

Árvores Balanceadas

Eduardo Furlan Miranda

2024-04-02

Adaptado dos materiais dos Profs. A. Manacero Jr., R. Alves,
G. T. de Assis,
e R. Guerra

Árvores Balanceadas

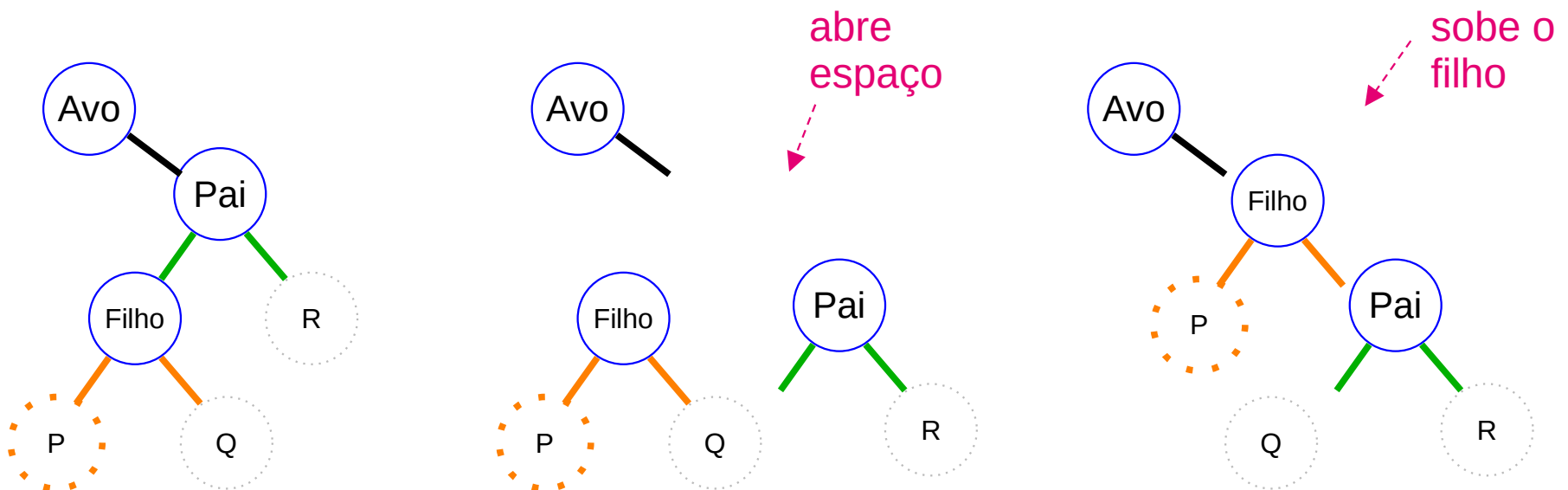
- Uma árvore deve ter seus componentes distribuídos de forma relativamente homogênea pela estrutura para que seja um mecanismo eficiente
- É necessário um **operador de balanceamento** capaz de manter os elementos distribuídos uniformemente entre as subárvores, pois as operações de inserção e remoção são aleatórias
- Existem duas maneiras de balancear uma árvore
 - **Global**, envolvendo toda a árvore
 - **Local**, envolvendo apenas uma parte

Árvores Balanceadas

- A árvore no balanceamento global pode ser construída a partir de uma estrutura externa
- Algoritmo simples, cria uma lista ordenada
 - Antes ou depois da existência da árvore
- Da lista constrói-se a árvore já balanceada
- O problema dessa estratégia é a necessidade de uma estrutura externa à árvore
- Solução → algoritmo DSW (C. Day, Q. Stout, e B. Warren)
- Transforma a árvore em uma espinha dorsal (formato lista)

Algoritmo DSW

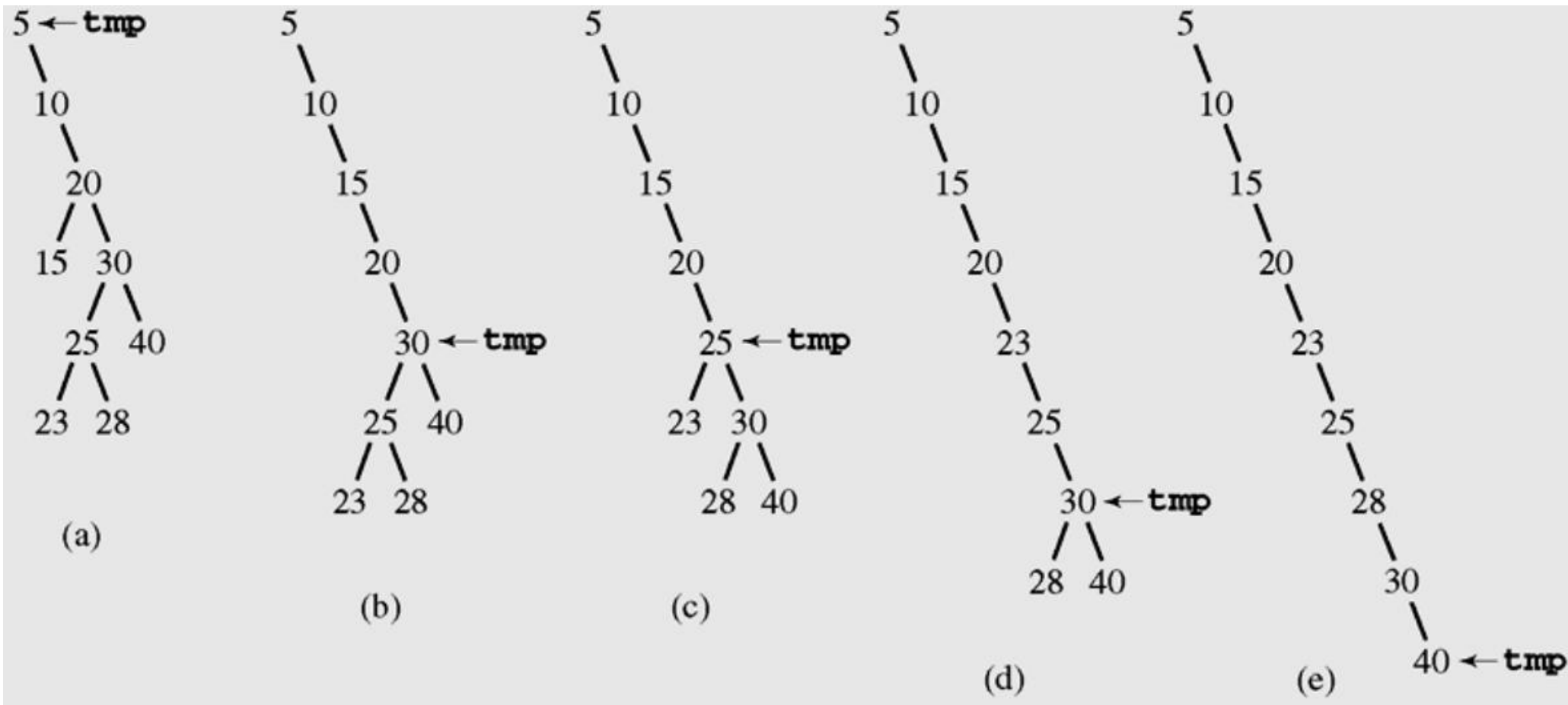
- Alicerce para transformações de árvore: é a rotação
- Existem dois tipos de rotação, esquerda e direita, que são simétricas entre si
 - **Rotação** direita do filho “Ch” em relação ao pai “Par”:



