CENTRO DE CIÊNCIA E TECNOLOGIA
LABORATÓRIO DE CIÊNCIAS MATEMÁTICAS
UNIVERSIDADE ESTADUAL DO NORTE FLUMINENSE

Arvores AVL

Casos de Balanceamento

Inserção de nós

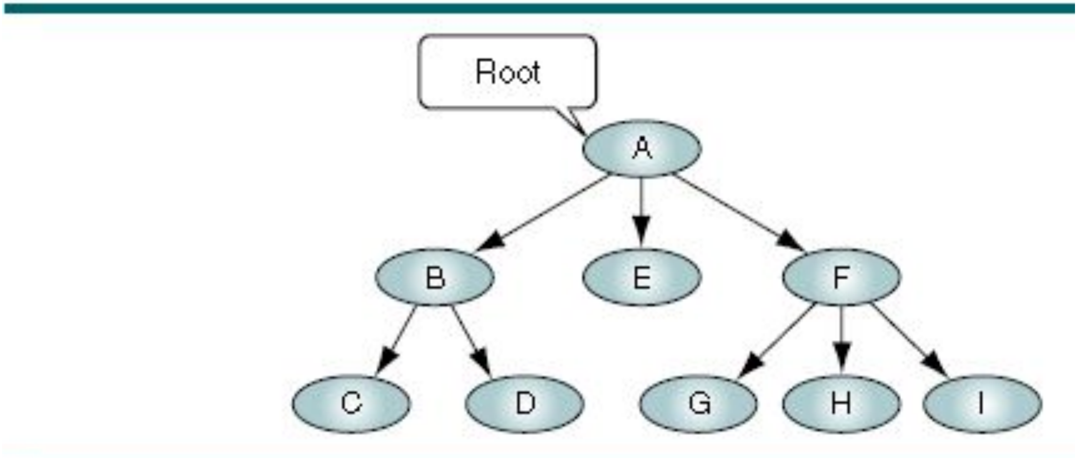
Disciplina: Estrutura de Dados II

Prof. Fermín Alfredo Tang Montané

Curso: Ciência da Computação

Árvores
Conceitos Básicos

- Uma árvore consiste de um conjunto finito de elementos, chamados **nós**, e um conjunto finito de conexões direcionadas, chamadas de **ramos**, que conectam os nós.

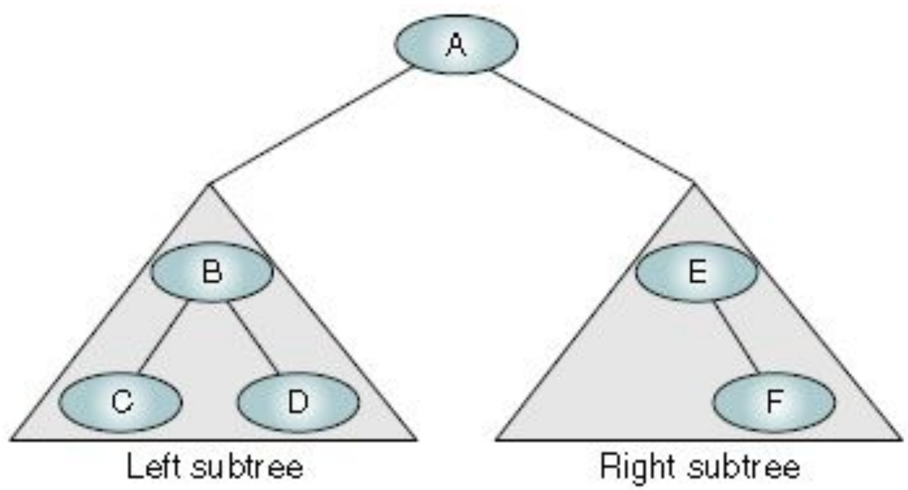


Tree

- O número de ramos associados a um nó é chamado de **grau do nó**. O grau de um nó pode ser decomposto em grau de entrada e grau de saída.
- O grau de entrada de um nó, corresponde ao número de ramos que entram nesse nó (ou apontados para esse nó)
- O grau de saída de um nó, corresponde ao número de ramos que saem desse nó.

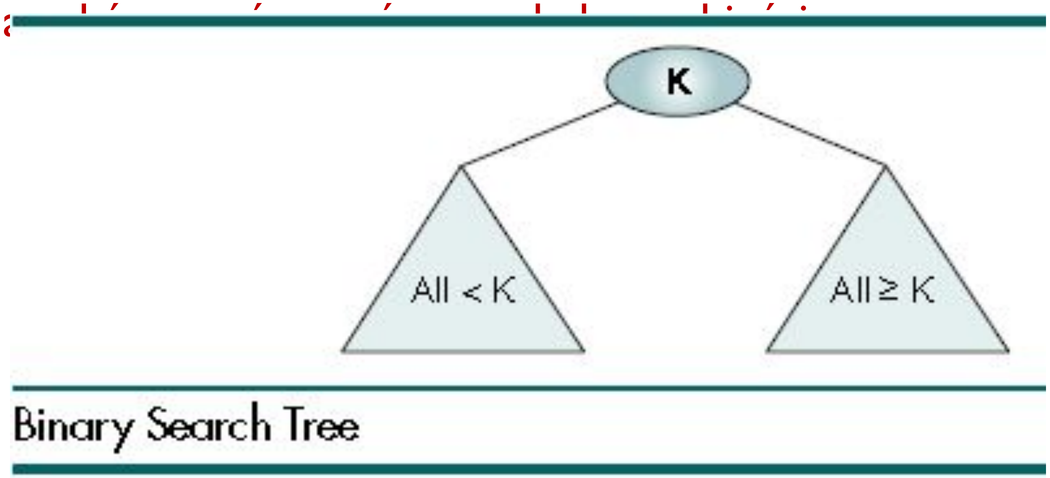
Árvores Binárias
Definição

- Uma **árvore binária** é uma árvore na qual cada nó pode ter até duas subárvores. O grau de saída de cada nó pode ser no máximo dois.
- Assim, cada nó pode ter zero, um ou duas subárvores.
- As subárvores são designadas como sub-arvore esquerda e direita.
- Na figura, mostra-se uma árvore binária com duas subárvores. Note que cada sub-árvore é pela sua vez uma árvore binária.



Árvore de Busca Binária
Definição

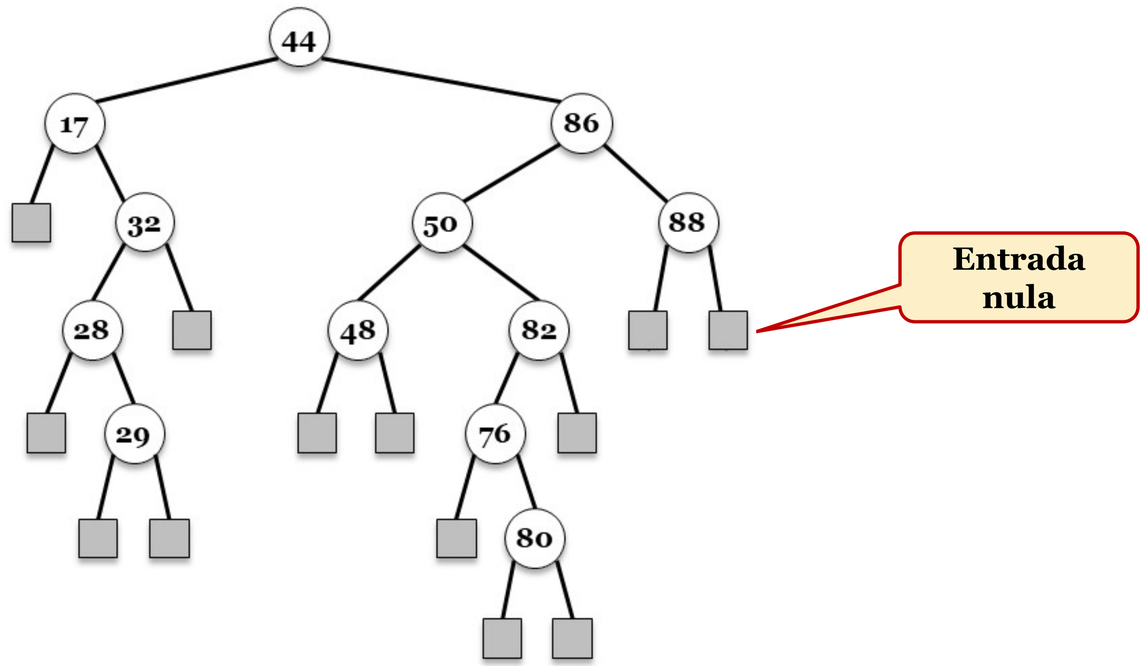
- Uma **Árvore de Busca Binária** (*Binary Search Tree BST*) é uma árvore binária que possui as seguintes propriedades:
  - Todos os elementos na subárvore esquerda são menores que a raiz;
  - Todos os elementos na subárvore direita são maiores ou iguais que a raiz;
  - Cada



- Em geral, a informação armazenada em cada nó é uma estrutura que compreende vários campos de dados. Um destes campos de dados deve ser utilizado como campo chave para realizar comparações entre os nós e decidir se um nó é menor ou maior ou igual que outro.

Árvore de Busca Binária
Exemplo

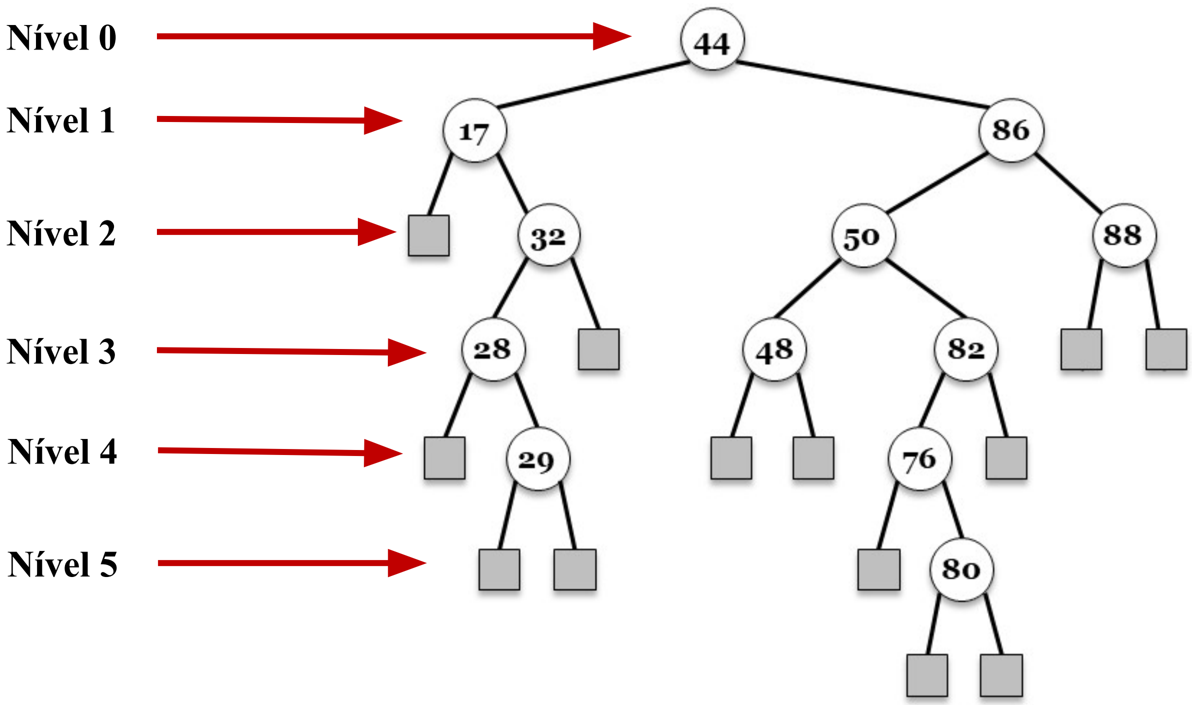
- Uma árvore de busca binária organiza um conjunto de informações com base em um campo chave que ser para identificar o conjunto.
- Cada nó pode ter até dois filhos, um filho esquerdo e um filho direito. Caso algum desses filhos não exista, cria-se uma entrada nula.



- Em algumas implementações, a entrada nula é representada como um nó sem informações. Já, é mais comum considerar essa entrada como um ponteiro nulo.

