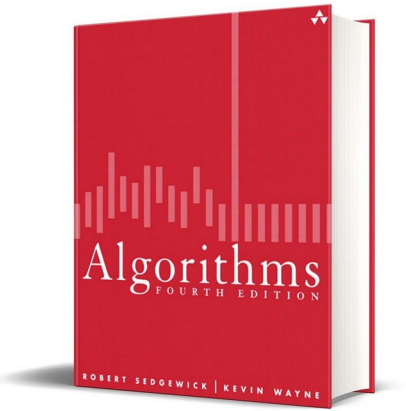
SCC0202 – Algoritmos e Estruturas de Dados I

Árvores Rubro-Negras

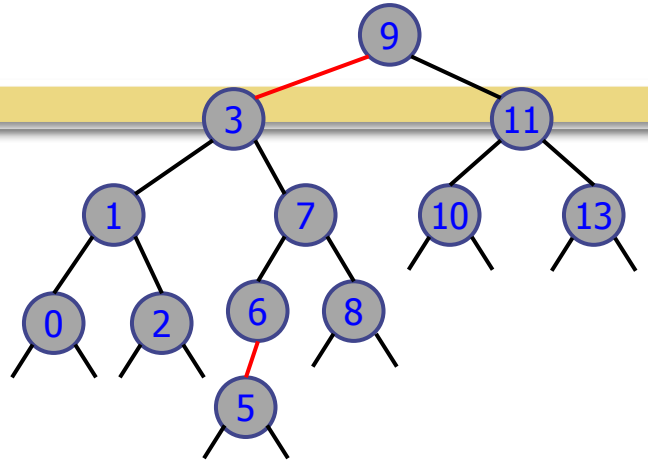
Prof.: Dr. Rudinei Goularte
(rudinei@icmc.usp.br)
Instituto de Ciências Matemáticas e de Computação - ICMC
Sala 4-229

Outras árvores balanceadas de busca

- 2-3 Trees, 2-3-4 Trees, Red-Black Trees,…
- Árvores Rubro-Negras
  - Guibas & Sedgewick, 1978
- **Left-Leaning Red-Black Trees**
  - Sedgewick, 2008



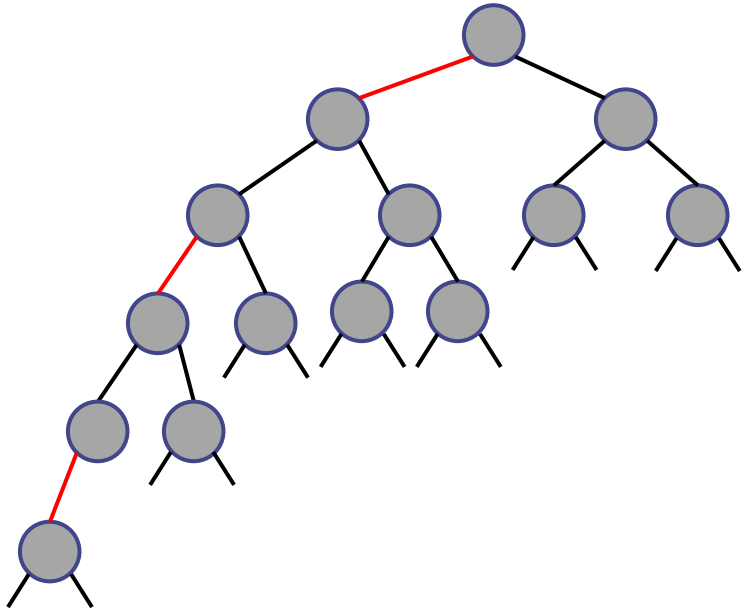
Árvore Rubro-Negra



- Estudaremos a versão LLRB
- É uma ABB
- Arestas são coloridas
  - Vermelha
  - Preta
- Regras
  - 1. Aresta vermelha sempre vai para o filho esquerdo
  - 2. Todo nó possui, no máximo, uma aresta vermelha
  - 3. Balanceamento negro perfeito
    - Todo filho nulo está à mesma distância da raiz – distância negra
      - Considerando apenas as arestas negras

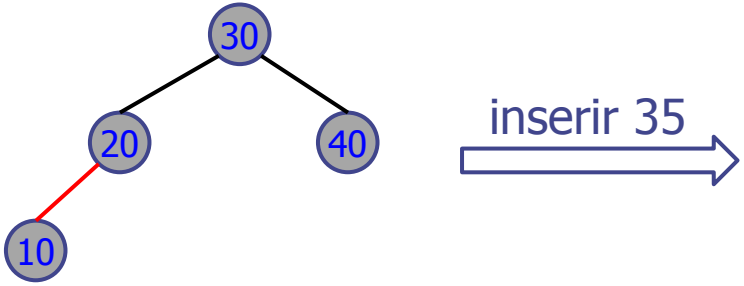
Árvore Rubro-Negra

- Altura no pior caso
  - h <= 2 log(n)



Árvore Rubro-Negra

- Inserção
  - Igual à ABB
  - Porém, todo nó inserido possui aresta incidente **vermelha**!
- Inserção em nó (pai) sem incidência vermelha



Árvore Rubro-Negra

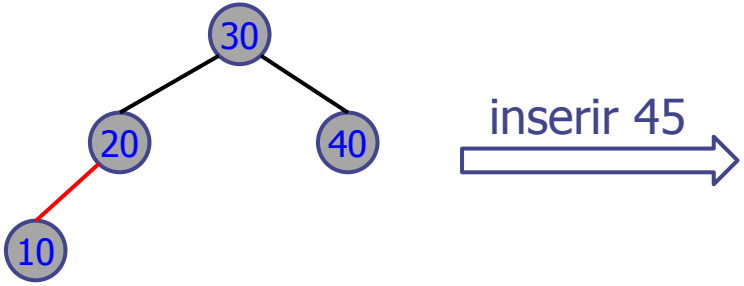
- Inserção em nó (pai) sem incidência vermelha

