

CSE 3010 – Data Structures & Algorithms

Lecture #40

What will be covered today

- Construction of an expression tree
- Self balancing BST

Self-Balancing Binary Search Trees

- Self-Balancing BSTs ensure that the height remains small as more and more items are inserted into the tree
- Can be used to implement other ADTs like priority queues, sets, dictionaries, etc.
- Several kinds of self-balancing BSTs based on how they balance the height of the tree

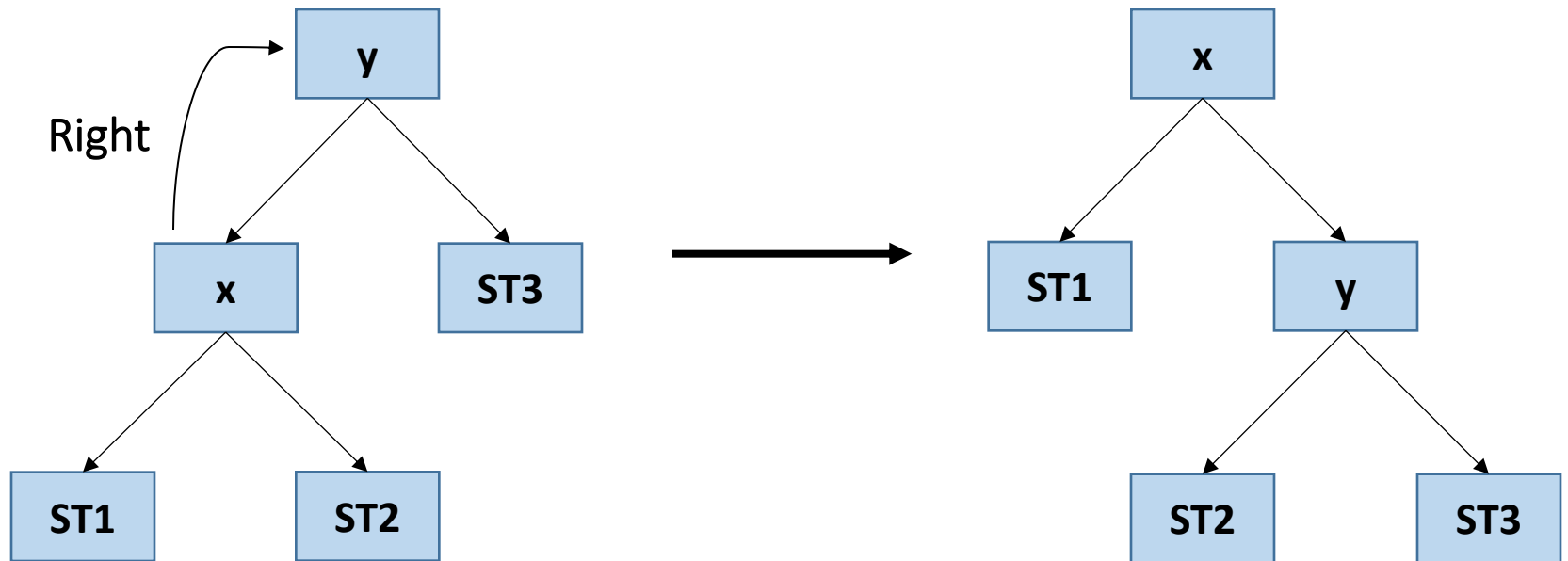
Self-Balancing BSTs

- **Red-Black Trees** – have an additional attribute that is used to keep the tree balanced during insertions and deletions
- **AVL Trees** – named so after their two inventors, Georgy Adelson-Velsky and Evgenii Landis
- **Splay Trees** – recently accessed items are made quick to access by a ‘splay’ operation
- **Scapegoat Trees** – balance trees without any additional node attribute

Tree Rotations

- Tree rotations can be performed on any BST
- They are most commonly used by self-balancing BSTs in order to keep the tree balanced
- Two basic rotations – left rotation and right rotation
- Rotations must ensure that the properties of a BST are met

Right Rotation



Left Rotation

