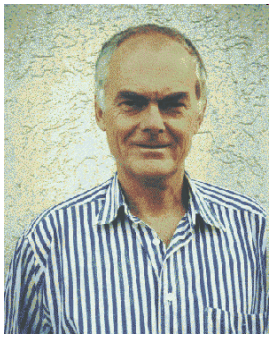
INF 1010
Estruturas de Dados Avançadas

Árvores Rubro Negras

Árvores rubro-negras ou vermelho e preto (red-black tree)



Rudolf Bayer (1939-...)

Rudolf Bayer (1972). "Symmetric binary B-Trees: Data structure and maintenance algorithms". *Acta Informatica* **1** (4): 290--306.

Árvores Rubronegras

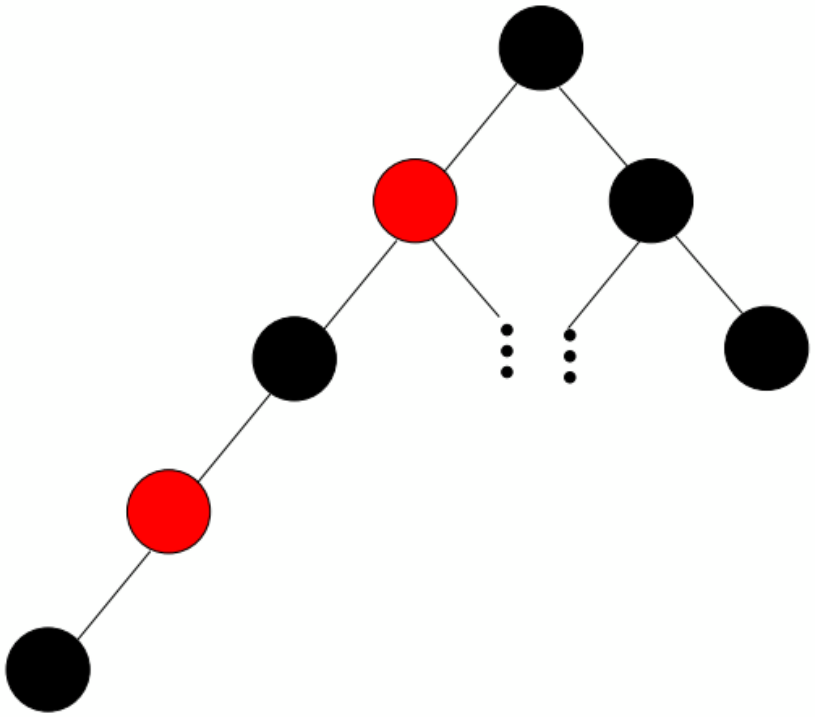
São árvores auto-ajustáveis, que procuram manter o balanceamento de forma automática mediante as seguintes restrições:

1. Todo nó é vermelho ou preto
2. A raiz é preta
3. As folhas são os nós NULL e são pretas
4. Nós vermelhos só tem filhos pretos
5. Todos os caminhos a partir da raiz da árvore até suas folhas passa pelo mesmo número de nós pretos

Árvores Rubronegras

Estas restrições garantem que o maior caminho entre a raiz e as folhas é no máximo o dobro do menor caminho.

