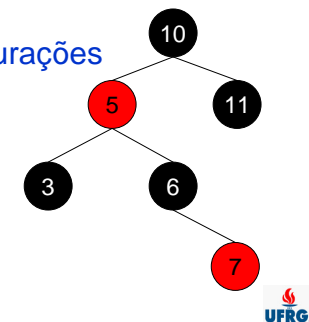


Árvores Rubro-Negras (Vermelho-Preta)

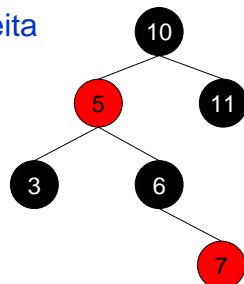
Árvores Rubro-Negras

- *Árvore Binária de Pesquisa* (ABP) com nodos coloridos de vermelho e preto
 - Árvore balanceada
 - Qualquer caminho da raiz até as folhas, **nenhum caminho será maior que duas vezes o comprimento de qualquer outro**
 - Aproximadamente balanceada
 - Número menor de rotações/reestruturações
 - Comparada com AVL



Estrutura da Árvore

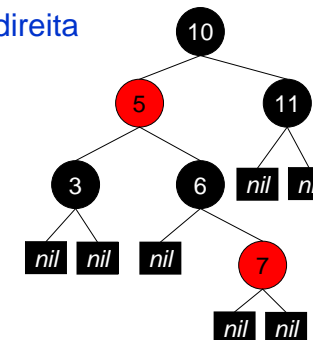
- Cada nodo contem os seguintes campos:
 - Chave
 - Ponteiro para subárvores esquerda
 - Ponteiro para subárvores direita
 - Cor



- Propriedade
 - Todo nodo da árvore é ou **vermelho** ou **preto**

Estrutura da Árvore

- Cada nodo contem os seguintes campos:
 - Chave
 - Ponteiro para subárvores esquerda
 - Ponteiro para subárvores direita
 - Cor

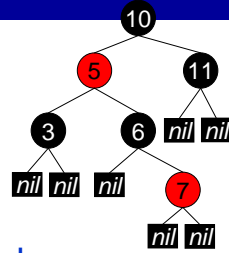


- Propriedade
 - Todo nodo da árvore é ou **vermelho** ou **preto**

Propriedades

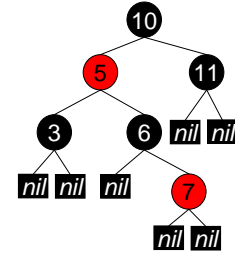
- I. Todo nodo é vermelho ou preto
- II. A raiz é preta
- III. Toda folha (*nil*) é preta
- IV. Se um nodo é vermelho, então ambos os seus filhos são pretos
- V. Para cada nodo, todos os caminhos desde um nodo até as folhas descendentes contêm o mesmo número de nodos pretos

BALANCEAMENTO



Propriedades - RESUMO

- RAIZ
 - A raiz é preta
- NODOS EXTERNOS
 - Todo nodo externo é preto
- NODOS INTERNOS
 - Os filhos de um nodo vermelho são pretos
- PROFUNDIDADE
 - Todos os *nodos externos* têm a mesma **profundidade preta** que é definida como o número de ancestrais pretos menos 1



INSERÇÃO

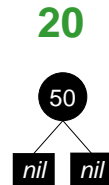
Inserção

- Encontra a posição na árvore
 - Substitui *nil* pelo nodo com 2 filhos *nil*
- Primeiro nodo (RAIZ)
 - **PRETO**
- demais nodos
 - **VERMELHO**

50

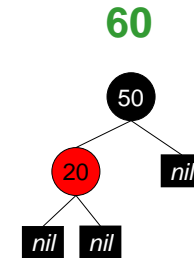
Inserção

- Encontra a posição na árvore
 - Substitui *nil* pelo nodo com 2 filhos *nil*
- Primeiro nodo (RAIZ)
 - PRETO
- demais nodos
 - VERMELHO



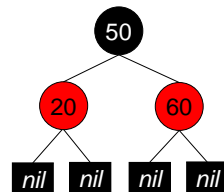
Inserção

- Encontra a posição na árvore
 - Substitui *nil* pelo nodo com 2 filhos *nil*
- Primeiro nodo (RAIZ)
 - PRETO
- demais nodos
 - VERMELHO