Árvore Rubro-Negra

- Inserção em nó (pai) sem incidência vermelha

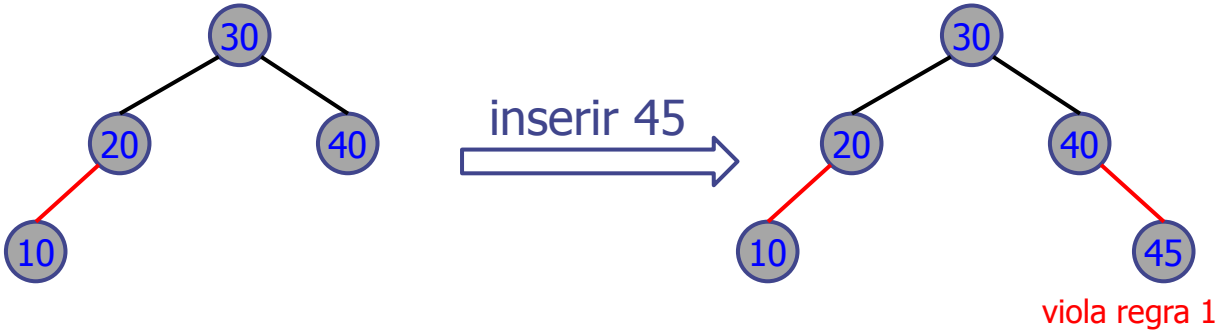
Árvore Rubro-Negra

- Inserção em nó sem incidência vermelha



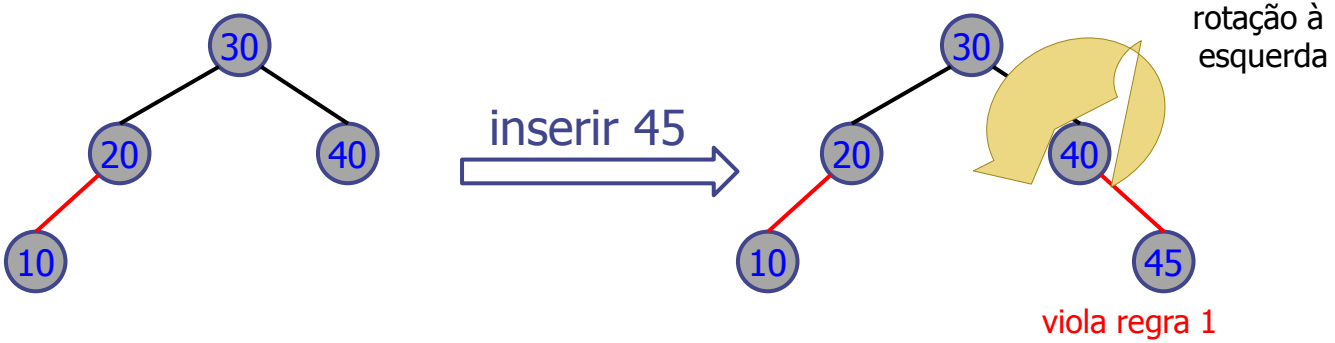
Árvore Rubro-Negra

- Inserção em nó sem incidência vermelha
  - Igual à ABB



Árvore Rubro-Negra

- Inserção em nó sem incidência vermelha
  - Igual à ABB

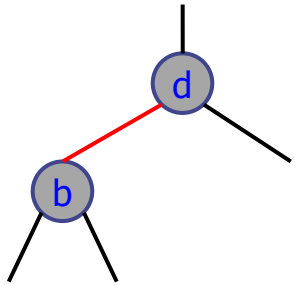


Árvore Rubro-Negra

- Inserção em nó sem incidência vermelha
  - Igual à ABB

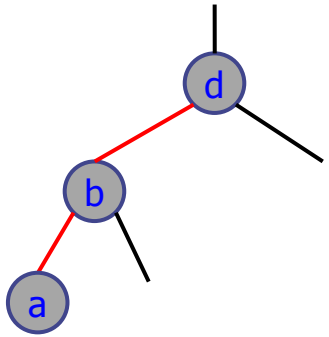
Árvore Rubro-Negra

- Inserção em nó com incidência vermelha
  - 3 casos
    - Todos violam a regra 2



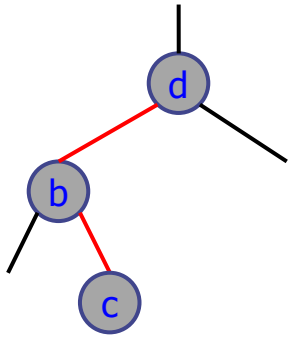
Árvore Rubro-Negra

- Inserção em nó (pai) com incidência vermelha
  - 3 casos
    - Todos violam a regra 2



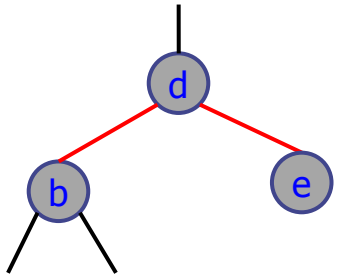
Árvore Rubro-Negra

- Inserção em nó com incidência vermelha
  - 3 casos
    - Todos violam a regra 2



Árvore Rubro-Negra

- Inserção em nó com incidência vermelha
  - 3 casos
    - Todos violam a regra 2

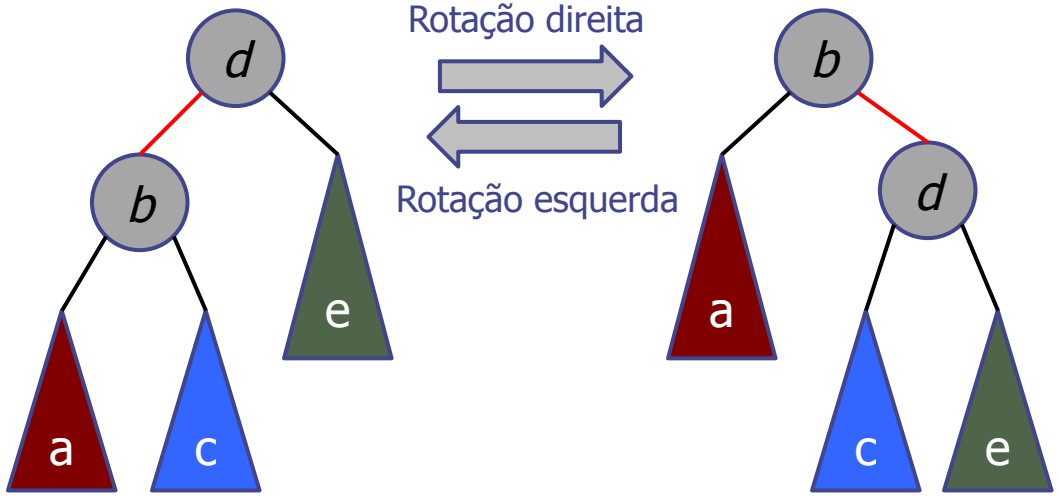


Árvore Rubro-Negra

- Manutenção das 3 regras após
  - Inserção
  - Remoção
- Operações locais
  - Rotação à esquerda
  - Rotação à direita
  - Inversão de cores

