

Algoritmos e Estrutura de Dados



Aula 16 – Estrutura de Dados:
Árvores Vermelho-Preto (Parte II)
Prof. Tiago A. E. Ferreira

Introdução

- Como estudado na aula passada, uma árvore vermelho-preto é uma árvore de busca binária quase balanceada
 - Desta forma, ao inserir e deletar elementos há a necessidade re-estruturar o “balanceamento” da árvore.
 - Já foi estudado o re-estruturamento da árvore para o caso de inserção de elemento.
 - É para remoção?

Nodo de Árvores Vermelho-Preto

- Como em uma árvore binária, a remoção de um elemento de uma árvore vermelho-preto tem custo em tempo de $O(\lg n)$
- Para Remover um elemento, o processo é bem semelhante a de uma árvore binária:

RB-DELETE(T, z)

```
1  if esquerda[z] = nil[T] or direita[z] = nil[T]
2    then y ← z
3    else y ← TREE-SUCCESSOR(z)
4  if esquerda[y] ≠ nil[T]
5    then x ← esquerda[y]
6    else x ← direita[y]
7  p[x] ← p[y]
8  if p[y] = nil[T]
9    then raiz[T] ← x
10   else if y = esquerda[p[y]]
11         then esquerda[p[y]] ← x
12         else direita[p[y]] ← x
13  if y ≠ z
14    then chave[z] ← chave[y]
15                                     copia dados satélite de y em z
16  if cor[y] = PRETO
17    then RB-DELETE-FIXUP(T, x)
18  return y
```

Deletando

- Observe que:
 - A função RB-Delete-FixUp só é invocada caso Z (o nodo deletado) seja preto.
 - Caso o nodo deletado seja vermelho as propriedades vermelho-preto ficam inalteradas, visto que:
 - Nenhuma altura preta foi alterada
 - Nenhum nodo vermelho ficou adjacente
 - Se Z é vermelho, então Z não é Raiz: Assim a Raiz continua preto.

Deletando um Elemento

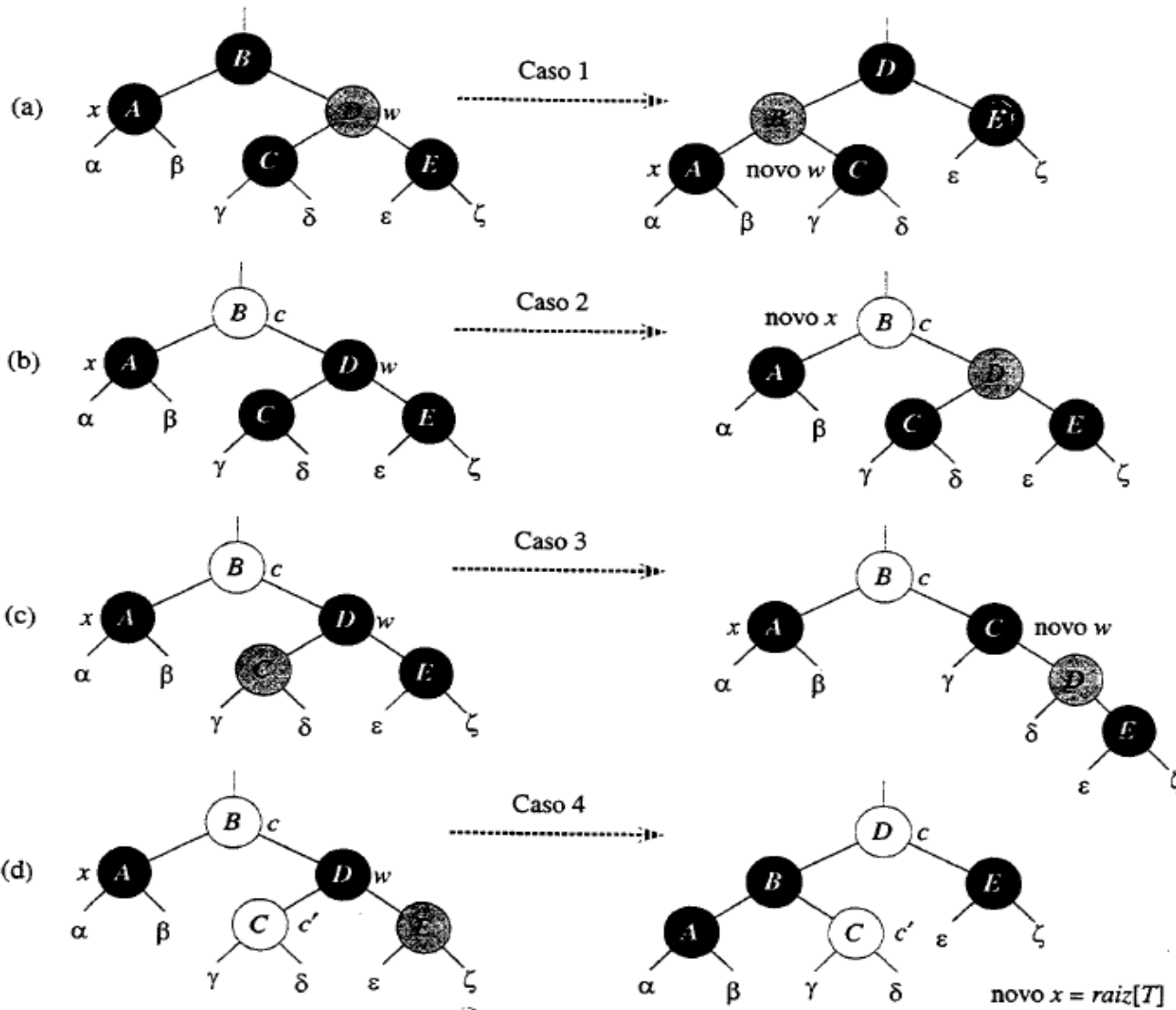
- Se o nodo y deletado é preto, podem ocorrer três problemas:
 - Se y era a raiz e um filho vermelho de y se torna a raiz (viola propriedade 2)
 - Se x e $P[y]$ (que agora tb é $P[x]$) eram vermelhos (viola propriedade 4)
 - A remoção de y faz com que o caminho que continha y tenha agora um nodo preto a menos (viola a propriedade 5)

