



# Conteúdo

1. Introdução
2. Listas
3. Pilhas e Filas
4. Árvores
5. Árvores de Pesquisa
  - Árvore Binária e **Árvore AVL**
  - Árvore N-ária e Árvore B
6. Tabelas de Dispersão (Hashing)
7. Métodos de Acesso a Arquivos
8. Métodos de Ordenação de Dados





# Árvores AVL





# Árvore Binária de Pesquisa

Uma árvore binária é uma árvore binária de pesquisa ou busca quando:

- todo elemento armazenado na subárvore esquerda é menor que a **raiz**  $r$ ;
- nenhum elemento armazenado na subárvore direita é menor que a **raiz**  $r$ ;
- as subárvores esquerda e direita também são árvores binárias de pesquisa.



# Problema da Árvore de Pesquisa Binária

- O **desbalanceamento progressivo** de uma Árvore de Pesquisa Binária tende a tornar linear a complexidade de pesquisa:

$$O(\log_2 n) \rightarrow O(n)$$

- Exemplo:
  - ordem de inclusão: 1, 13, 24, 27, 56
  - complexidade da pesquisa:  $O(n)$

⇒ Alternativa de solução: **Árvore AVL**

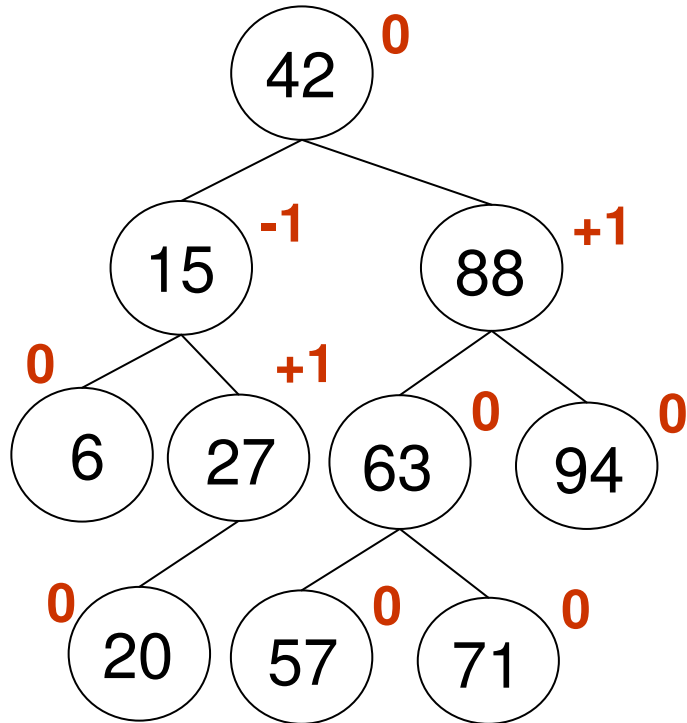
# Árvore AVL

- Uma **árvore AVL**  $A$  é uma Árvore de Pesquisa Binária tal que:

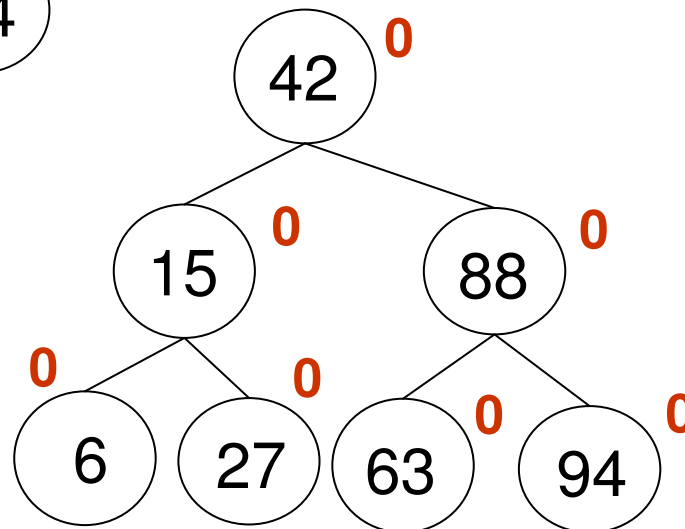
$$\forall \text{ subárvore } A' \in A \ (h(A'.esq) - h(A'.dir) \in [-1, 1])$$

- Uma árvore AVL é uma Árvore de Pesquisa Binária Balanceada
- AVL (Adelson-Velskii e Landis)

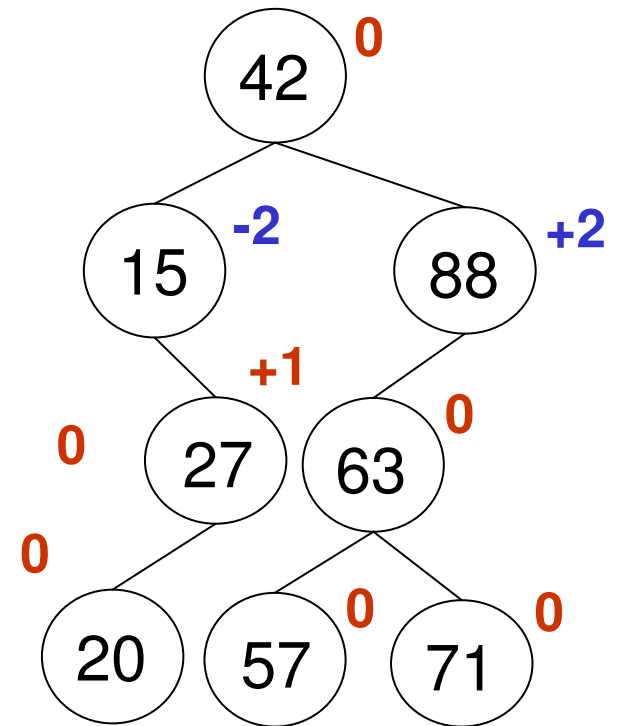
# Exemplos



Árvore AVL



Árvore AVL

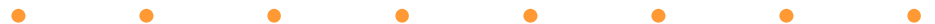


Não é árvore AVL!



# Operação de Rotação

- Como manter uma árvore AVL sempre balanceada após uma inclusão ou exclusão?
  - através de uma operação de Rotação
- Características da operação:
  - preserva a ordem das chaves
  - basta uma execução da operação de rotação para tornar a árvore novamente AVL





# Operação de Rotação

Tipos de rotação:

⇒ Rotação simples:

- para a direita
- para a esquerda

⇒ Rotação dupla:

- para a direita (1 simples para a esq. na subárvore esq. e, depois, 1 simples para a dir.)
- para a esquerda (1 simples para a dir. na subárvore dir. e, depois, 1 simples para a esq.)







## Rotação Simples

Executada toda vez que uma árvore fica desbalanceada com um fatorB:

- **positivo** e sua subárvore ESQ também tem um fatorB positivo  
└─ Rotação Simples para a Direita