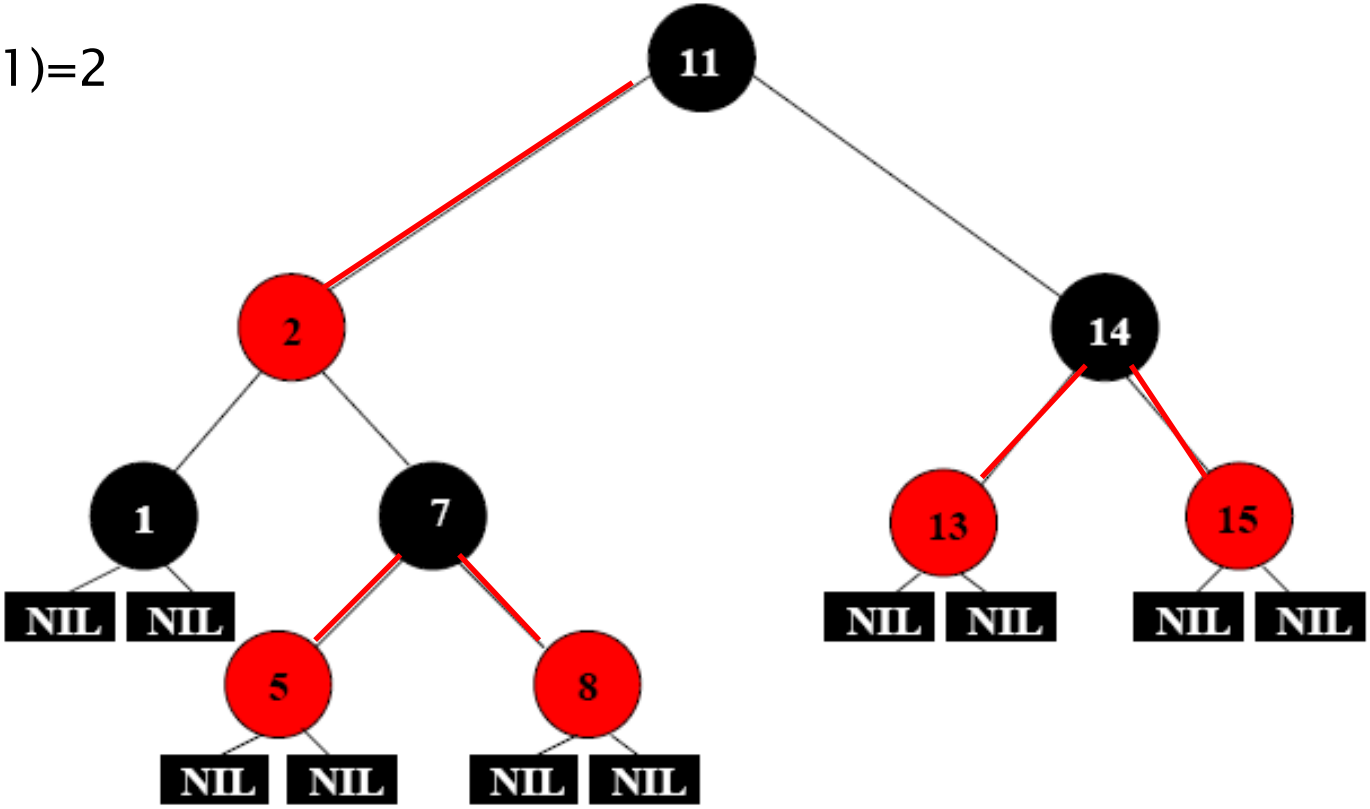
Elas garantem um balancemanto razoável



Formas de representação

h(11)=4

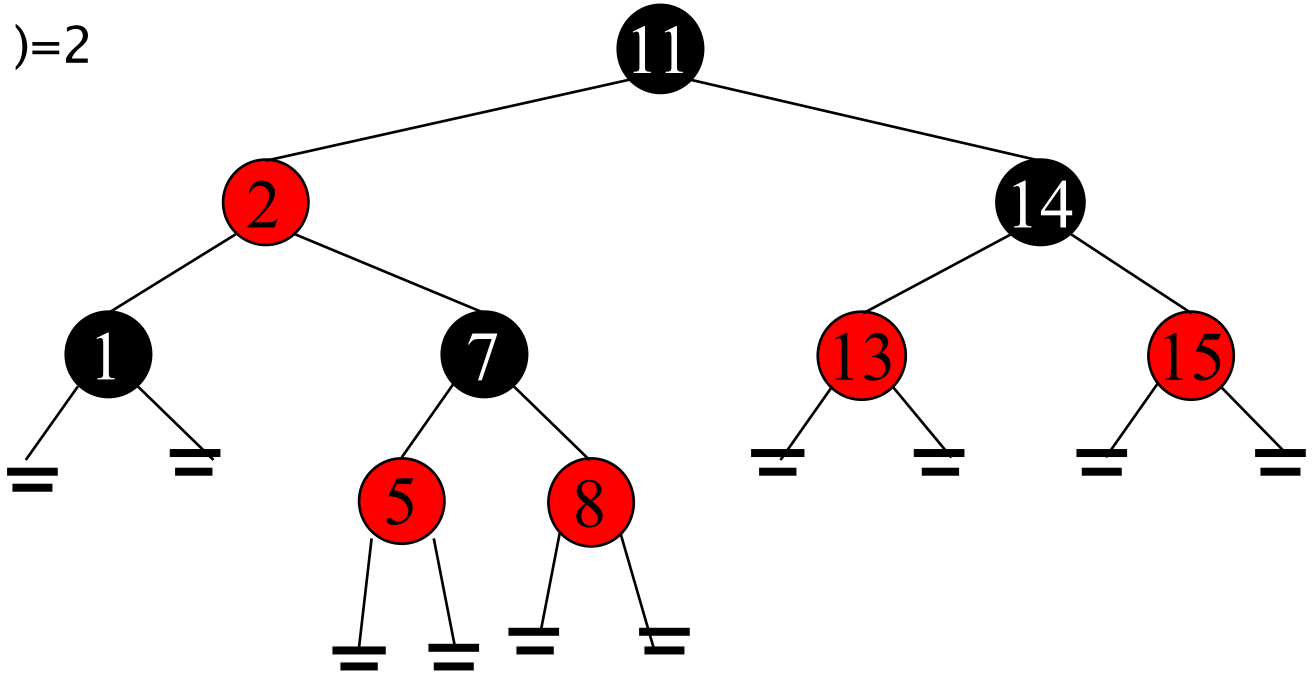
bh(11)=2



Formas de representação

h(11)=4

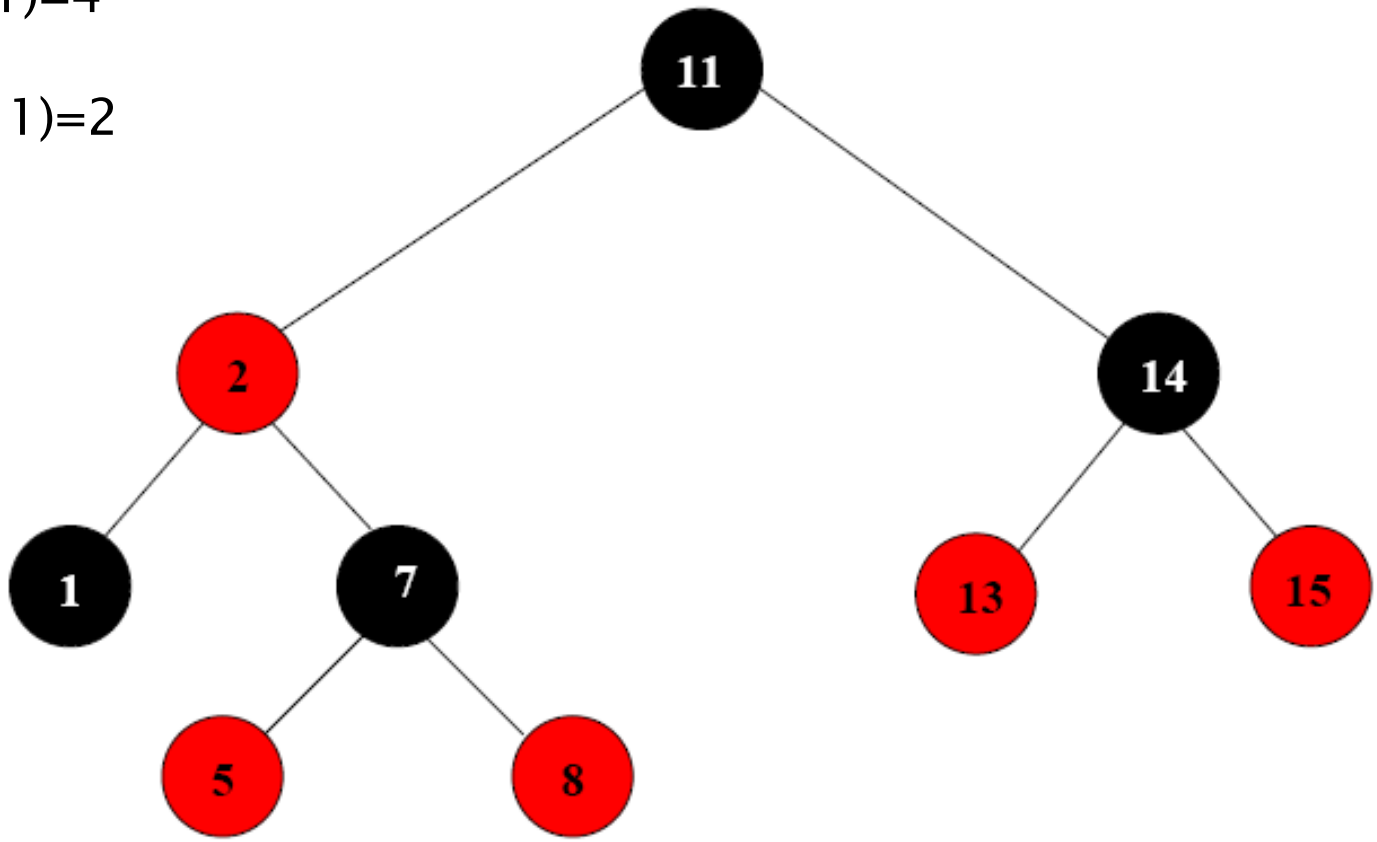
bh(11)=2



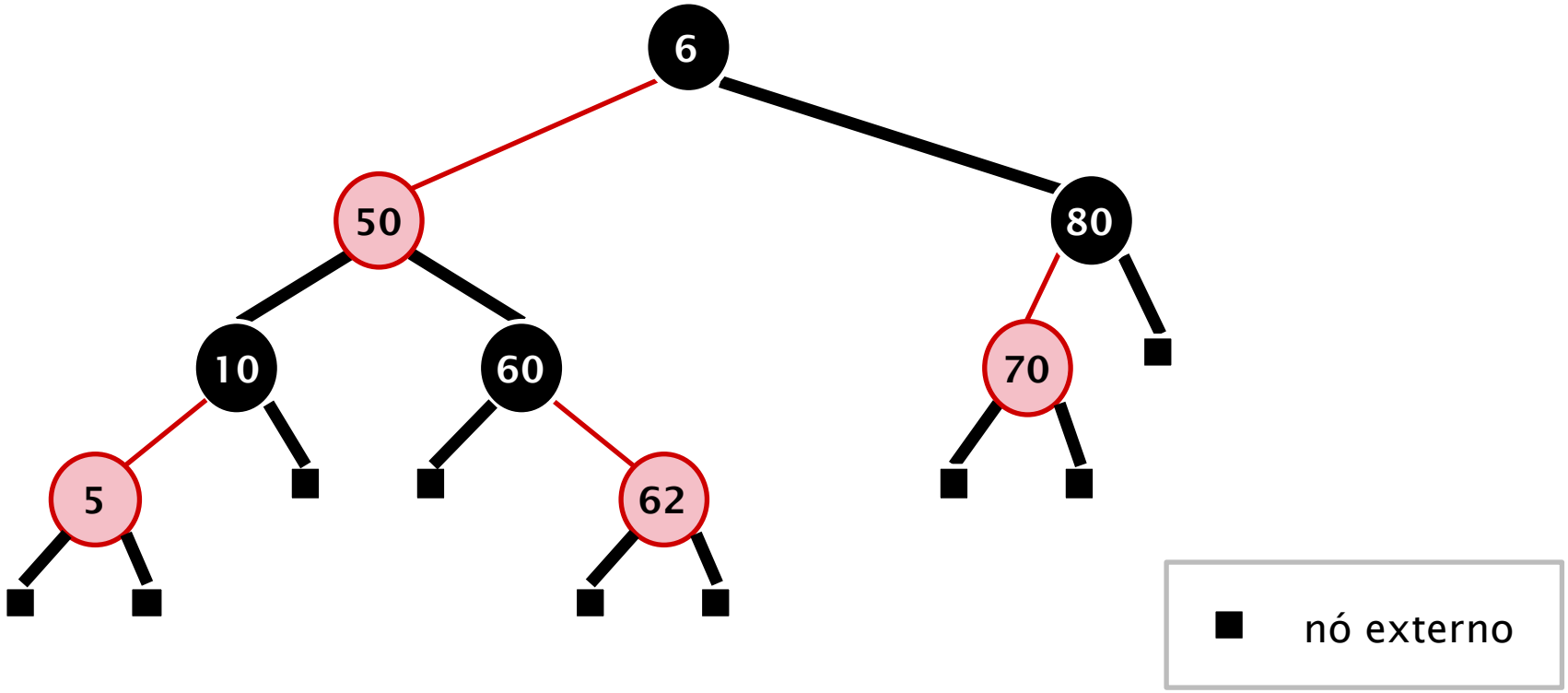
Formas de representação

h(11)=4

bh(11)=2



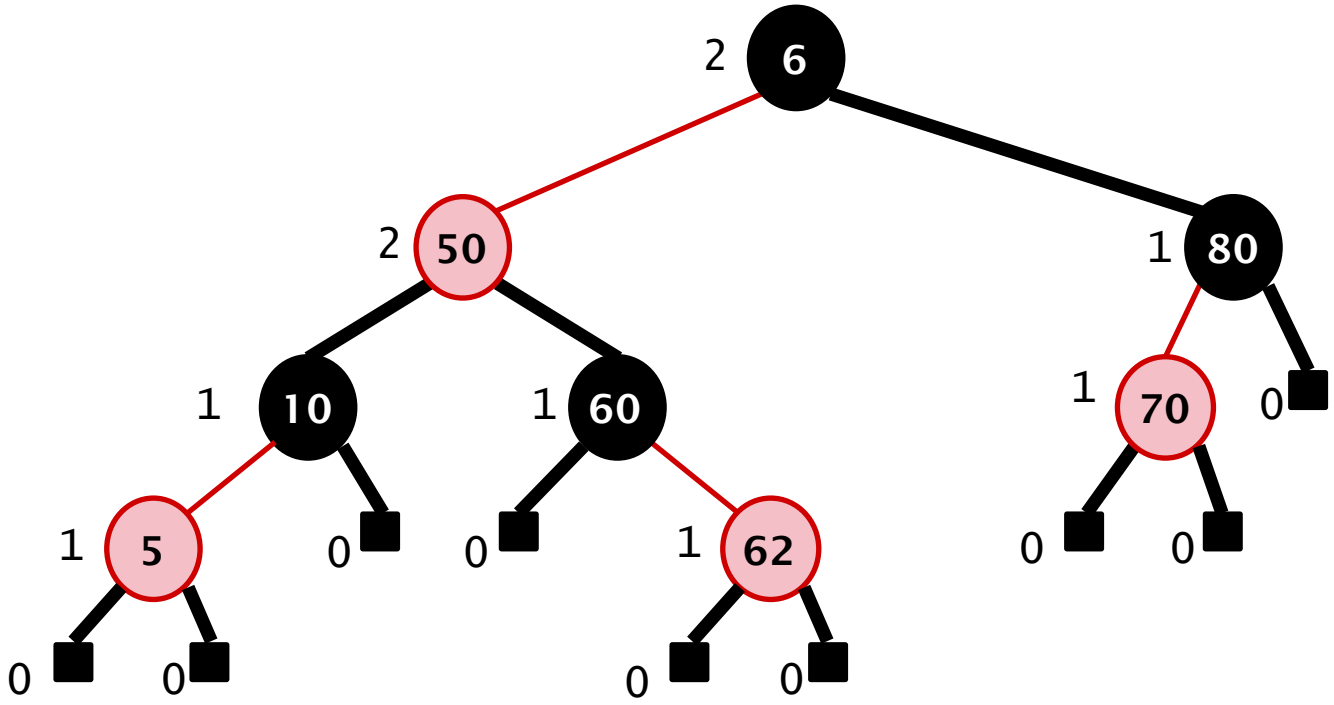
Árvore Red-Black



Árvore Red-Black - definição

Altura negra (rank) de um nó:

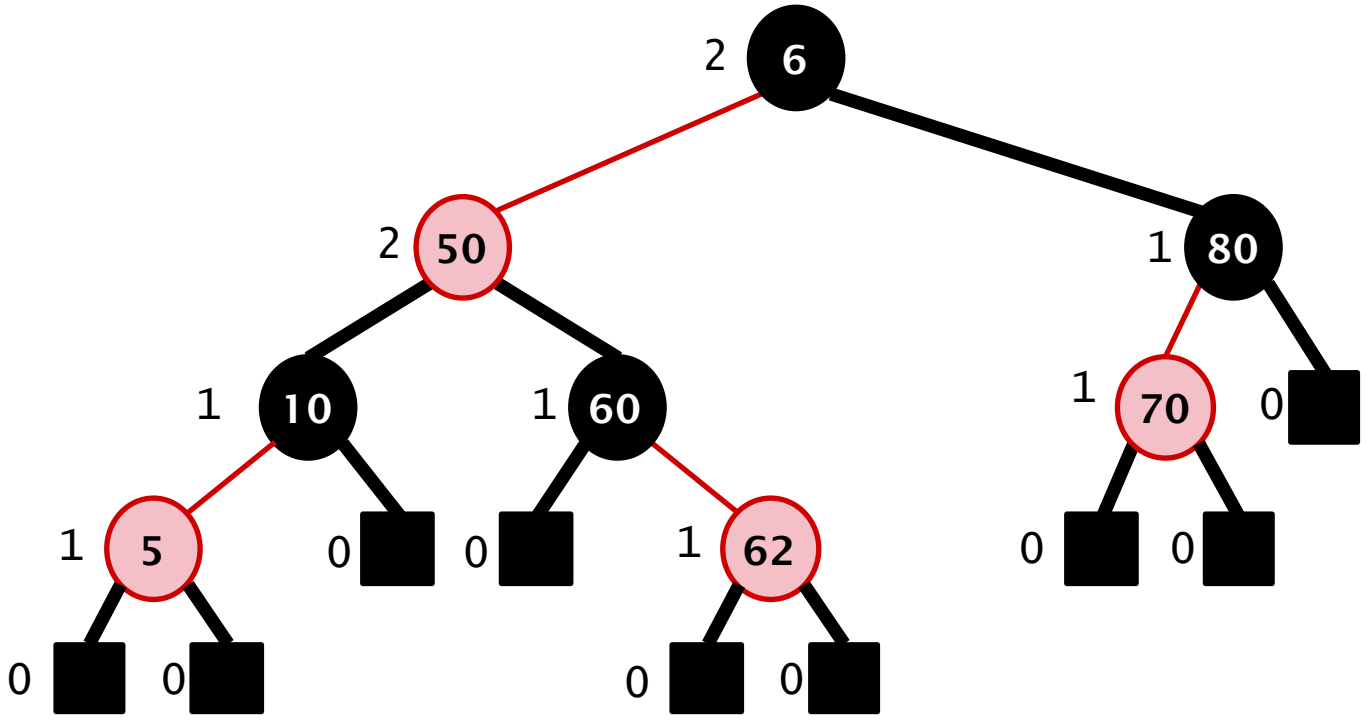
número de **arestas pretas** de qualquer caminho desde o nó até um nó externo



Árvore Red-Black - conceitos

Sejam P e Q dois caminhos da raiz até nós externos:

comprimento(P) ≤ 2×comprimento(Q)



Árvore Red-Black - conceitos

r = *rank* da raiz (2)

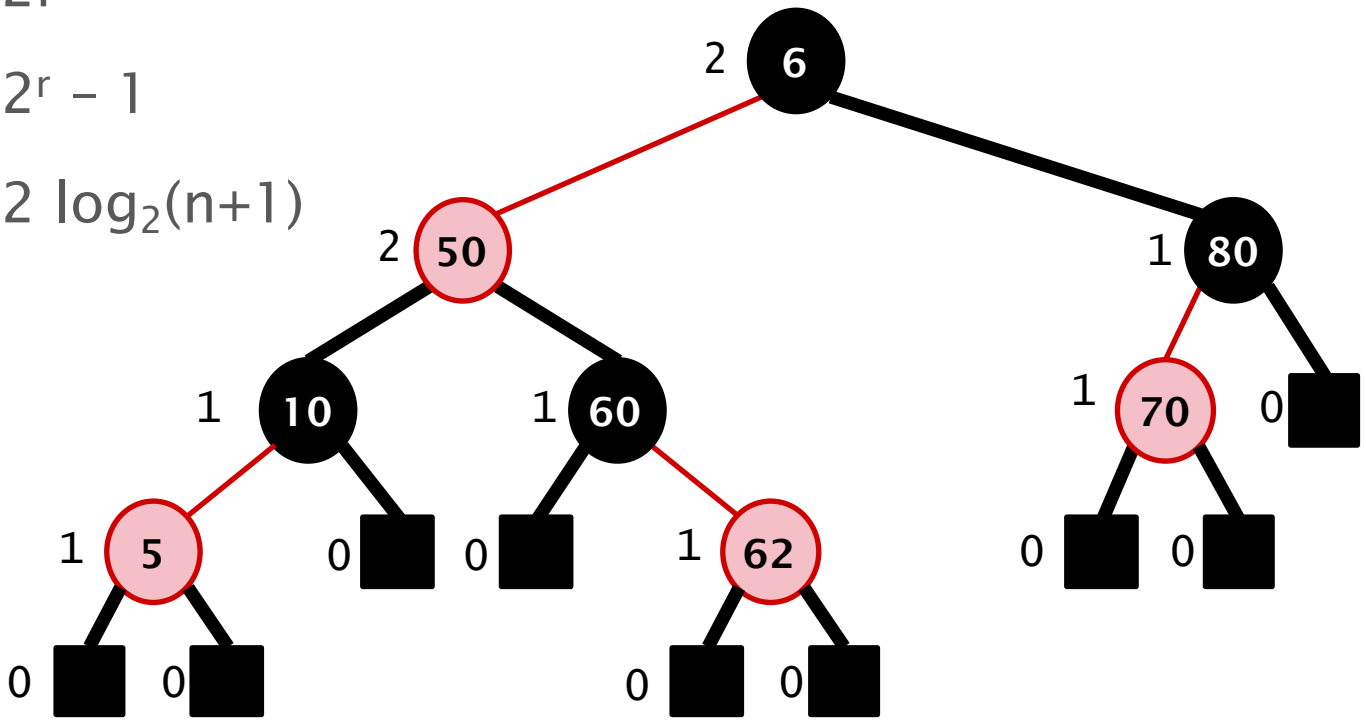
h = altura da árvore sem nós externos (3 = 2r-1)

n = número de nós (8)