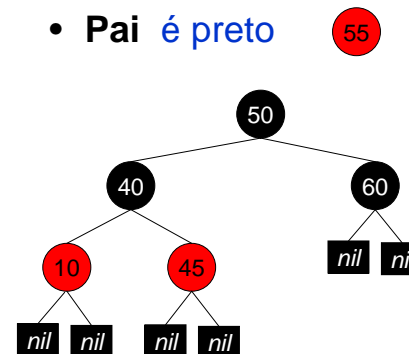
Inserção – regra básica

- Encontra a posição na árvore
 - Substitui *nil* pelo nodo a ser inserido
 - Novo nodo possui 2 filhos *nil*
- Primeiro nodo (RAIZ)
 - PRETO
- demais nodos
 - VERMELHO



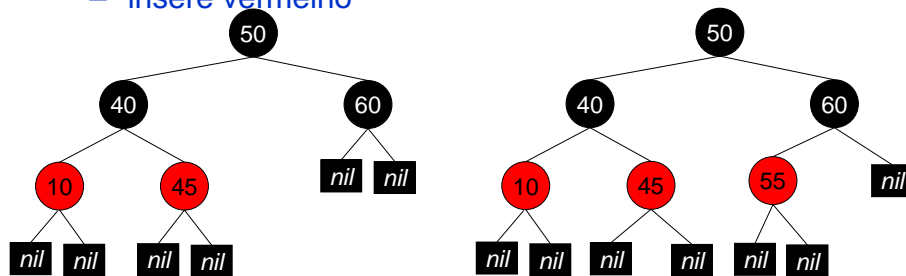
Caso 1

- Pai é preto



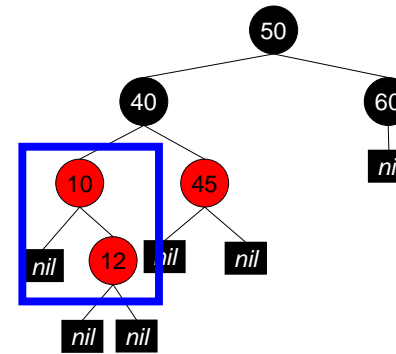
Caso 1

- **pai** é preto – 55
– insere vermelho



Caso 2

- **pai** é vermelho 12



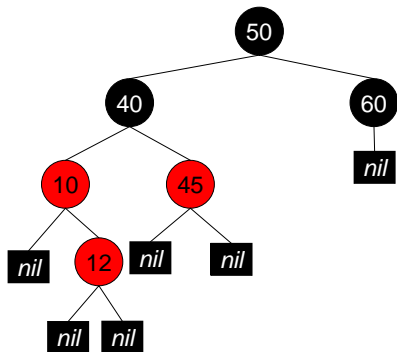
Se **pai** não é raiz

- seu **avô** é preto (*óbvio*)
- verificar a cor do **tio**

⇒ **2 casos**

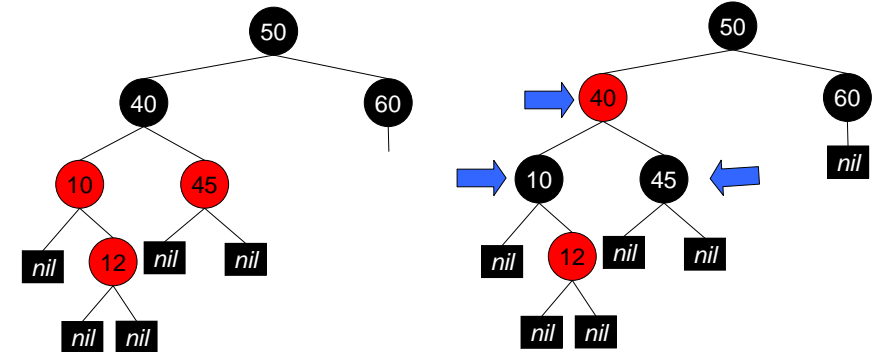
Caso 2.1

- **tio** é vermelho 12
- Alterar as cores **pai**, **tio** e do **avô**



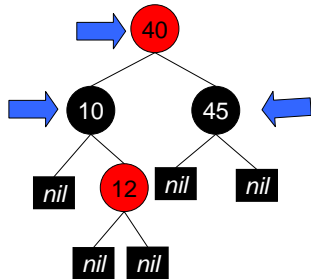
Caso 2.1

- **tio** é vermelho 12
- Alterar as cores **pai**, **tio** e do **avô**



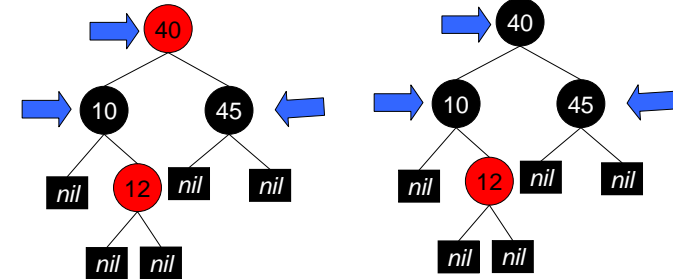
Caso 2.1 – Exceção RAIZ é avô

- **tio** é vermelho 12 Alterar as cores **pai**, **tio** e do **avô**



Caso 2.1 – Exceção RAIZ é avô

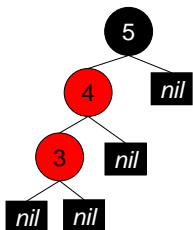
- **tio** é vermelho 12 Alterar as cores **pai**, **tio** e do **avô**