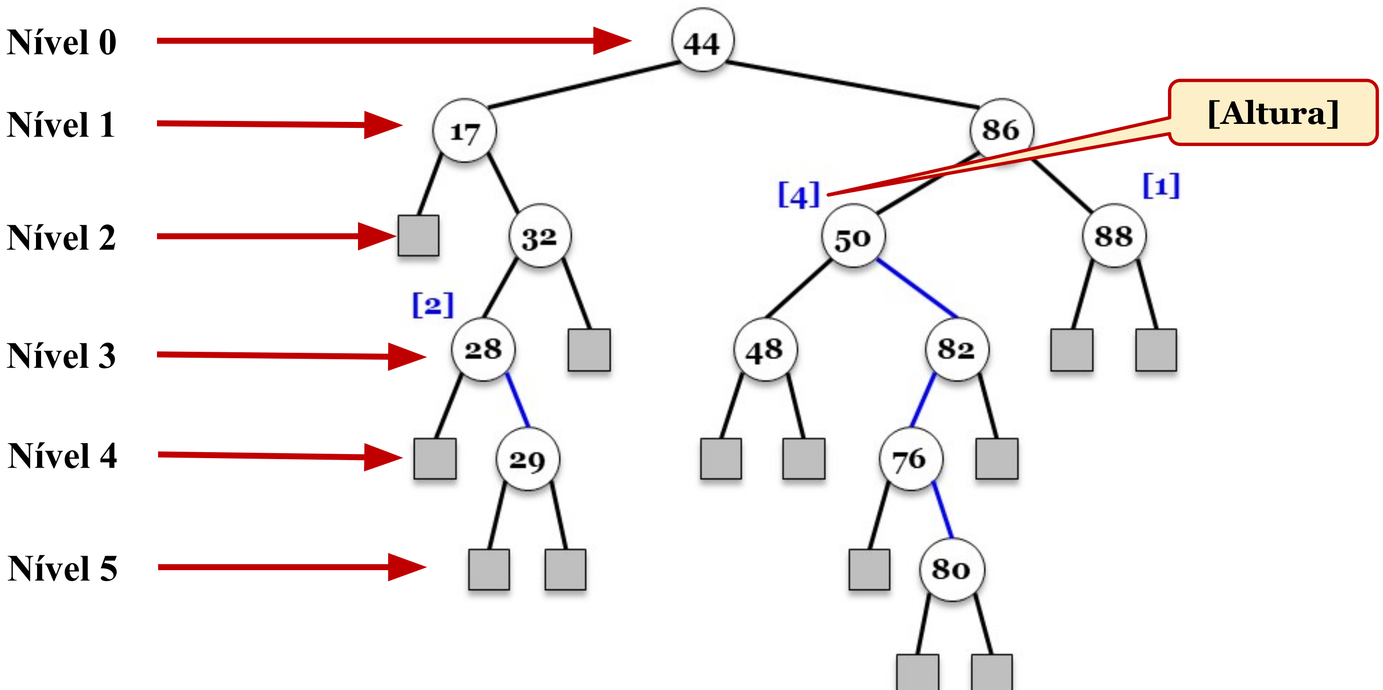
Árvore de Busca Binária
Nível dos nós

- O **nível de um nó.-** Número que mede a distância de um nó desde a raiz. Como a raiz tem distância zero desde si própria, a raiz se encontra no nível 0.



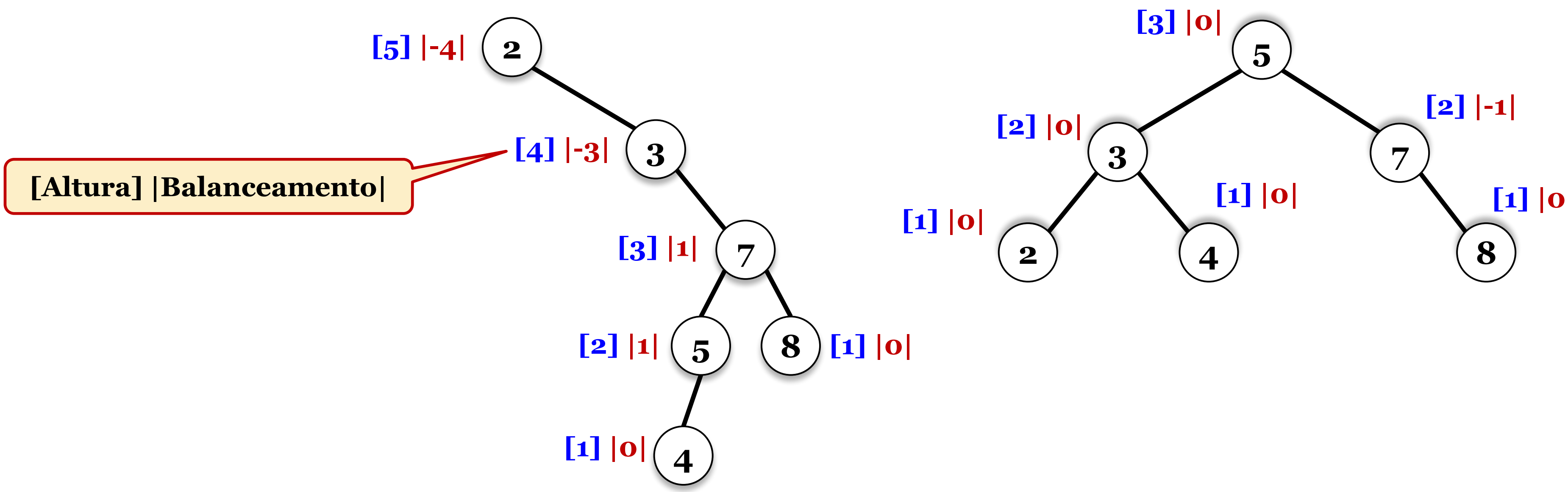
Árvore de Busca Binária
Altura de um nó

- **Altura de um nó.-** A altura de um nó v em uma árvore corresponde número de níveis no caminho de maior comprimento do nó v até uma folha.



Árvore de Busca Binária
Problema do Balanceamento

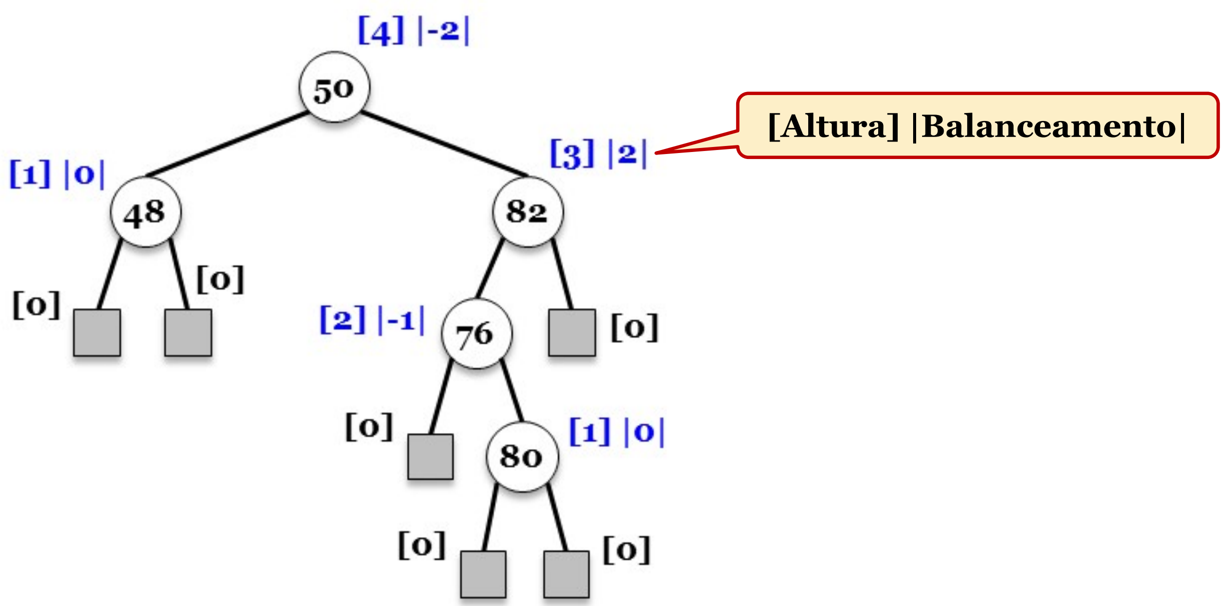
- O principal problema de uma árvore de busca binária é que como resultado de uma sequência de inserções e/ou remoções a árvore pode ficar desbalanceada. Ou seja, para pelo menos algum nó da árvore a altura da sua sub-árvore esquerda é muito diferente da altura da sua sub-arvore direita.



- **Fator de balanceamento de um nó.-** Corresponde a diferença de alturas das subárvores esquerda e direita do nó.

Árvores de Busca Binária
Fator de Balanceamento

- **Fator de Balanceamento.-** Para cada nó da árvore, calcula-se o valor absoluto da diferença das alturas dos filhos esquerdo e direito.



Árvore de Busca Binária
Problema do Balanceamento

- Mas porque esta diferença de alturas é um problema?
- Lembre que o esforço necessário para realizar qualquer operação em uma árvore, seja uma busca, inserção ou remoção depende da altura da árvore.
- Esta altura H no caso de uma árvore binária com N elementos pode ser como mínimo $H_{min} = \lfloor log_2 N \rfloor + 1$ e como máximo $H_{max} = N$.
- Acontece que para N suficientemente grandes $log_2 N$ é um número muito pequeno se comparado com N.

- Por exemplo, se todos os níveis estão preenchidos:

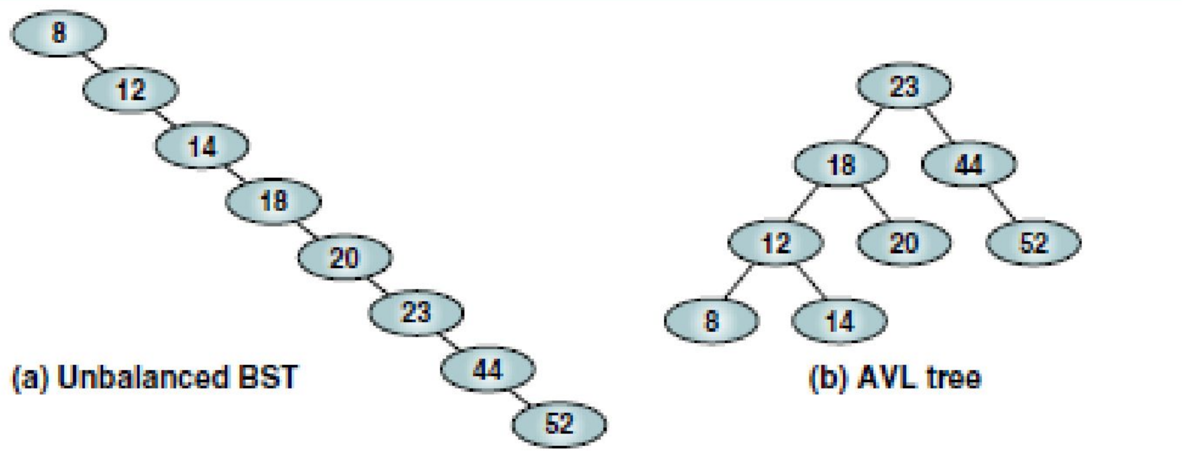
$$N = 2^H - 1$$

$$H = \log_2(N+1)$$

1	1	1
3	2	3
7	3	7
15	4	15
31	5	31
1.023	10	1.023
32.767	15	32.767
1.048.575	20	1.048.575
33.554.431	25	33.554.431
1.073.741.823	30	1.073.741.823

Árvore de Busca Binária
Desbalanceamento: Caso Extremo

- No caso extremo de desbalanceamento a árvore binária se torna uma lista.