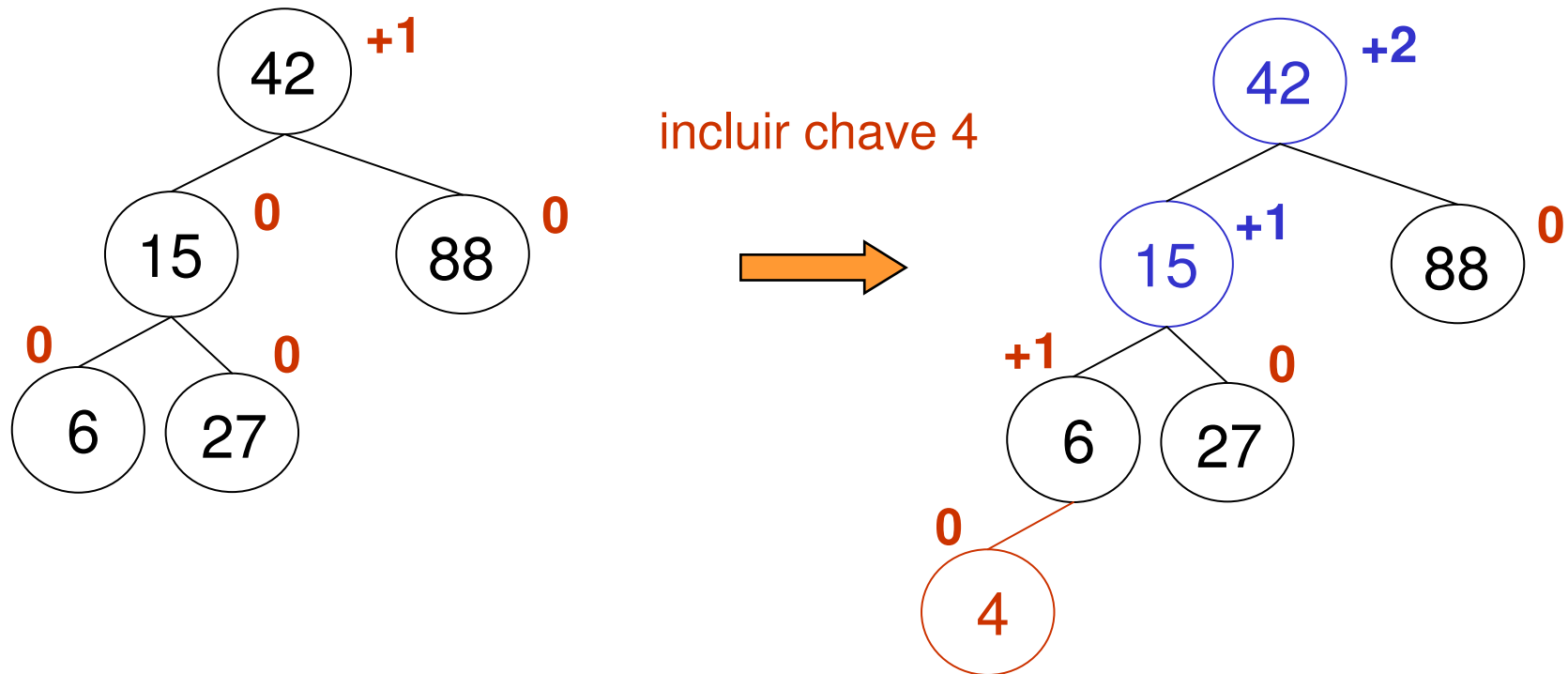
ou

- **negativo** e sua subárvore DIR também tem um fatorB negativo  
└─ Rotação Simples para a Esquerda