- Lembra movimentação em “pares”, par de vértices e par de arestas

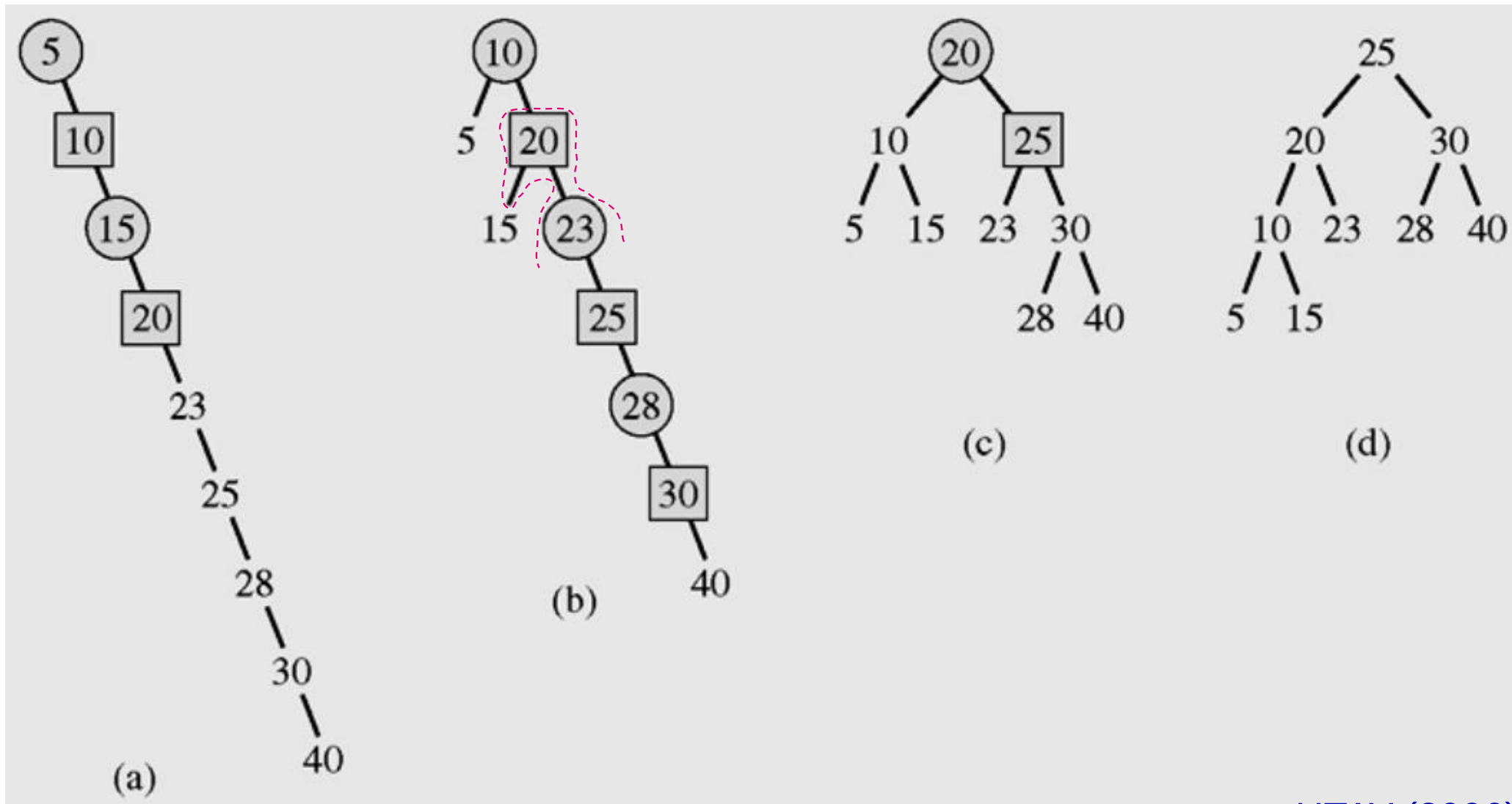
Algoritmo DSW

Transformando uma árvore de pesquisa binária em um backbone



Algoritmo DSW

Caminho inverso, transformando uma espinha dorsal em uma árvore perfeitamente equilibrada



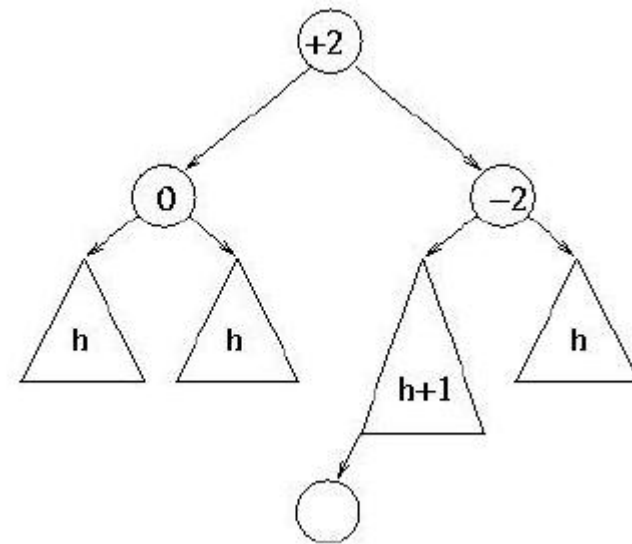
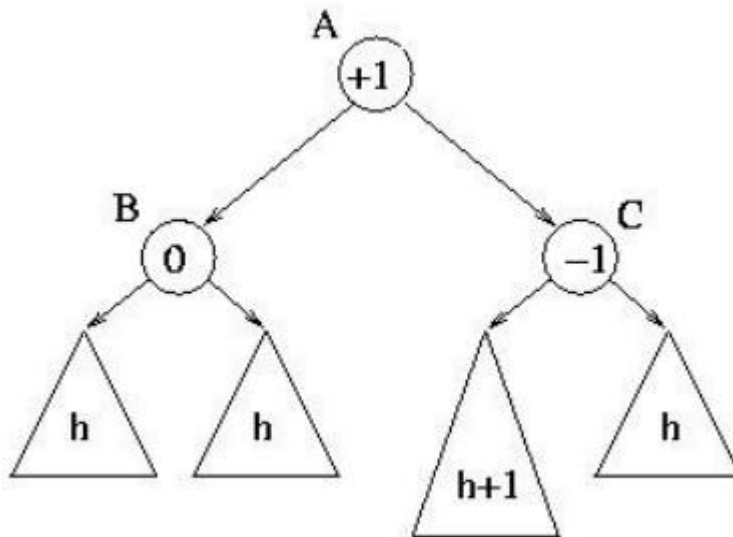
Algoritmo DSW

- O algoritmo DSW trabalha sobre a árvore toda
 - Se houver a necessidade de manter o balanceamento a cada inserção ou remoção, a eficiência fica comprometida
- Uma solução para esse problema é fazer uso de algoritmos que trabalhem apenas em parte da árvore, a cada inserção ou remoção
 - Árvores AVL
 - Árvores Rubro-negras

Árvores AVL (Algoritmo DSW)

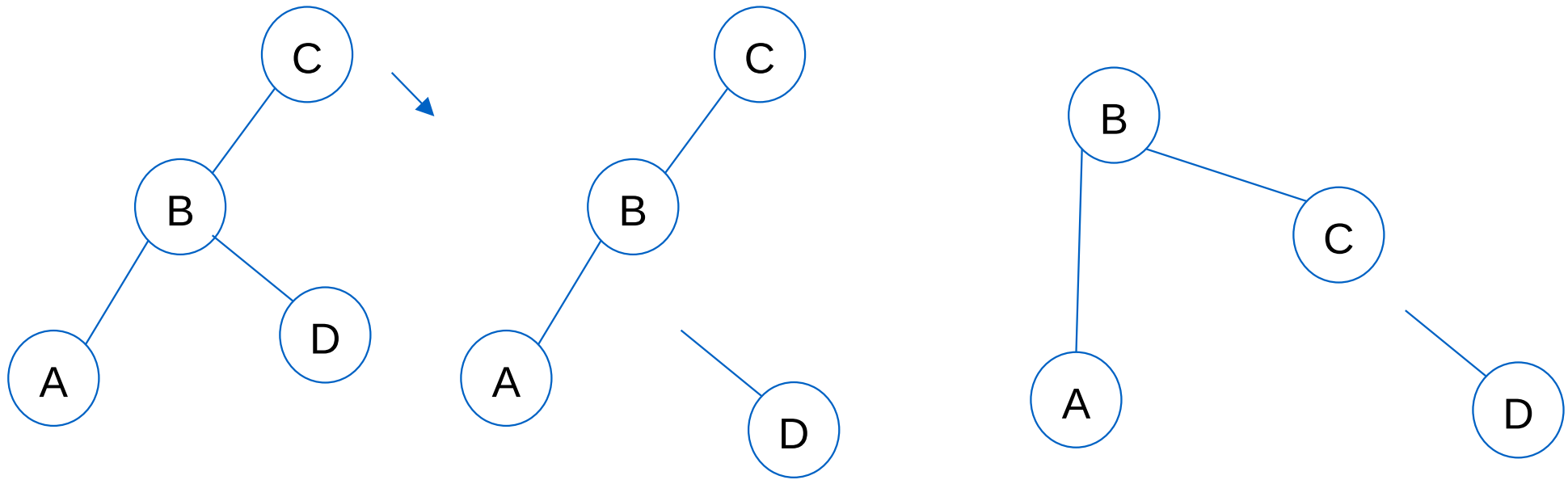
Árvores AVL

- Recebem esse nome a partir de seus criadores, os matemáticos G. M. Adelson Velskii, e E. M. Landis
- A base do algoritmo está na manutenção dos índices de balanceamento em torno de 1
- Quando o índice chega a 2 ou -2 , deve ser feito o balanceamento através de rotações