$h \leq 2r$

$n \geq 2^r - 1$

$h \leq 2 \log_2(n+1)$



altura de árvore rubro-negra com n nós

h <= 2*bh

até profundidade bh, árvore cheia:

n >= 2^{bh}-1 (n = número nós na árvore)

log (n+1) >= bh => h <= 2*log(n+1)

Árvore Red-Black - busca

igual à busca numa árvore binária de busca

Árvore Red-Black - inserção

árvore vazia → novo nó preto

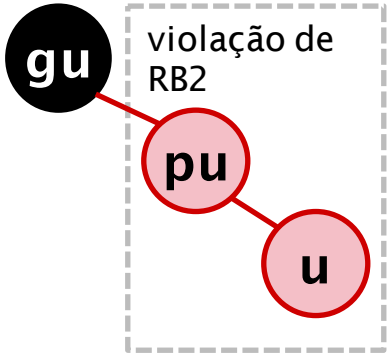
árvore não vazia → novo nó vermelho

se houver violação de RB2, a árvore está desequilibrada

u e pu vermelhos → pu não pode ser raiz (RB1)

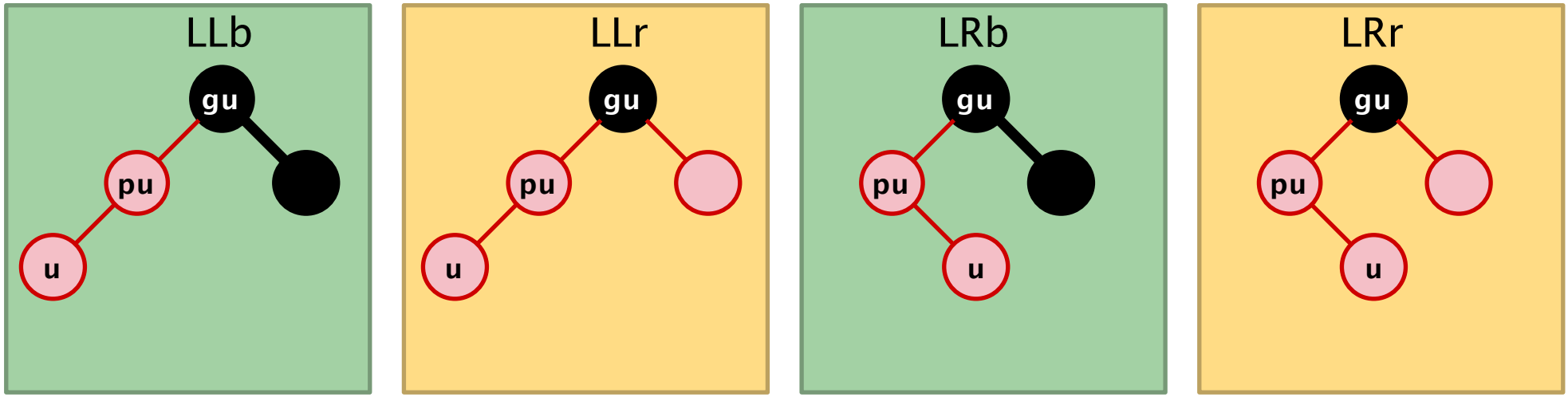
gu existe e é preto (RB2)

(u = novo nó; pu = progenitor de u;
gu = avô/avó de u)



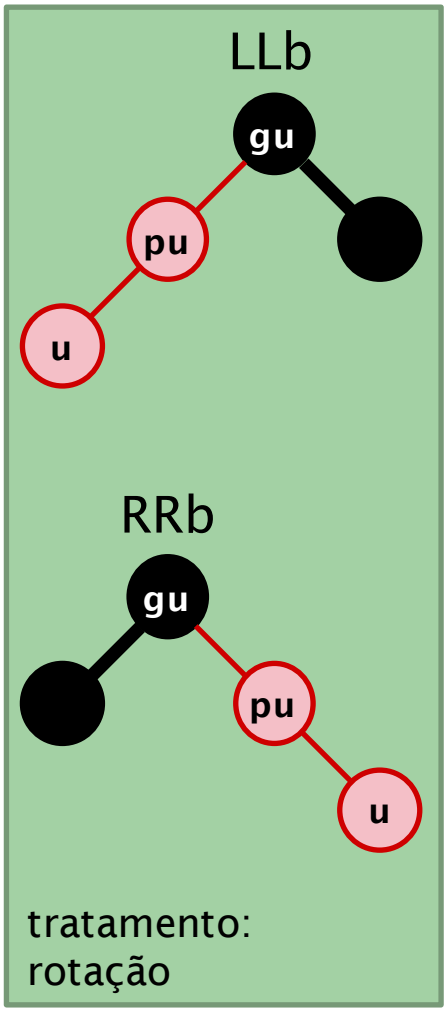
Tipos de desbalanceamento

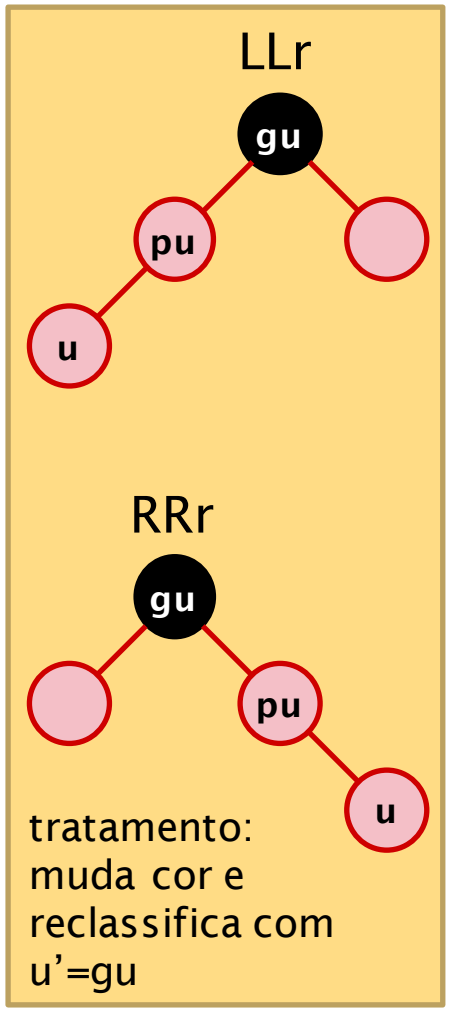
notação XYc: pu é filho X {L,R} de gu
u é filho Y {L,R} de pu
outro filho de gu é de cor c {b,r}

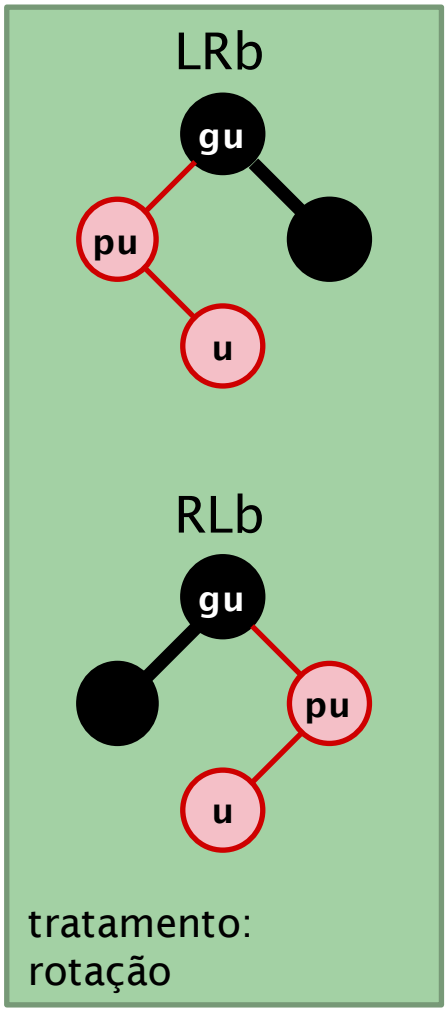


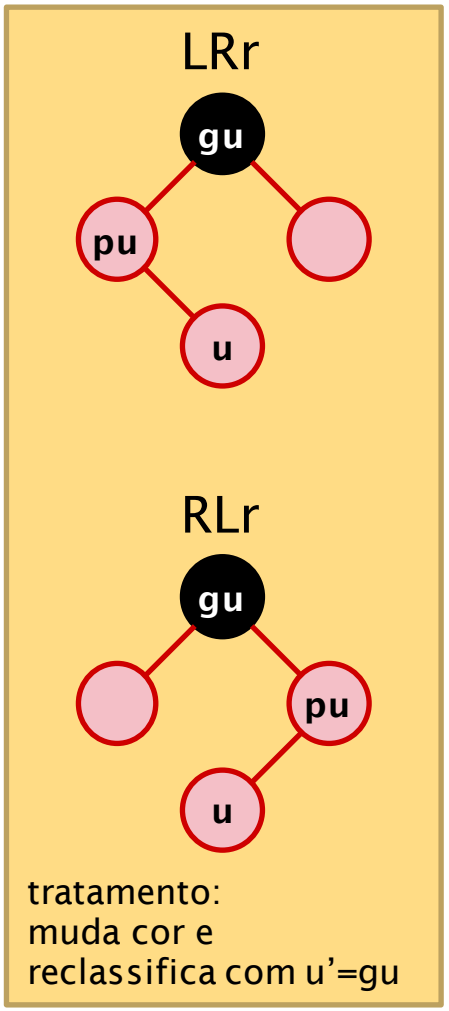
Tipos de desbalanceamento

notação XYc: pu é filho X {L,R} de gu
u é filho Y {L,R} de pu
outro filho de gu é de cor c {b,r}