Avô é raiz – altera para preto

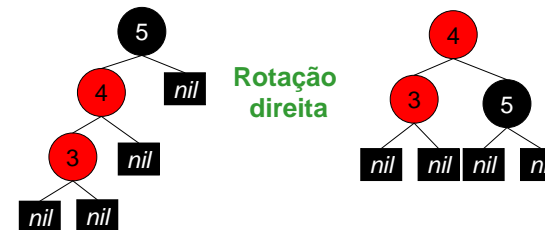
Caso 2.2 (A) - rotação direita

- **tio** é preto



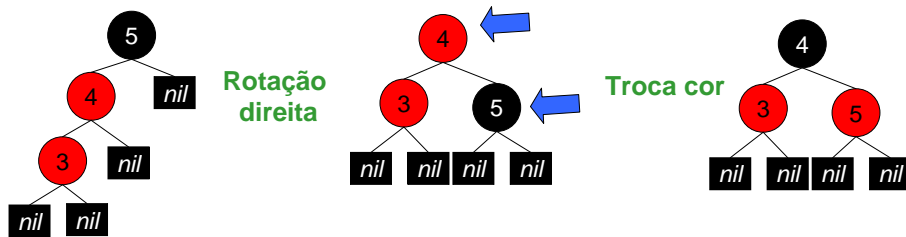
Caso 2.2 (A) - rotação direita

- **tio** é preto



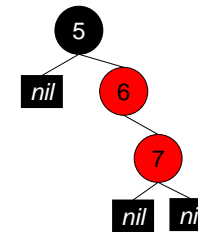
Caso 2.2 (A) - rotação direita

- tio é preto



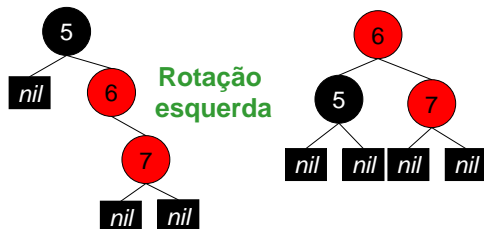
Caso 2.2 (B) - rotação esquerda

- tio é preto



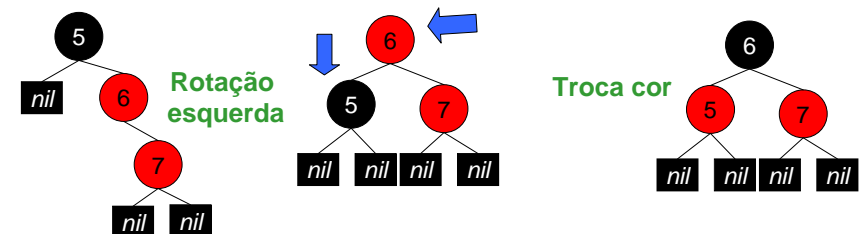
Caso 2.2 (B) - rotação esquerda

- tio é preto



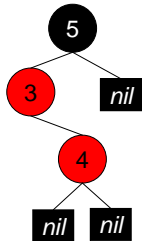
Caso 2.2 (B) - rotação esquerda

- tio é preto



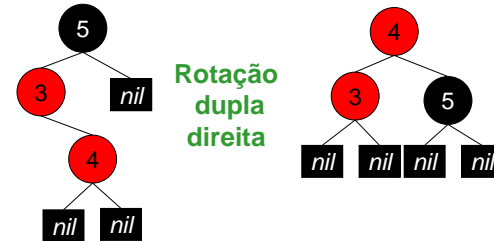
Caso 2.2 (C) - rotação dupla direita

- tio é preto



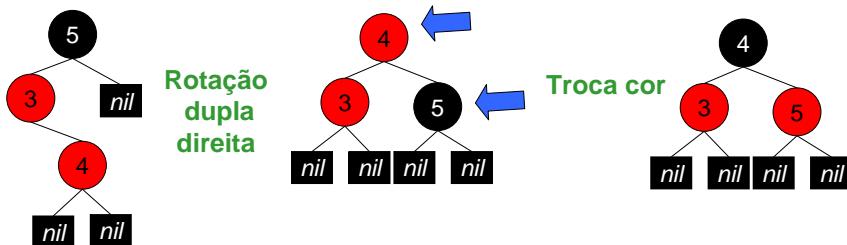
Caso 2.2(C) - rotação dupla direita

- tio é preto



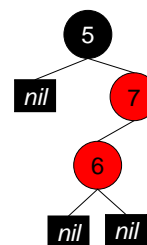
Caso 2.2 (C) - rotação dupla direita

- tio é preto