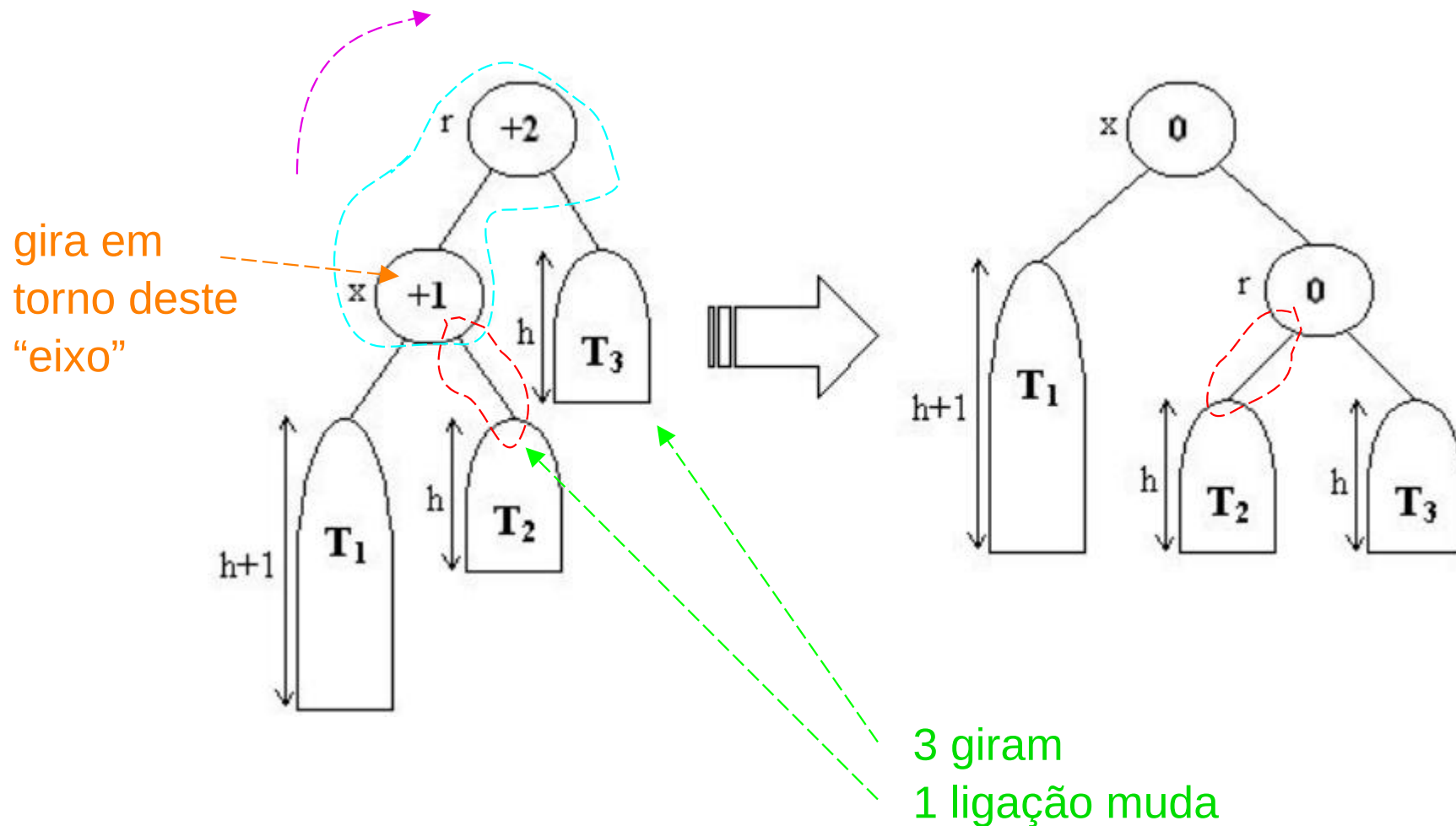
Árvores AVL

- Rotação à direita LL (sentido horário)



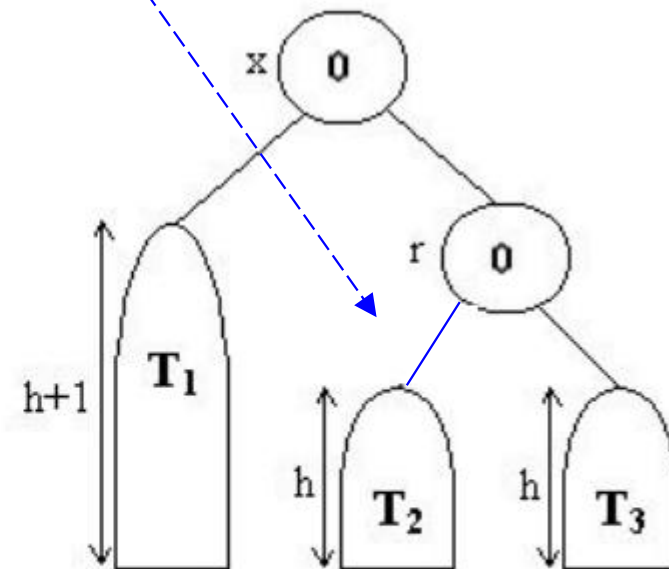
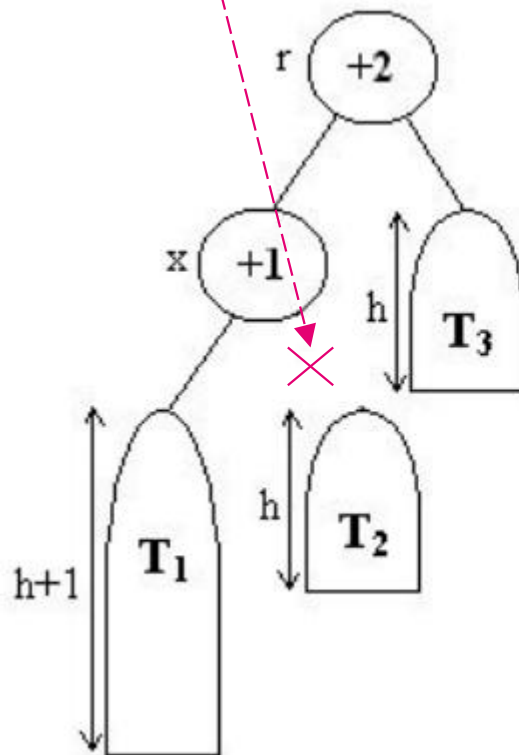
Árvores AVL

- Rotação à direita LL (sentido horário) - outro exemplo



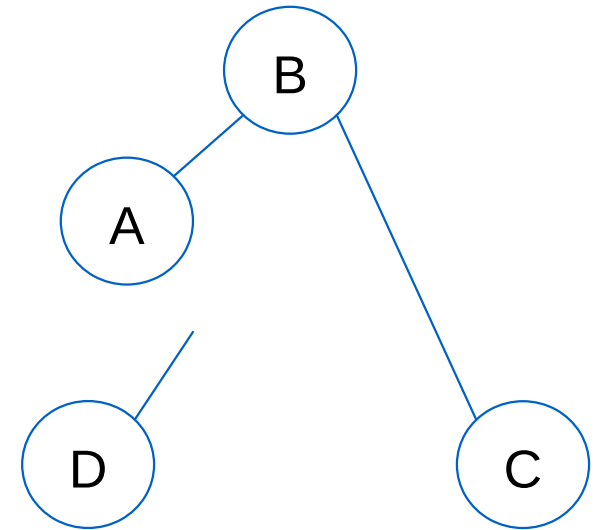
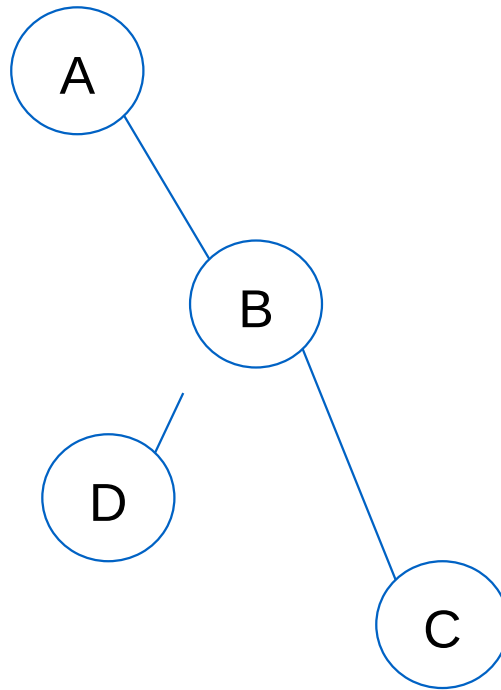
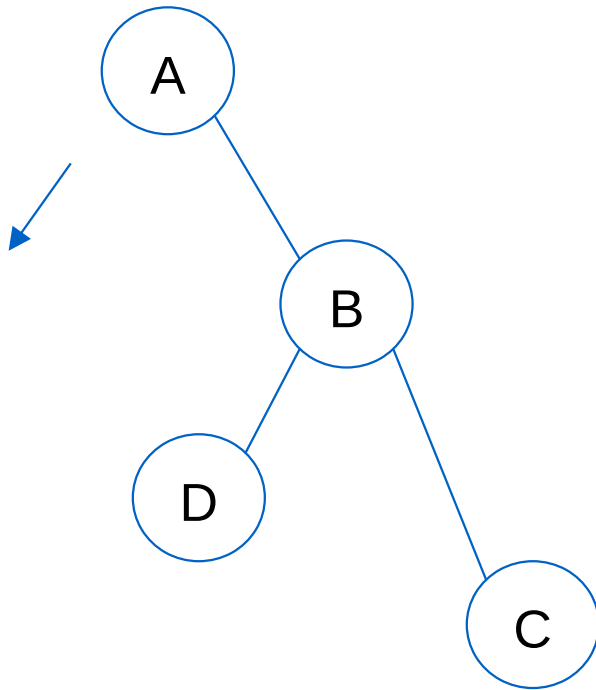
Árvores AVL - outra forma de pensar