Função RB-DELETE-FIXUP

RB-DELETE-FIXUP(T, x)

```
1 while  $x \neq \text{raiz}[T]$  e  $\text{cor}[x] = \text{PRETO}$ 
2   do if  $x = \text{esquerda}[p[x]]$ 
3     then  $w \leftarrow \text{direita}[p[x]]$ 
4         if  $\text{cor}[w] = \text{VERMELHO}$ 
5             then  $\text{cor}[w] \leftarrow \text{PRETO}$                                 ▷ Caso 1
6                  $\text{cor}[p[x]] \leftarrow \text{VERMELHO}$                         ▷ Caso 1
7                 LEFT-ROTATE( $T, p[x]$ )                                ▷ Caso 1
8                  $w \leftarrow \text{direita}[p[x]]$                             ▷ Caso 1
9         if  $\text{cor}[\text{esquerda}[w]] = \text{PRETO}$  e  $\text{cor}[\text{direita}[w]] = \text{PRETO}$ 
10            then  $\text{cor}[w] \leftarrow \text{VERMELHO}$                             ▷ Caso 2
11                 $x \leftarrow p[x]$                                         ▷ Caso 2
12            else if  $\text{cor}[\text{direita}[w]] = \text{PRETO}$ 
13                then  $\text{cor}[\text{esquerda}[w]] \leftarrow \text{PRETO}$                 ▷ Caso 3
14                     $\text{cor}[w] \leftarrow \text{VERMELHO}$                         ▷ Caso 3
15                    RIGHT-ROTATE( $T, w$ )                                ▷ Caso 3
16                     $w \leftarrow \text{direita}[p[x]]$                             ▷ Caso 3
17                     $\text{cor}[w] \leftarrow \text{cor}[p[x]]$                         ▷ Caso 4
18                     $\text{cor}[p[x]] \leftarrow \text{PRETO}$                             ▷ Caso 4
19                     $\text{cor}[\text{direita}[w]] \leftarrow \text{PRETO}$                     ▷ Caso 4
20                    LEFT-ROTATE( $T, p[x]$ )                                ▷ Caso 4
21                     $x \leftarrow \text{raiz}[T]$                                 ▷ Caso 4
22            else (igual à cláusula then com “direita” e “esquerda” trocadas)
23  $\text{cor}[x] \leftarrow \text{PRETO}$ 
```

Caso de RB-DELETE-FIXUP(T,x)



Caso de RB-DELETE-FIXUP(T, x)

- **Caso 1:**
 - O irmão w de x é vermelho
- **Caso 2:**
 - O irmão w de x é preto e ambos os filhos de w são pretos
- **Caso 3:**
 - O irmão w de x é preto e o filho da esquerda de w é vermelho e o da direita é preto
- **Caso 4**
 - O irmão w de x é preto e o filho da direita de w é vermelho

Inserção e Exclusão de Elementos

- Dada uma árvore vermelho-preta
 - Ao inserir um novo elemento ou ao deletar um elemento, as propriedades das árvores vermelho-preto podem ser comprometidas
 - Utiliza-se as operações de rotação para corrigir a distribuição de nodos vermelhos e pretos.

Exercícios Práticos

- **Exercício** : Implementar a funcionalidade de deletar um nodo em uma árvore vermelho-preta.