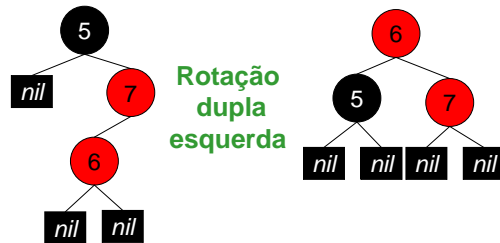
Caso 2.2 (D) - rotação dupla esquerda

- tio é preto



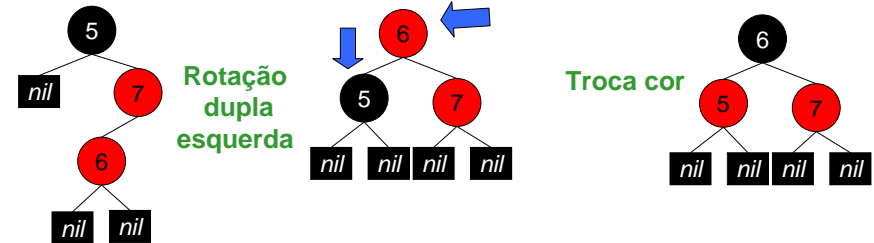
Caso 2.2 (D) - rotação dupla esquerda

- tio é preto



Caso 2.2 (D) - rotação dupla esquerda

- tio é preto



Inserção - resumo

- Raiz
 - Preto
- Demais filhos
 - vermelhos
- Caso 01 – pai preto
 - Insere vermelho
- Caso 02 – pai vermelho
 - Tio Vermelho
 - troca cor pai, tio, avô
 - Tio Preto
 - rotação e troca de cores

Problema:

DUPLO VERMELHO

REMOÇÃO

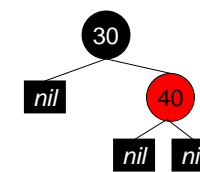
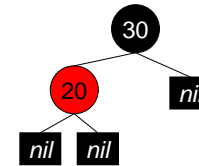
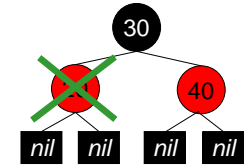
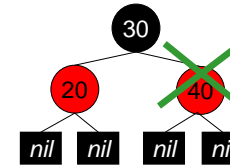
Remoção

- Remoção nodo intermediário
 - Não há problema porque as cores permanecem iguais
Existe apenas a troca de valores
- Nodo Vermelho
 - Ok! Não altera o balanceamento da árvore
- Nodo Preto
 - **PROBLEMA**