- Outra forma de pensar, é que se vai girar para a **direita**
 - Primeiro a ligação da **direita** é temporariamente desfeita
 - Depois a ligação é refeita no nó **pai** que trocou de lugar
 - T2 fica aproximadamente na mesma posição



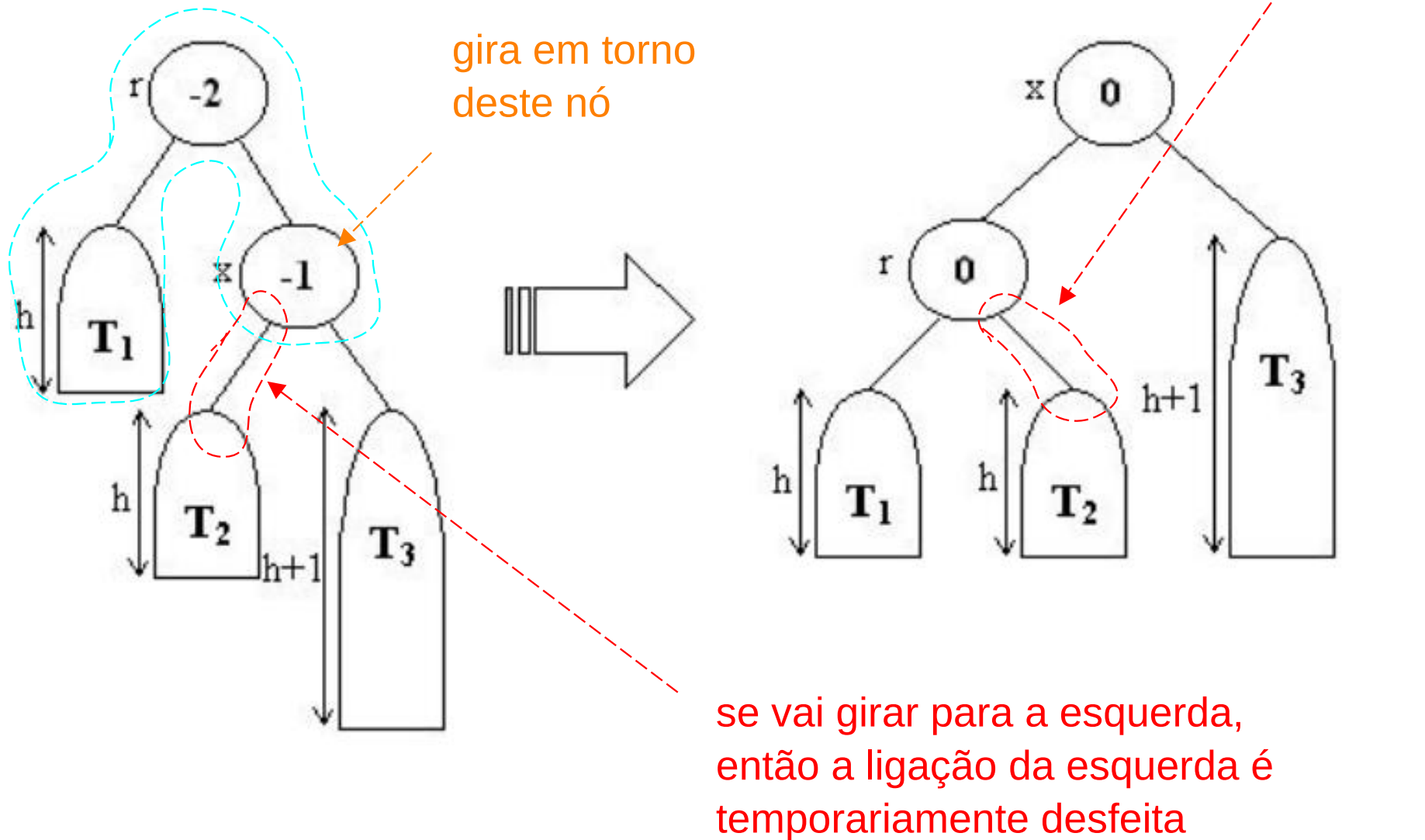
Árvores AVL

- Rotação à esquerda RR (sentido anti-horário)



Árvores AVL

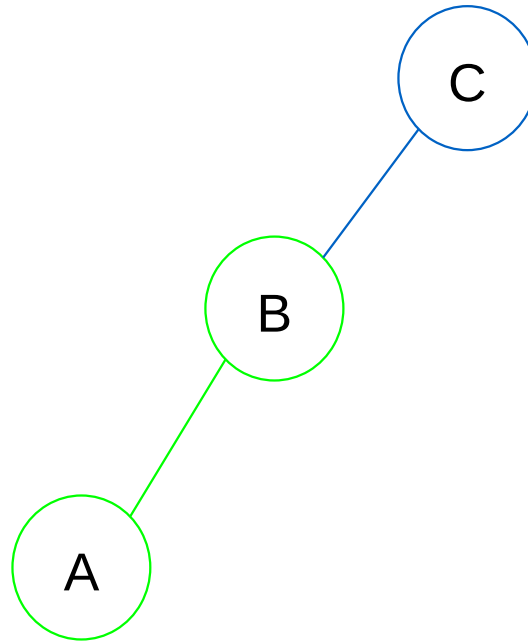
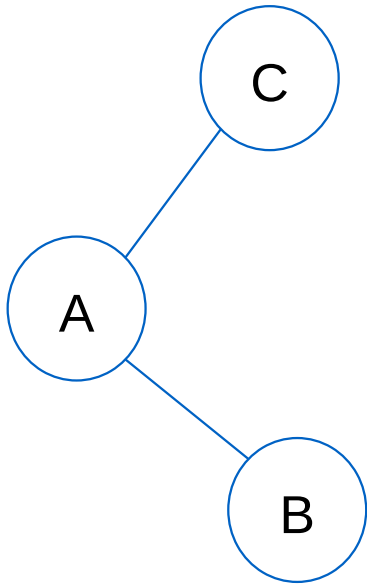
- Rotação RR (no sentido anti-horário)



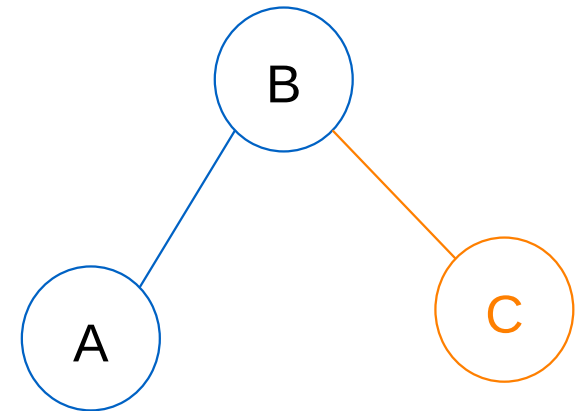
Árvores AVL

15

- Rotação dupla à direita LR (rotação e troca, horário)