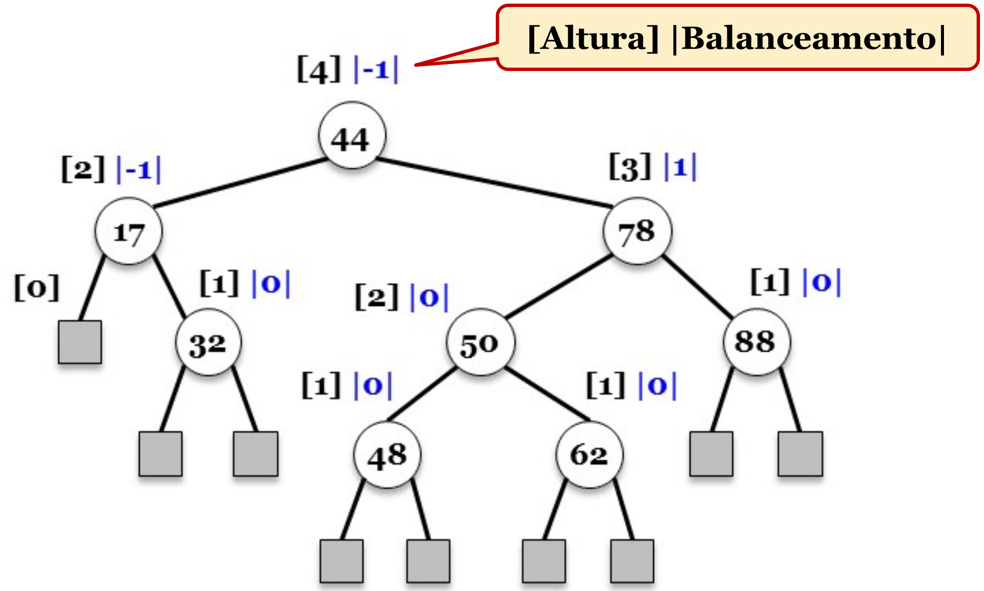


Two Binary Trees

Árvores AVL
Definição

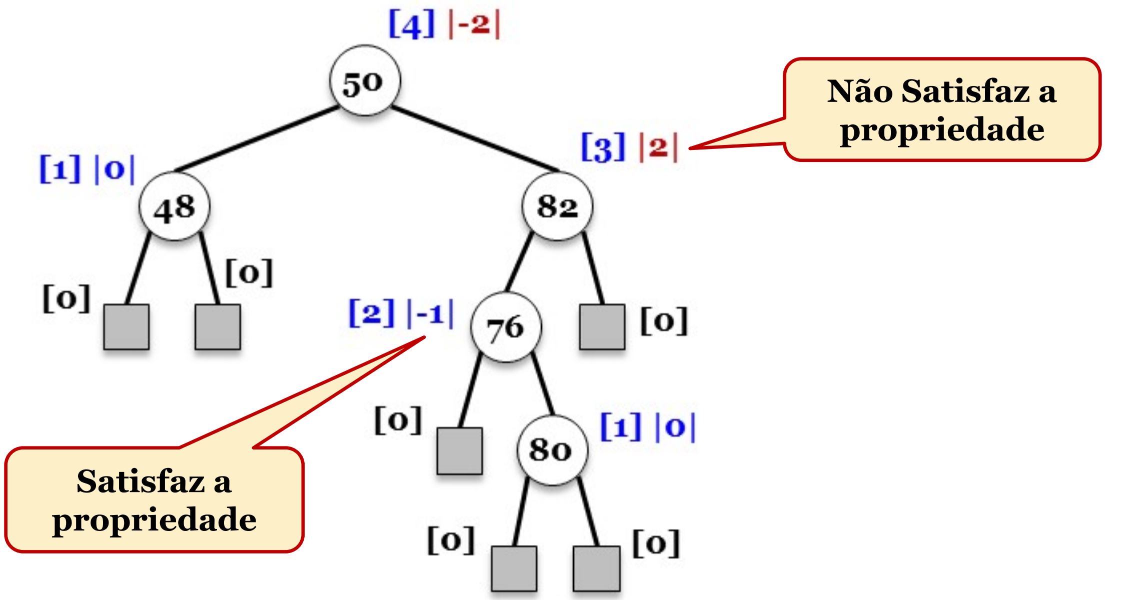
- Em 1962, dois matemáticos russos, Adelson-Velskii e Landis propuseram um tipo de árvore binária balanceada, assim como os mecanismos para preservar esse balanceamento.
- Trata-se de uma árvore de busca binária na qual cada nó satisfaz uma propriedade de balanceamento entre as alturas das suas subárvores esquerda e direita. Tal balanceamento garante que a árvore possua altura de ordem logarítmica com relação ao número de nós.

- **Propriedade de balanceamento.-** Para cada nó v da árvore T, a altura dos filhos de v diferem em no máximo uma unidade.



Árvores AVL
Propriedades

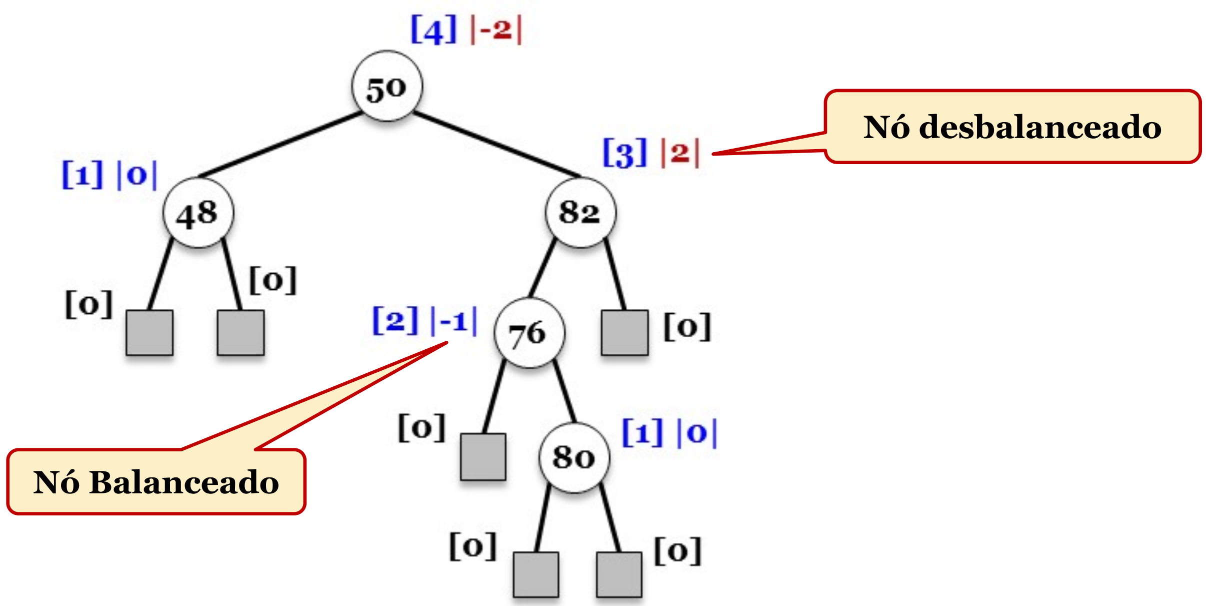
- **Propriedade de balanceamento.-** Para cada nó v da árvore T, a altura dos filhos de v **diferem em no máximo uma unidade**.



- Toda subárvore de uma árvore AVL também é uma árvore AVL.
- A altura de uma árvore AVL com N nós é de ordem $O(log_2 \text{ N})$.

Árvores AVL
Nós Balanceados e Desbalanceados

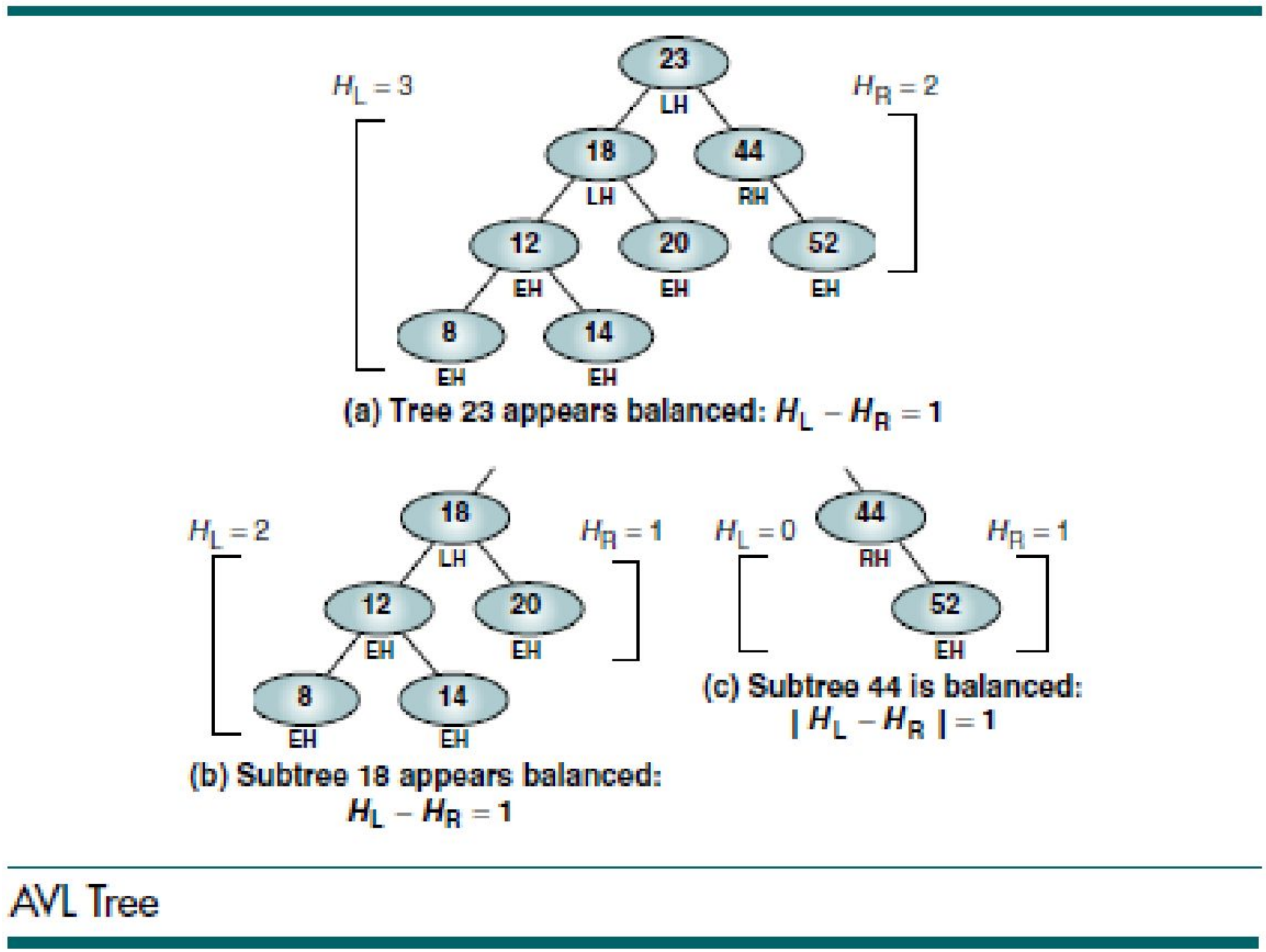
- Dada uma árvore binária T, se diz que um nó v de T está balanceado se o valor absoluto da diferença entre as alturas dos filhos de v, é no máximo igual a um. Caso contrário, se diz que o nó está desbalanceado.
- Esta propriedade de balanceamento de altura caracteriza as árvores AVL.



Arvores AVL (AVL Trees)

Fator de Balanceamento

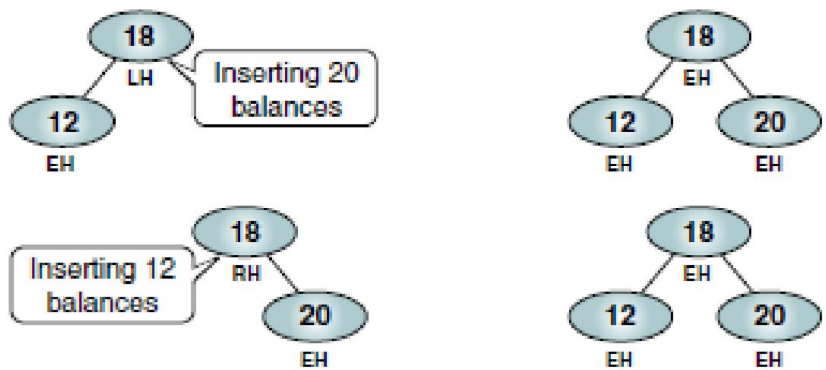
- O fator de balanceamento em árvores AVL também é representado pelas constantes LH, RH e EH (*Left Height*, *Right Height* e *Equal Height*), que indicam se um nó tem altura esquerda predominante, altura direita predominante ou alturas iguais nas duas subárvores.



Arvores AVL (AVL Trees)

Observação

- Foi mencionado que as possíveis causas do desbalanceamento de uma árvore AVL são as inserções ou remoções de nós. No entanto, é importante mencionar que nem todas as inserções geram desbalanceamento, assim como nem todas as remoções geram desbalanceamento.
- Por exemplo, as duas inserções mostradas na figura abaixo, não causam desbalanceamento. Pelo contrário, causam um balanceamento perfeito.