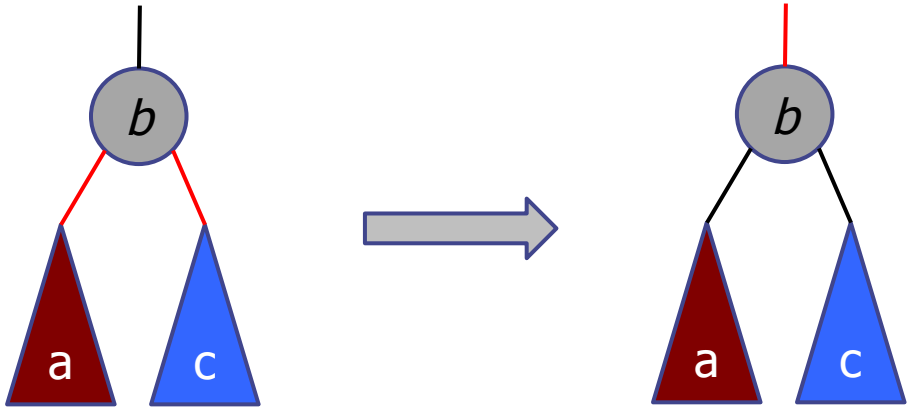
Árvore Rubro-Negra

- Rotações à esquerda e à direita
  - Parecidas com AVL



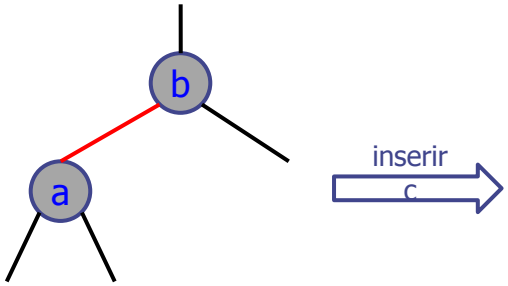
Árvore Rubro-Negra

- Inversão de cores (arestas)

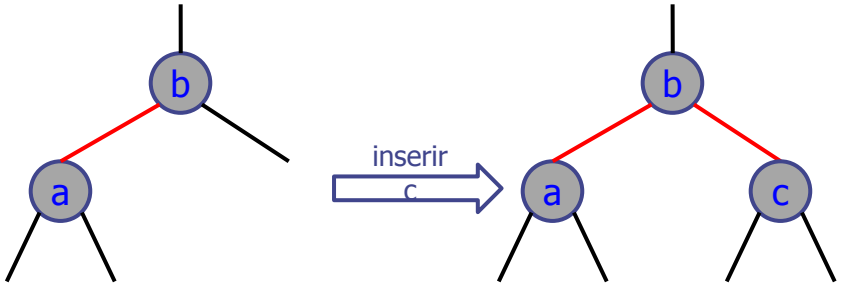


nenhuma das 3 operações infringe a regra 3!

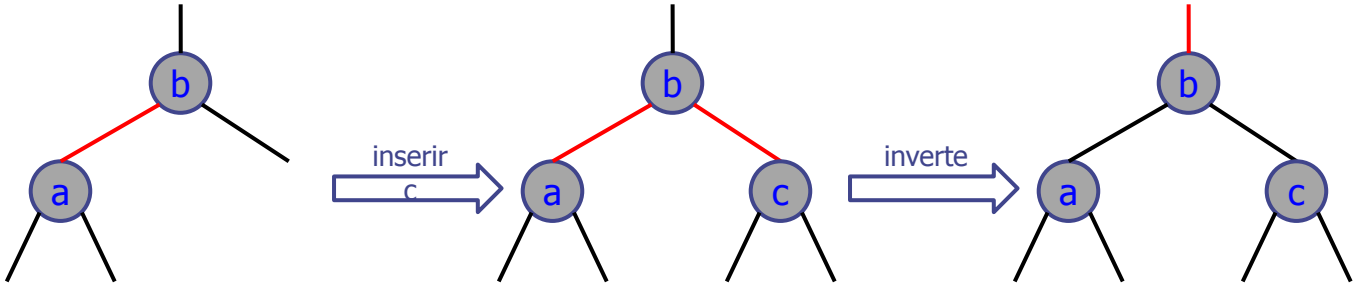
Inserção em nó com incidência vermelha (1o. caso)



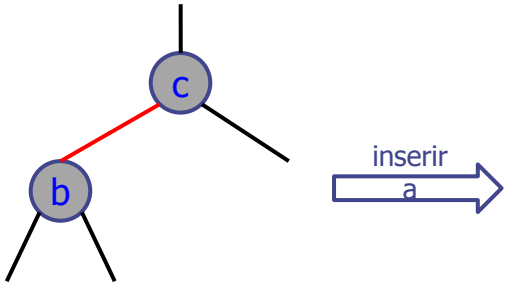
Inserção em nó com incidência vermelha (1o. caso)



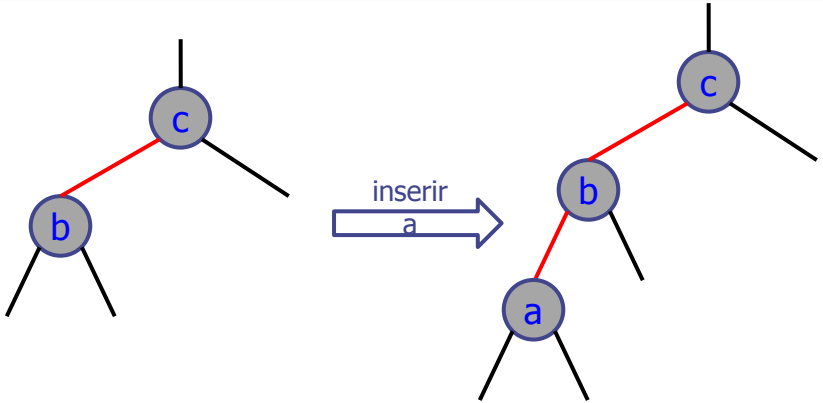
Inserção em nó com incidência vermelha (1o. caso)



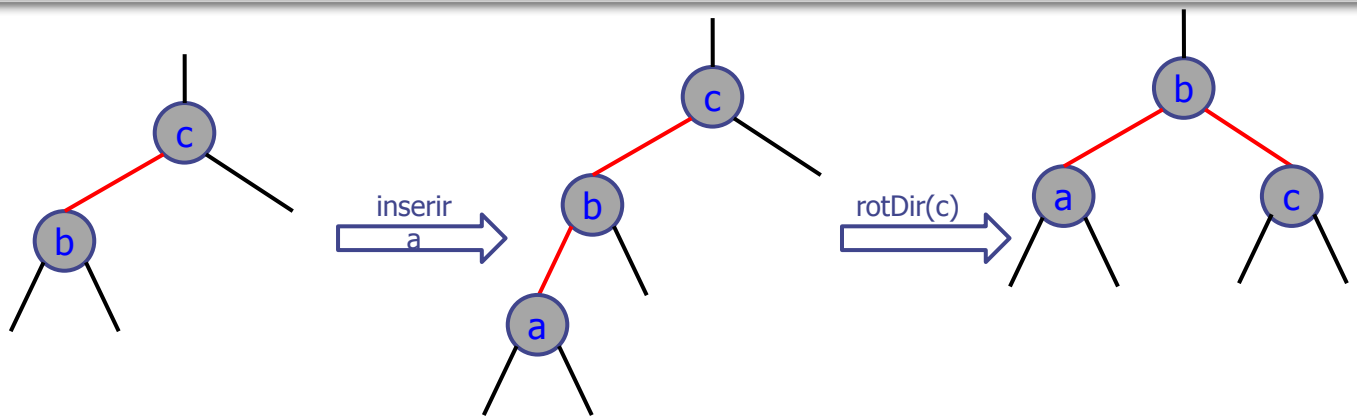
Inserção em nó com incidência vermelha (2o. caso)