⇒ Complexidade:  $O(1)$

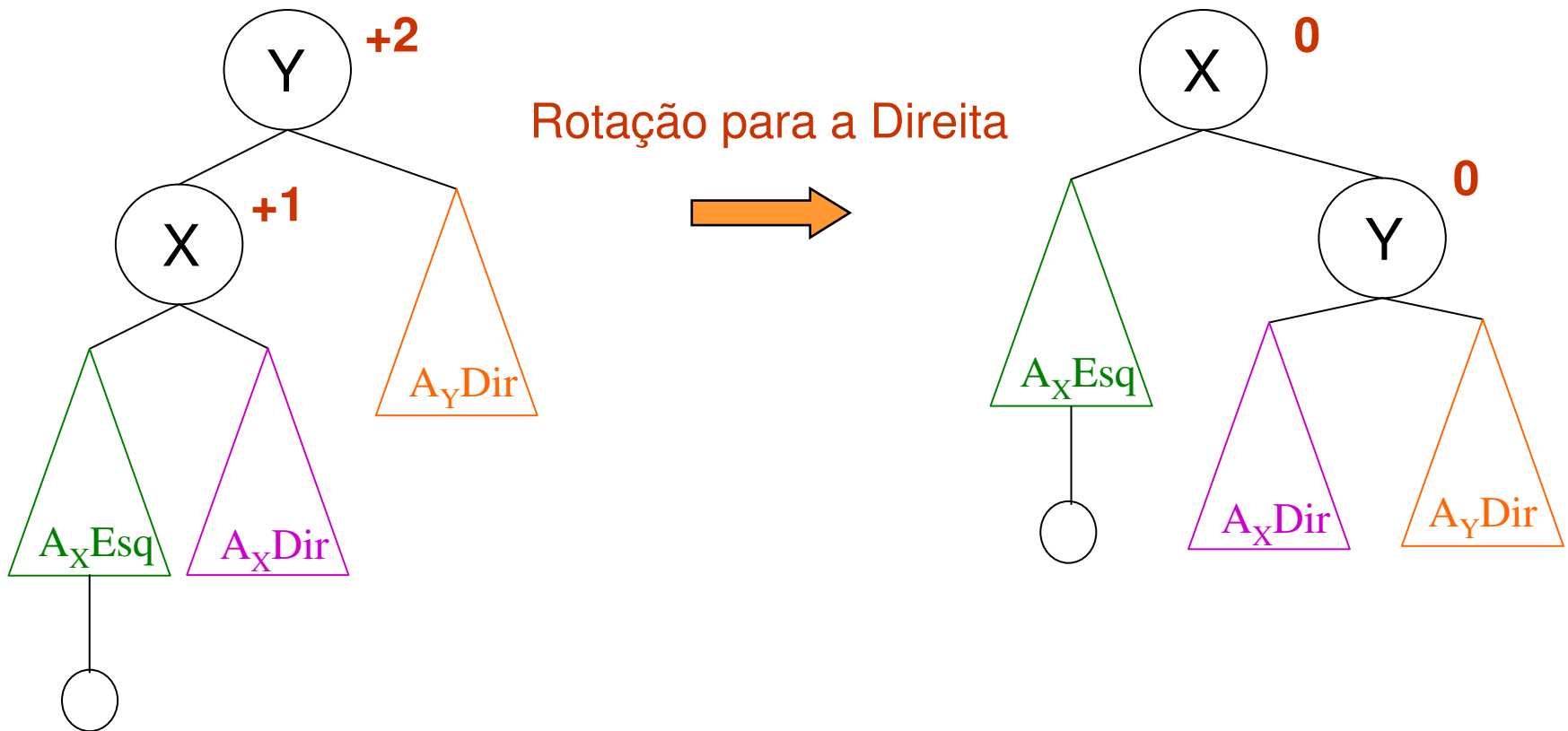


## Rotação Simples para a Direita

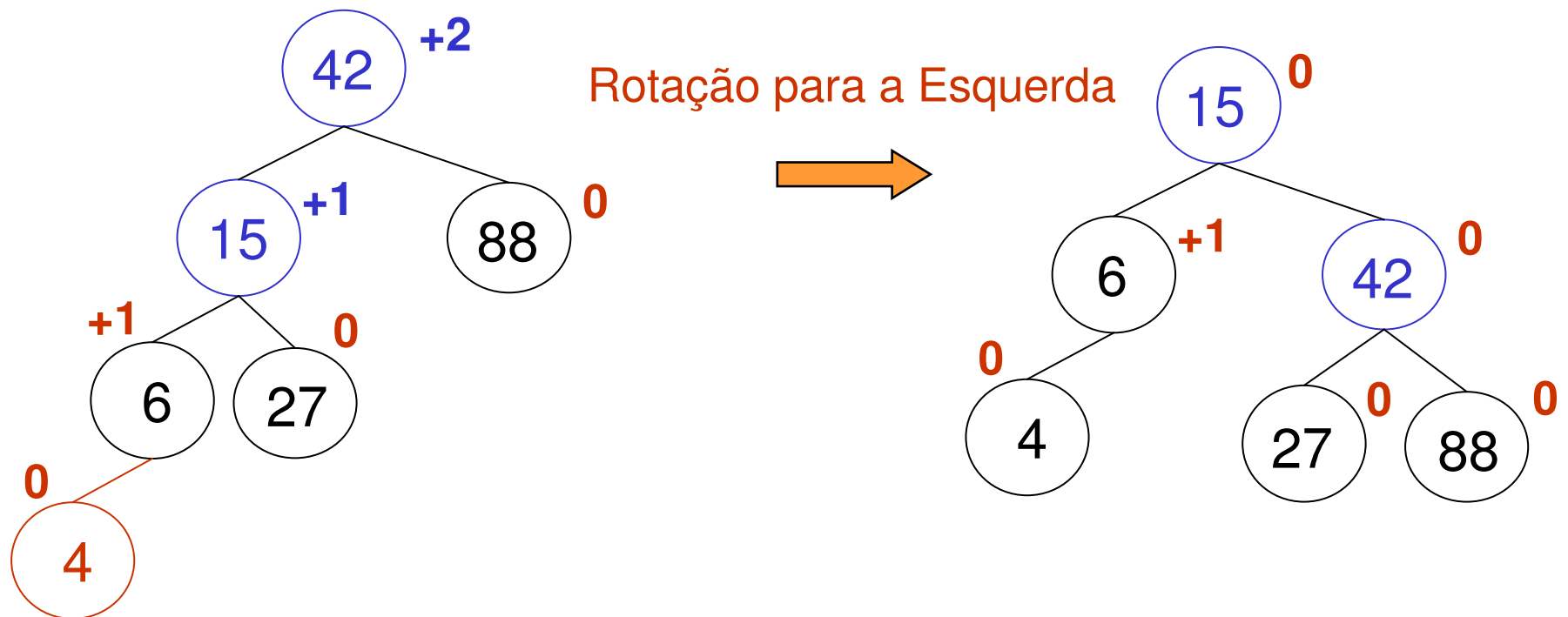


↳ uma rotação simples para a direita é necessária

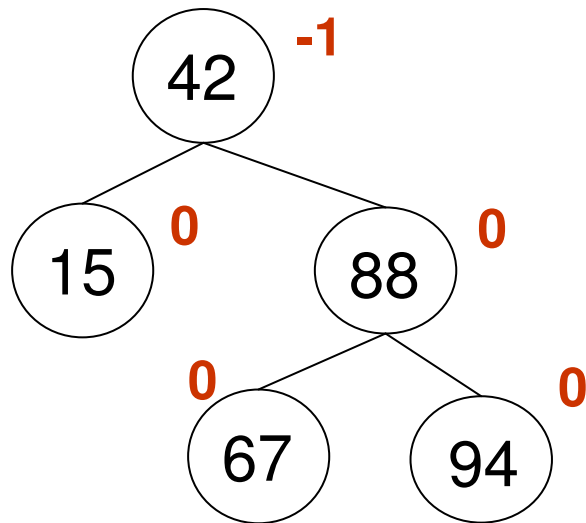
# Rotação Simples para a Direita



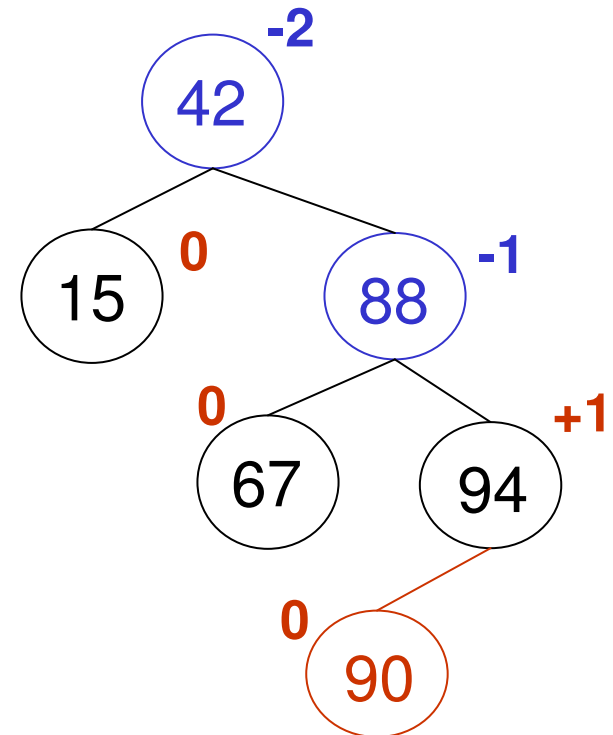
## Exemplo - Rotação Simples para a Direita