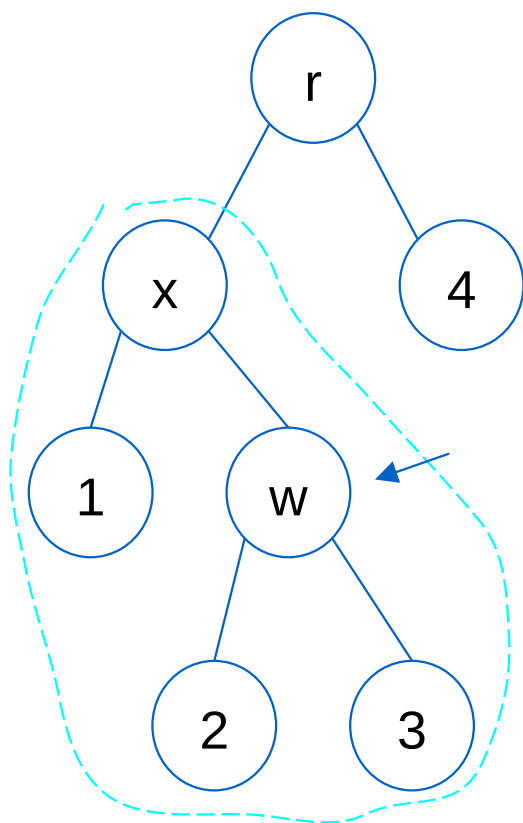
Rotação esquerda na
subárvore



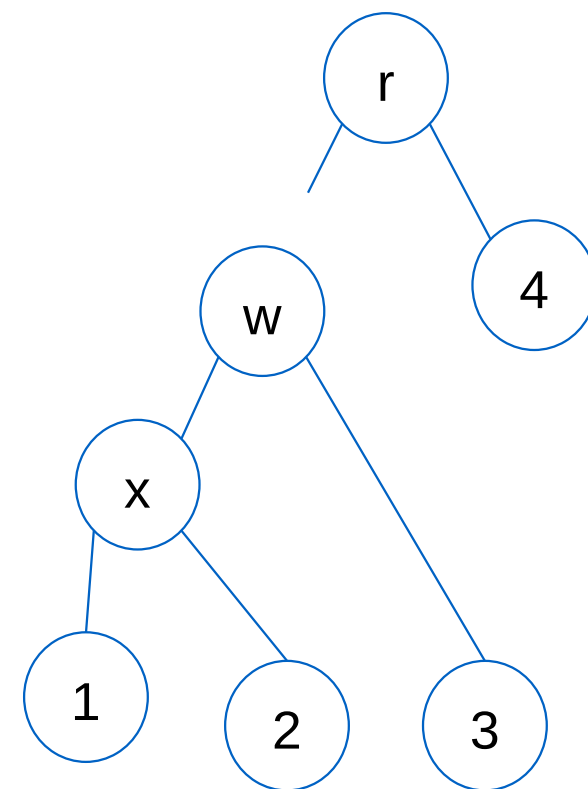
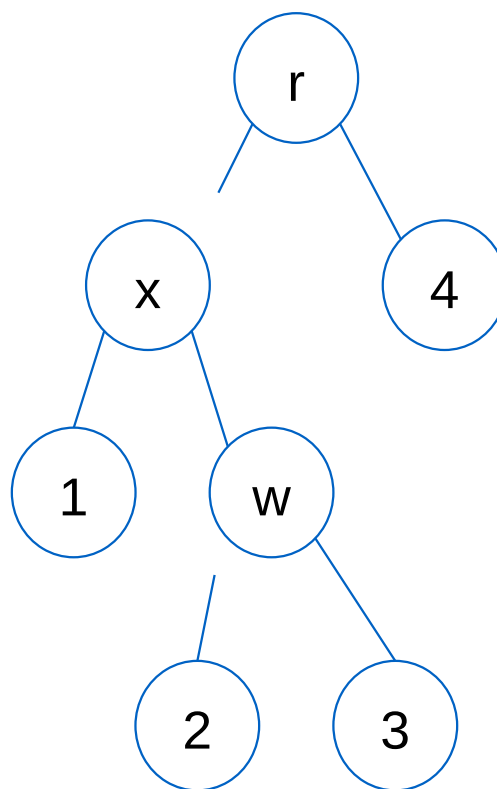
Rotação direita na árvore
original

(continua)

- Rotação dupla à direita LR (rotação e troca, horário)

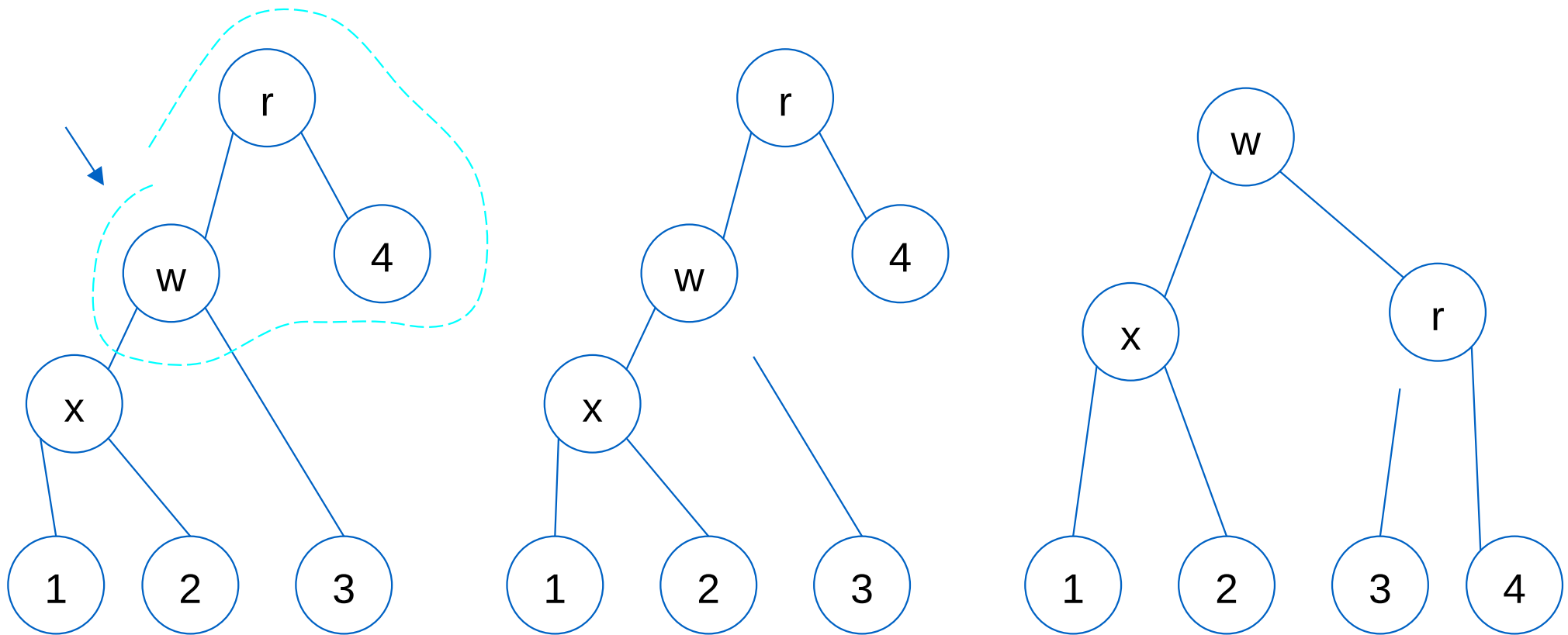


Rotação esquerda na subárvore



(continua)

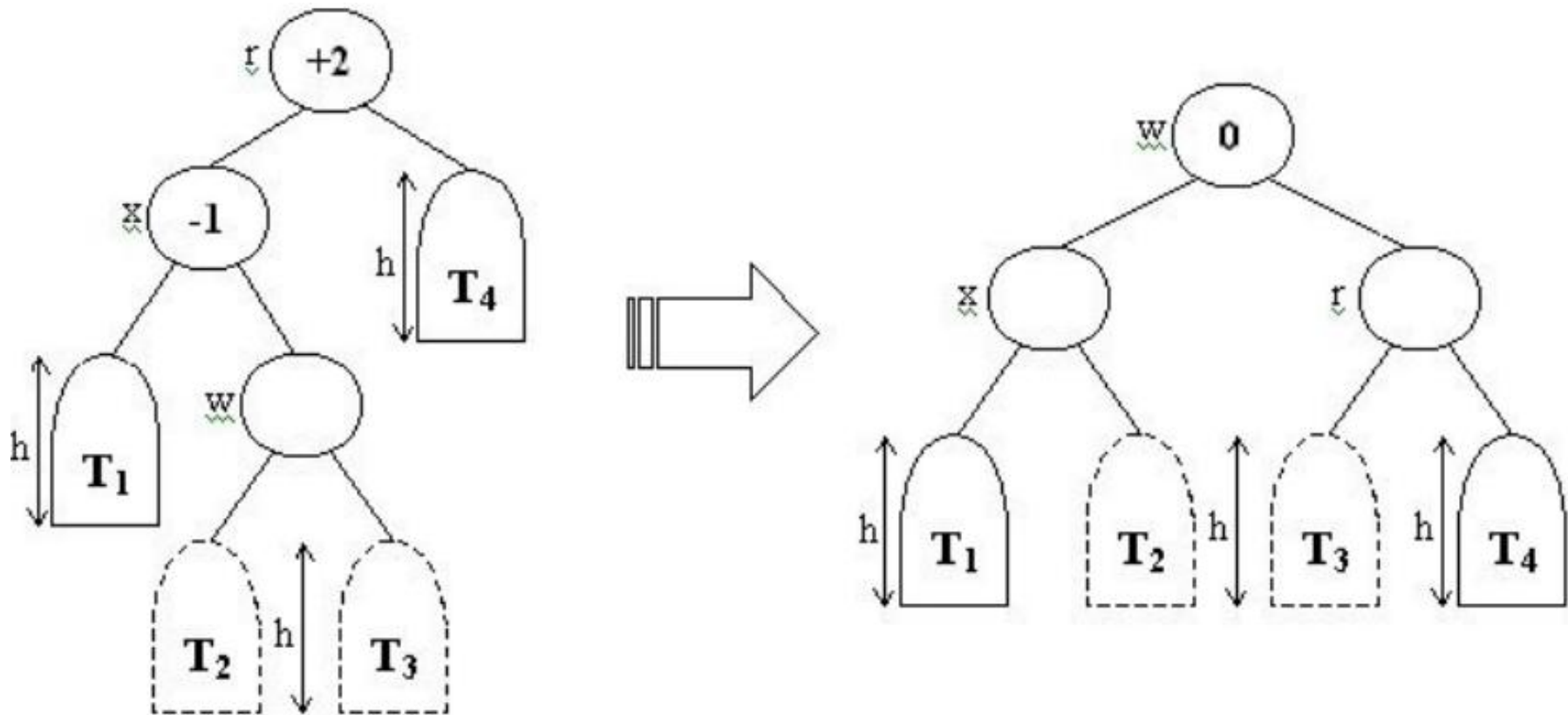
- Rotação dupla à direita LR (rotação e troca, horário)



