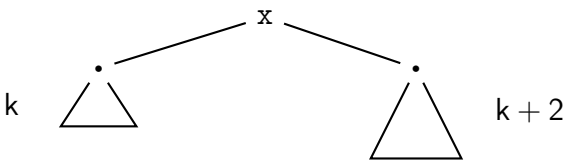


Inserção na árvore AVL

Agora está na hora de ver como as coisas funcionam na prática.

Quer dizer, após uma operação de inserção, a propriedade da árvore AVL pode ser violada.

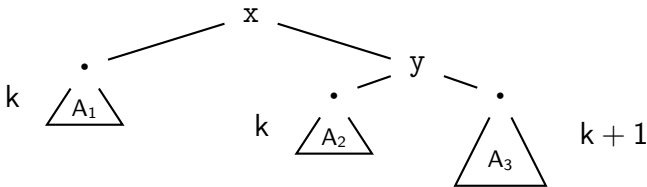
Por exemplo,



A ideia aqui é que o desbalanceamento foi provocado por uma inserção no lado direito.

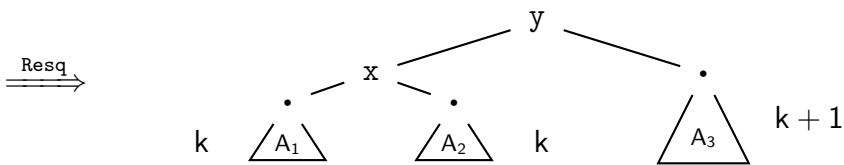
E para resolver o problema, é preciso examinar a estrutura da subárvore direita.

O caso mais simples é o seguinte

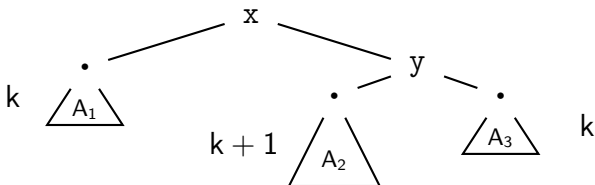


onde a inserção ocorreu à direita de y

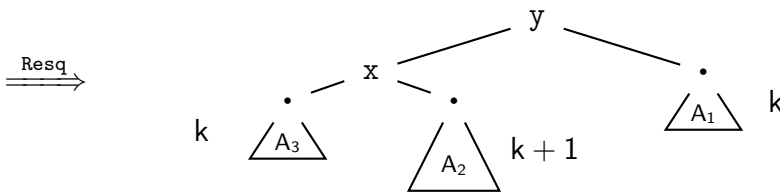
Esse caso é fácil porque basta uma rotação à esquerda para restaurar o balanceamento



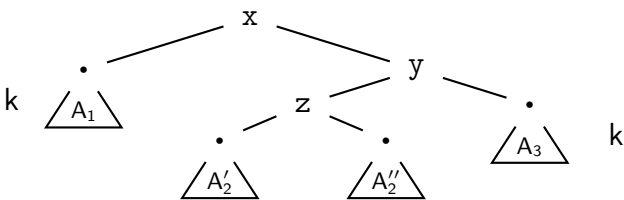
Mas, pode ser que a inserção tenha ocorrido à esquerda de y, produzindo a seguinte situação



E nesse caso, a rotação à esquerda apenas leva o problema para o outro lado



Daí que agora é preciso examinar a estrutura da subárvore esquerda de y

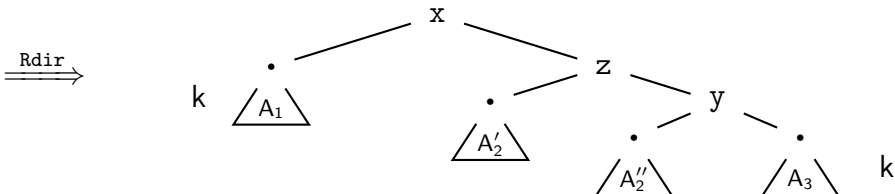


onde A_2' tem altura k e A_2'' tem altura $k + 1$ — (ou vice-versa)

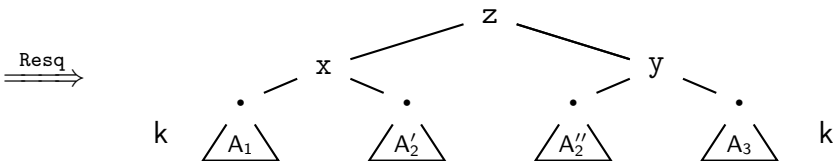
Em qualquer caso, uma operação de *rotação dupla* restaura o balanceamento de x .

Quer dizer,

- primeiro nós aplicamos uma rotação à direita sobre y



- e depois nós aplicamos uma rotação à esquerda sobre x



E daí, como

- A_1 e A_3 possuem altura k
- A_2' tem altura k e A_2'' tem altura $k + 1$ — (ou vice-versa)

após a rotação dupla a árvore se encontra balanceada.

Quando o desbalanceamento ocorre para o outro lado, a situação é basicamente análoga.

E agora nós estamos prontos para escrever o algoritmo de inserção.