## Rotação Simples para a Esquerda



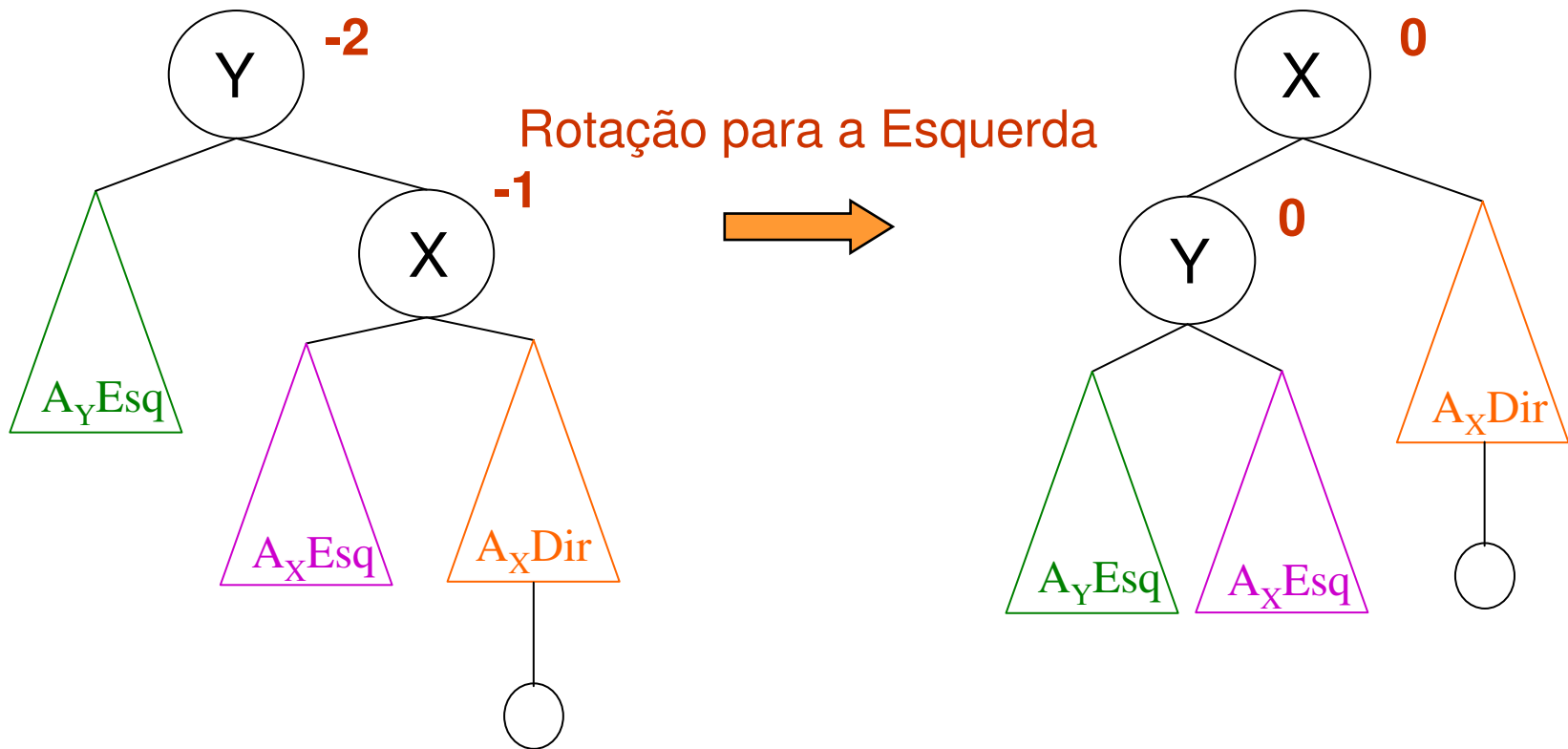
incluir chave 90



uma rotação simples para a esquerda é necessária

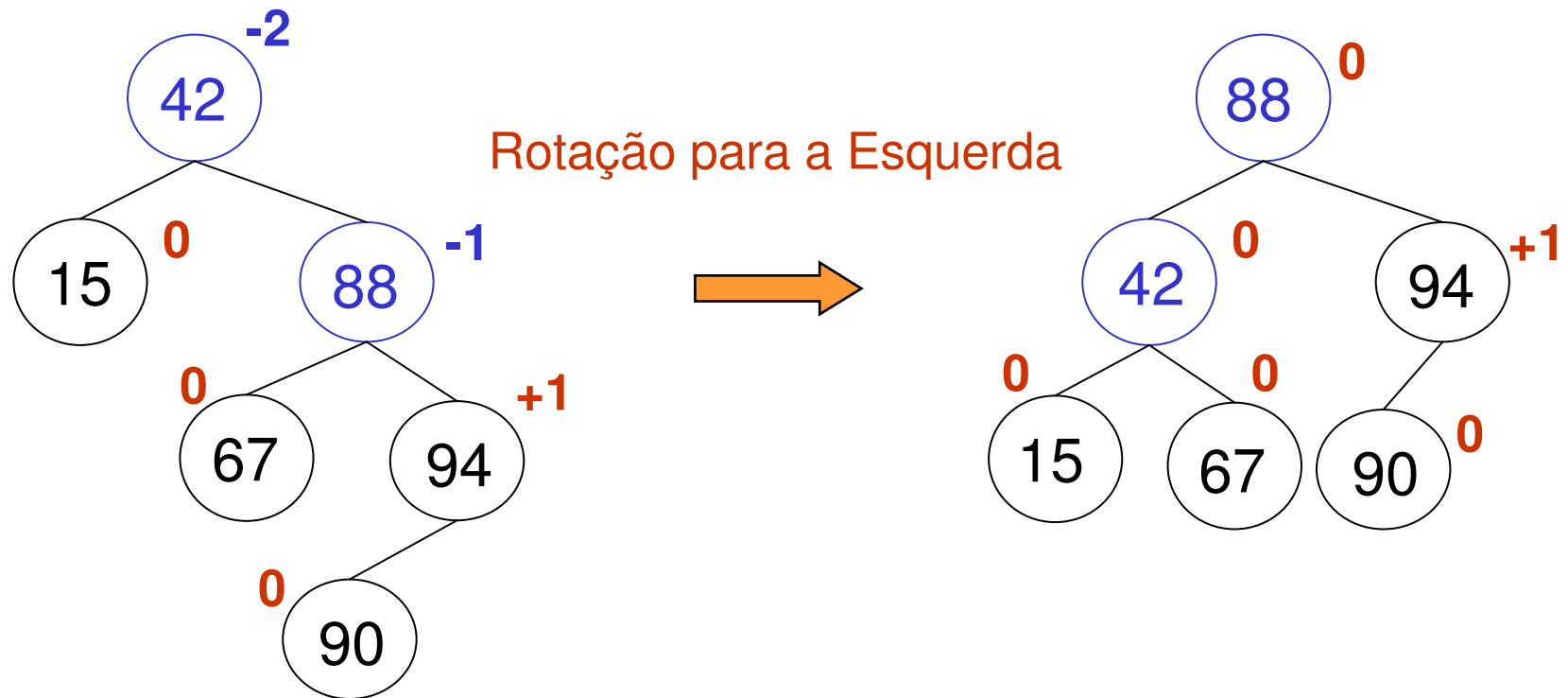
...

# Rotação Simples para a Esquerda



.....

## Exemplo - Rotação Simples para a Esquerda



# Rotação Dupla

Executada toda vez que uma árvore fica desbalanceada com um fatorB:

- **positivo** e sua subárvore ESQ tem um *fatorB* negativo

└ Rotação Dupla para a Direita: 1 simples para a esq. na subárvore esq. e, depois, 1 simples para a dir.

ou

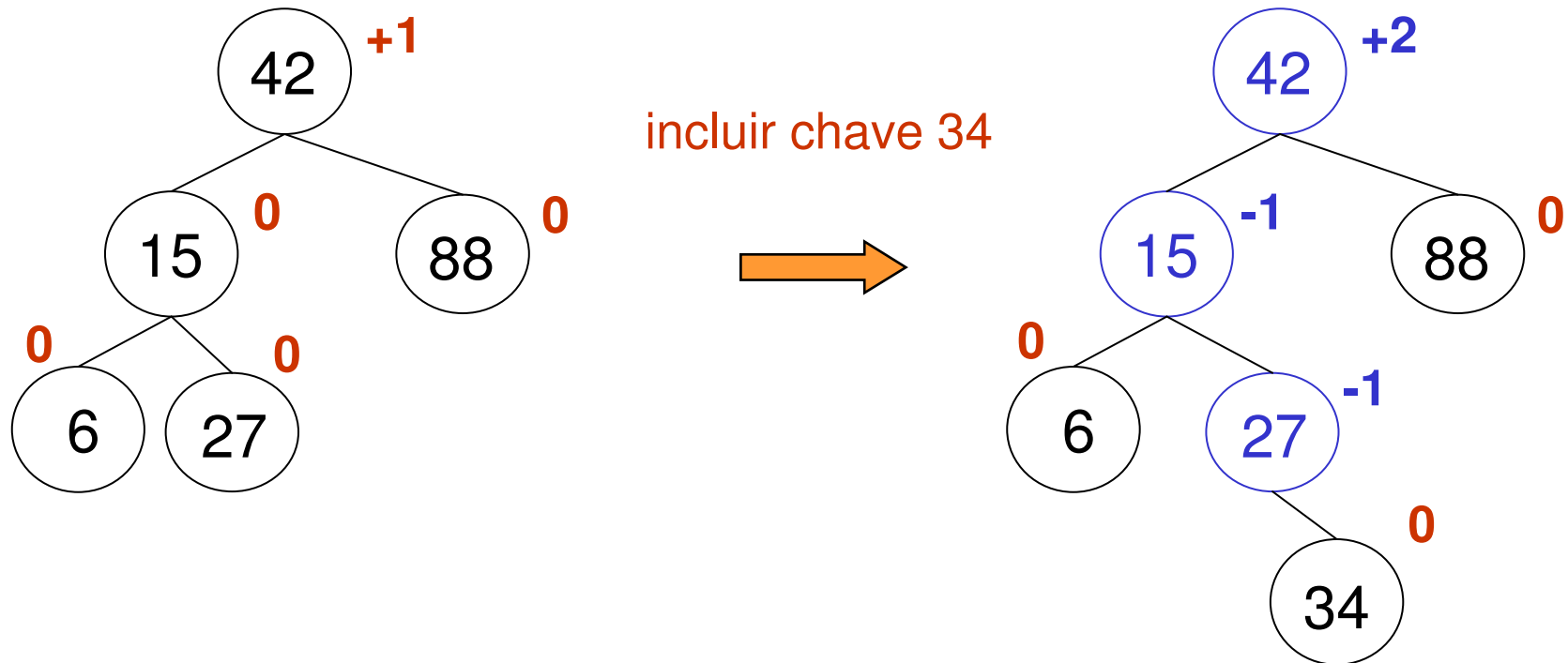
- **negativo** e sua subárvore DIR tem um *fatorB* positivo

└ Rotação Dupla para a Esquerda: 1 simples para a dir. na subárvore dir. e, depois, 1 simples para a esq.