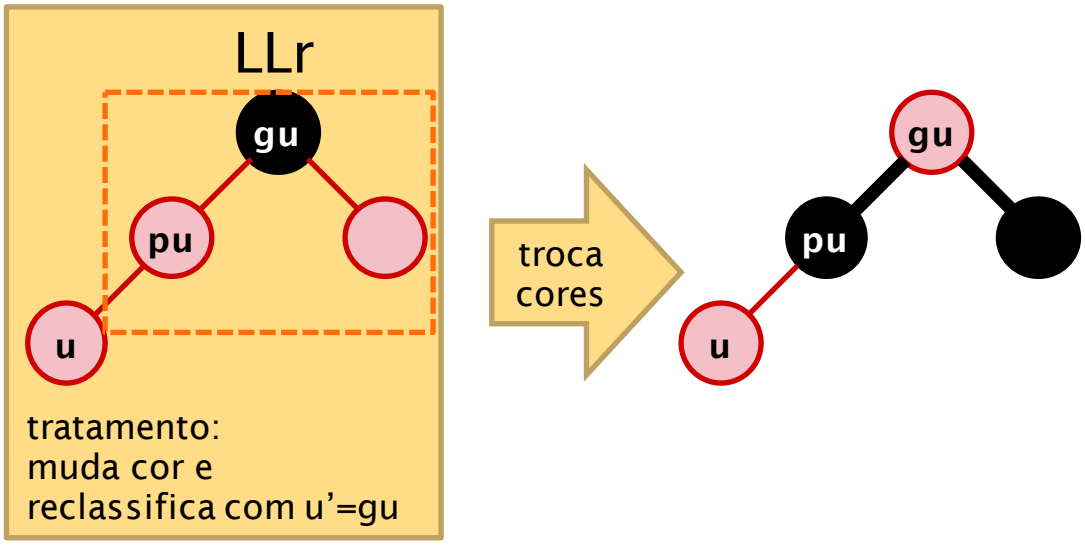




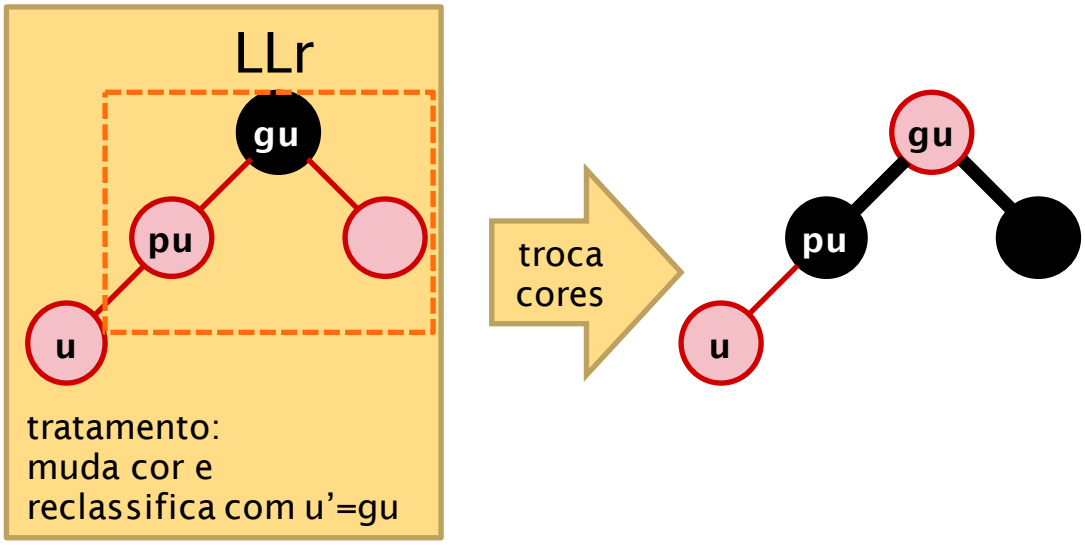




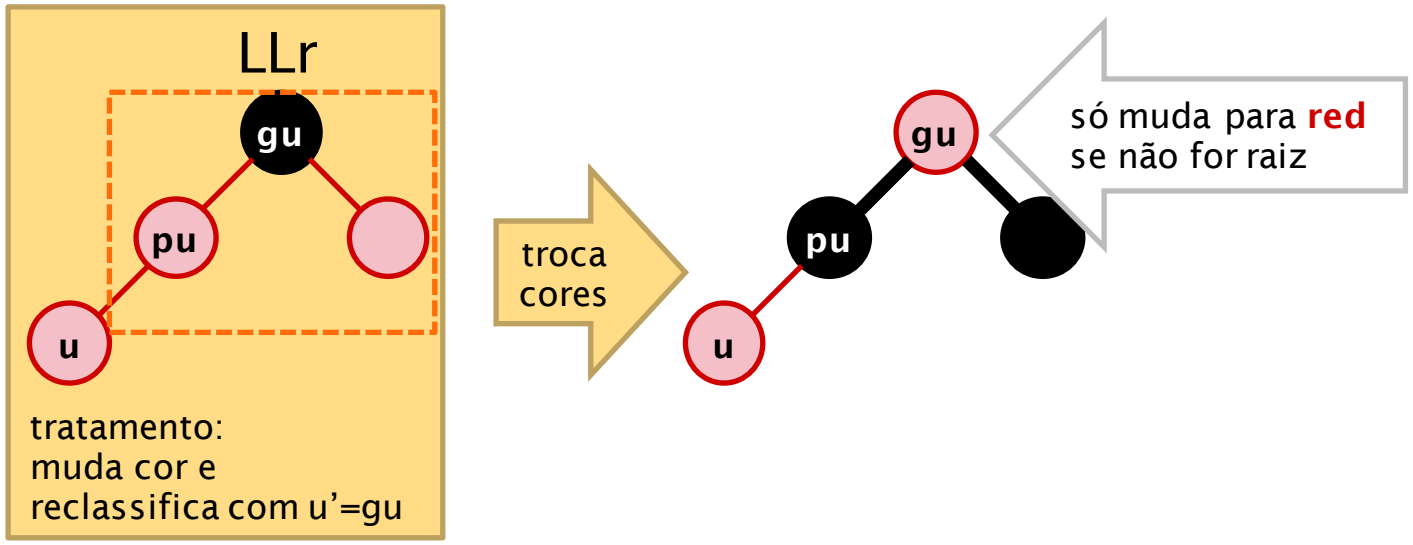
Tratamento por mudança de cor



Tratamento por mudança de cor

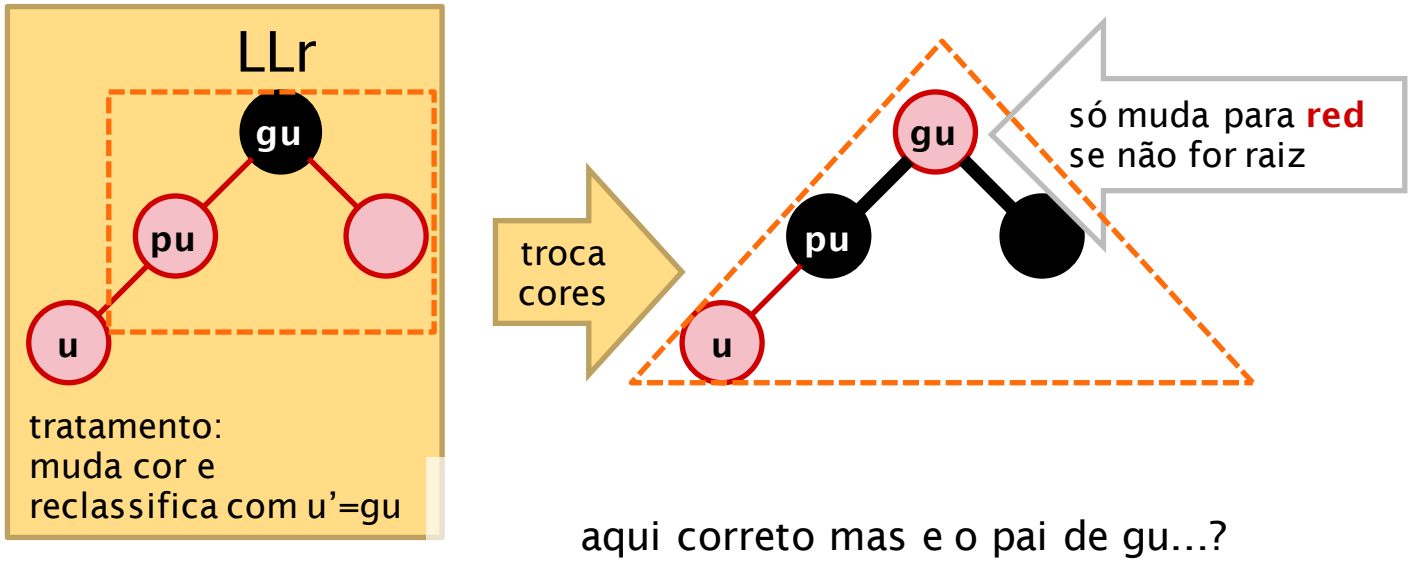


Tratamento por mudança de cor



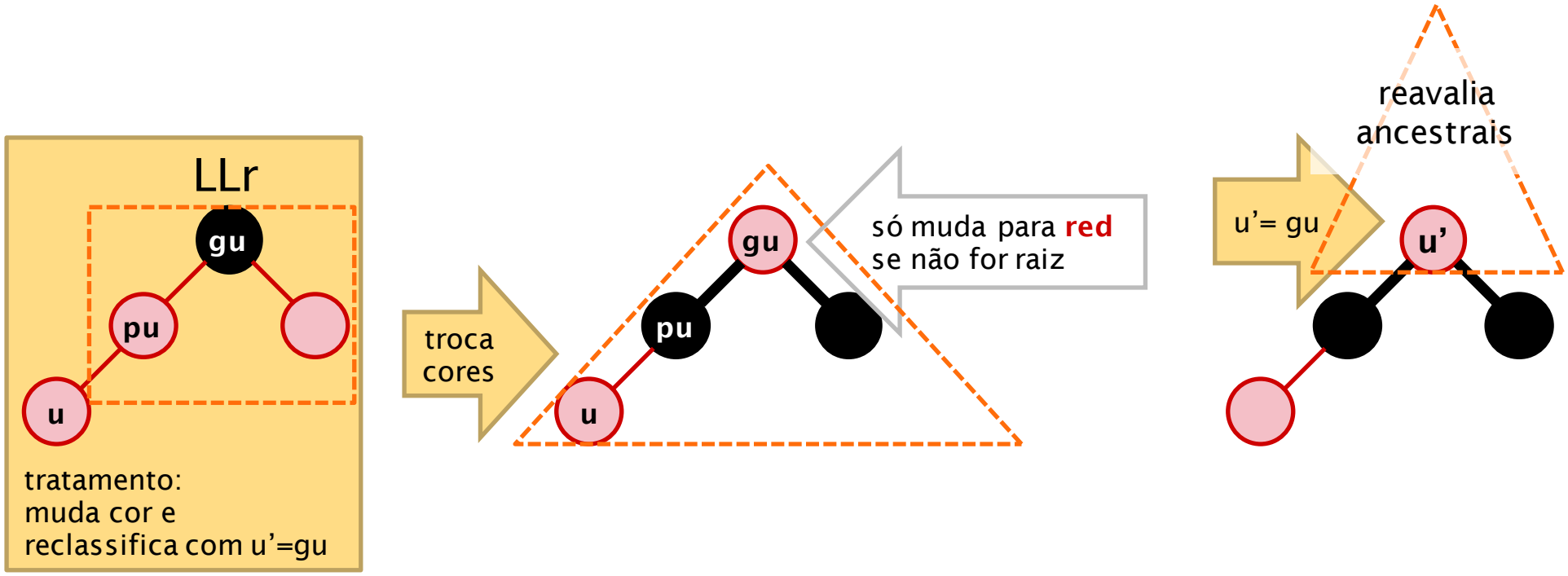
se raiz pode acrescentar um nó negro pois vai ser acrescentado em todos os caminhos!

Tratamento por mudança de cor

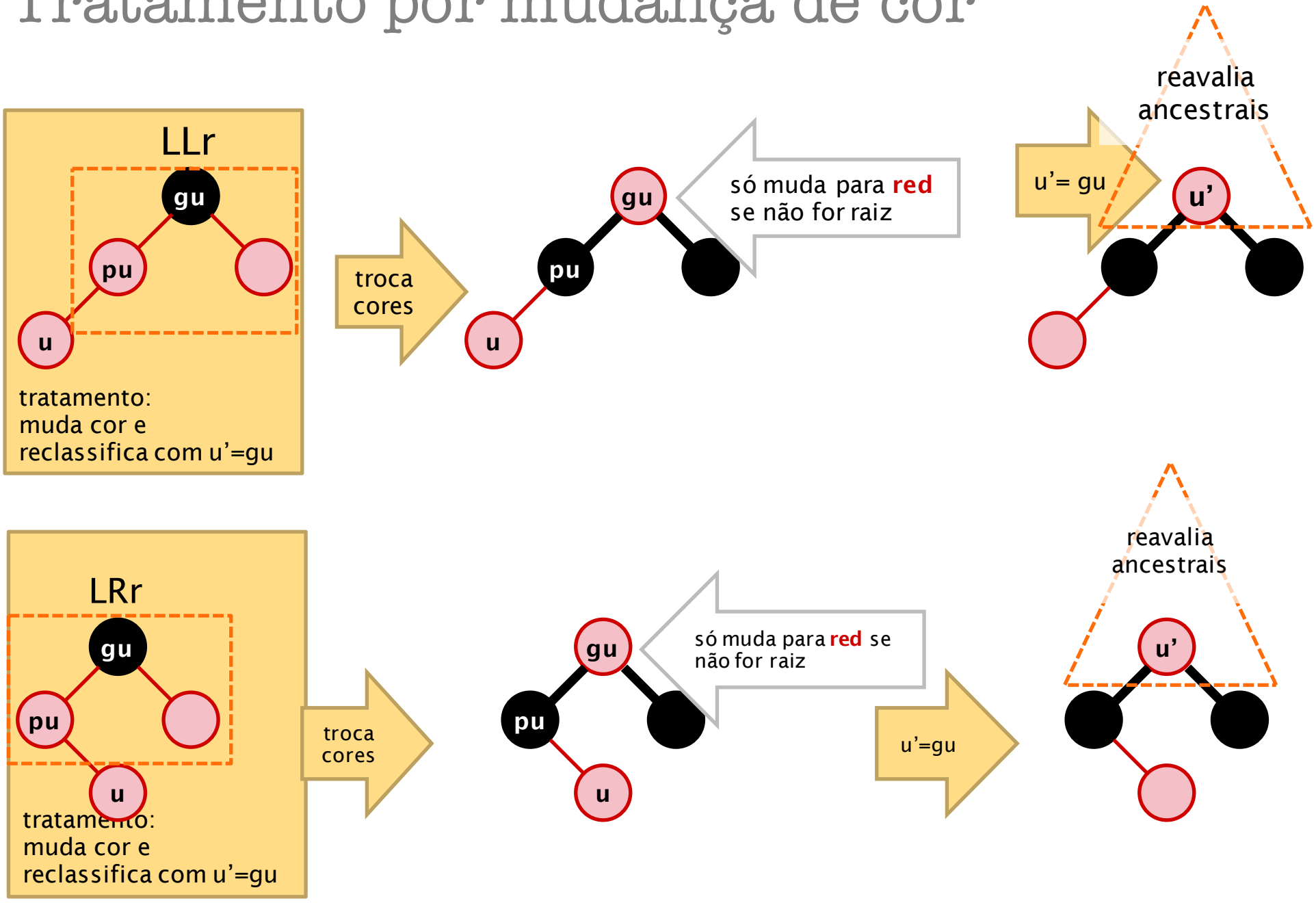


aqui correto mas e o pai de gu...?