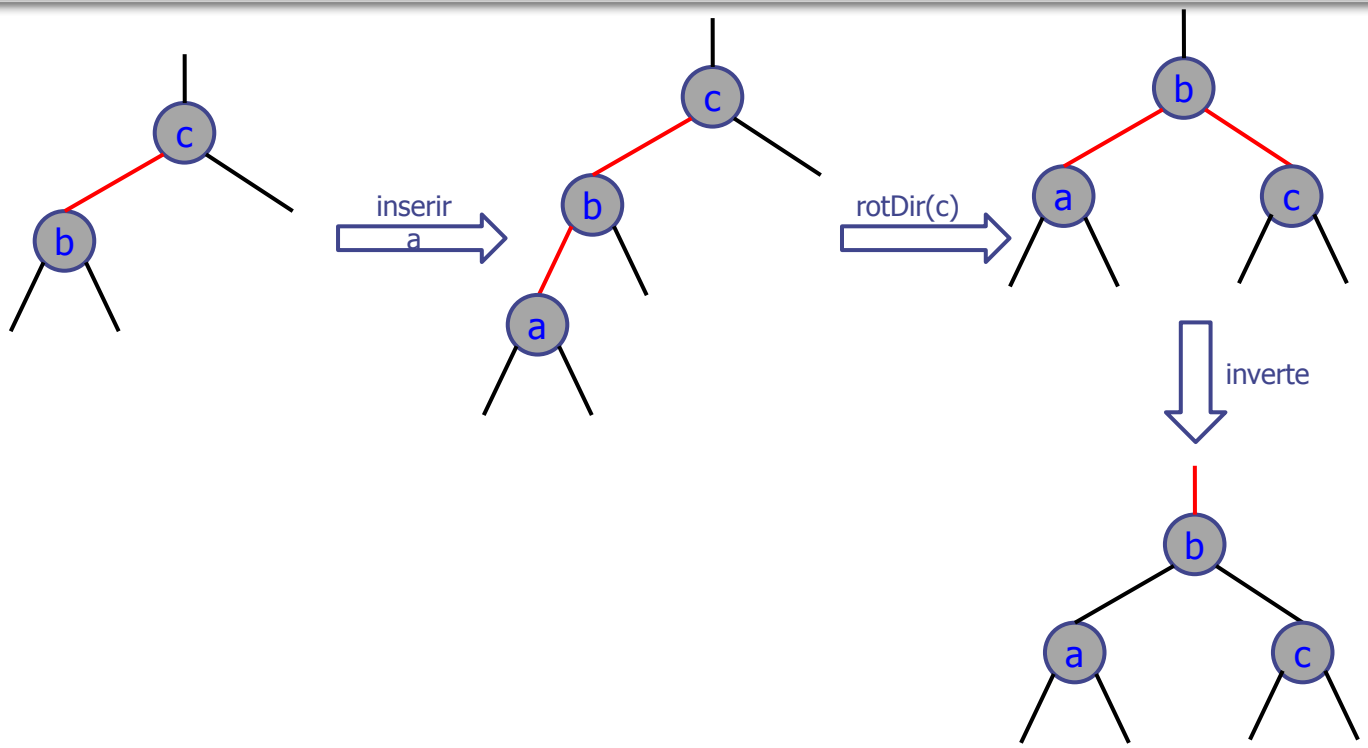
Inserção em nó com incidência vermelha (2o. caso)



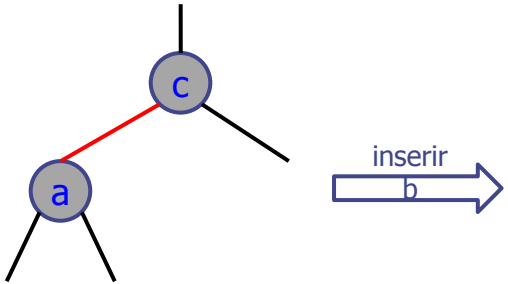
Inserção em nó com incidência vermelha (2o. caso)



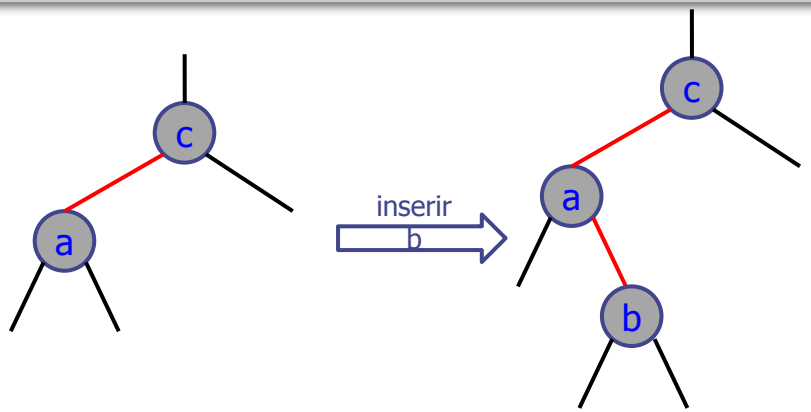
Inserção em nó com incidência vermelha (2o. caso)



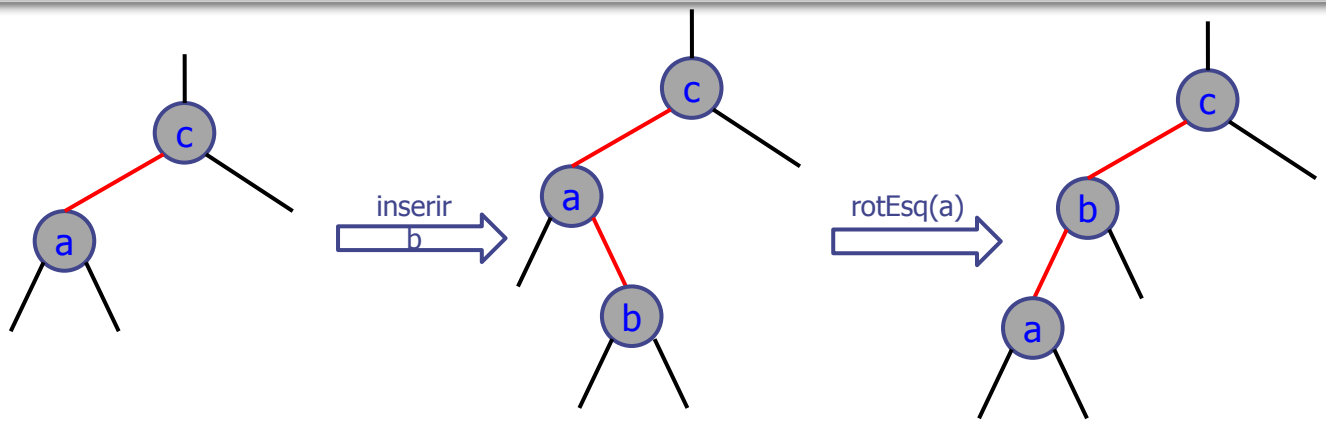
Inserção em nó com incidência vermelha (3o. caso)