Rotação direita na árvore original

(continua)

- Rotação dupla à direita LR (rotação e troca, horário)

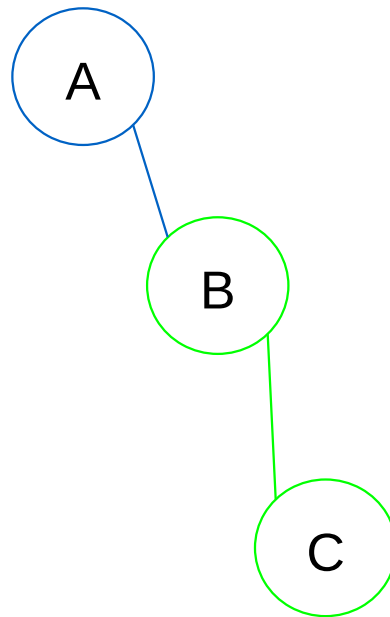
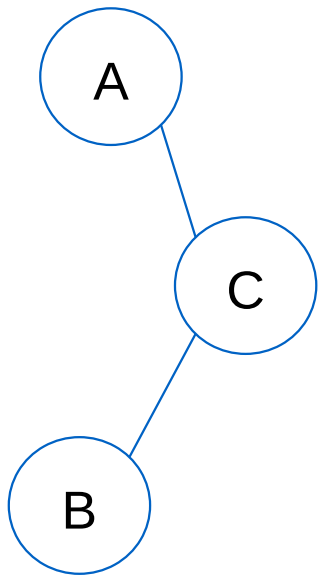