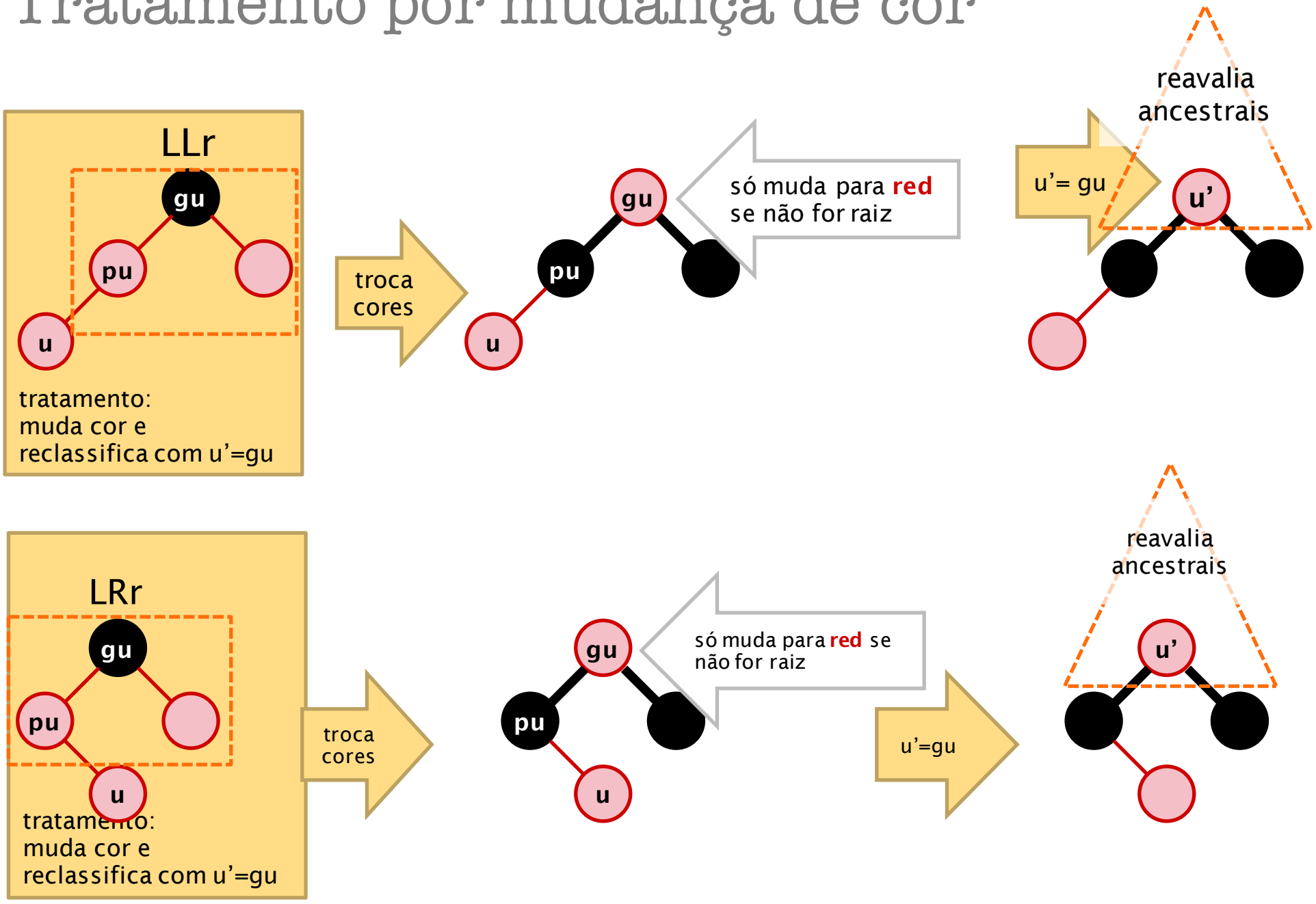
Tratamento por mudança de cor



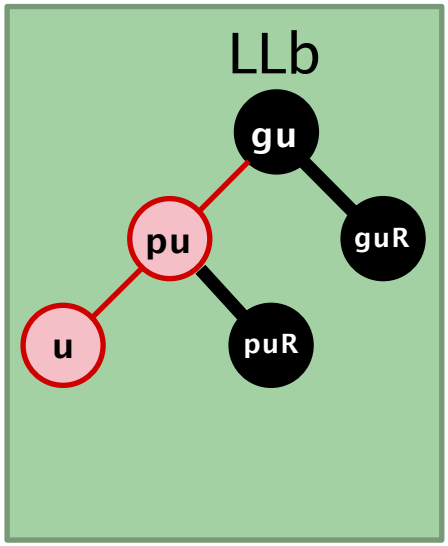
Tratamento por mudança de cor



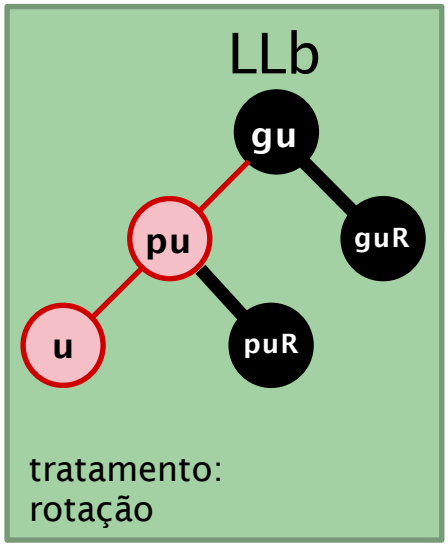
Tratamento por mudança de cor



Tratamento por rotação

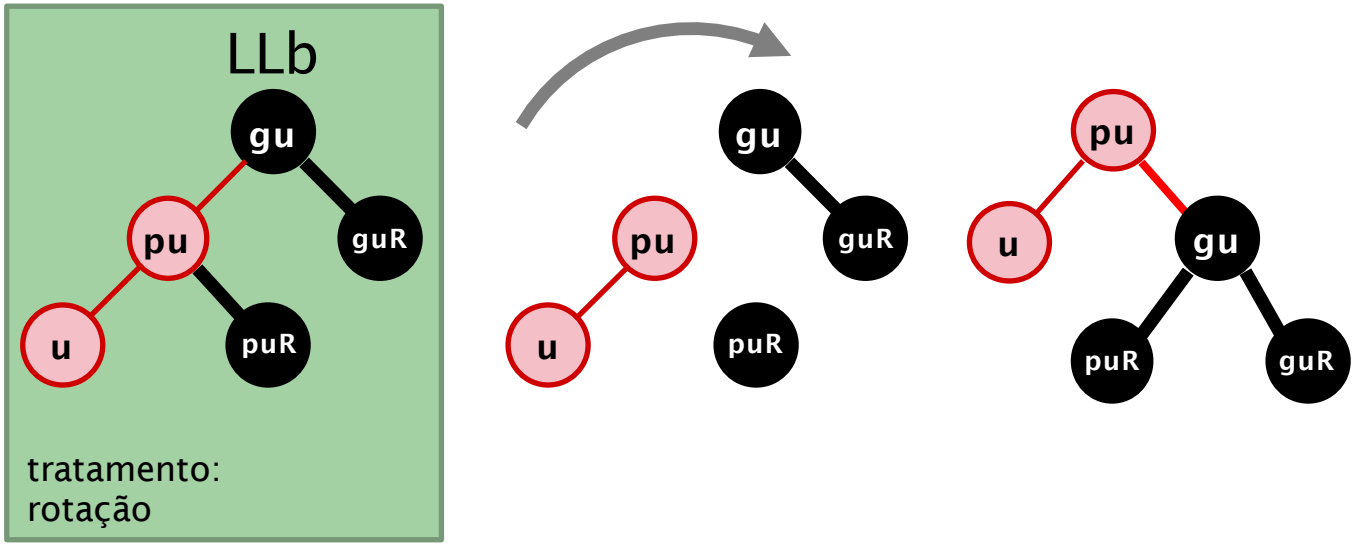


Tratamento por rotação

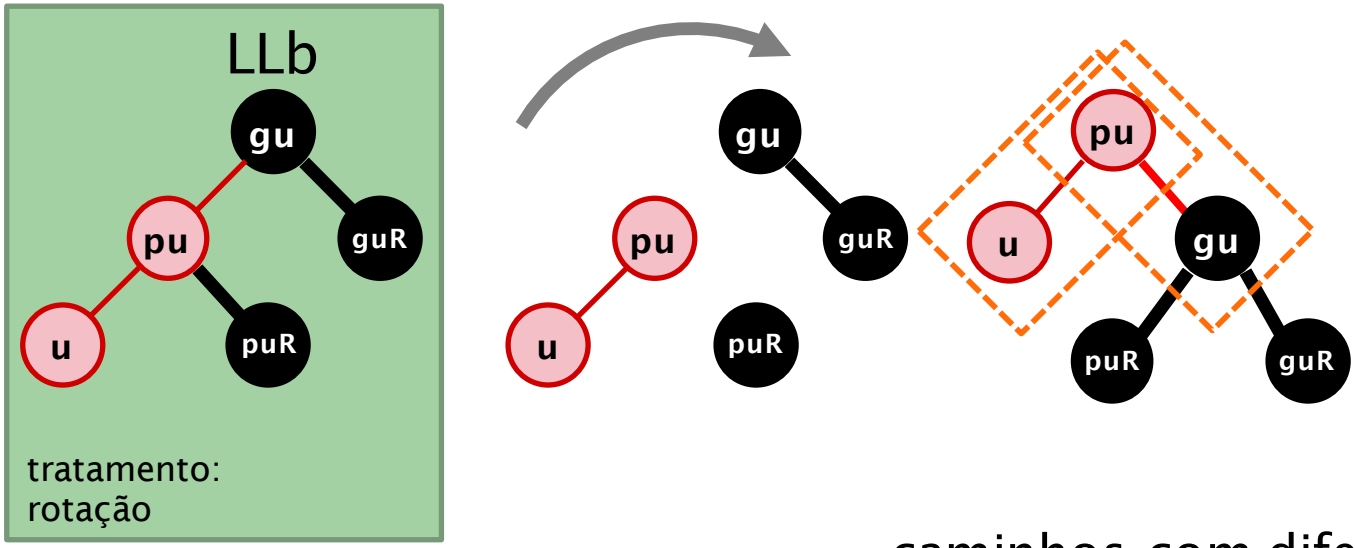


aqui não tem como mudar as cores sem alterar a contagem de nós negros em caminhos!!!