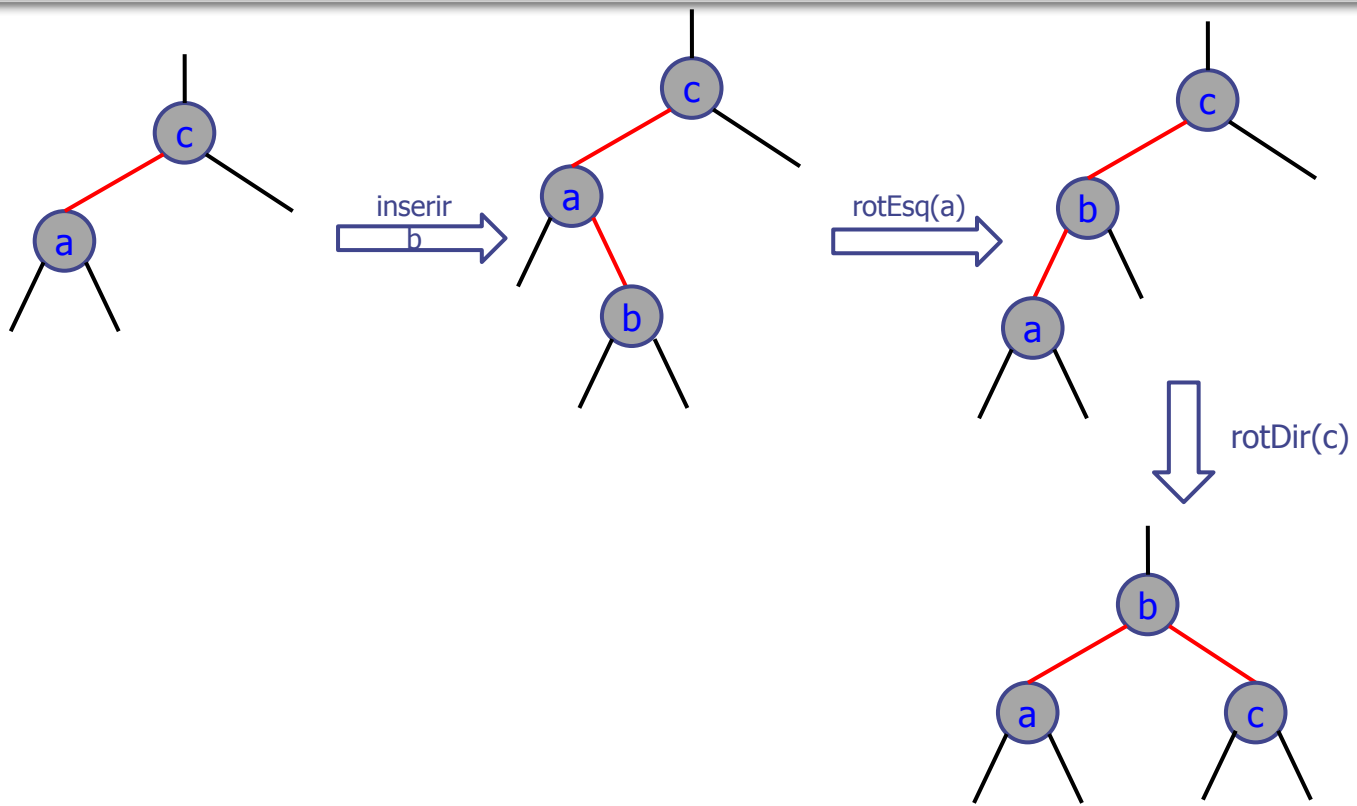
Inserção em nó com incidência vermelha (3o. caso)



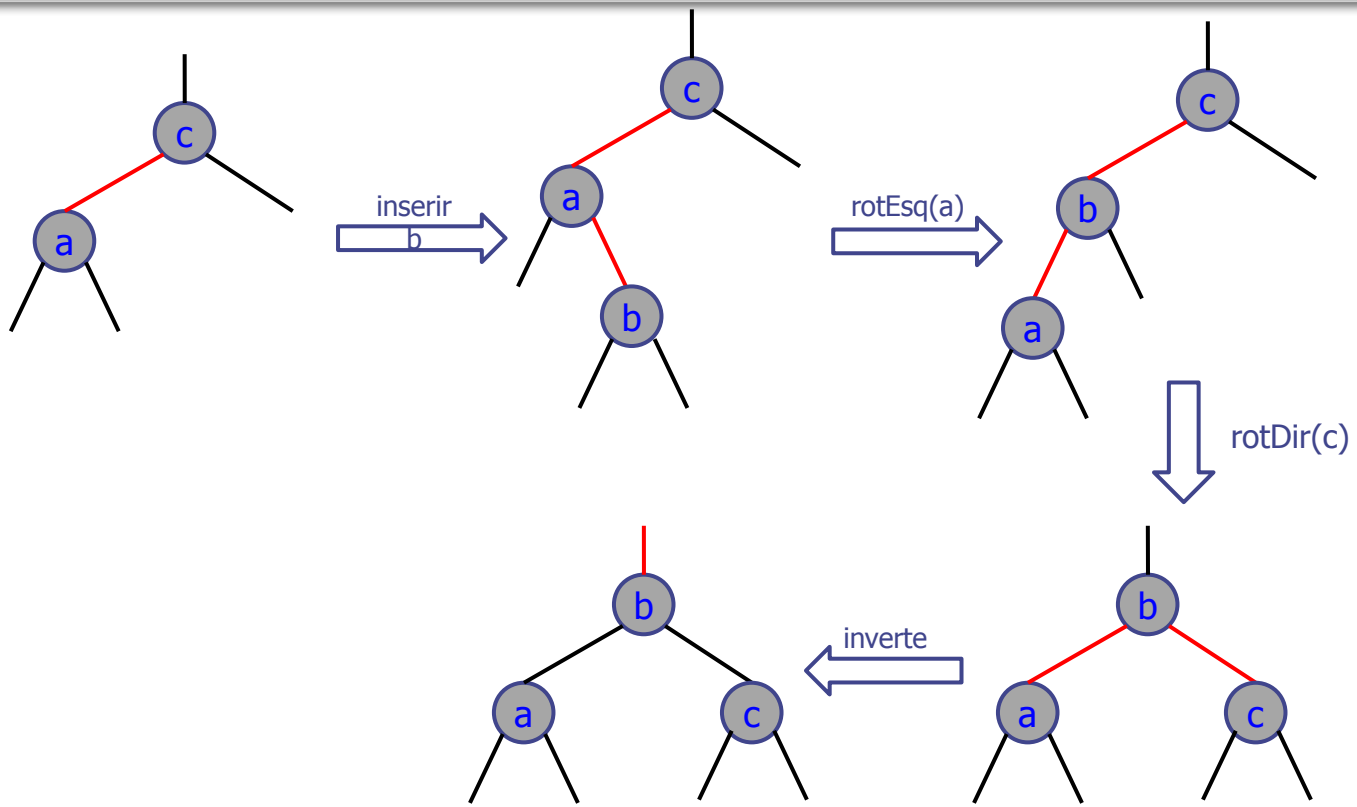
Inserção em nó com incidência vermelha (3o. caso)



