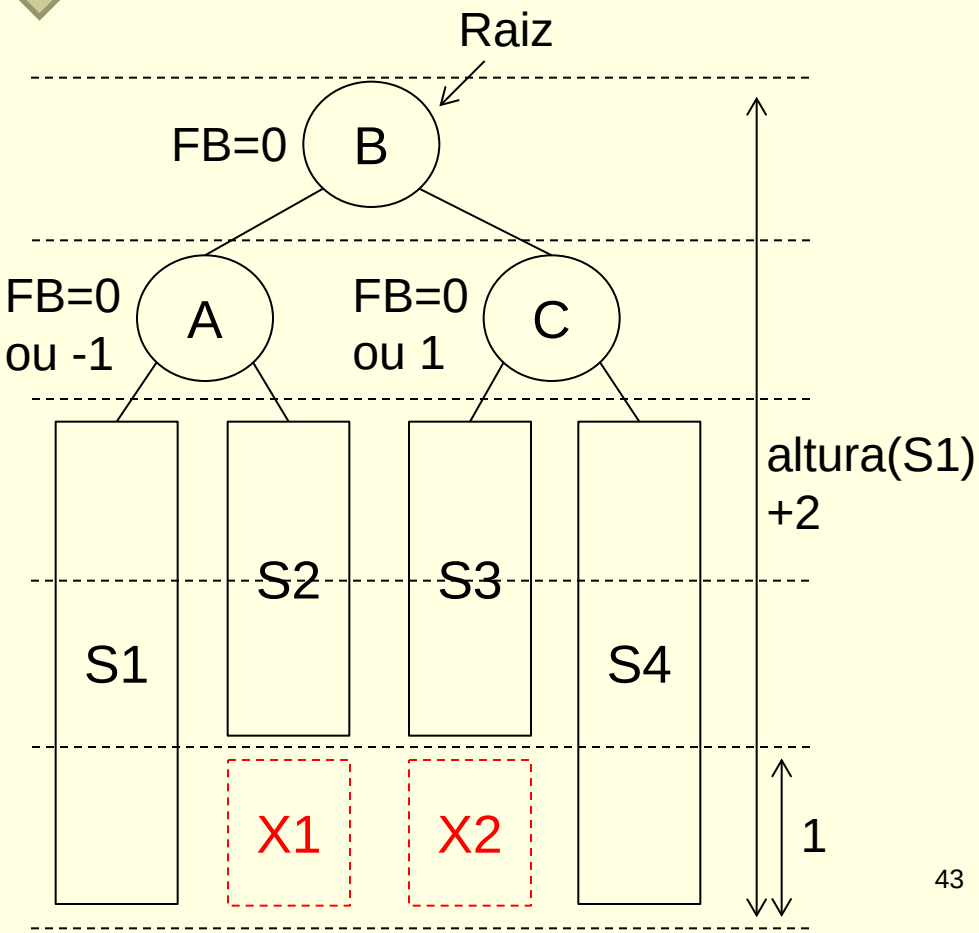
AVL: segundo caso

- Quando subárvores do pai e filho pendem para lados opostos
  - Rotação dupla
    - Primeiro, rotaciona-se o filho para o lado do desbalanceamento do pai
    - Em seguida, rotaciona-se o pai para o lado oposto do desbalanceamento
      - ED ou DE (com raciocínio inverso)
  - Às vezes, é necessário realocar algum elemento, pois ele perde seu lugar na árvore