⇒ Complexidade:  $O(1)$

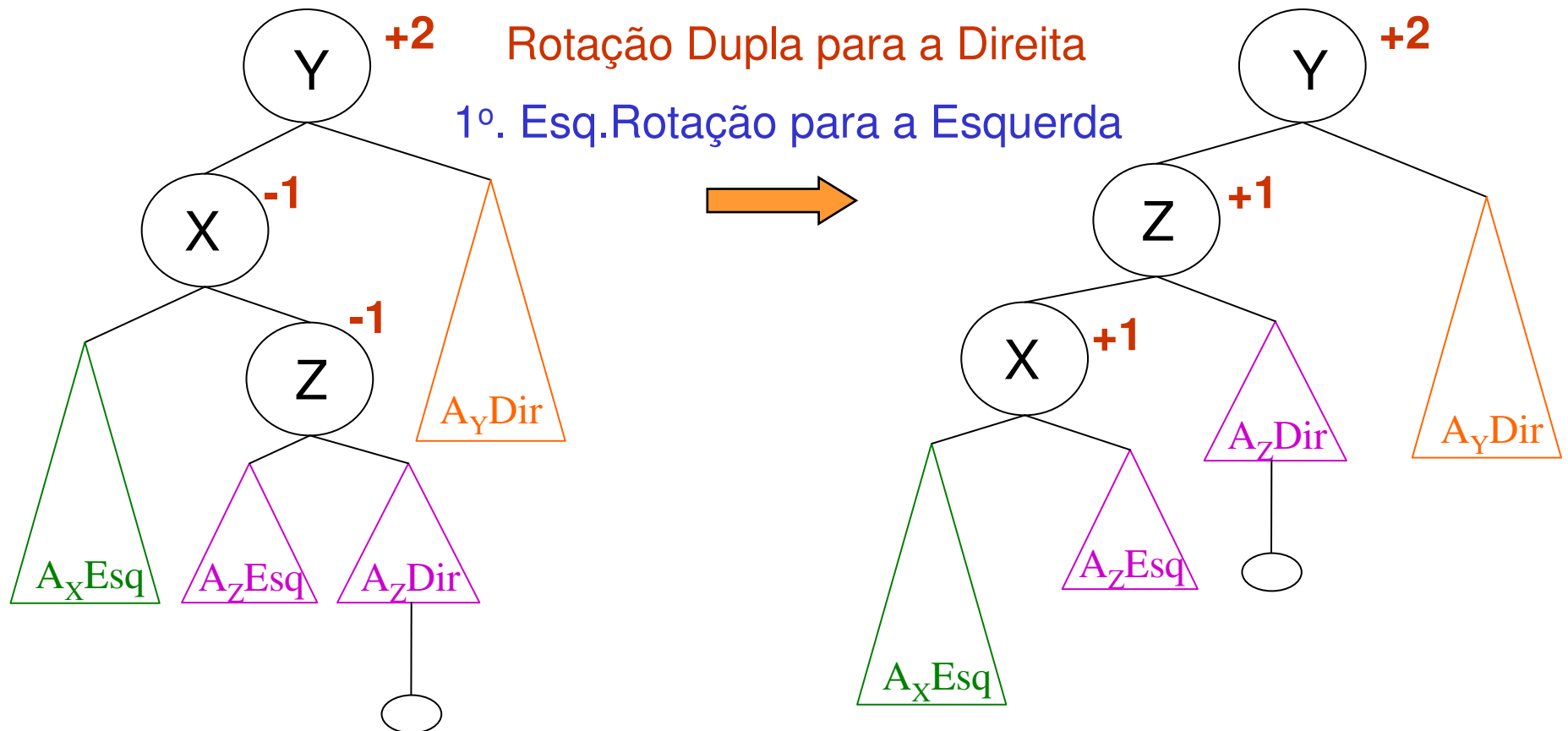


## Rotação Dupla para a Direita



↳ uma rotação dupla para a direita é necessária

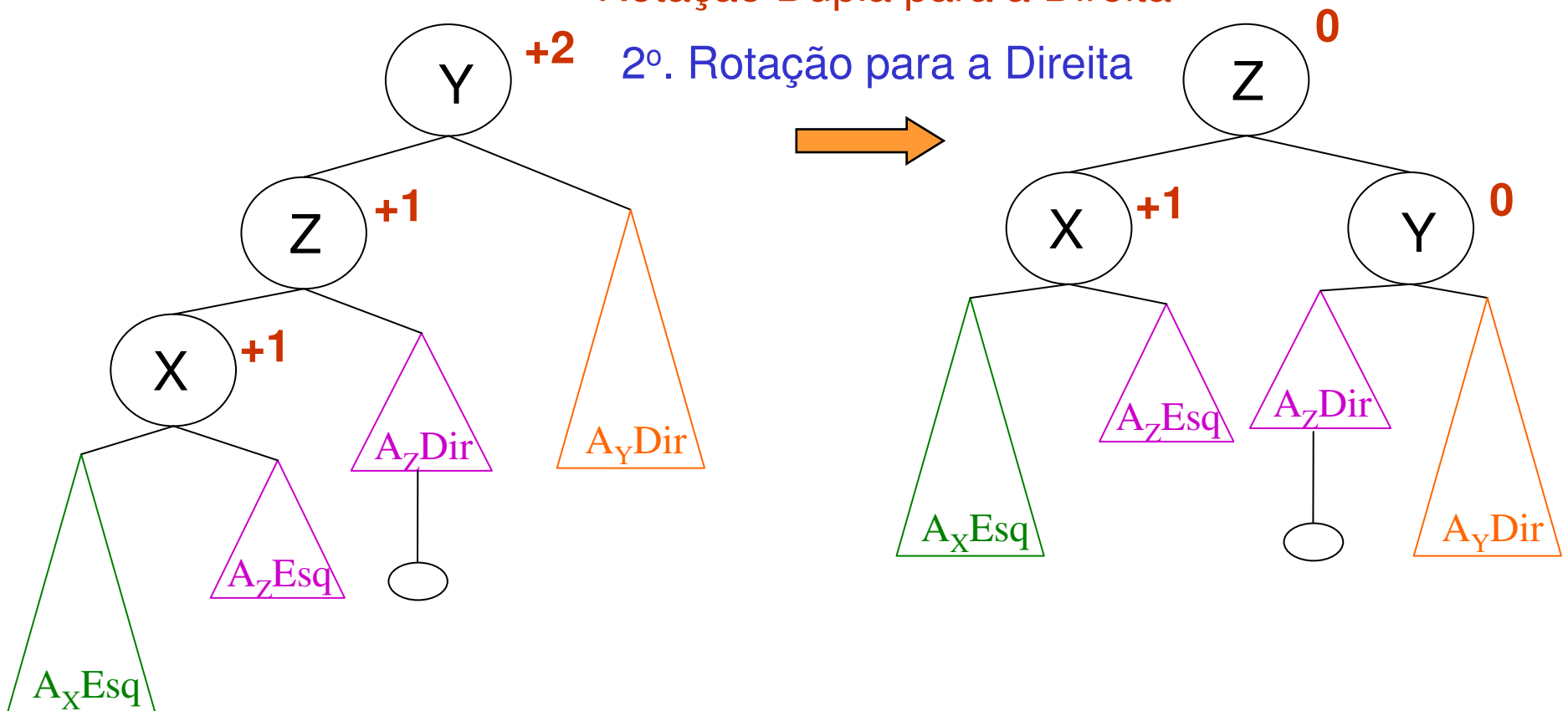
# Rotação Dupla para a Direita



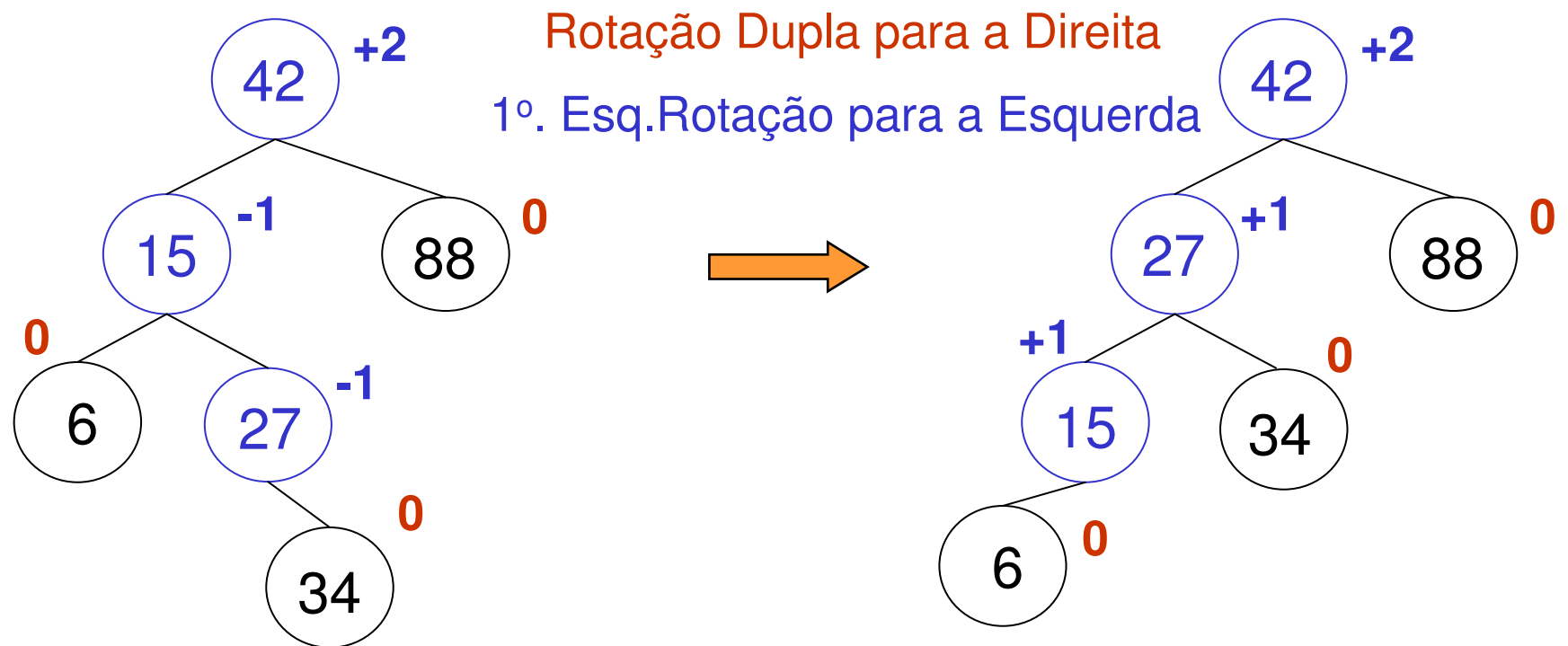