Tratamento por rotação

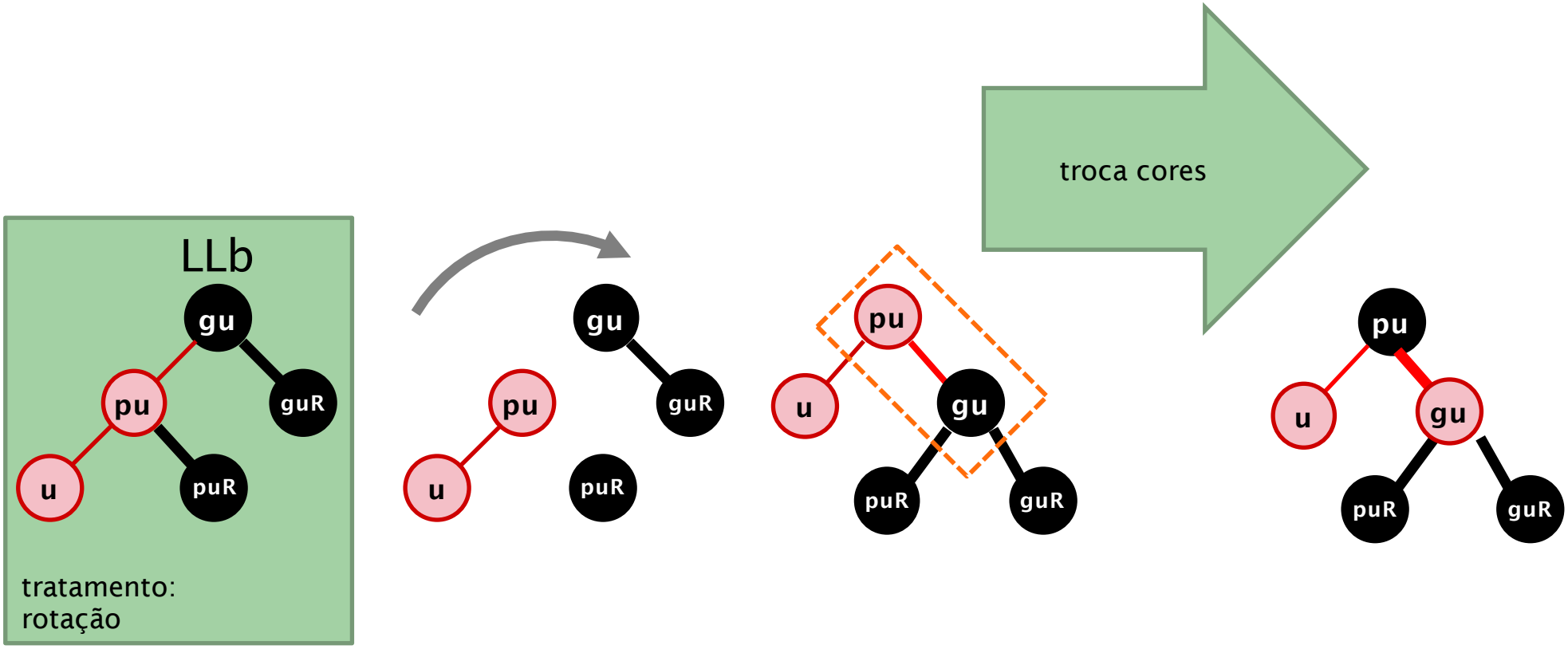


Tratamento por rotação

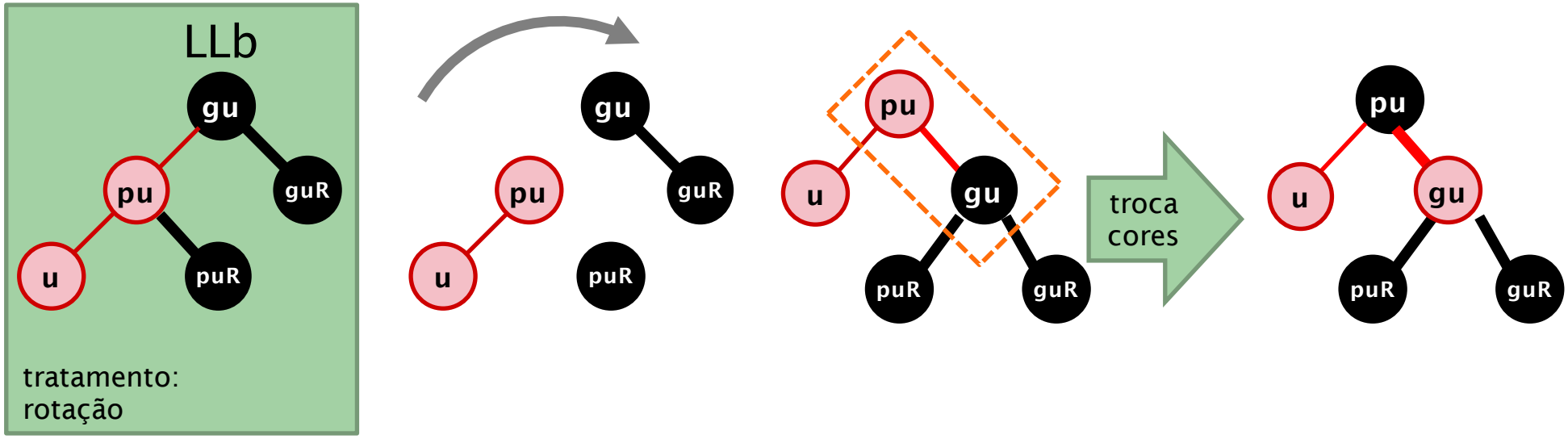


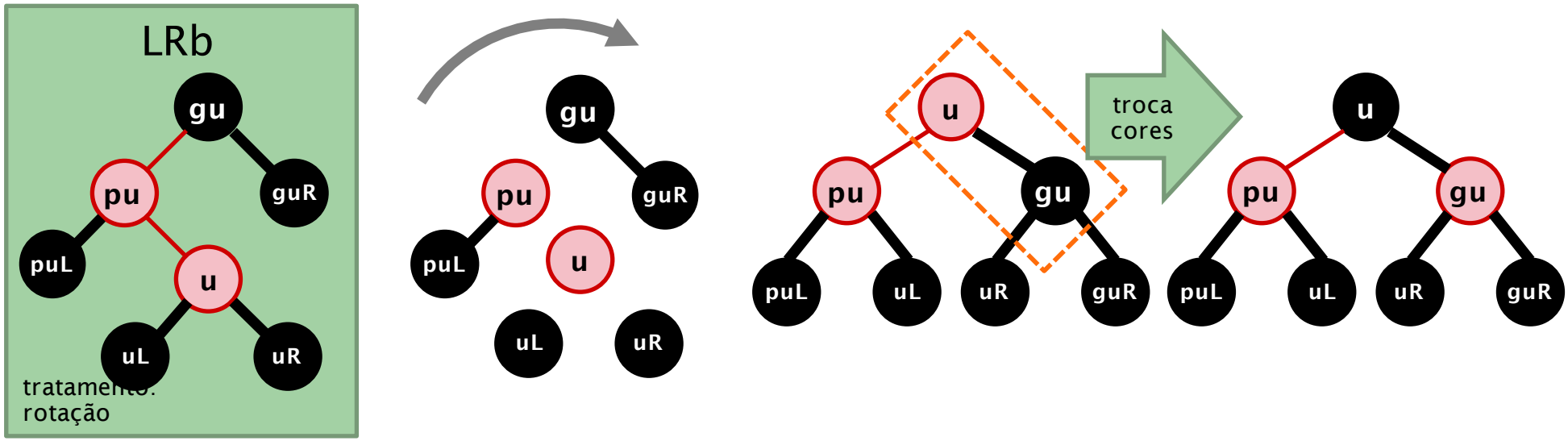
caminhos com diferentes números de nós negros até a raiz!