Formalmente: ED

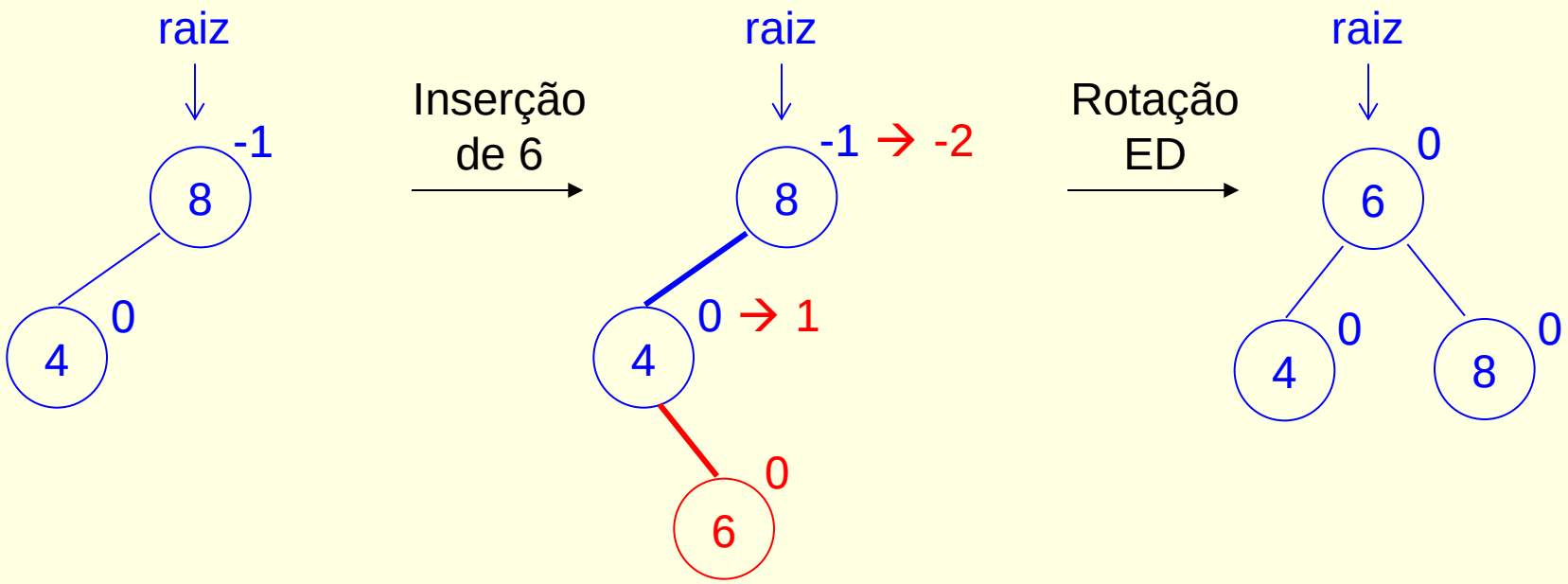
Generalizando





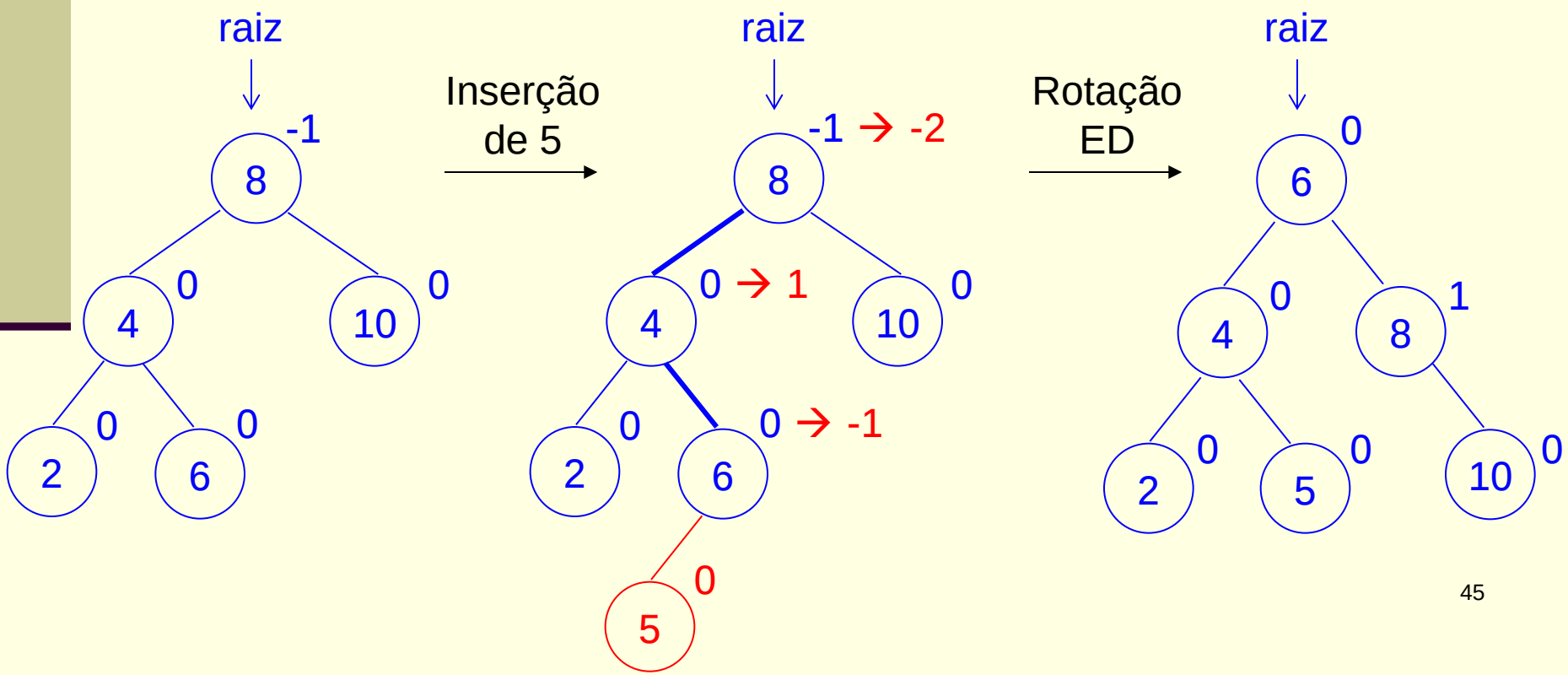
Exercício

- Passo a passo: implementar rotação ED



Exercício

- Testar sub-rotina no caso abaixo



Formalmente: DE

- Raciocínio similar

AVL

- As transformações dos casos anteriores diminuem em 1 a altura da subárvore com raiz desbalanceada p

- Assegura-se o rebalanceamento de todos os ancestrais de p e, portanto, o rebalanceamento da árvore toda

AVL

- Implementar sub-rotina de inserção de elementos na AVL
  - Função principal de inserção
    - Usando sub-rotinas de rotações e conferindo balanceamento recursivamente

AVL

- Exercício: teste a sub-rotina anterior inserindo os elementos **5** e **40** na árvore abaixo