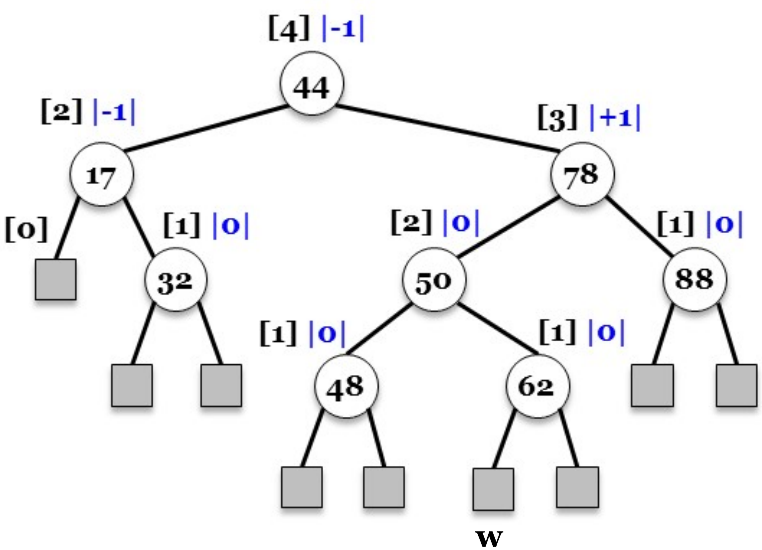


Automatic AVL Tree Balancing

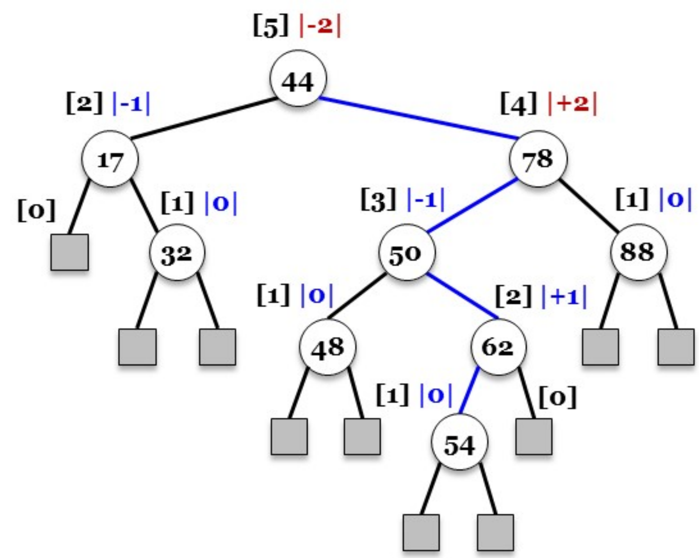
Árvores AVL
Inserção

- A inserção de um novo nó na árvore T, acontece sempre em um nó folha.
- Tal operação pode violar a propriedade de balanceamento de altura, ao incrementar a altura de alguns nós ancestrais de w (no caminho de w até a raiz). Neste caso é preciso restaurar o balanceamento da altura.

- Ilustra-se o caso da inserção do nó 54.



Árvore Inicial



Árvore Após Inserção

Arvores AVL (AVL Trees)

Desbalanceamento por Inserção

- Os casos de desbalanceamento por inserção somente acontecem quando um novo nó é inserido na subárvore de maior altura de um nó z, fazendo com que essa subárvore aumente de altura. Temos duas possibilidades:
  - Se um nó z é LH, tem altura esquerda, ou seja a subárvore esquerda é maior e a inserção do novo nó acontece nesta subárvore.
  - Ou, se um nó z é RH, tem altura direita, ou seja a subárvore direita é maior e a inserção do novo nó acontece nesta subárvore.

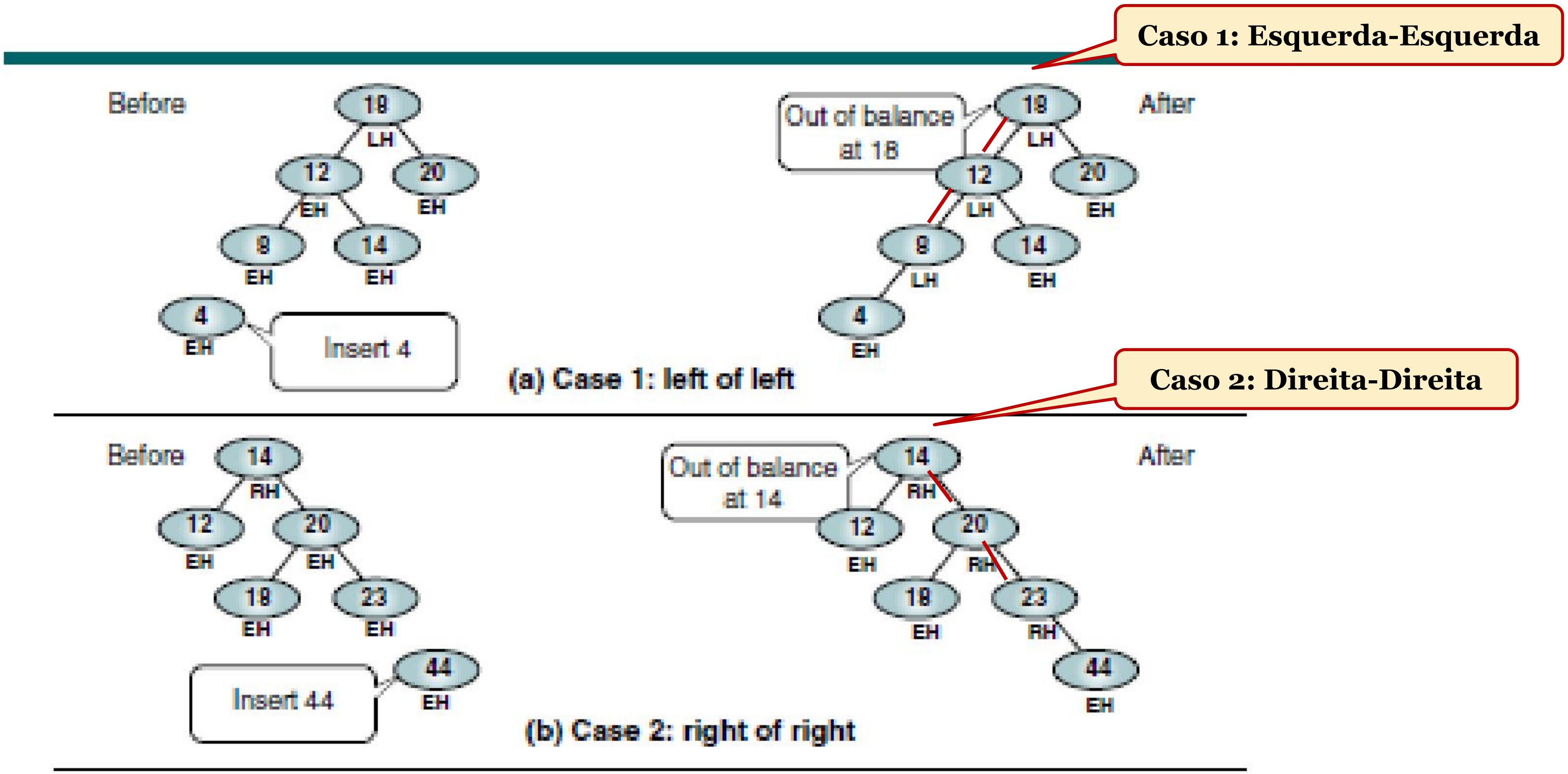
Arvores AVL (AVL Trees)

Casos de Desbalanceamento

- Como consequência da inserção, o nó z ficará desbalanceado.
- Identificam-se 4 casos possíveis **(De cima para baixo)**:
  - **Caso1.- Esquerda-Esquerda.-** O nó z tem altura esquerda e seu filho y de maior altura, tem altura esquerda (Exige rotação simples à direita).
  - **Caso2.- Direita-Direita.-** O nó z tem altura direita e seu filho y de maior altura, tem altura direita (Exige rotação simples à esquerda).
  - **Caso3.- Esquerda-Direita.-** O nó z tem altura esquerda e seu filho y de maior altura, tem altura direita (Exige rotação dupla à direita).
  - **Caso4.- Direita-Esquerda.-** O nó z tem altura direita e seu filho y de maior altura, tem altura esquerda (Exige rotação dupla à esquerda).

Arvores AVL (AVL Trees)

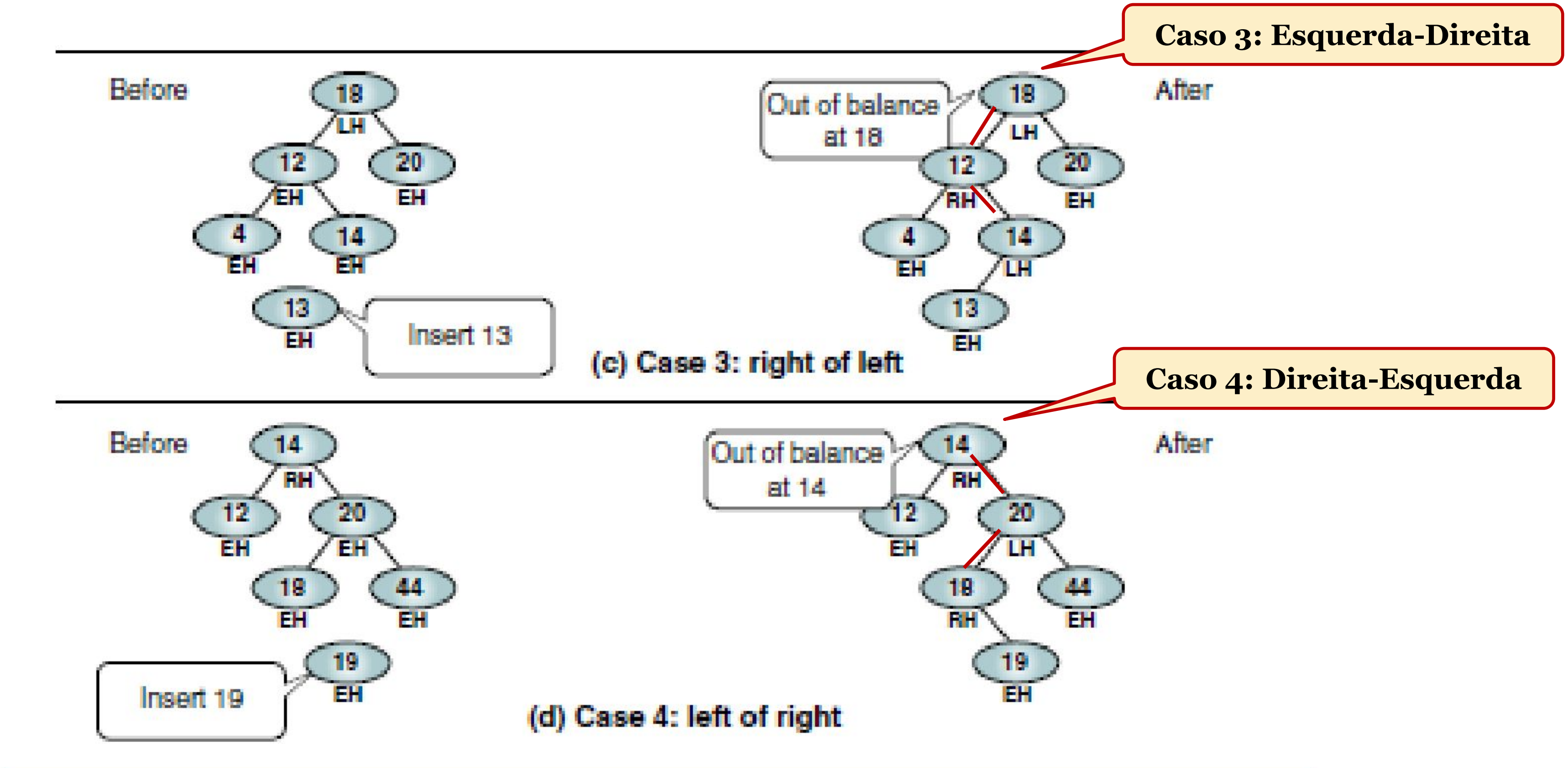
Desbalanceamento por Inserção: Casos 1 e 2



- Obs. Na figura, a notação dos casos está invertida (de baixo para cima).

Arvores AVL (AVL Trees)

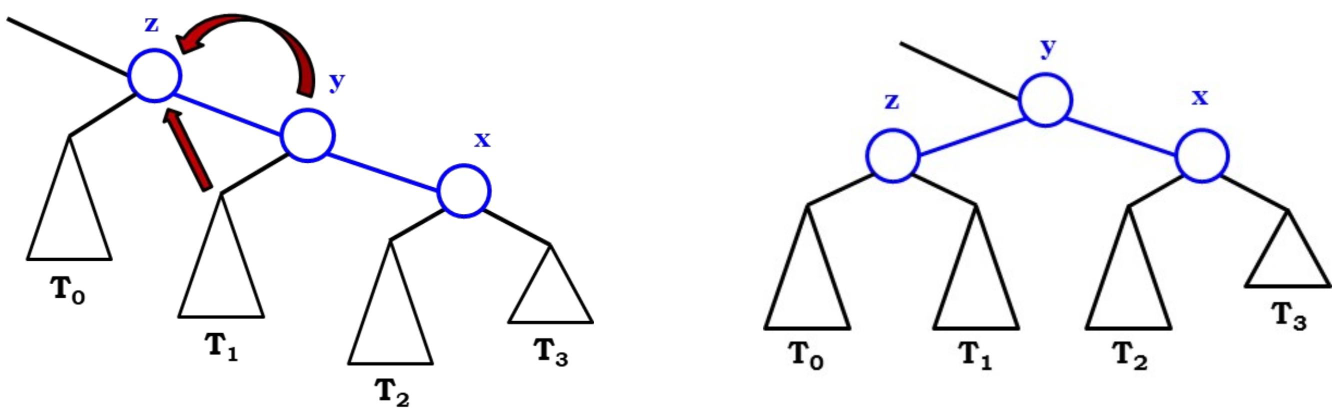
Desbalanceamento por Inserção: Casos 3 e 4



- Obs. Na figura, a notação dos casos está invertida (de baixo para cima).