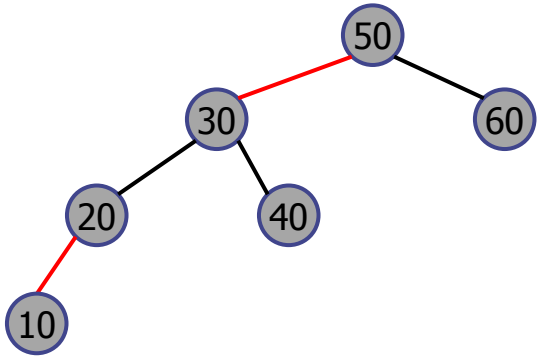
Inserção em nó com incidência vermelha (3o. caso)



Inserção em nó com incidência vermelha (3o. caso)

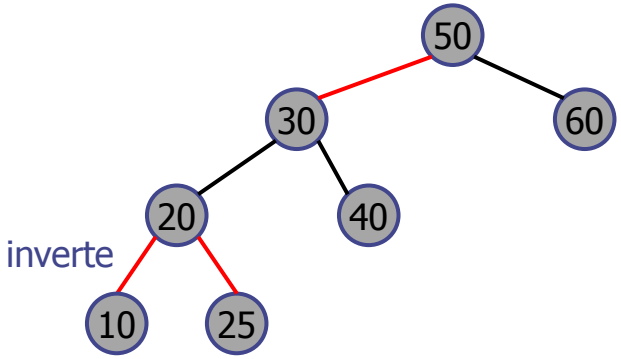


Exemplo



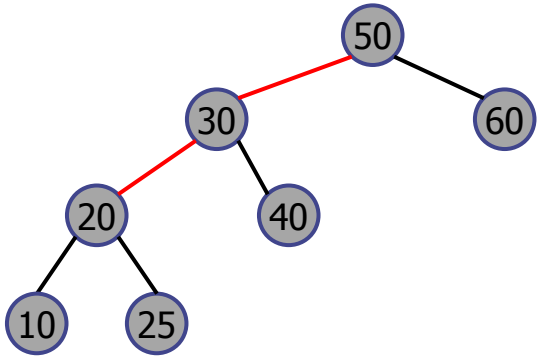
inserir 25

Exemplo



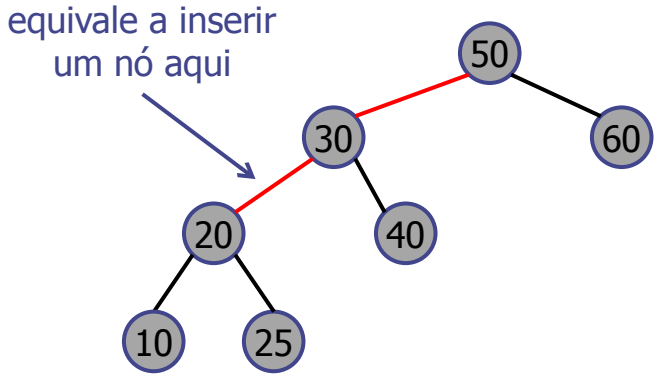
inserir 25

Exemplo



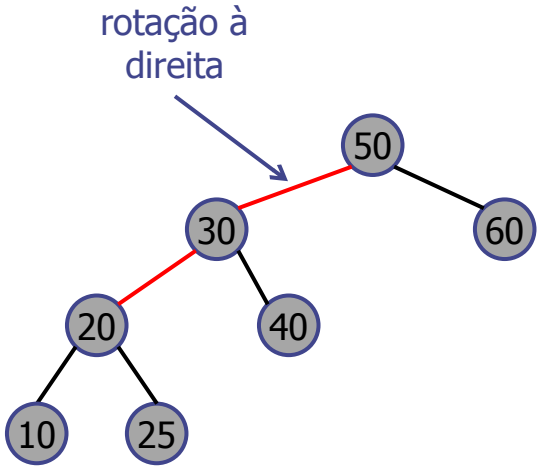
inserir 25

Exemplo



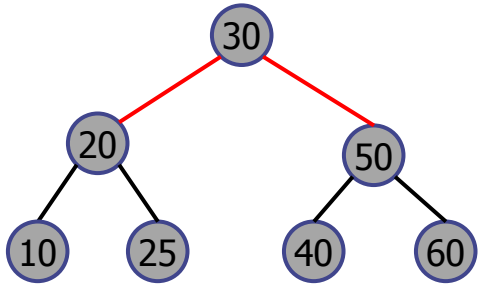
inserir 25

Exemplo



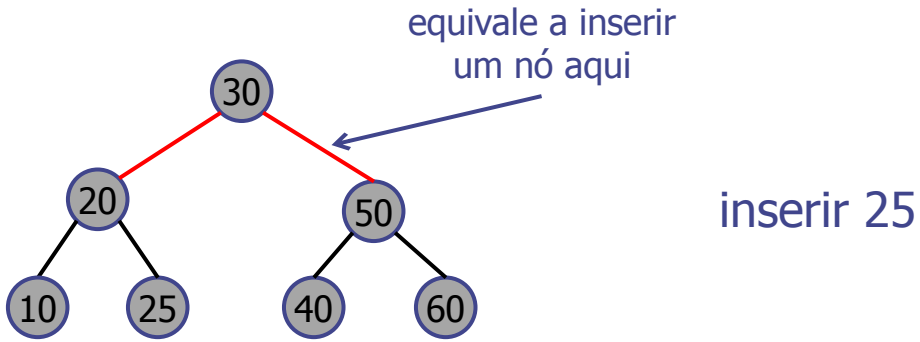
inserir 25

Exemplo



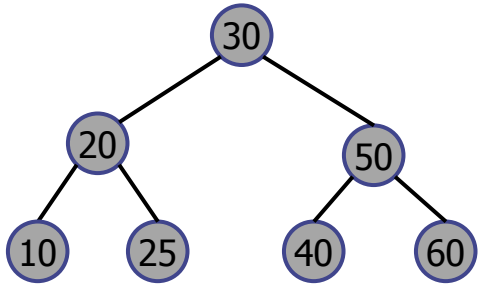
inserir 25

Exemplo



inserir 25

Exemplo



inserir 25

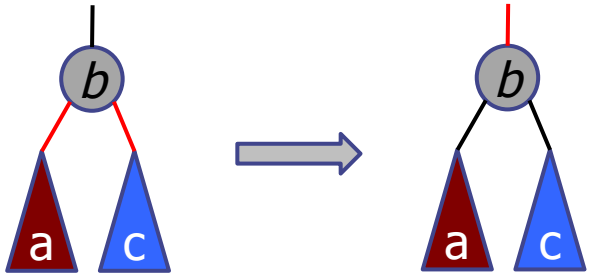
Exercício

- Inserir em uma árvore rubro-negra as seguintes chaves:
  - 10, 15, 20, 17, 25, 5, 12