# Rotação Dupla para a Direita

Rotação Dupla para a Direita

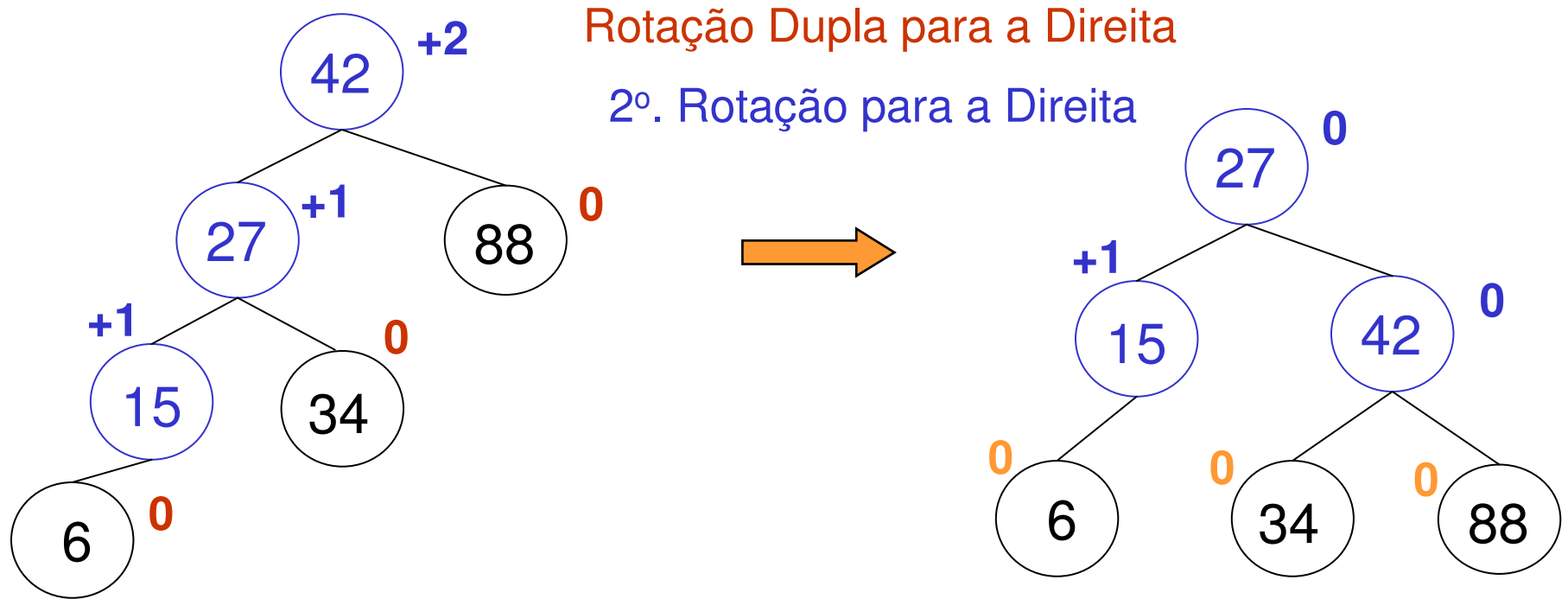
2º. Rotação para a Direita



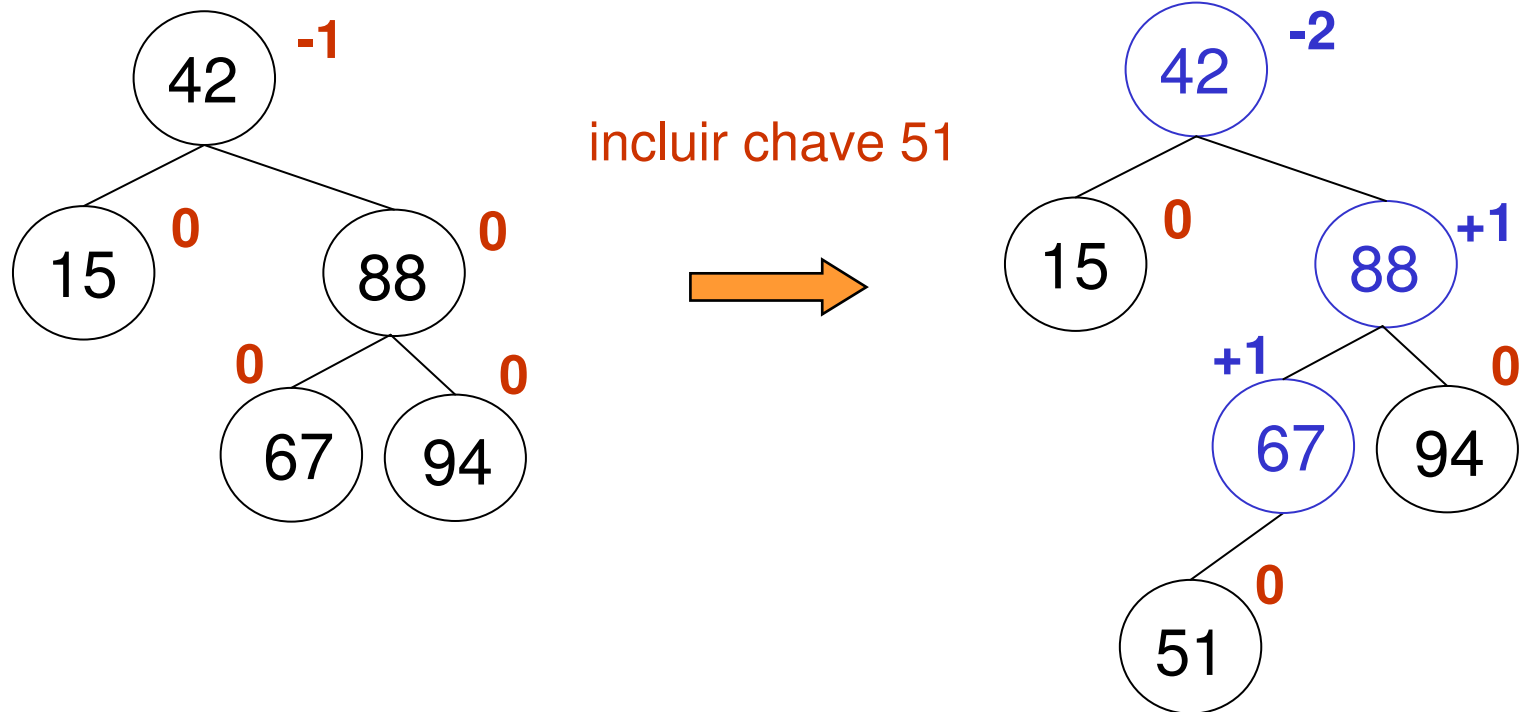
## Exemplo - Rotação Dupla para a Direita