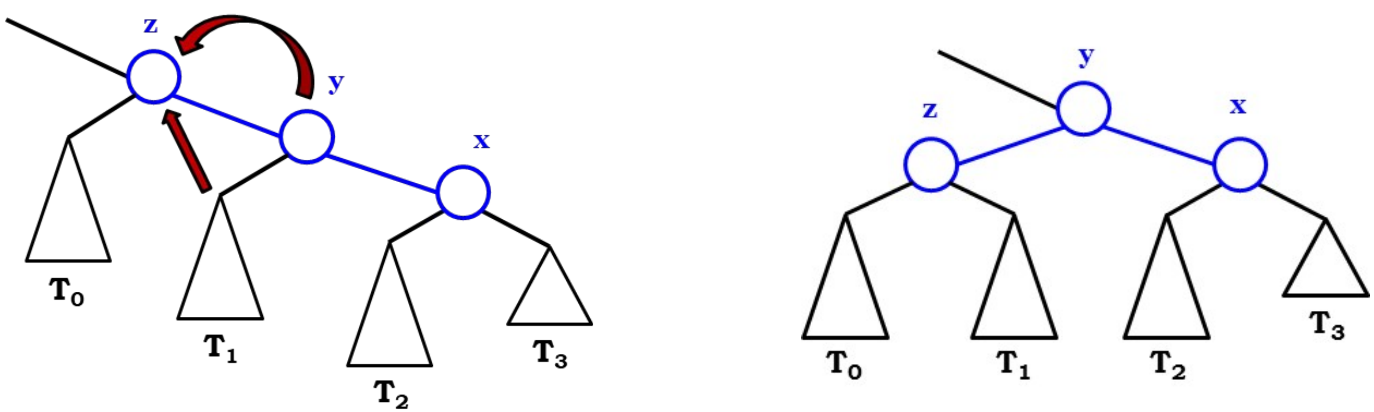
Árvores AVL
Balanceamento após a Inserção

- O processo requer identificar três nós x, y, z. A reestruturação da árvore é realizada com base nesses nós.
  - O nó z corresponde ao primeiro nó desbalanceado encontrado no caminho ascendente do w até a raiz da árvore T.
  - O nó y corresponde ao filho de z com maior altura (ancestral de w).
  - O nó x corresponde ao filho de y com maior altura (sem empate, ancestral de w).
- Além disso, identificam-se quatro árvores denotadas como T_0, T_1, T_2 e T_3, filhas dos nós x, y, z. Identificadas na ordem de percurso em-ordem.



Árvores AVL
Re-estruturação da àrvore: Rotações

- A inserção de um elemento novo na arvore AVL produz o desbalanceamento da mesma. Como consequência disso, certos nós ficarão desbalanceados apresentando assim uma diferença maior do que um entre as alturas de seus filhos (esquerdo e direto).
- Para realizar o balanceamento da árvore é preciso identificar o nó desbalanceado de maior nível. O balanceamento consiste em aumentar o nível desse nó e seus descendentes (descer o nó) e ao mesmo tempo diminuir o nível de outros e seus descendentes (subir o nó).
- Este processo é chamado de rotação.



Árvores AVL
Casos possíveis de balanceamento

- Existem 4 casos possíveis de balanceamento em um árvore de busca binária.
- Esses casos serão denotados como caso1, caso 2, caso 3 e caso 4.
- A identificação de cada caso é realizada com base no relacionamento entre os nós z, y e x. A seguir identifica-se cada caso:

Caso 1:
- o nó y é filho esquerdo do nó z;
- o nó x é filho esquerdo do nó y;

Caso 2:
- o nó y é filho direito do nó z;
- o nó x é filho direito do nó y;

Caso 3:
- o nó y é filho esquerdo do nó z;
- o nó x é filho direito do nó y;

Caso 4:
- o nó y é filho direito do nó z;
- o nó x é filho esquerdo do nó y;

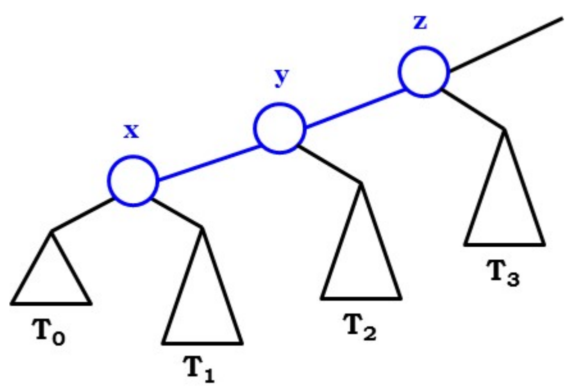
- Observa-se as seguintes simetrias:
  - i) Os casos 1 e 2 são simétricos;
  - ii) Os caso 3 e 4 são simétricos.

Árvores AVL

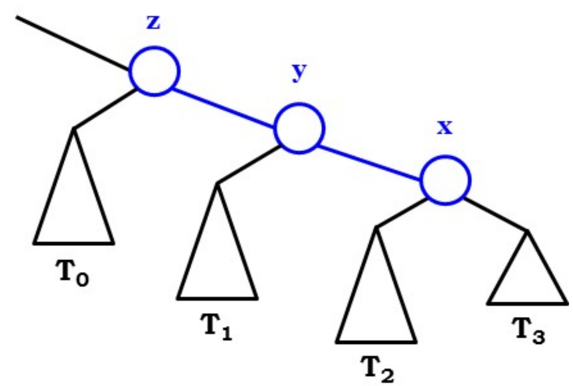
Casos possíveis de balanceamento

- Ilustra-se de maneira esquemática os 4 casos possíveis de balanceamento em um árvore de busca binária. Destaca-se os nós z, y e x, que serão utilizados na reestruturação da árvore. Os triângulos representam sub-árvores de alturas diversas.

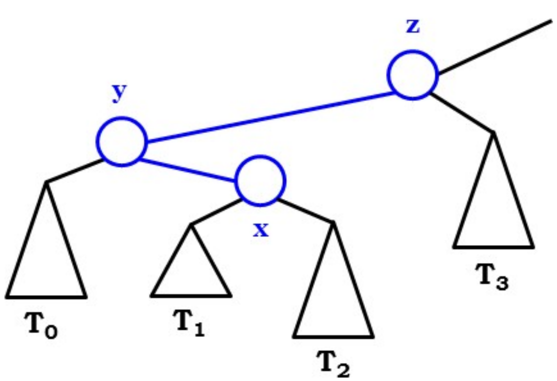
Caso 1



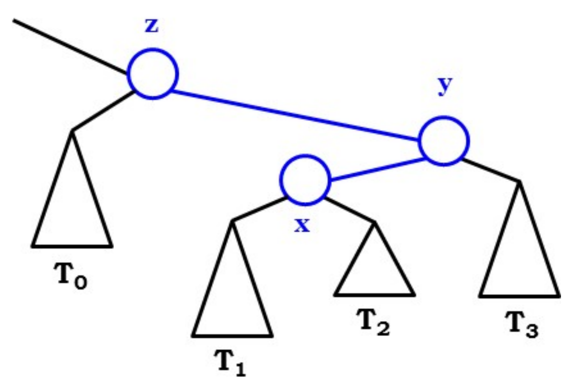
Caso 2



Caso 3



Caso 4



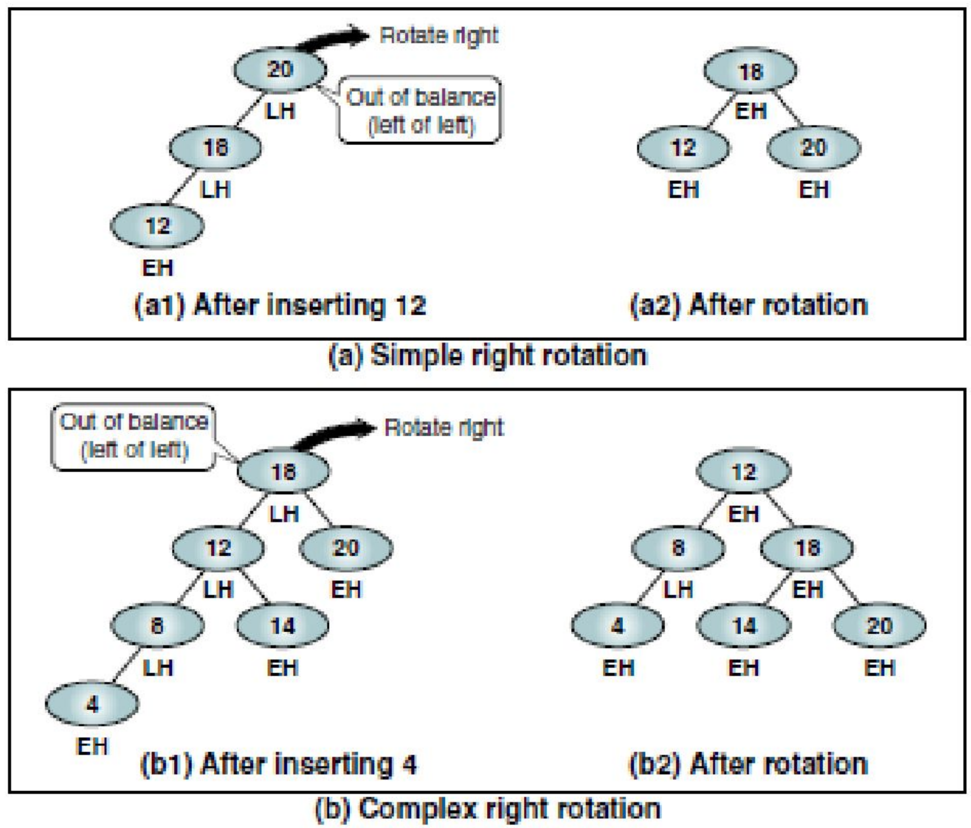
Árvores AVL
Balanceamento da Árvore: Rotações

- O processo de balanceamento de uma árvore pode ser melhor compreendido através da operação de rotação.
- A rotação é uma operação que permite redistribuir os nós da árvore preservando a sua condição de árvore de busca binária.
- A rotação consiste em aumentar o nível (descer o nó na árvore) do nó desbalanceado de maior nível ao mesmo tempo que diminui-se o nível do filho de maior altura.
- Informalmente podemos dizer que o nó z desce na árvore na árvore enquanto o nó y sobe.
- Dependendo do caso de balanceamento, a árvore pode demandar uma ou até duas rotações:
  - Os casos 1 e 2 demanda apenas uma rotação, assim o processo de balanceamento é chamado de **rotação simples**.
  - Os casos 3 e 4 demandam duas rotações, sendo que o processo de balanceamento é chamado de **rotação dupla.**

Arvores AVL (AVL Trees)

Caso 1: Esquerda-Esquerda – Rotação à Direita

- Mostra-se dois exemplos de rotação à direita.