Operações

Inverte

Obs.: preto = 0
vermelho = 1

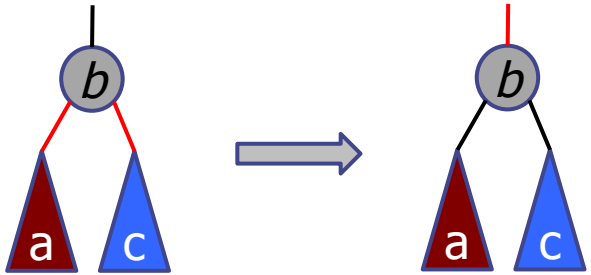



```c
typedef struct no no_t;

struct no {
    no_t *esq, *dir;
    int cor;
    int info;
};
```

Operações

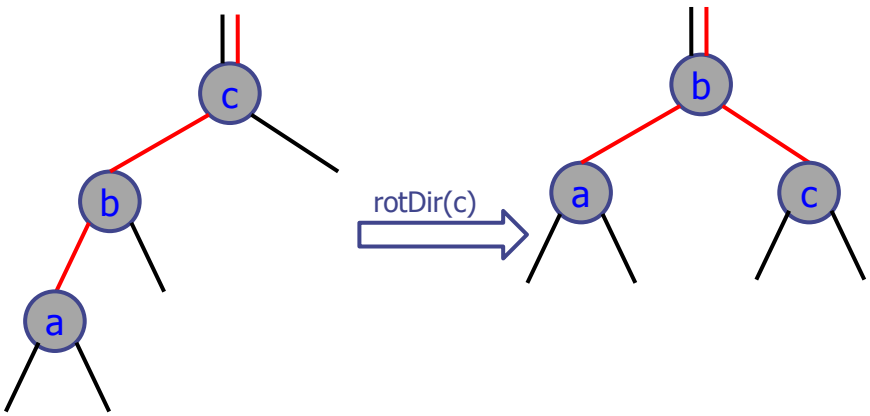
Inverte




```
void inverte(no_t *r){
    r->cor = !r->cor;
    if (r->esq)
        r->esq->cor = !r->esq->cor;
    if (r->dir)
        r->dir->cor = !r->dir->cor;
}
```

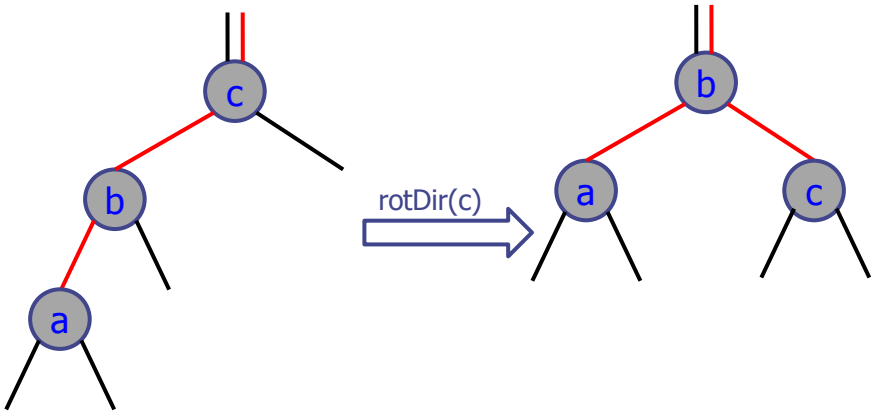
Operações

Rotação Direita



Operações

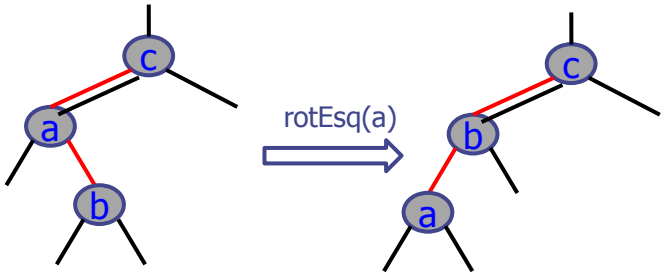
Rotação Direita




```
no_t *rotacaoDireita(no_t *c)
{
    no_t *b = c->esq;
    c->esq = b->dir;
    b->dir = c;
    b->cor = c->cor;
    c->cor = 1;
    return b;
}
```

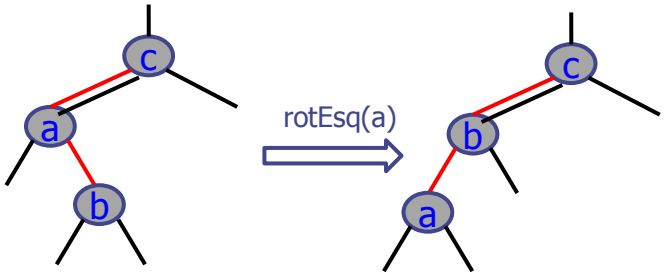
Operações

Rotação Esquerda



Operações

Rotação Esquerda




```c
no_t *rotacaoEsquerda(no_t *a)
{
    no_t *b = a->dir;
    a->dir = b->esq;
    b->esq = a;
    b->cor = a->cor;
    a->cor = 1;
    return b;
}
```

Inserção: resumo

- Segue algoritmo da ABB
- Sempre insere aresta vermelha
- Pode quebrar regras 1 e 2
- Operações locais corrigem problemas
  - Localmente
- Operação **inverte**
  - Propaga aresta vermelha para cima
  - Pode ser tratada **recursivamente** como inserção
  - Até alcançar um nó
    - Sem incidência vermelha; ou
    - Raiz da árvore

Inserção: resumo

```
no_t *insere_llrb(no_t *h, int data)
{
    if (h == NULL)
        return criarNo(data);
    if (data < h->info)
        h->esq = insere_llrb(h->esq, data);
    else if (data > h->info)
        h->dir = insere_llrb(h->dir, data);

    if (Vermelho(h->dir) && !Vermelho(h->esq))
        h = rotacaoEsquerda(h);
    if (Vermelho(h->esq) && Vermelho(h->esq->esq))
        h = rotacaoDireita(h);
    if (Vermelho(h->esq) && Vermelho(h->dir))
        inverte(h);

    return h;
}
```

