Aula dia 09/02/22 - Árvores Rubro Negras

Uma árvore rubro-negra é uma árvore binária de busca que satisfaz às seguintes propriedades

1) Todo nó é vermelho ou preto
2) A raiz é preta
3) Toda folha é preta (*) folha é nó vazio

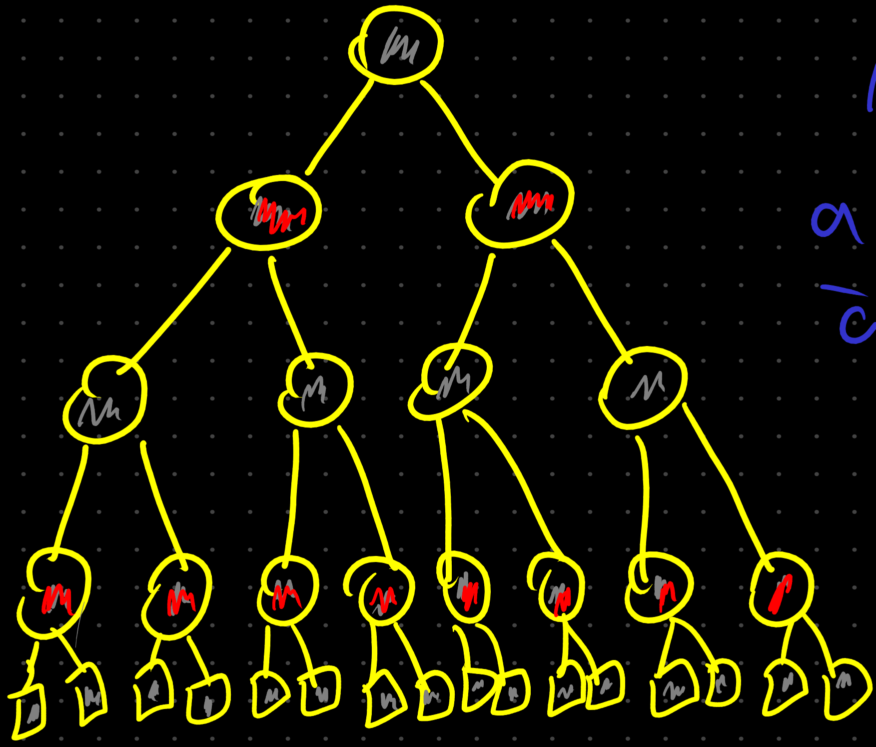
subárvores que são vazias

4) Se um nó é vermelho, então seus filhos são pretos
5) Para cada nó, todos os caminhos simples do nó até as folhas descendentes contém o mesmo número de nós pretos.

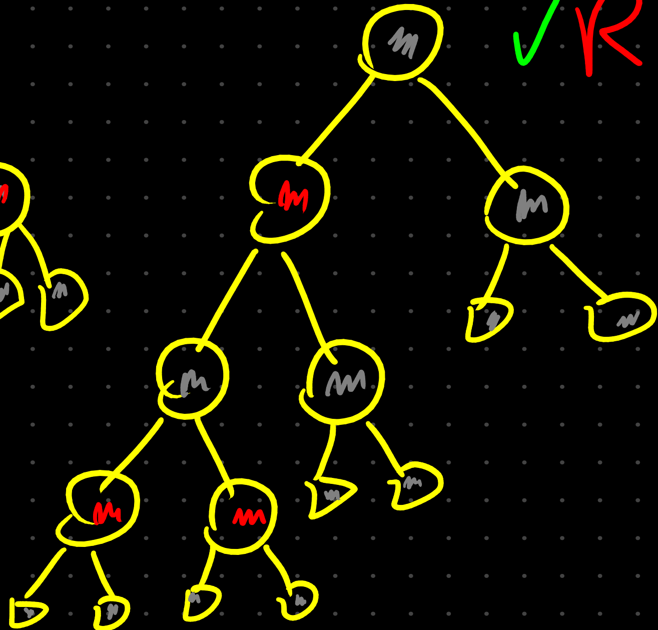
Lema: Uma árvore rubro-negra com n nós internos (não é folha) tem, no máximo, a altura $\boxed{2\log_2(n+1)} = O(\log n)$

no mínimo, é sempre $\log_2 n + 1$ → Árvore completa

Mostramos que uma ABB é RB apresentando uma coloração que satisfaz às propriedades.



✓ R N



A complexidade da busca, inserção e remoção em uma ABB é $O(h)$, onde h é altura da árvore.

A altura de uma ABB com n nós é no mínimo $\log n$ no máximo n. Se garantirmos que a altura de uma ABB for $O(\log n)$, a busca, a inserção e a remoção ficam $O(\log n)$. Porém, se após uma remoção ou inserção as propriedades que proporcionam esse $O(\log n)$ para a altura forem violadas, perdemos essa vantagem nas operações seguintes. Por isso as operações de inserção e remoção em árvores balanceadas precisam ser revistas.