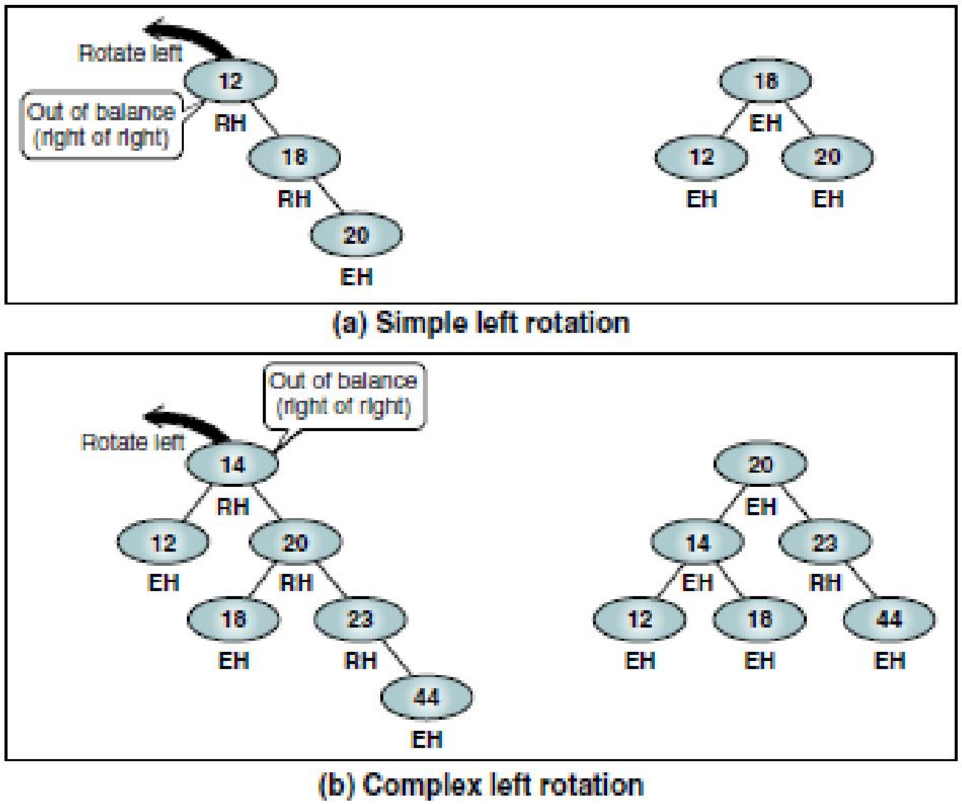


Left of Left—Single Rotation Right

Arvores AVL (AVL Trees)

Caso 2: Direita-Direita – Rotação à Esquerda

- Mostra-se dois exemplos de rotação à esquerda.

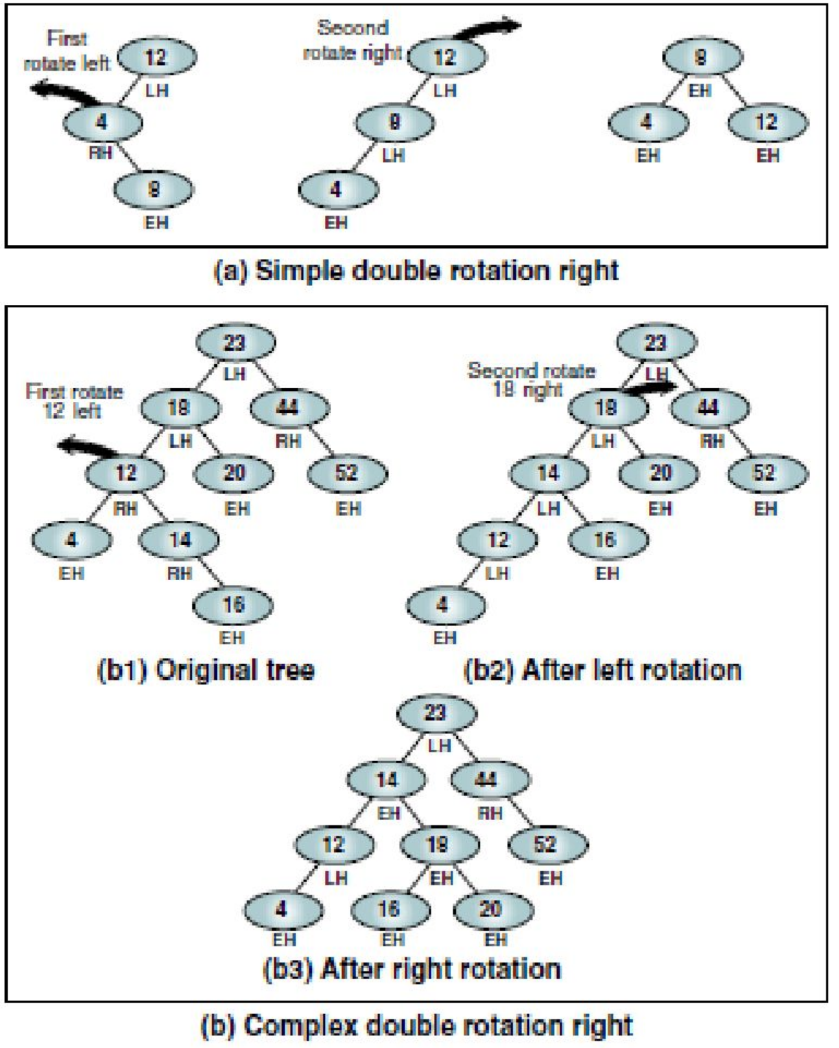


Right of Right—Single Rotation Left

Arvores AVL (AVL Trees)

Caso 3: Esquerda-Direita – Rotação Dupla à Direita

- Mostra-se dois exemplos de rotação dupla à direita.
- Toda rotação dupla realiza duas rotações. Neste caso, a primeira é a esquerda e a segunda é a direita.

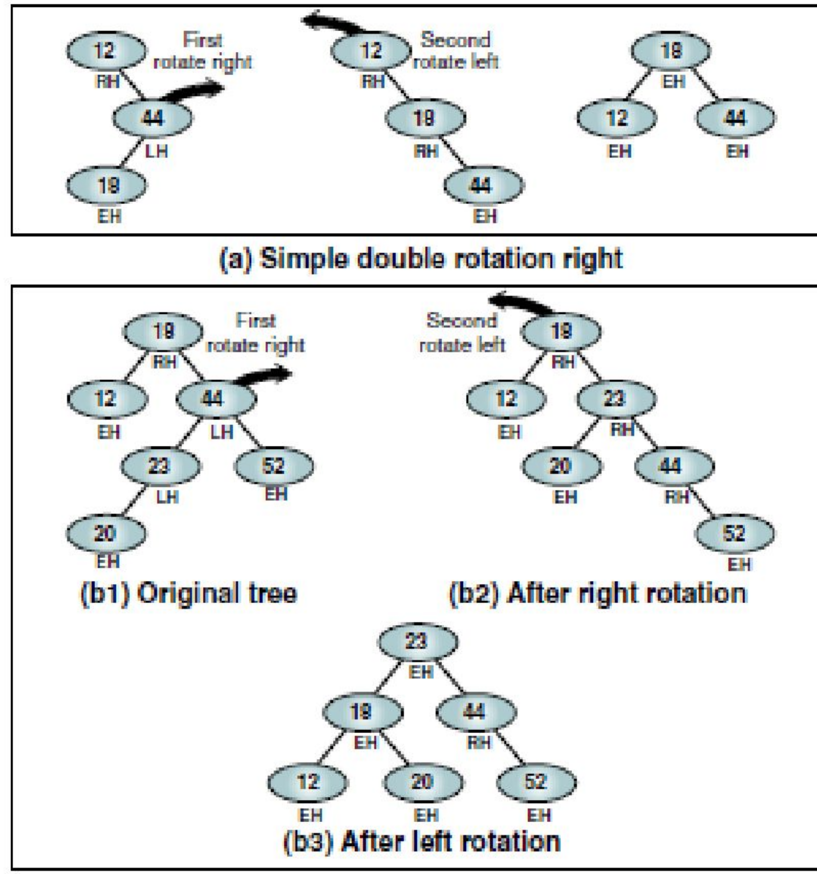


Right of Left—Double Rotation Right

Arvores AVL (AVL Trees)

Caso 4:Direita-Esquerda – Rotação Dupla à Esquerda

- Mostra-se dois exemplos de rotação dupla à esquerda.
- Toda rotação dupla realiza duas rotações. Neste caso, a primeira é a direita e a segunda é a esquerda.



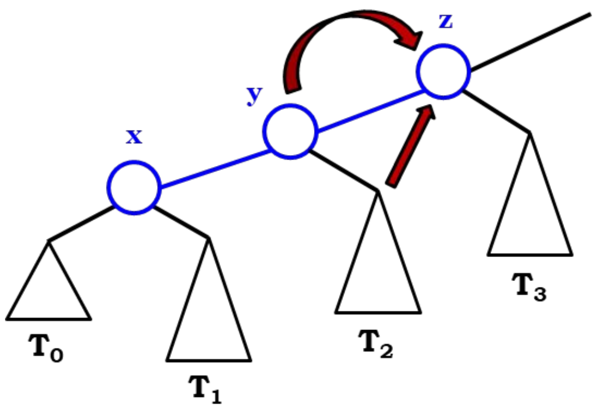
(a) Simple double rotation right

(b) Complex double rotation right

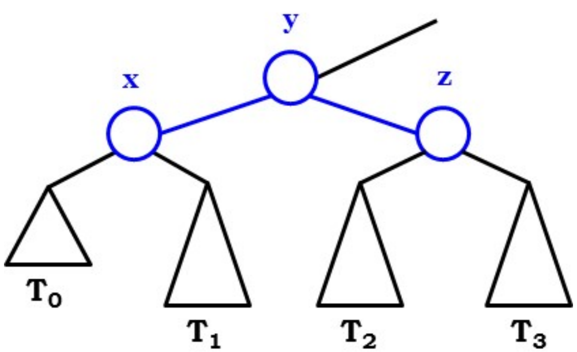
Left of Right—Double Rotation Right

Árvores AVL
Caso 1: Rotação Direita

- A rotação direita consiste em:
  - substituir a sub-árvore com raiz em z pela sub-árvore com raiz em y;
  - fazer T_2 filho esquerdo de z;
  - fazer z filho direito de y.



Rotação Direita

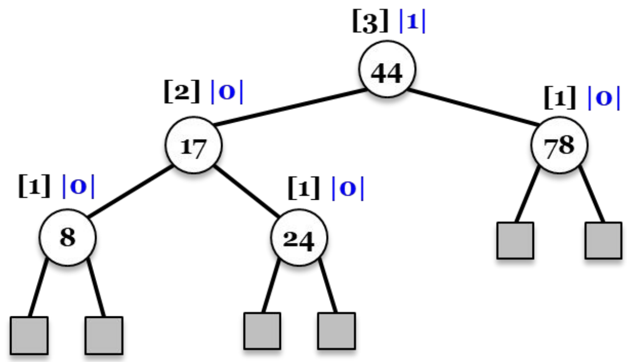


Arvore Balanceada

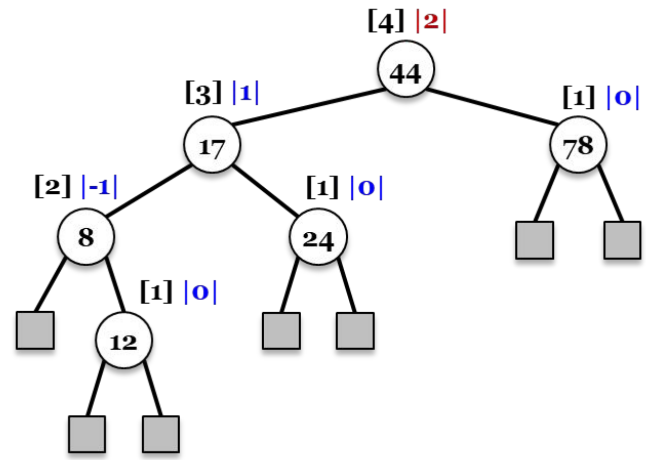
Árvores AVL

Caso 1: Rotação Direita – Exemplo1

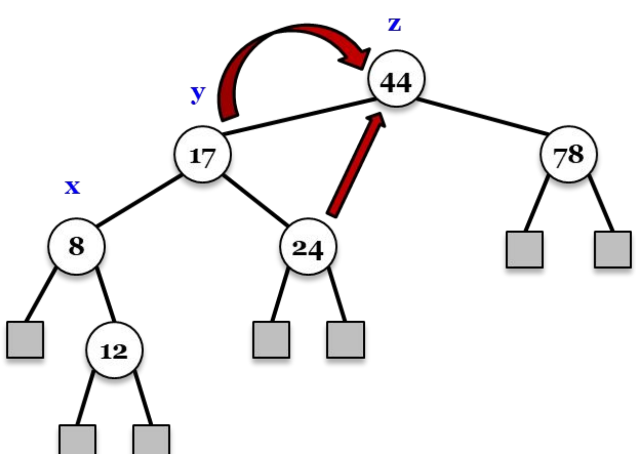
- Ilustra-se o caso da inserção do nó 12.