Outra forma de representar

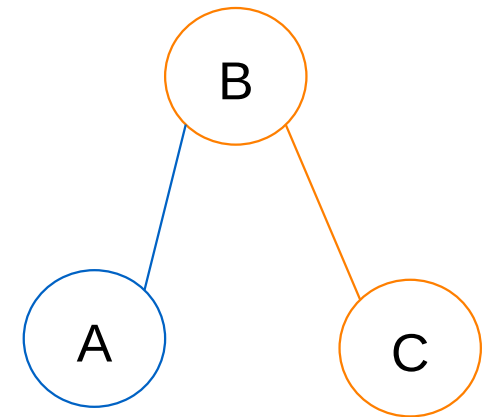
Árvores AVL

19

- Rotação dupla à esquerda RL



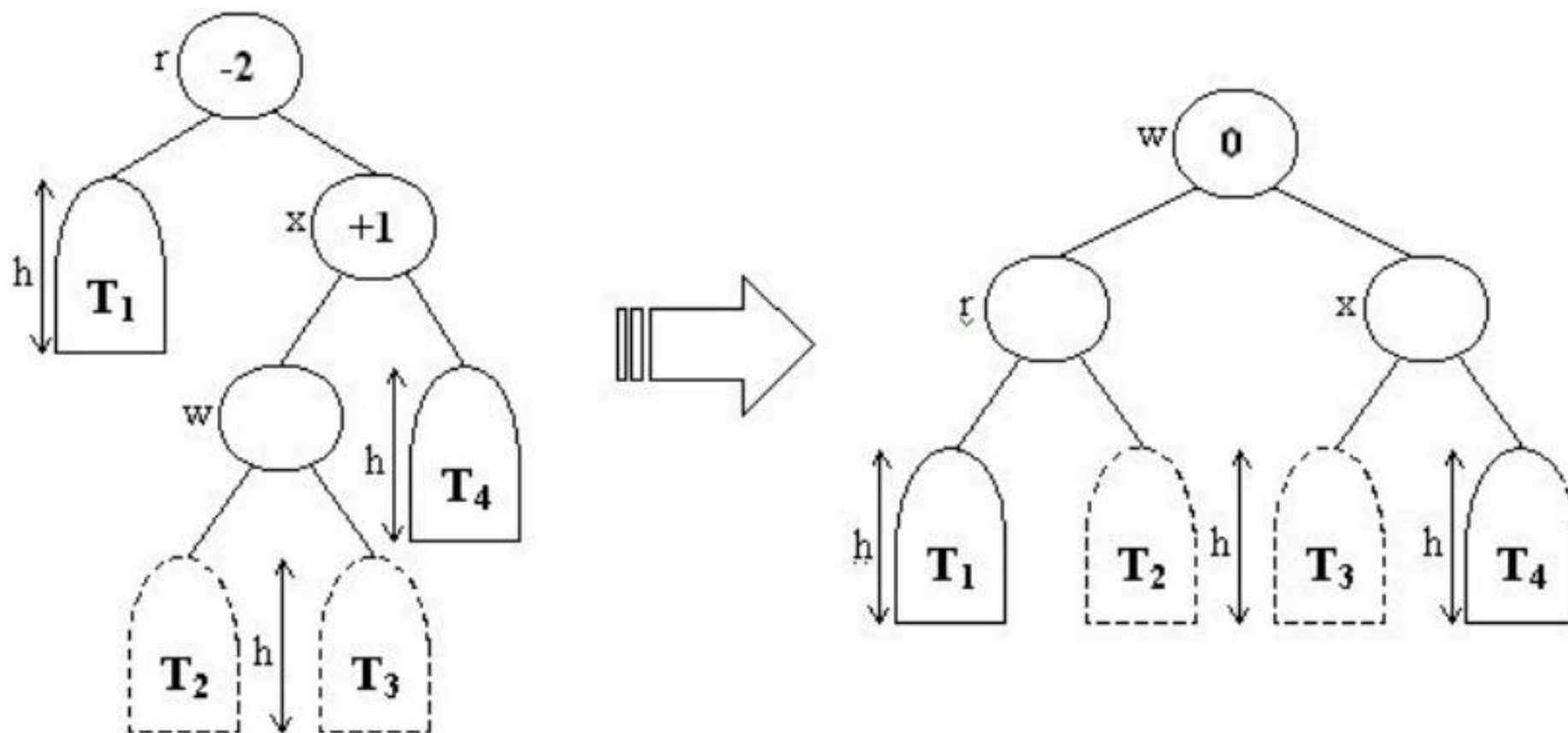
Rot. direita na
subárvore



Rot. esquerda
na árvore
original

Árvores AVL

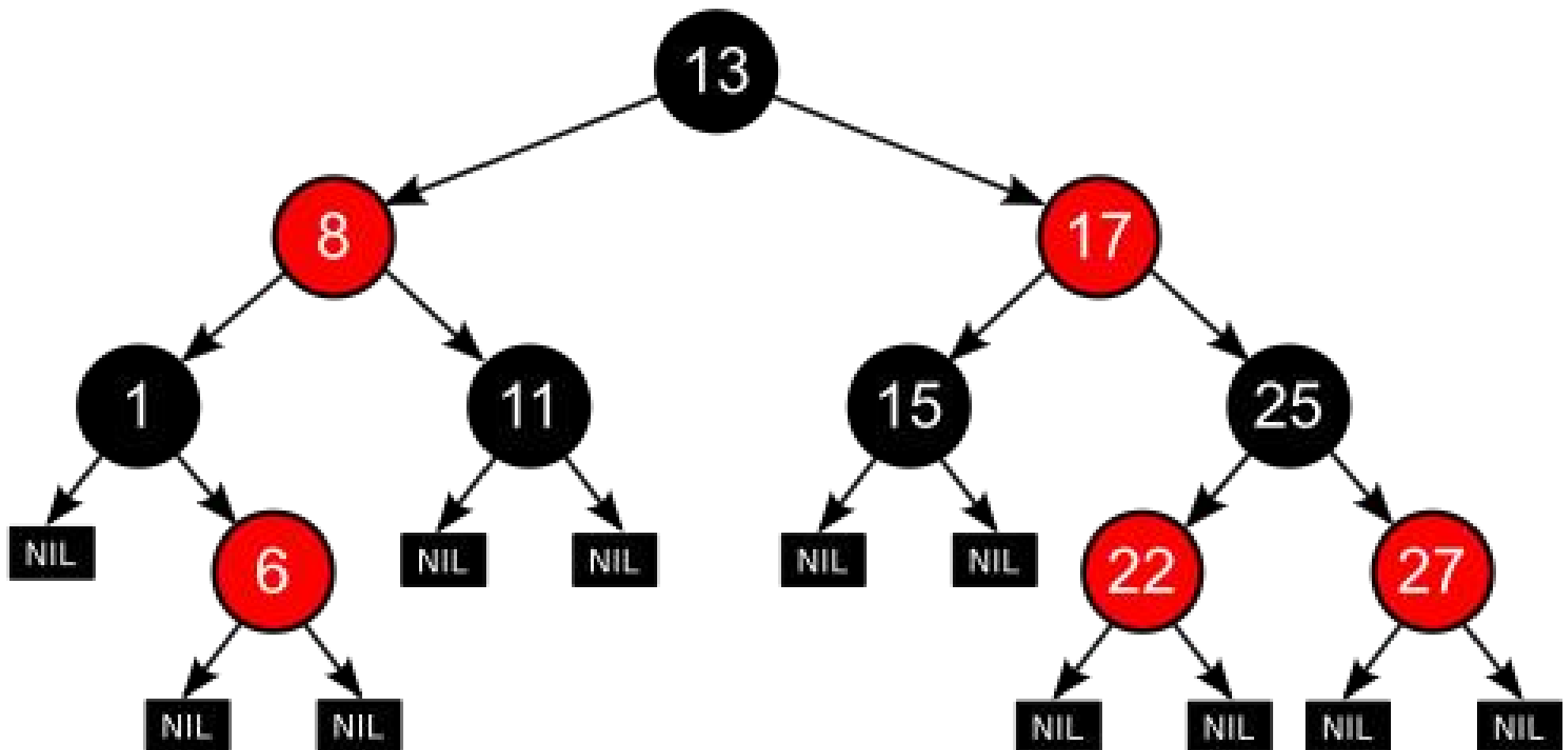
- Rotação RL (rotação e troca, anti-horário)



Outra forma de representar

Árvores Rubro-negras (Algoritmo DSW)

Árvores Rubro-negras



Árvores Rubro-negras

- São árvores de busca binária autoajustáveis, que procuram manter o balanceamento de forma automática
- Devem ser atendidas as restrições:
 - Um nó é vermelho ou preto
 - A raiz é preta
 - Todas as folhas são pretas
 - Os dois filhos de um nó vermelho são pretos
 - Todos os caminhos de algum nó para suas folhas contêm o mesmo número de nós pretos

Árvores Rubro-negras

- O caminho mais longo da raiz até uma folha será, no máximo, o dobro do menor caminho da raiz até qualquer folha
- Isso garante um balanceamento mínimo razoável
- As operações de inserção, remoção, e busca de valores necessitam de tempo de pior caso proporcional à altura da árvore
- Este limite proporcional a altura permite que sejam eficientes no pior caso

Árvores B

Árvores B

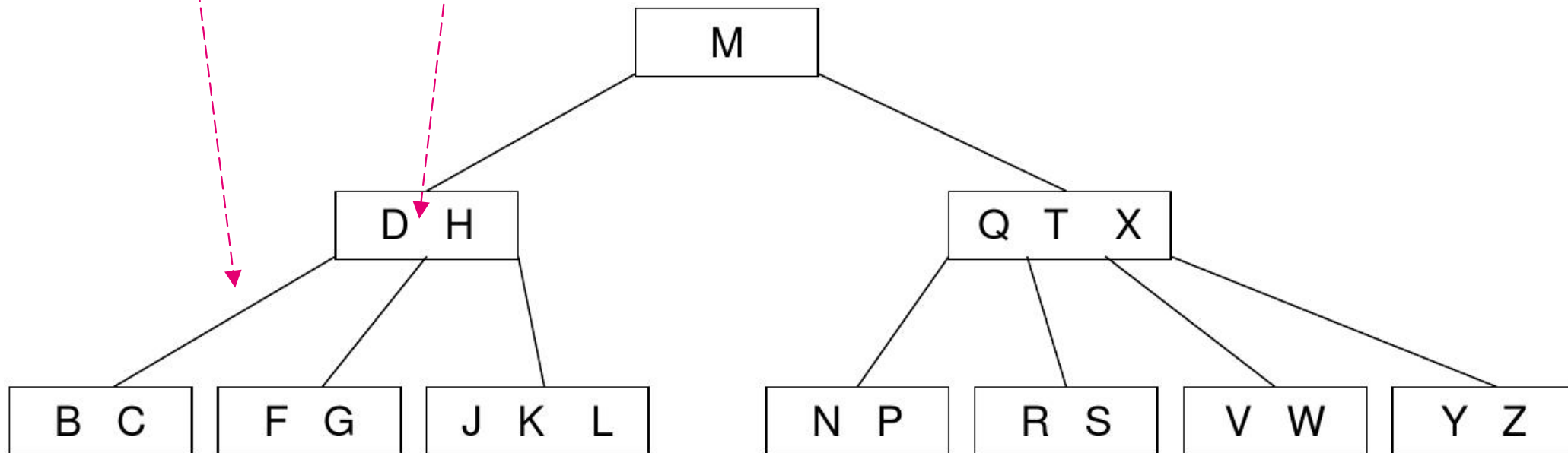
- Criadas por Bayes e McCreight em 1972
- Usada em aplicações onde a quantidade de dados é muito grande para que as chaves possam ser armazenadas somente em memória
- É necessário o uso de memória secundária, o que ocasiona um gasto de tempo significativo para acesso a um só nó dos dados
- Utiliza mais de uma chave em cada nó da estrutura
- Assegura que as folhas estão sempre no mesmo nível
- Utilizada como forma de armazenamento em memória secundária

Árvores B

- Definição:
 - Seja t a ordem da árvore B. Uma árvore B de ordem t é uma árvore ordenada que é vazia, ou que satisfaz as seguintes condições
 - A raiz é uma folha ou tem no mínimo dois filhos
 - Todas as folhas estão no mesmo nível
 - Cada nó diferente do raiz e das folhas possui no mínimo t filhos
 - Cada nó tem no máximo $2t$ filhos
- Como consequência da definição:
 - Cada nó (página) possui entre $t-1$ e $2t-1$ chaves, exceto o raiz que possui entre 1 e $2t-1$ chaves

Árvores B

- Exemplo $t=3$, cada nó tem
 - Entre $t-1$ e $2t-1$ chaves (2 e 5)
 - Entre t e $2t$ filhos (3 e 6)