## Exemplo - Rotação Dupla para a Direita

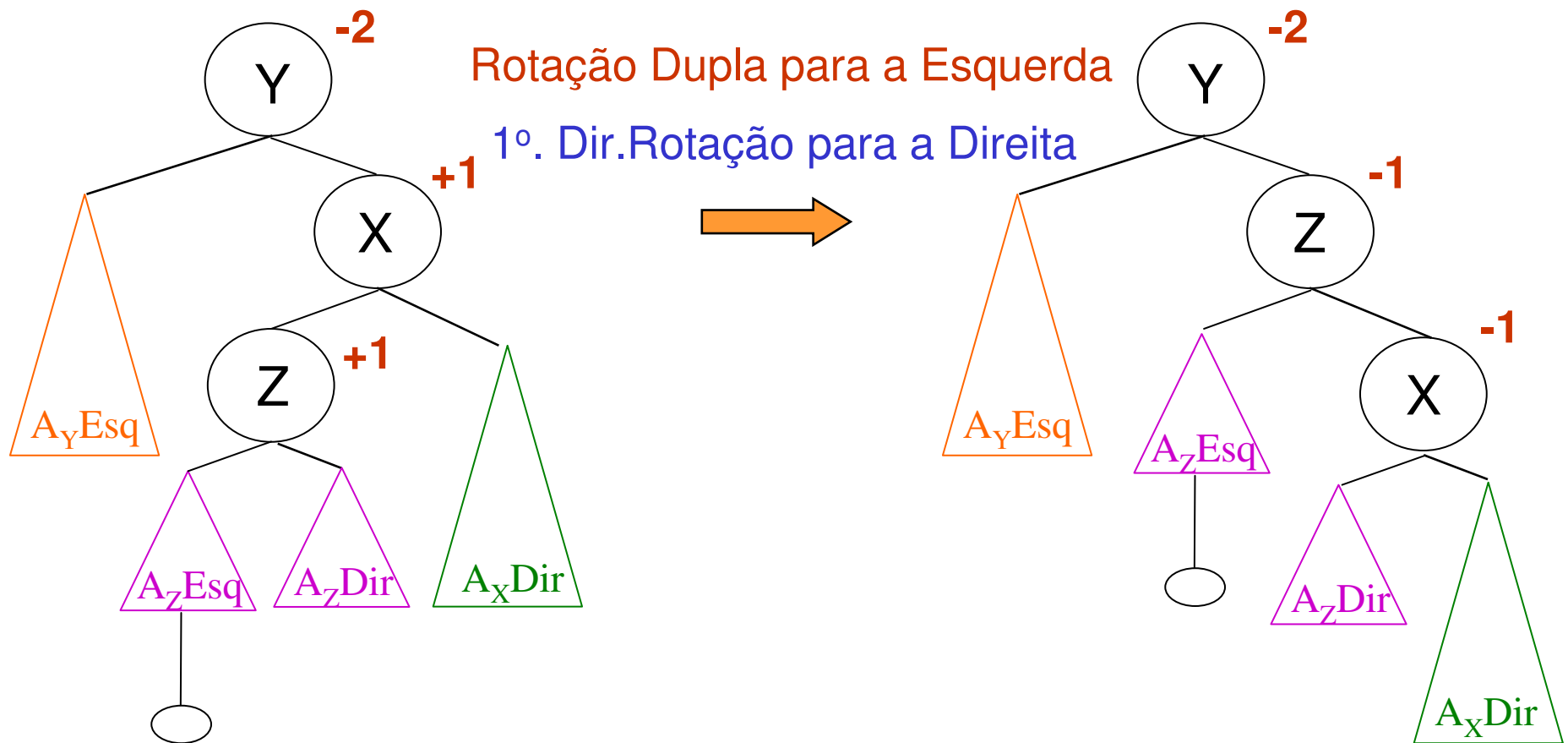


## Exemplo - Rotação Dupla para a Esquerda

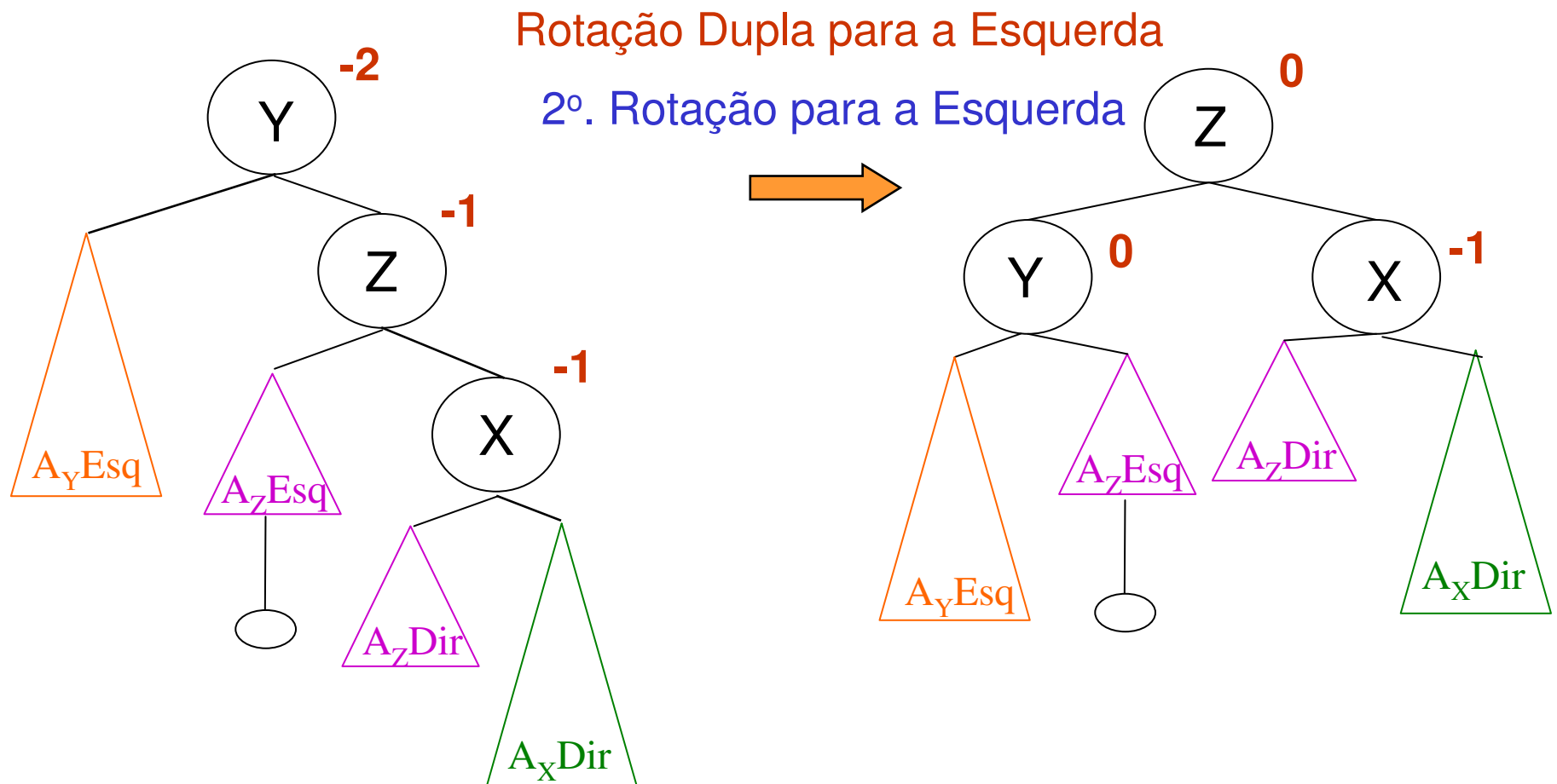


↳ uma rotação dupla para a esquerda é necessária

# Rotação Dupla para a Esquerda



# Rotação Dupla para a Esquerda

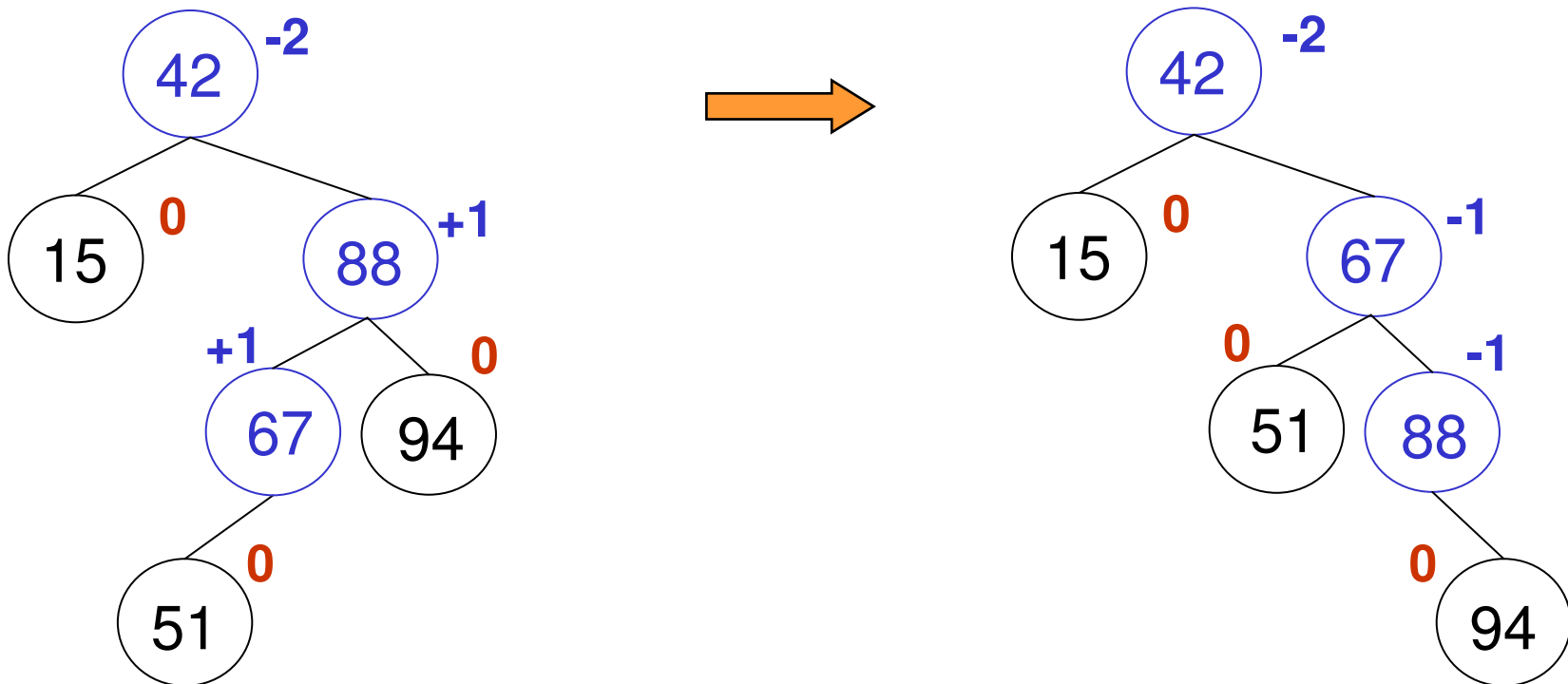




## Exemplo - Rotação Dupla para a Esquerda

Rotação Dupla para a Esquerda

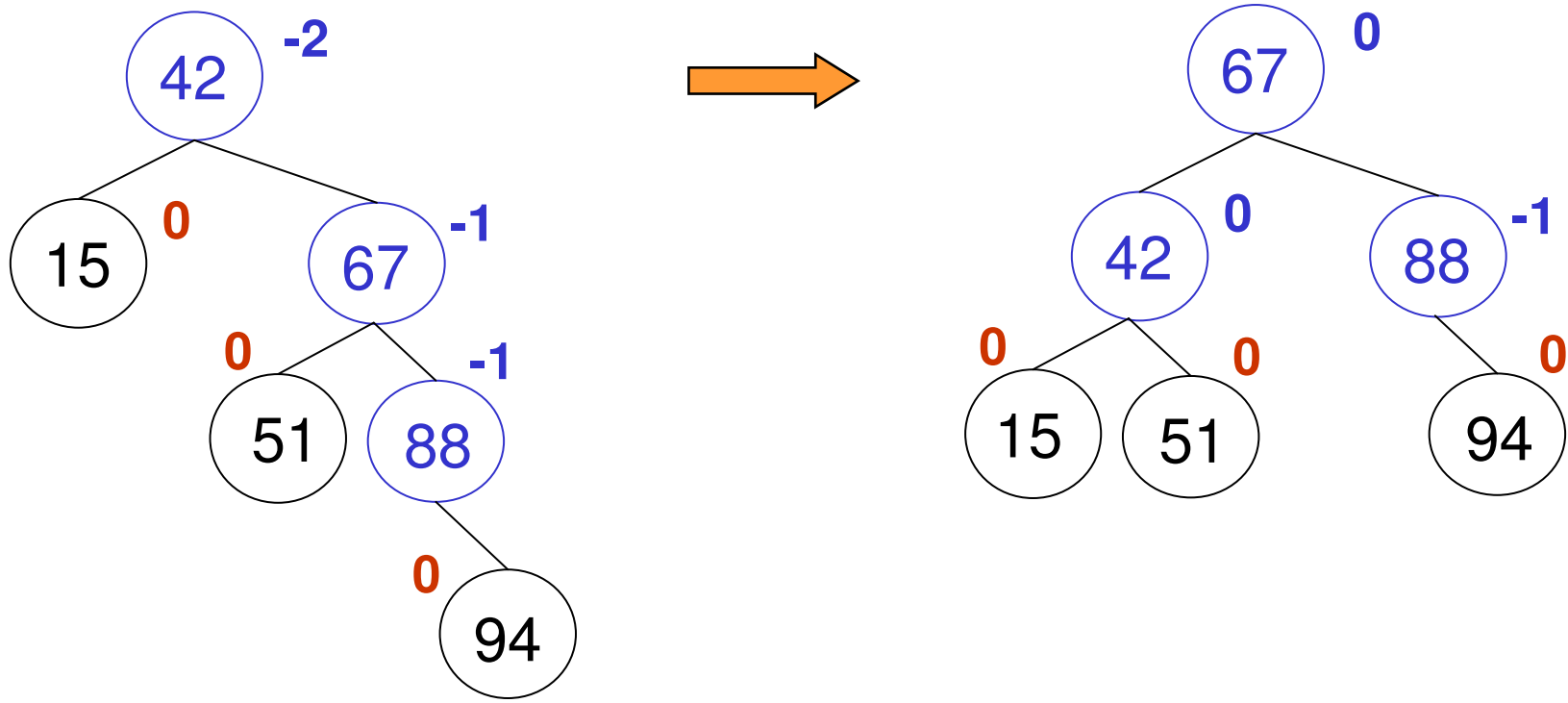
1º. Dir.Rotação para a Direita



## Exemplo - Rotação Dupla para a Esquerda

Rotação Dupla para a Esquerda

2º. Rotação para a Esquerda





# Balanceamento

Quando realizar o balanceamento de uma árvore AVL?

- Sempre que a árvore apresentar um fatorB fora do intervalo  $[-1,1]$ .
- O valor do fatorB deve ser testado sempre **após** uma operação de **inclusão** ou **exclusão** de nodo na árvore.





## Interface da Árvore AVL

```
public interface ArvoreAVL<C extends Comparable<C>,V> extends  
    ArvoreBinariaPesquisa<C,V> {  
  
}
```