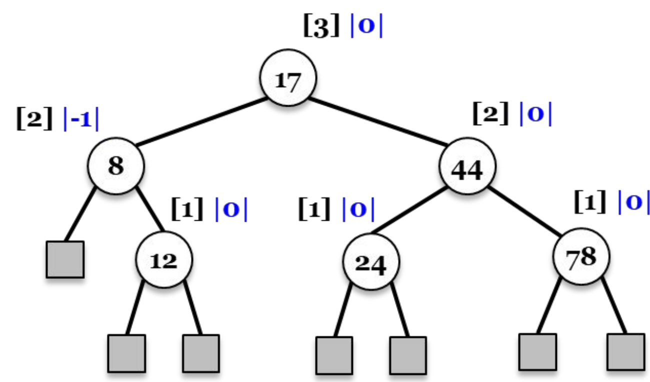
Árvore Inicial



Árvore Após Inserção



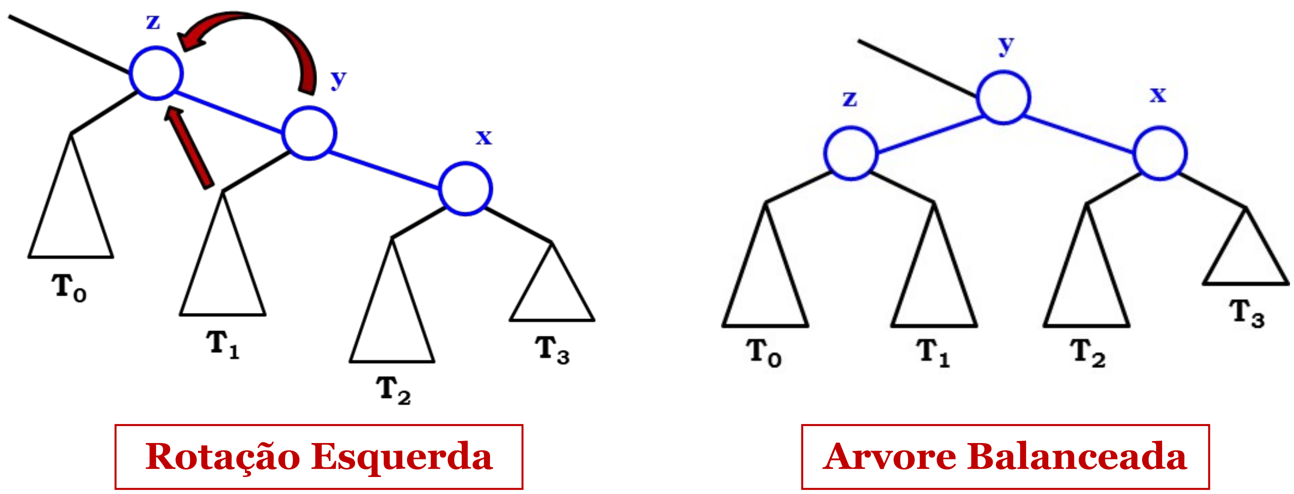
Rotação Direita



Arvore Balanceada

Árvores AVL
Caso 2: Rotação Esquerda

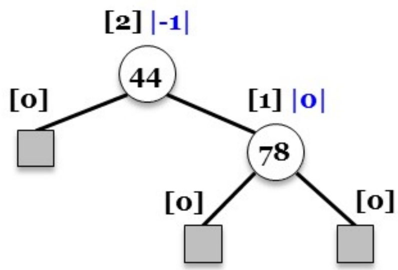
- A rotação esquerda consiste em:
  - substituir a sub-árvore com raiz em z pela sub-árvore com raiz em y;
  - fazer T_1 filho direito de z;
  - fazer z filho esquerdo de y.



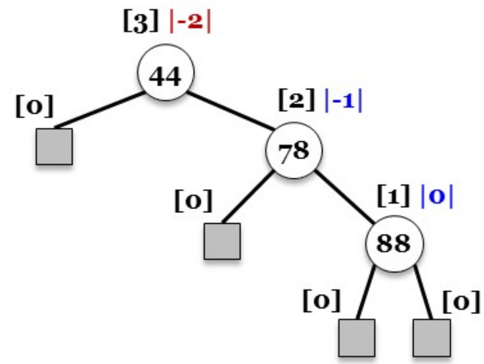
- Vale observar que z, y e x e T_0, T_1 ,T_2 e T_3 são utilizados são parâmetros do método.

Árvores AVL
Caso 2: Rotação Esquerda – Exemplo2-A

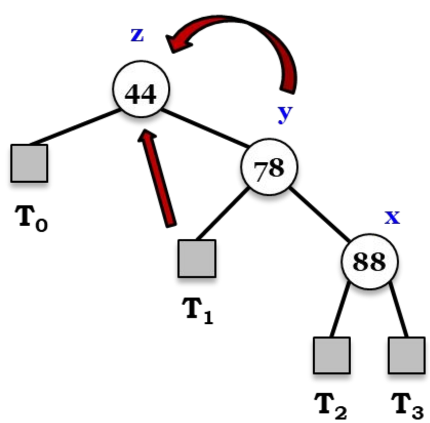
- Ilustra-se o caso da inserção do nó 88.



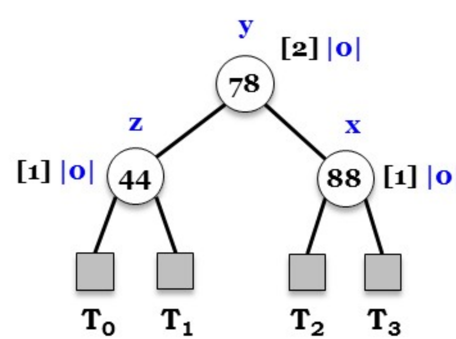
Árvore Inicial



Árvore Após Inserção



Rotação Esquerda

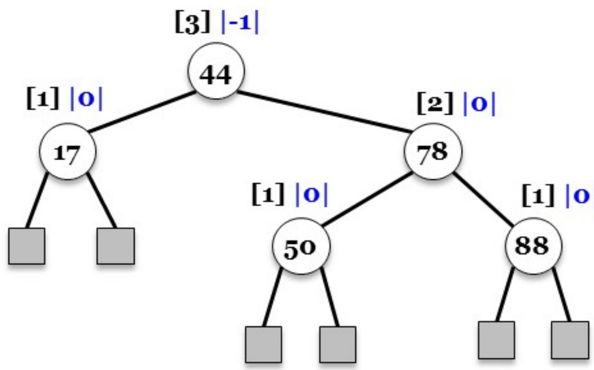


Arvore Balanceada

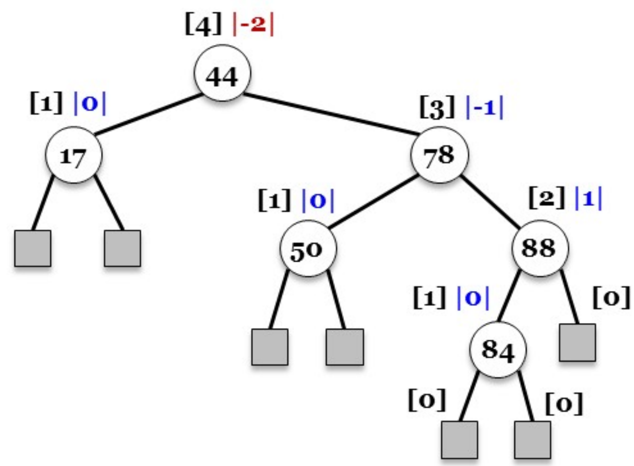
Árvores AVL

Caso 2: Rotação Esquerda – Exemplo2-B

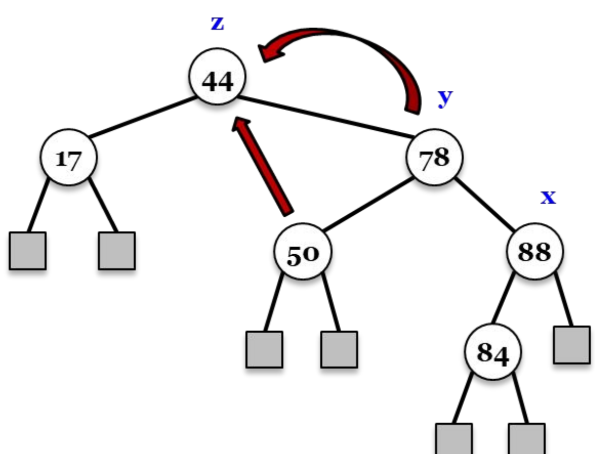
- Ilustra-se o caso da inserção do nó 84.



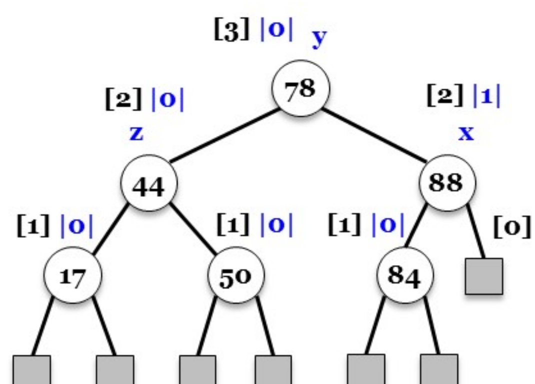
Árvore Inicial



Árvore Após Inserção



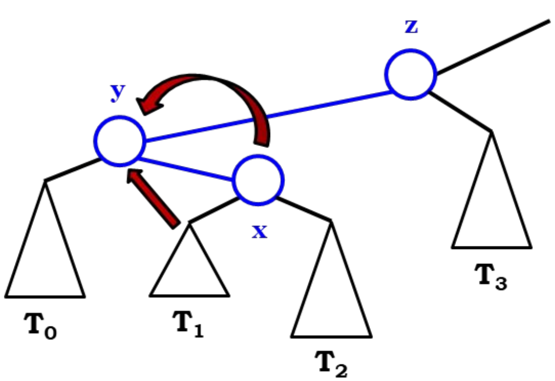
Rotação Esquerda



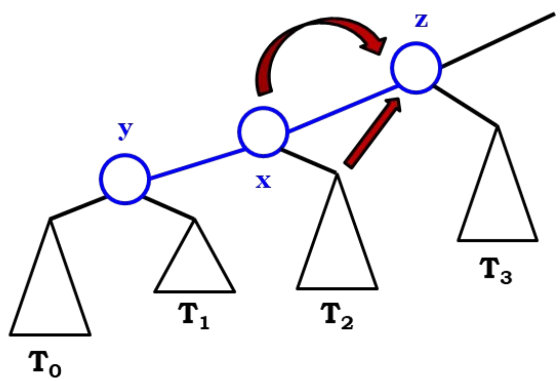
Arvore Balanceada

Árvores AVL
Caso 3: Rotação Dupla: Esquerda-Direita

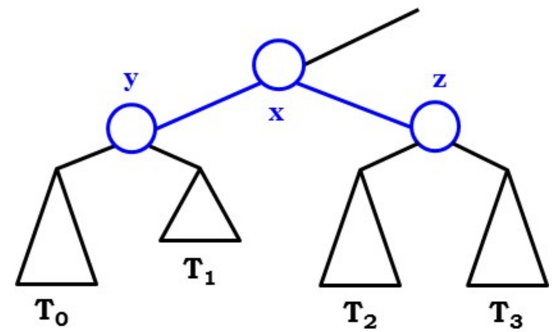
- A rotação esquerda-direita é uma rotação dupla que compreende uma rotação à esquerda seguida de uma rotação à direita.



Rotação Esquerda



Rotação Direita

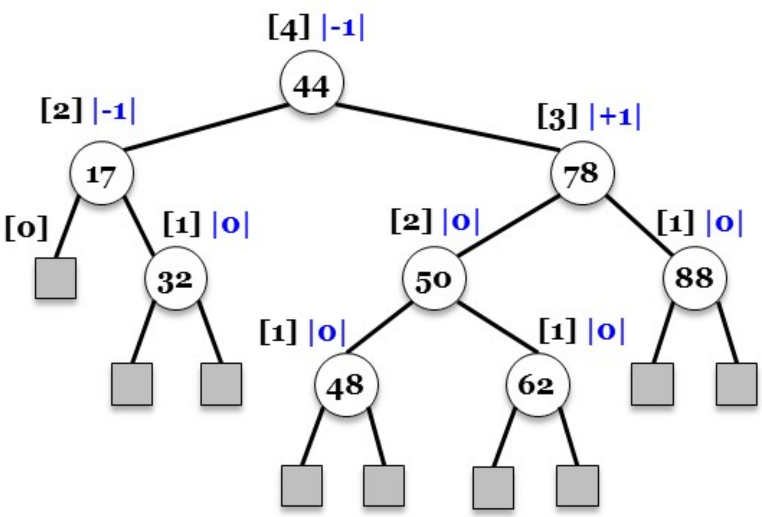


Arvore Balanceada

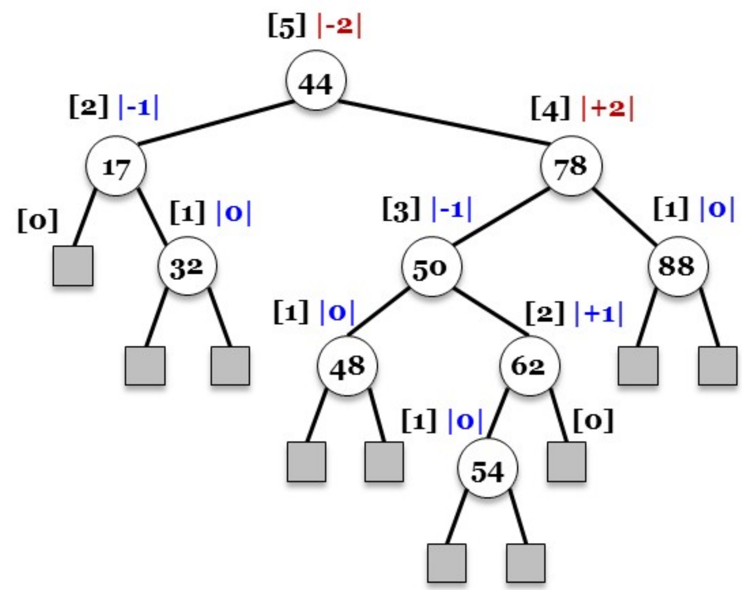
Árvores AVL

Caso 3: Rotação Dupla: Esquerda-Direita –Exemplo 3

- Ilustra-se o caso da inserção do nó 54.