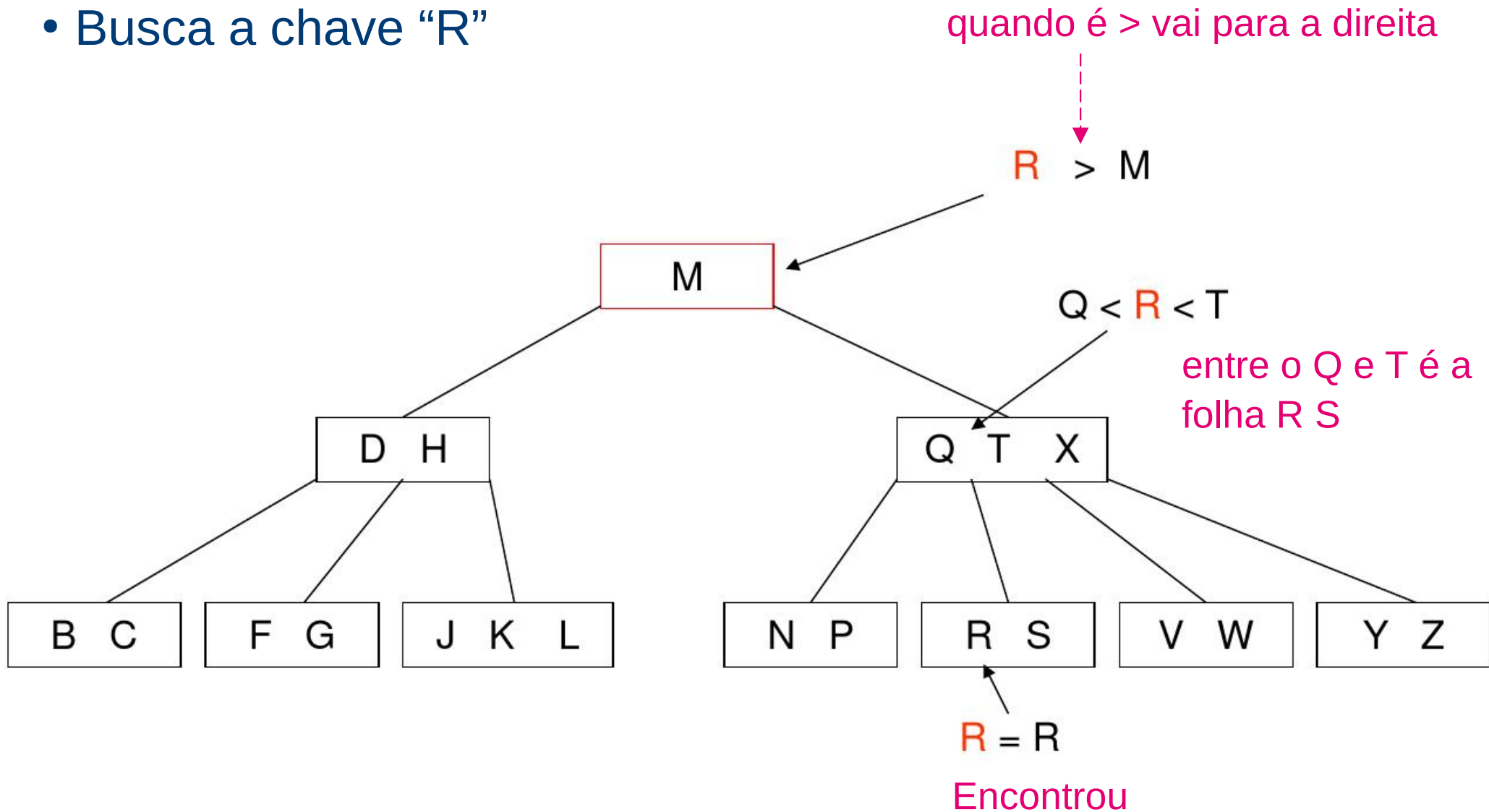


Árvores B

- Busca
 - Compara a chave **x** com a(s) chave(s) do nó raiz
 - Se a chave não estiver no nó raiz
 - A busca deve prosseguir em um certo filho dessa página
 - Escolhe-se qual filho explorar de forma parecida com a pesquisa realizada na árvore binária de pesquisa

Árvores B

- Busca a chave "R"



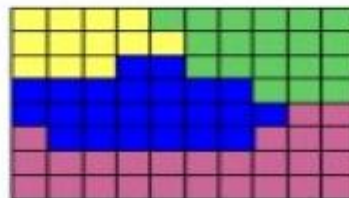
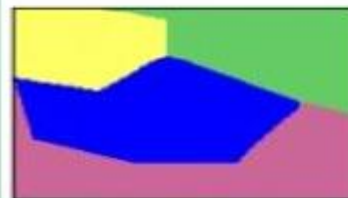
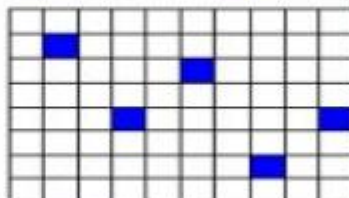
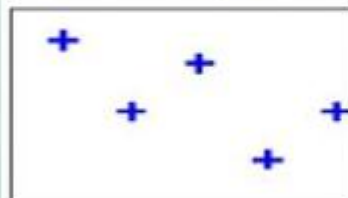
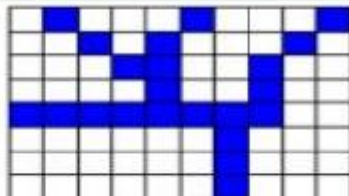
Estruturas multidimensionais

Estruturas multidimensionais

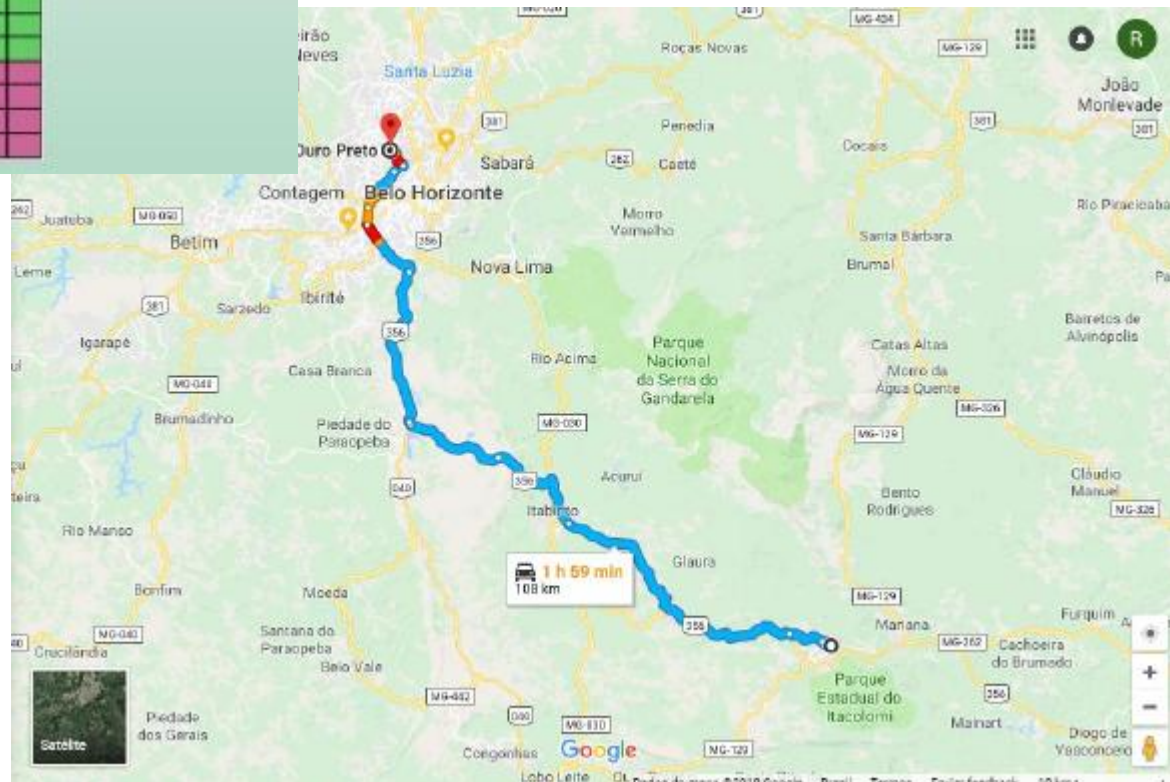
Vetorial & Matricial

Elementos representados de forma mais precisa

- Ponto
- Linha
- Polígono



Espaço subdividido em células (ou pixels)



Estruturas de dados espaciais - exemplos

- Quadtree
 - Extensão multidimensional da árvore de busca binária
- Grid files
 - É um método de acesso a pontos que divide um espaço em uma grade não periódica onde uma ou mais células da grade se referem a um pequeno conjunto de pontos
- Árvore k -d
 - É uma árvore binária em que cada nó é um ponto k -dimensional
- Árvore R
 - Similar à árvore B, mas que são usadas para métodos de acesso no espaço com o fim de indexar informação multidimensional

Referências

ALVES, R. Árvores B. [S. l.]: Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte, IFRN, 2010. Disponível em: <https://docente.ifrn.edu.br/robinsonalves/disciplinas/estruturas-de-dados-nao-lineares/ArvBrl.pdf>

ASSIS, G. T. de. Introdução à Estrutura de Dados Espaciais & QuadTree. [S. l.]: Universidade Federal de Ouro Preto, UFOP, 2018. Disponível em: http://www.decom.ufop.br/guilherme/BCC203/geral/ed2_introducao-estruturas-dados-espaciais_victor.pdf

HTAY, M. M. The DSW Algorithm. [S. l.]: Radford University, 2006. Disponível em: https://sites.radford.edu/~mhtay/ITEC360/webpage/Lecture/06_p2.pdf.

MANACERO JR., A. Árvores Balanceadas. [S. l.]: Unesp, 2007. Disponível em: <https://www.dcce.ibilce.unesp.br/~aleardo/cursos/ed2/arvbalanc.pdf>

GUERRA, R. Aula - Árvore AVL (vídeo). [S. l.]: Universidade Federal de Rio Grande, FURG, 2014. Disponível em: <https://youtu.be/3zmjQlJhBLM>