Algoritmo rotação_esquerda(T, x) 0

1 y ← x.dir

2 x.dir ← y.esq

3 se y.esq ≠ NULO então

  | y.esq.pai ← x

fimse

4 y.pai ← x.pai

se x.pai ≠ NULO então

5 S1 T.raiz ← y

senão se x = x.pai.esq então

S.2 x.pai.esq ← y

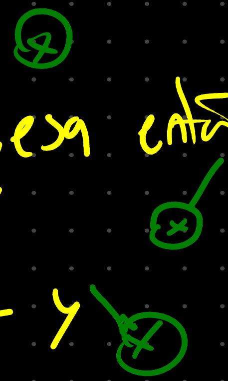
S.3 senão x.pai.dir ← y

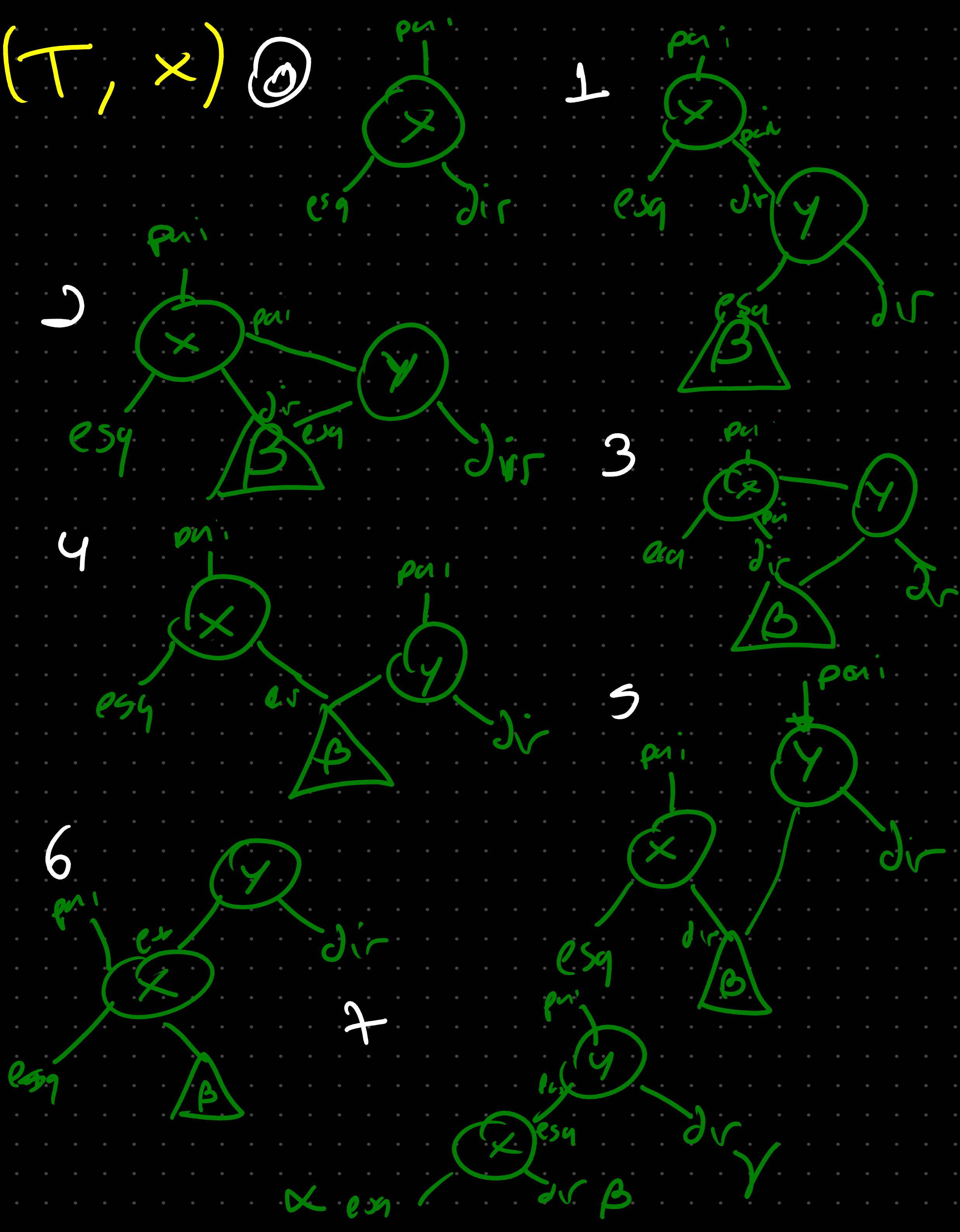
fimse

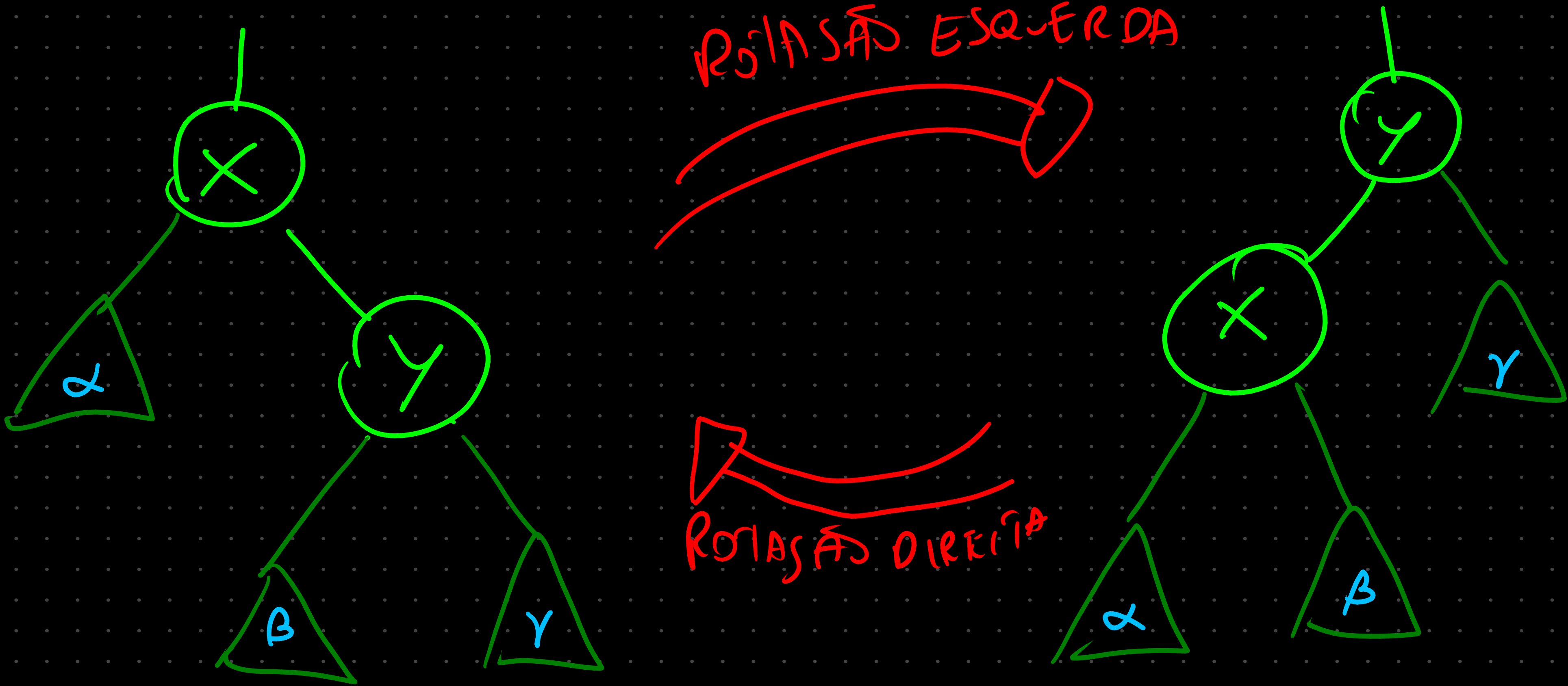
6 y.esq ← x

7 x.pai ← y

fim



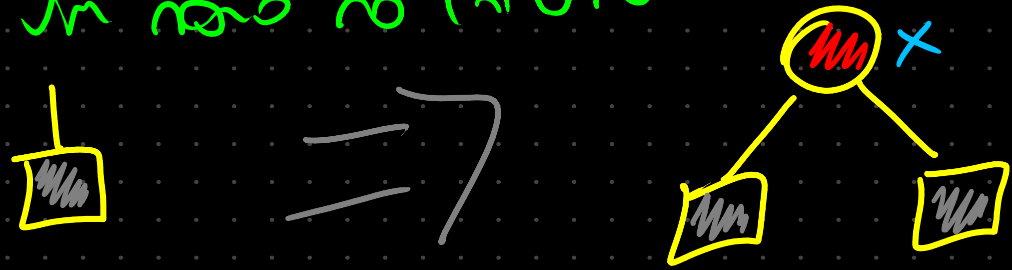




Essas operações elas transformam a árvore preservando as propriedades de árvore binária de busca

Inserção na ARB

A inserção vai sempre substituir um nó folha por um novo nó interno



Se o pai de x for preto. Ok não há violação alguma

Se o pai for vermelho, a gente olha para o tio