Tratamento por rotação

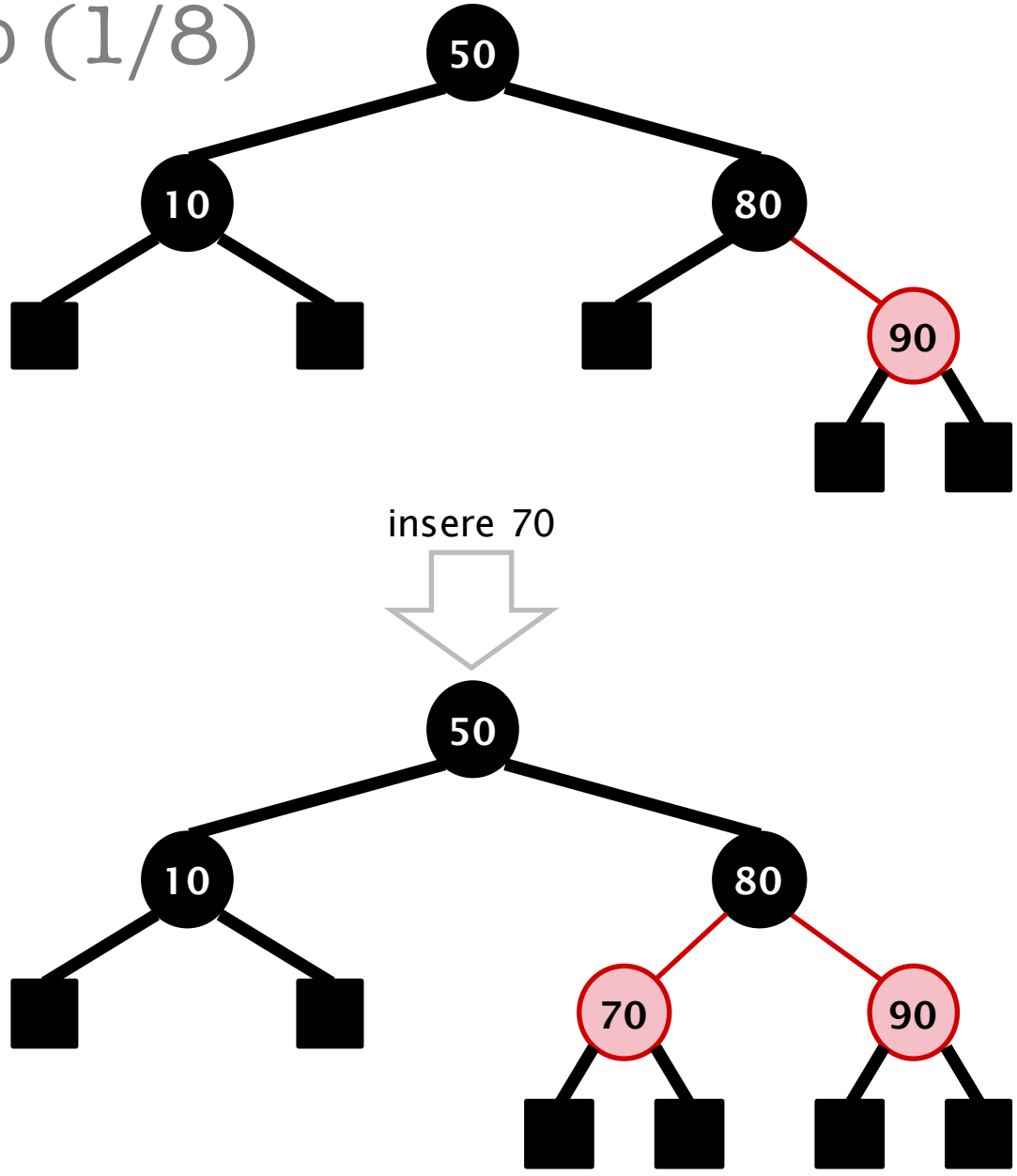


Tratamento por rotação

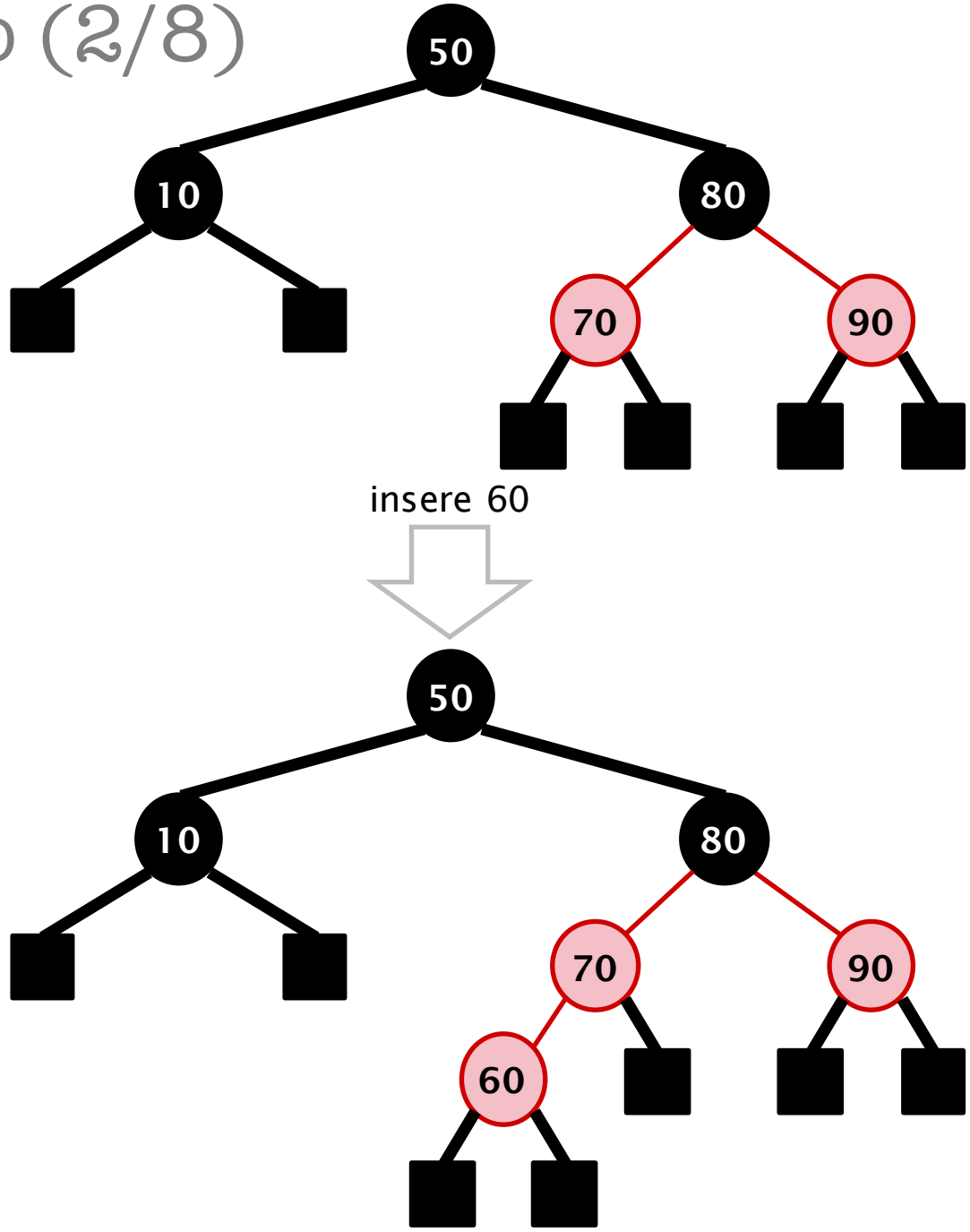




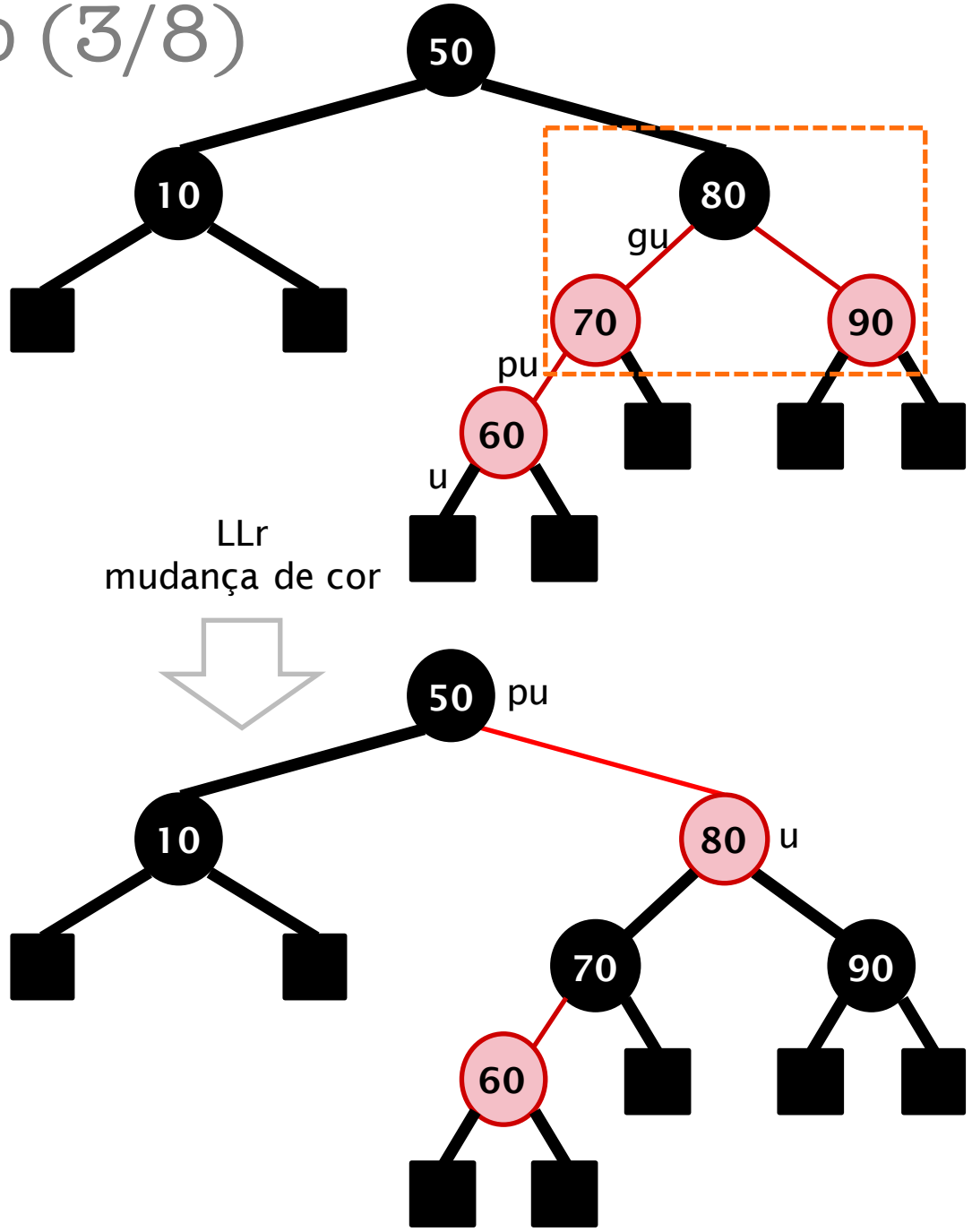
Exemplo (1/8)



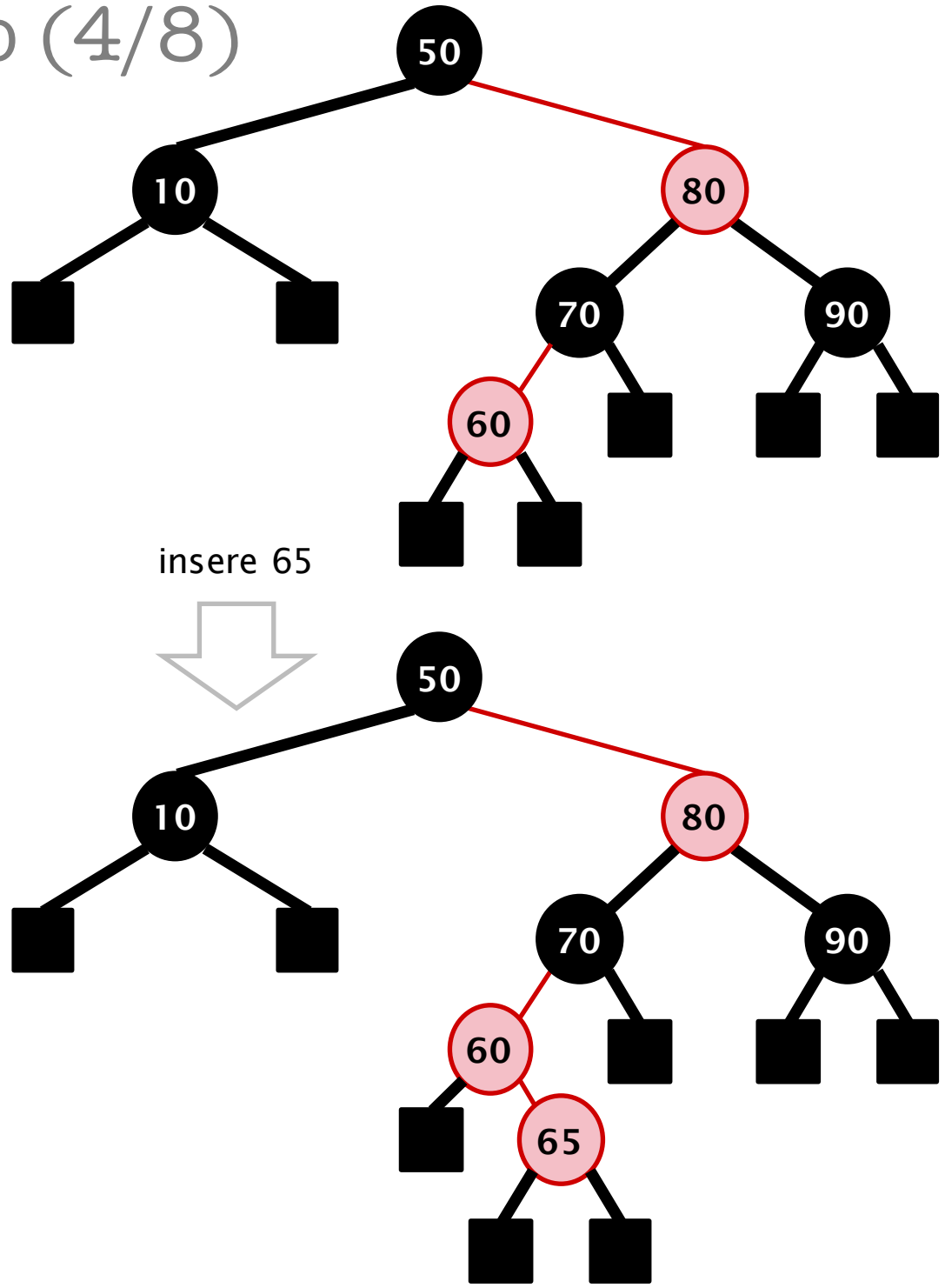
Exemplo (2/8)



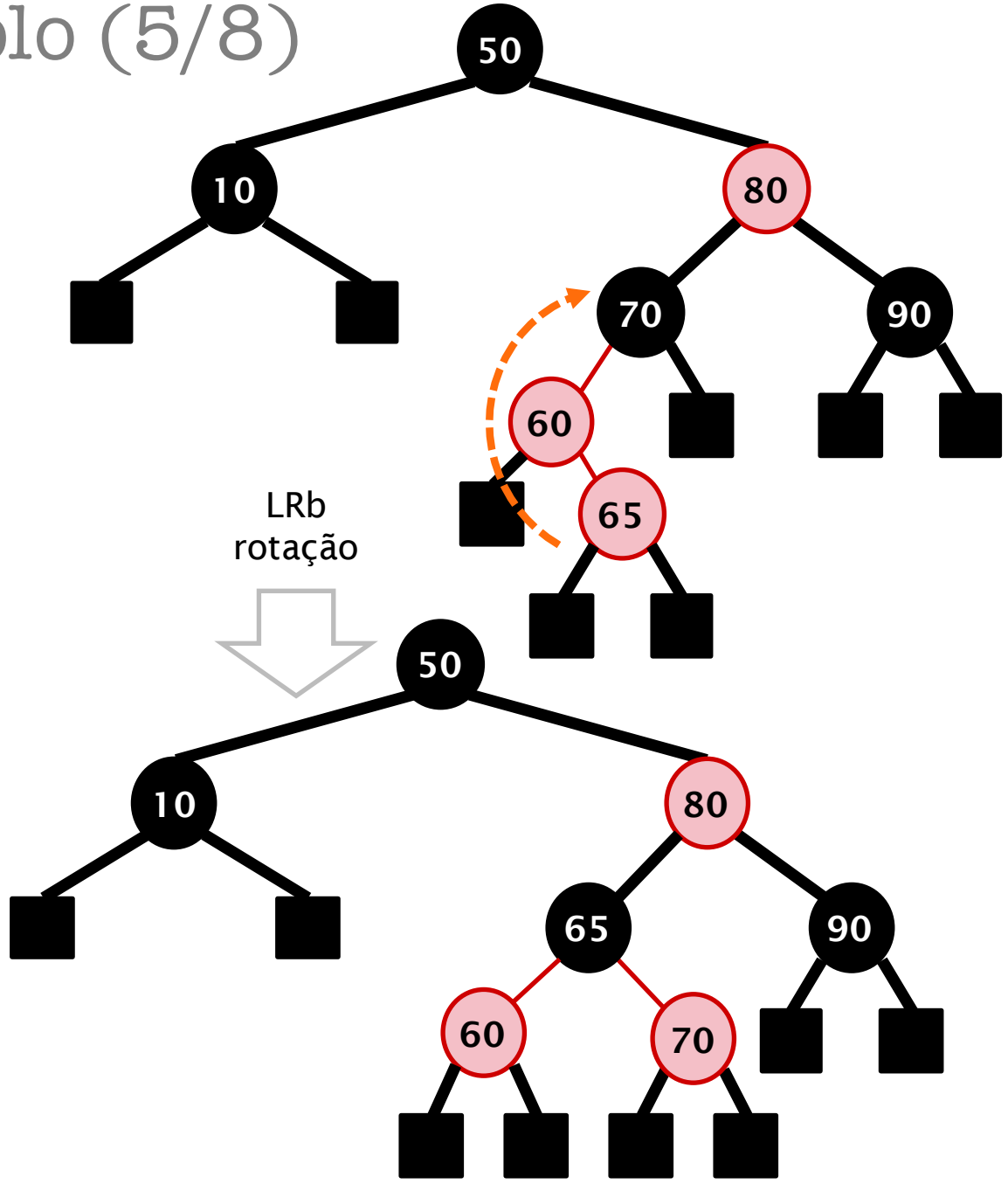
Exemplo (3/8)