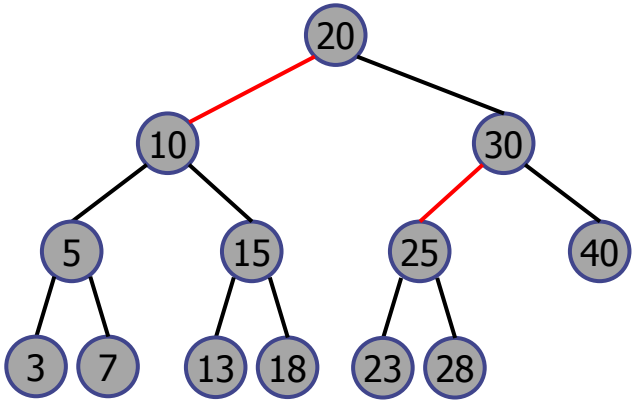
```
int Vermelho(no_t *h)
{
    if (h == NULL)
        return 0;
    return (h->cor == 1);
}
```

Remoção

- Remoção sempre de um nó folha com aresta vermelha
  - Aplicar conceitos de remoção de ABB no caso de nó não folha
    - Substituição de chave não altera formato/propriedades da árvore
  - Propagar aresta vermelha da raiz até a folha se necessário:
    - Se busca pela esquerda: moveRedLeft
    - Se busca pela direita: moveRedRight
  - Na volta da recursão: operações inverte, rotação direita e esquerda para corrigir violações

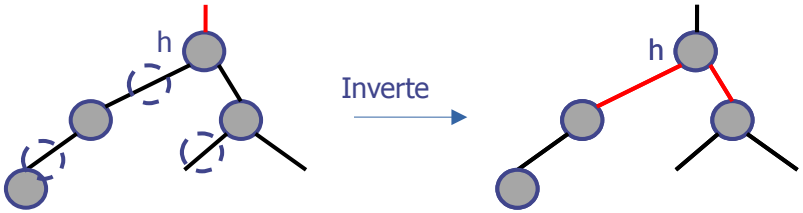
Exemplo 1

- Remover a chave 3 da árvore abaixo:

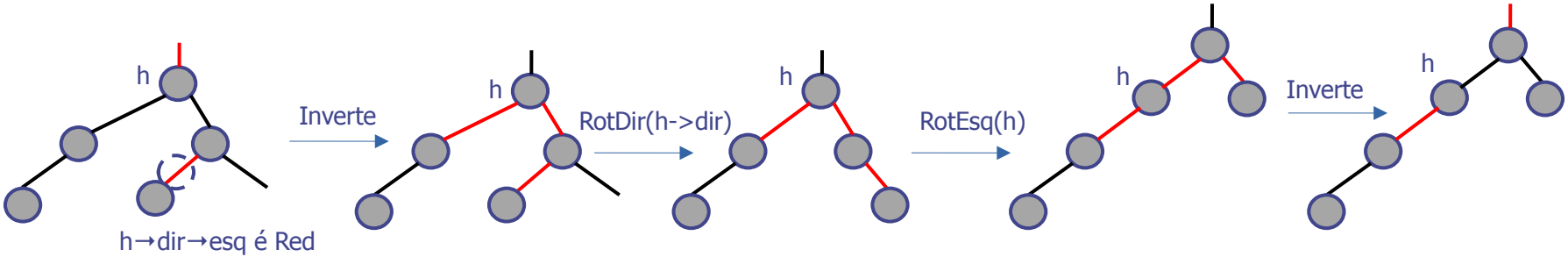


Exemplo 1

MoveRedLeft: 2 casos



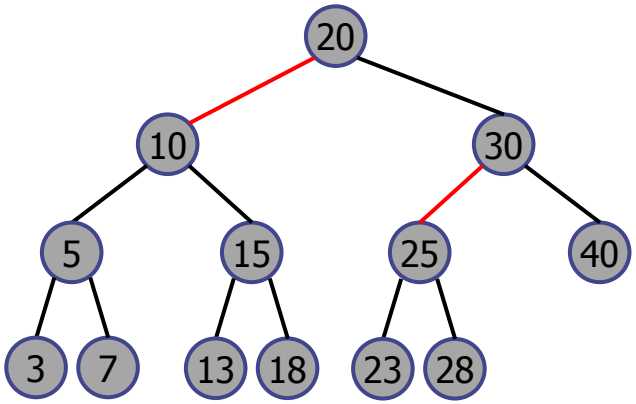
h→esq é Black
h→esq→esq é Black



h→dir→esq é Red

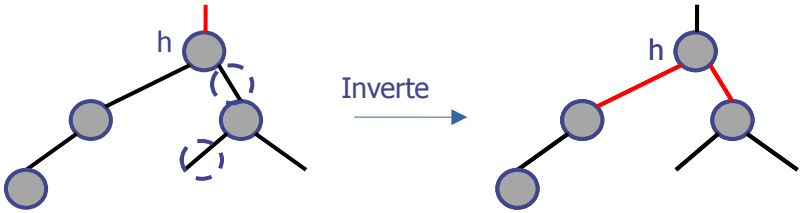
Exemplo 2

- Remover a chave 40 da árvore abaixo:



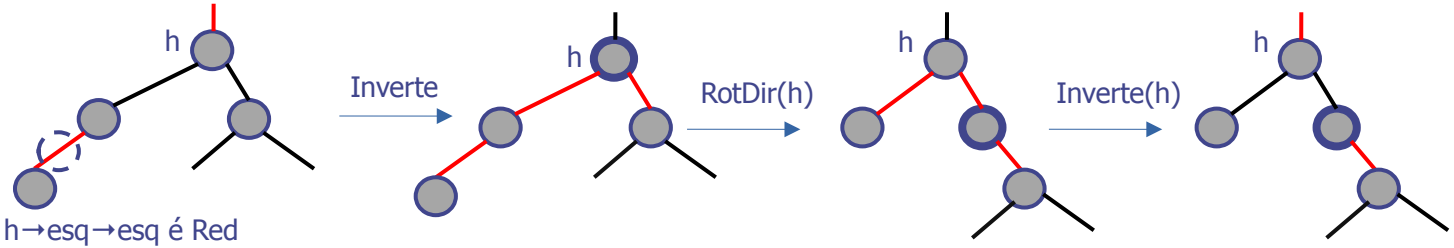
Exemplo 2

MoveRedRight: 2 casos



h→dir é Black

h→dir→esq é Black



h→esq→esq é Red

Referências

- Árvores Rubro-Negras
  - https://www.youtube.com/watch?v=YBacFFv4Mbo
  - http://www.cs.princeton.edu/~rs/talks/LLRB/RedBlack.pdf
  - https://www.cs.princeton.edu/~rs/talks/LLRB/LLRB.pdf

- Material baseado nos originais do prof. Marcelo G. Manzato