AA Trees: Definition

- AA Trees are bottom-up trees.
- Instead of storing a color with each node, we store the node's level.
- The *level* of a node represents the number of **left** links on the path to the null node sentinel.

AA Trees: Level

- The level of a node is:
  - o 1, if the node is a leaf
  - o In all other cases, one more than the level of its *left* child.
- A horizontal link is a connection between two nodes at the same level.

1-2

AA Trees: Definition

- A *left* child must be one level lower than its parent, i.e. there are no horizontal left links.
- A right child may be the same level or one level lower than its parent.
- There may not be two consecutive horizontal links.
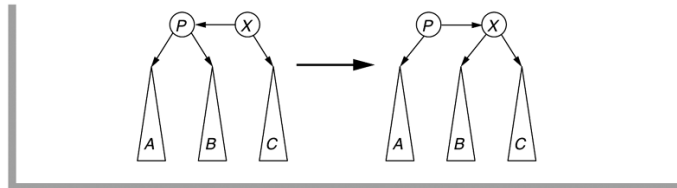
1-3

Rotations

- If an insertion or removal produces a tree that violates the definition of an AA tree, we use rotations to fix this.
- There are **exactly** two rotations.

1-4

Rotation: Skew

- Fixes left arrows
- Corresponds to single rotation on left child.

figure 19.55
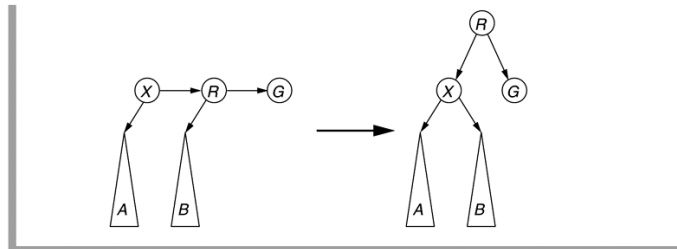The skew procedure is a simple rotation between *X* and *P*.

1-5

Rotation: Split

- Fixes two consecutive right arrows
- Corresponds to single rotation on right child

figure 19.56
The split procedure is a simple rotation between *X* and *R*; note that *R*'s level increases.

1-6

Insertion

- As always, nodes are inserted as leafs.
- A new node is always connected by a horizontal link to its parent.
- Two scenarios:
  - Left link. In this case, perform a skew and fix any problems introduced by it
  - Right link. May have to do a split and fix any problems introduced by it.

1-7

Class exercise on Insertion

- Insertion sequence requiring only splits: 1,2,3,4,5,6,7
- Insertion sequence requiring skews followed by splits: 7,6,5,4,3,2,1
- Insertion sequence requiring a mix of skews and splits: 10, 85, 15, 70, 20, 60, 30, 50, 65, 80, 90, 40, 5, 55, 35.

1-8

Observation about levels

- The insertion algorithm is designed so that:
  - Nodes at level 2 or higher must have two children.
  - If a node does not have a right horizontal link, its two children are at the same level.

1-9

Removal

- Same as in all other binary trees.
- The case where we remove a node with one child only occurs when the sub-tree has exactly two nodes. In particular, the node has just a right child. Treat as special case.
- When removing a leaf, the level of the parent and ancestors may have to be adjusted.
- Violation of AA Tree properties need to be fixed with skews and splits.

1-10

Removal

- Recall that:
  - A left child must be one level lower than its parent.
  - A right child may be the same level or one level lower than its parent.
- If the levels of **a parent and a child** differ by more than one, then lower the level of the parent.
- If that parent has a right child at the same level, then also lower the level of that right child.

1-11

Class exercise on Removal

- From the tree built by 1,2,3, remove: 1
- From same tree, remove: 3
- From the tree built earlier, remove: 5
- From the tree built earlier, remove: 7
- Example from original paper.

1-12
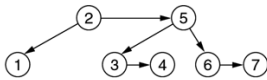
Removal: Interesting Case

figure 19.63

When 1 is deleted, all nodes become level 1, thereby introducing horizontal left links.

1-13