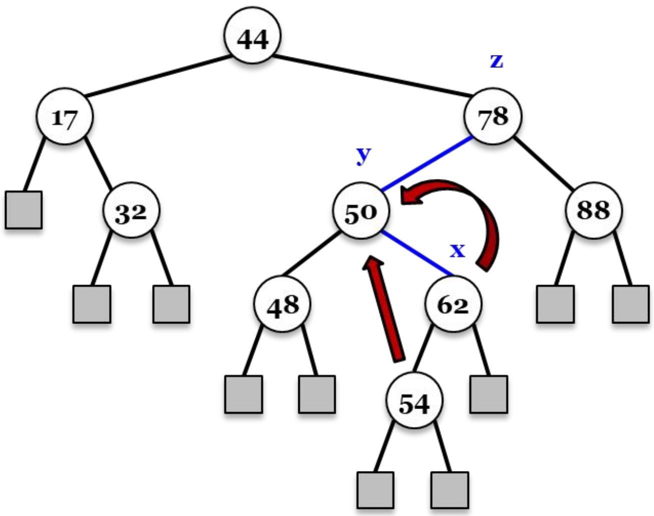
Árvore Inicial

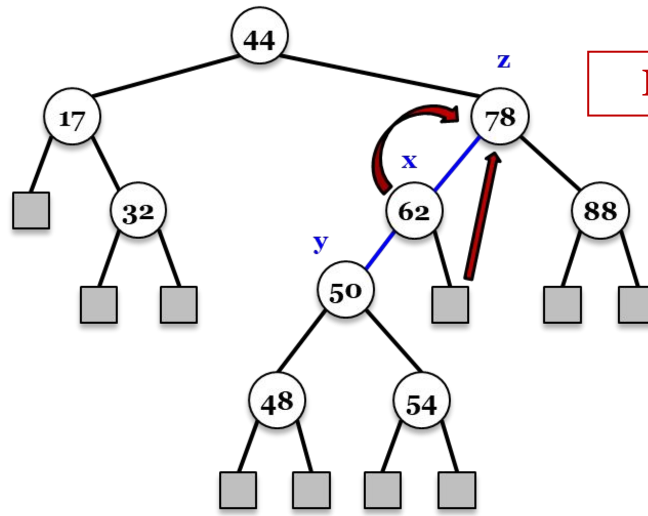


Árvore Após Inserção

Árvores AVL

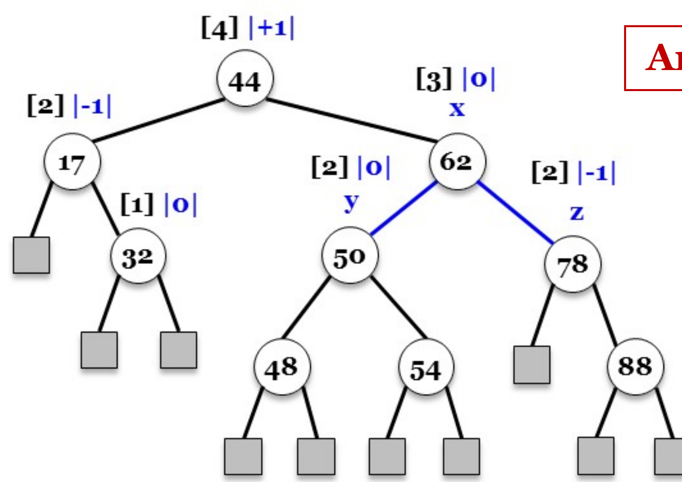
Caso 3: Rotação Dupla: Esquerda-Direita –Exemplo 3





Rotação Direita

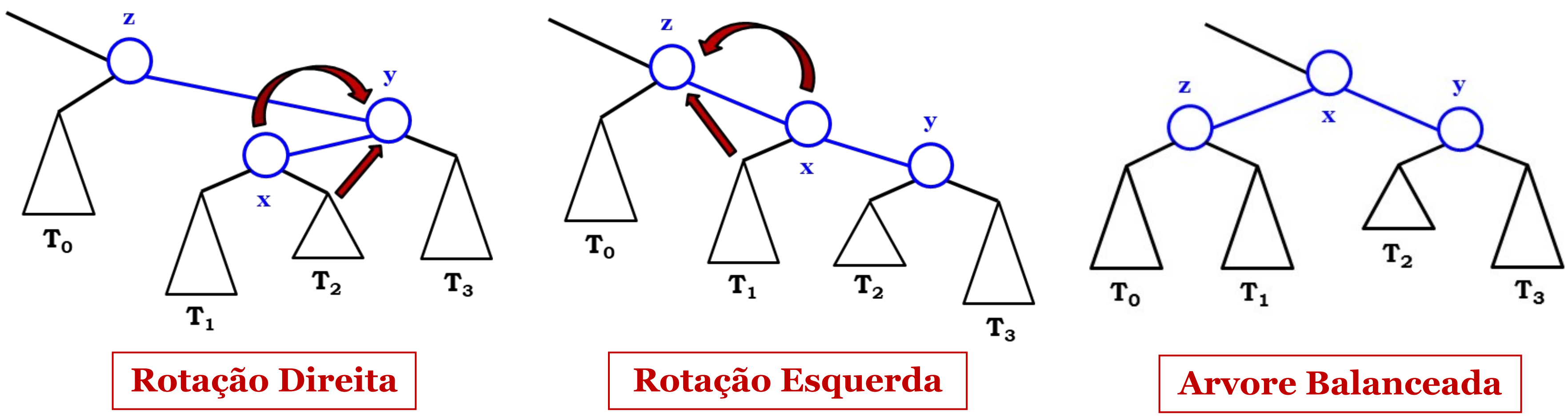
Rotação Esquerda



Arvore Balanceada

Árvores AVL
Caso 4: Rotação Dupla: Direita-Esquerda

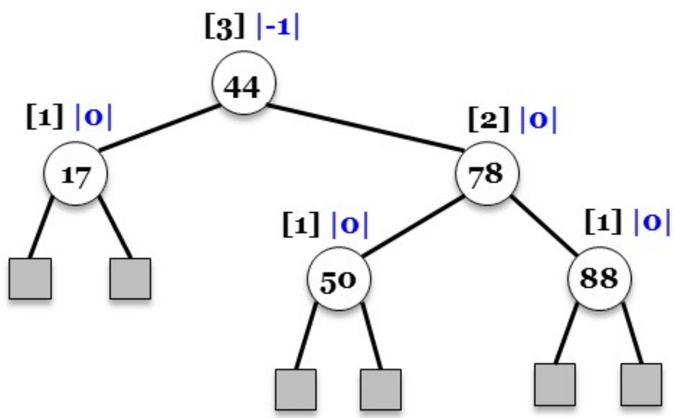
- A rotação direita-esquerda é uma rotação dupla que compreende uma rotação à direita seguida de uma rotação à esquerda.



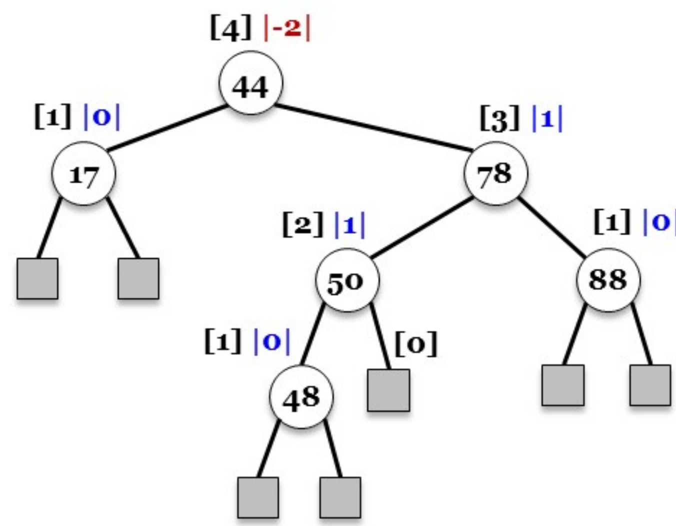
Árvores AVL

Caso 4: Rotação Dupla: Direita-Esquerda –Exemplo 4

- Ilustra-se o caso da inserção do nó 48.



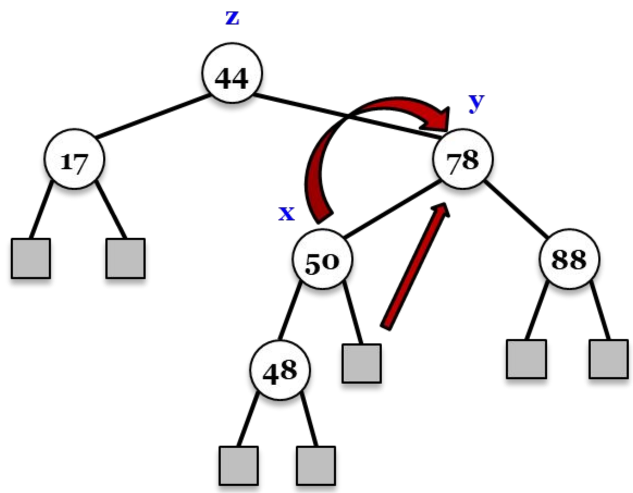
Árvore Inicial



Árvore Após Inserção

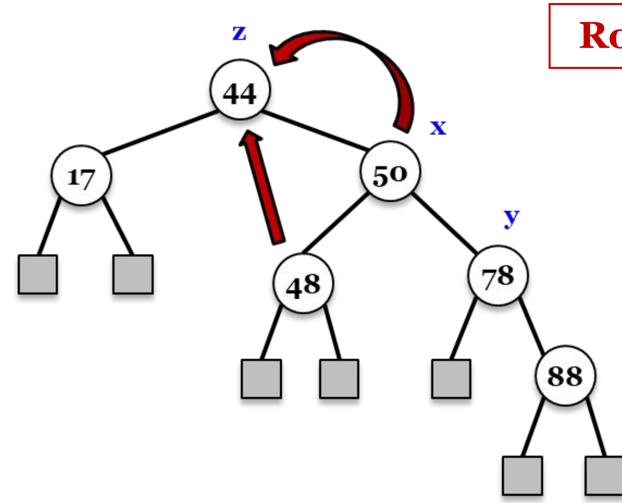
Árvores AVL

Caso 4: Rotação Dupla: Direita-Esquerda –Exemplo 4

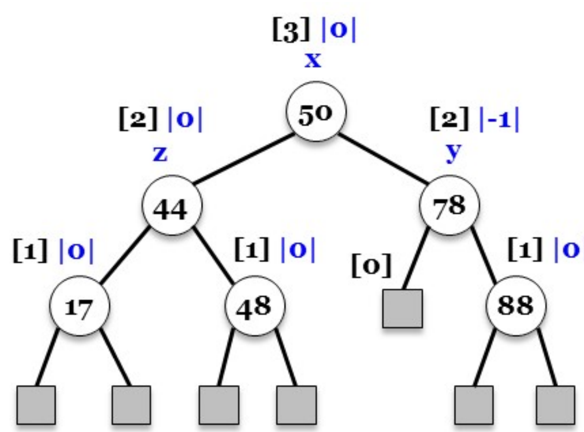


Rotação Direita

Rotação Esquerda



Arvore Balanceada



Referências

- Gilberg, R.F. e Forouzan, B. A. Data Structures_A Pseudocode Approach with C. Capítulo 8. AVL Search. Segunda Edição. Editora Cengage, Thomson Learning, 2005.
- Michael T. Goodrich, Roberto Tamassia, David Mount. Data Structures and Algorithms in C++. Cap10 Search Trees. 2ª Edição. 2011.