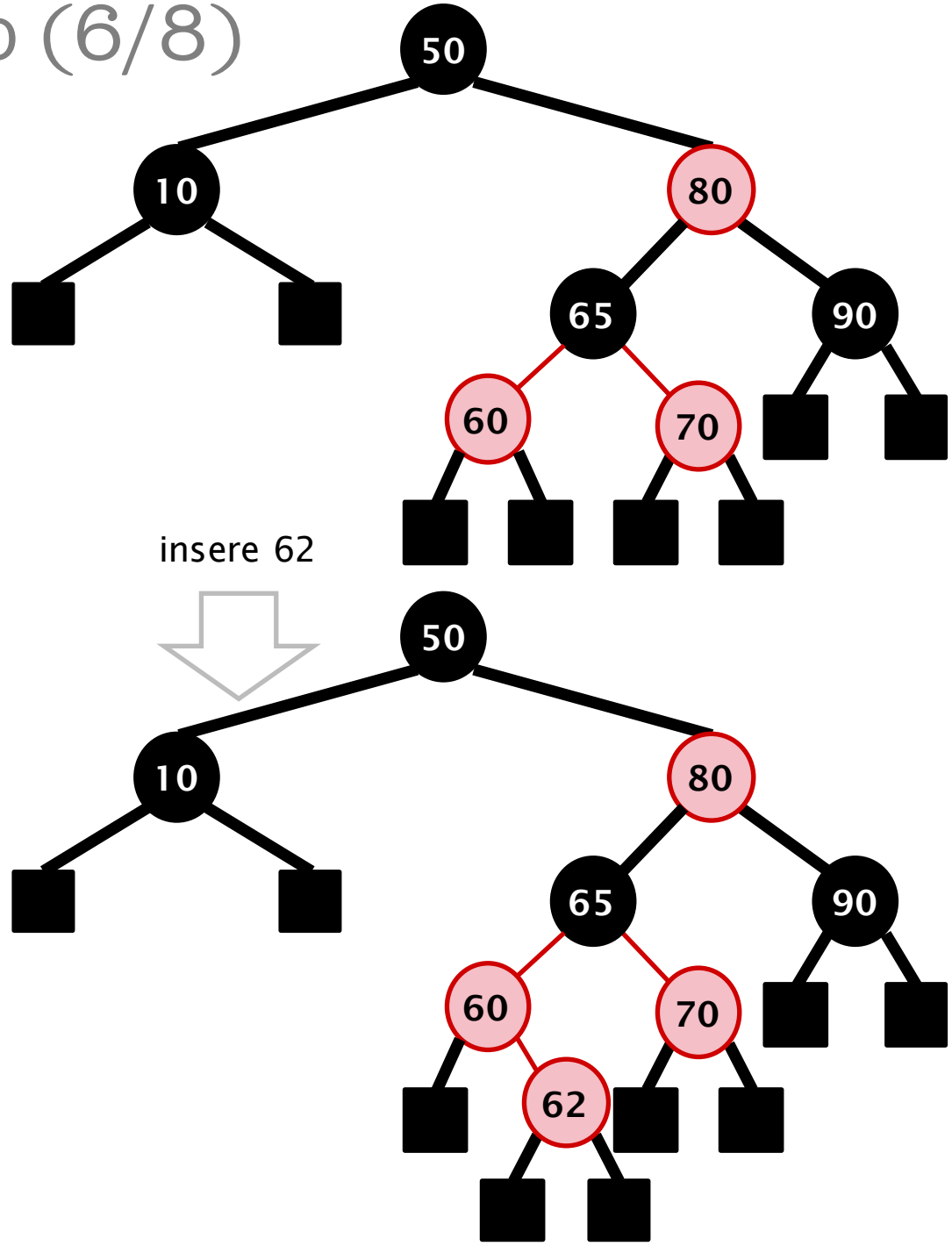
Exemplo (4/8)



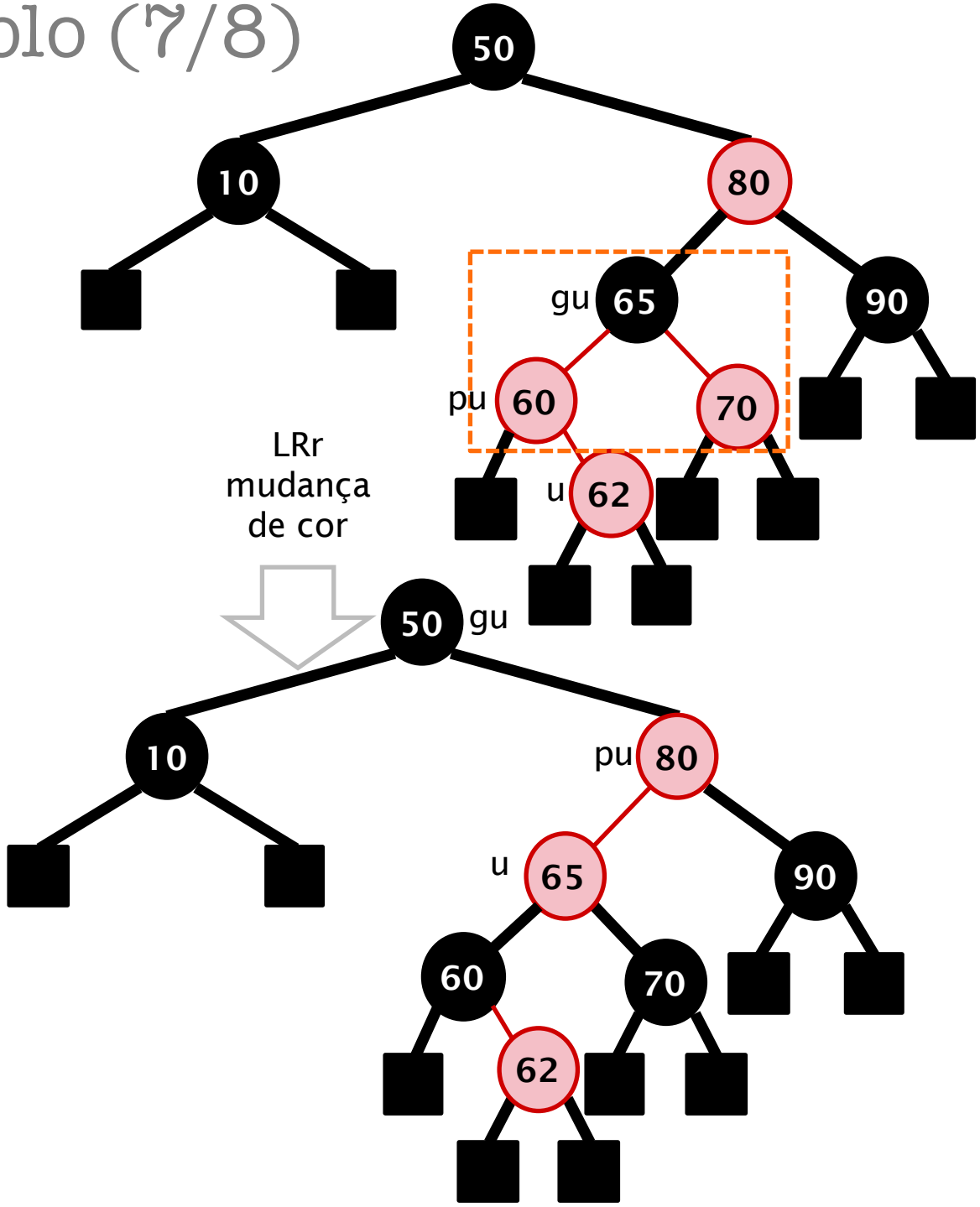
Exemplo (5/8)



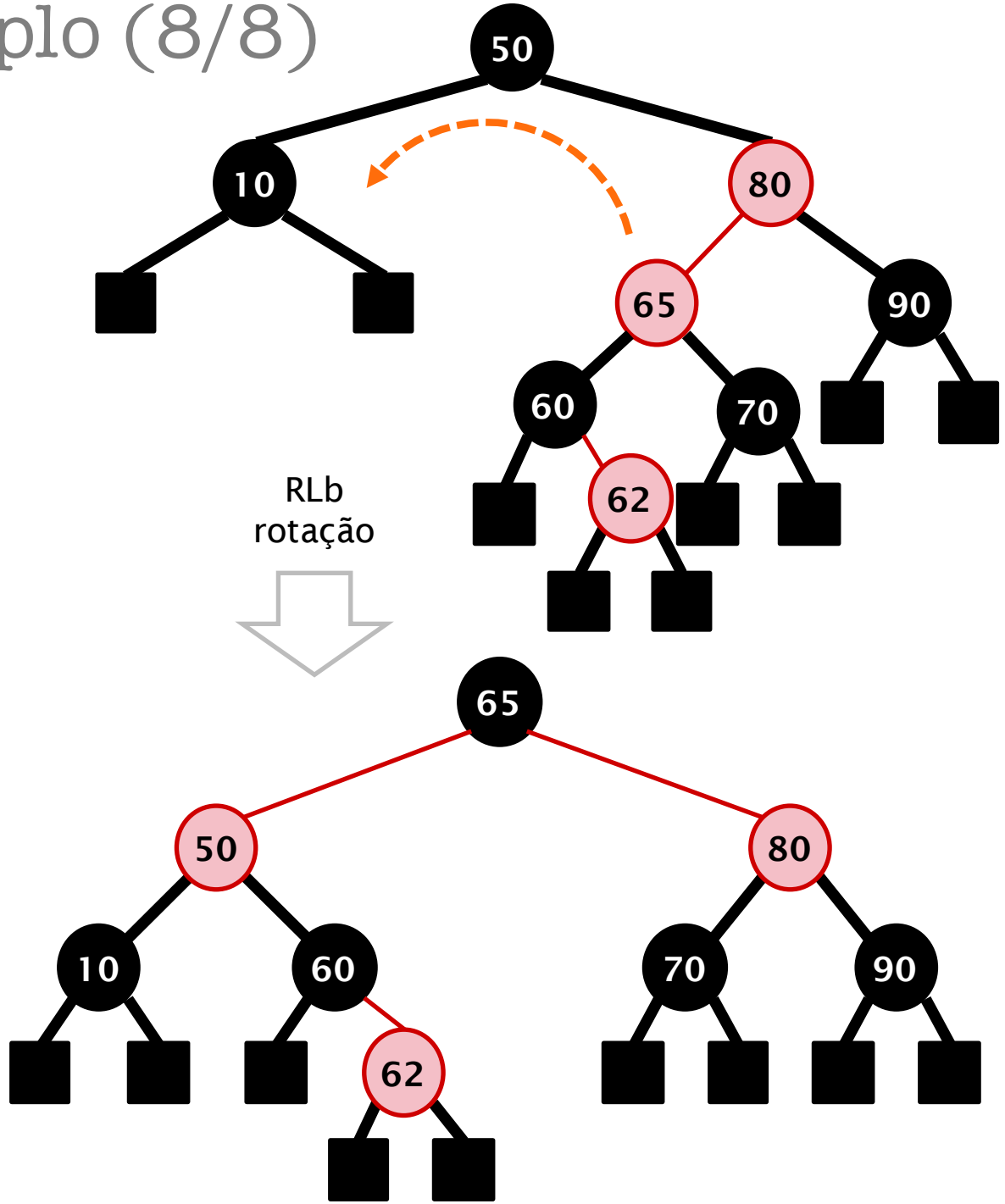
Exemplo (6/8)



Exemplo (7/8)


50
10
80
gu
65
90
pu
60
70
u
62
LRr
mudança
de cor
50
gu
10
pu
80
u
65
90
60
70
62

Exemplo (8/8)



visualização em:

https://www.cs.usfca.edu/~galles/visualization/RedBlack.html

(vá inserindo chaves em ordem crescente para entender como a árvore se comporta!