Problema:
DUPLO PRETO

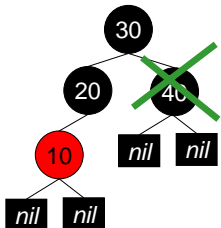
Caso 0

- Nodo vermelho



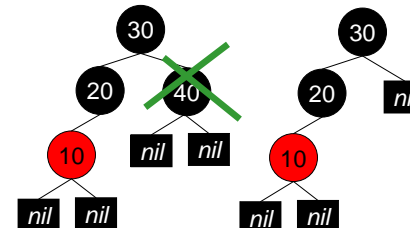
Caso 01

- Irmão preto – filho vermelho



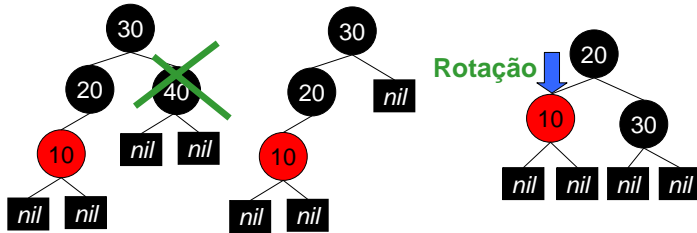
Caso 01

- Irmão preto – filho vermelho



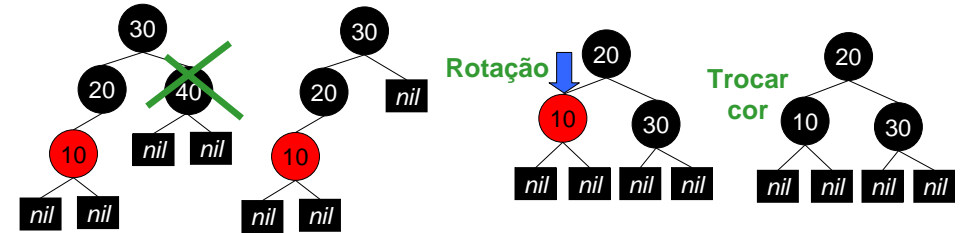
Caso 01

- Irmão preto – filho vermelho



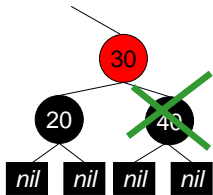
Caso 01

- Irmão preto – filho vermelho



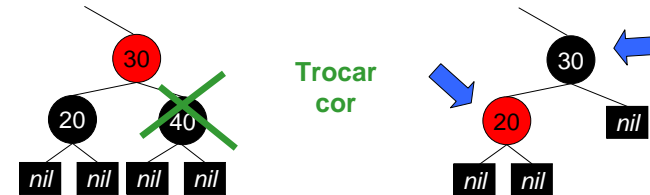
Caso 02

- Irmão preto – dois filhos pretos



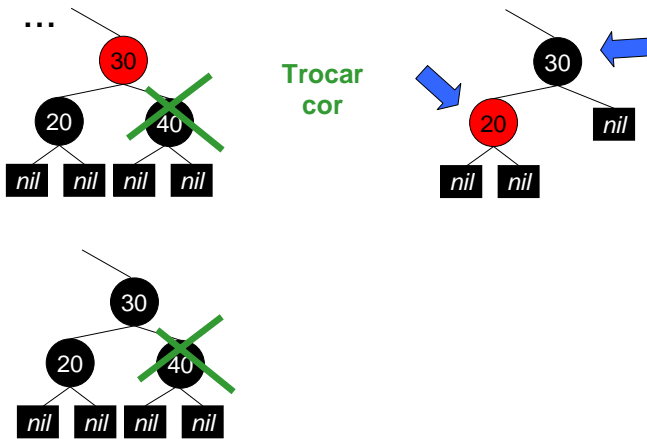
Caso 02

- Irmão preto – dois filhos pretos



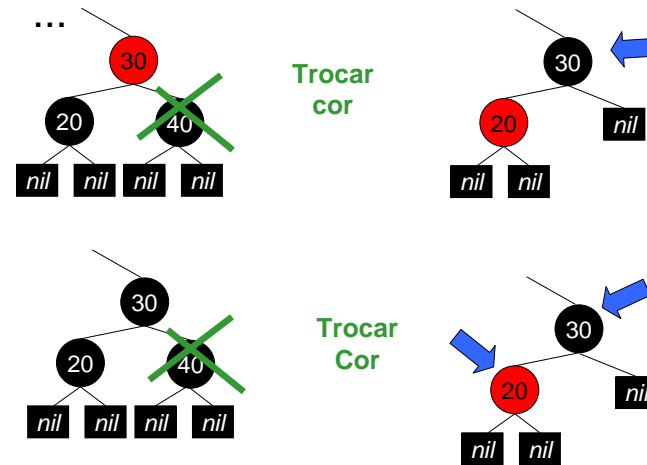
Caso 02

- Irmão preto – dois filhos pretos



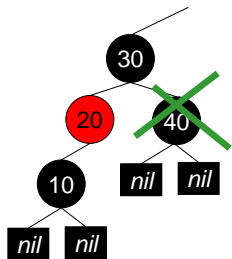
Caso 02

- Irmão preto – dois filhos pretos



Caso 03

- Irmão vermelho



Caso 03

- Irmão vermelho



Demos

- <http://webpages.ull.es/users/jriera/Docencia/AVL/AVL%20tree%20applet.htm>
- <http://people.ksp.sk/~kuko/bak/>