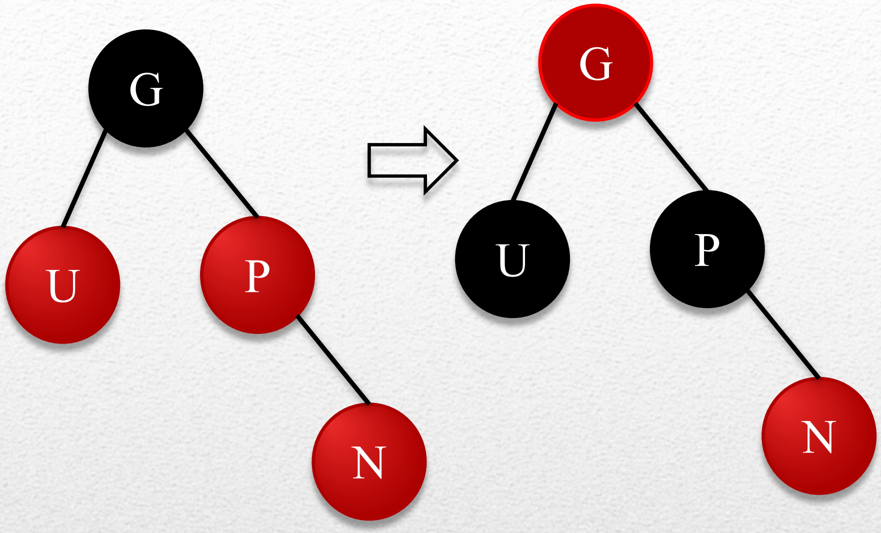
Red Black Trees:Insertion

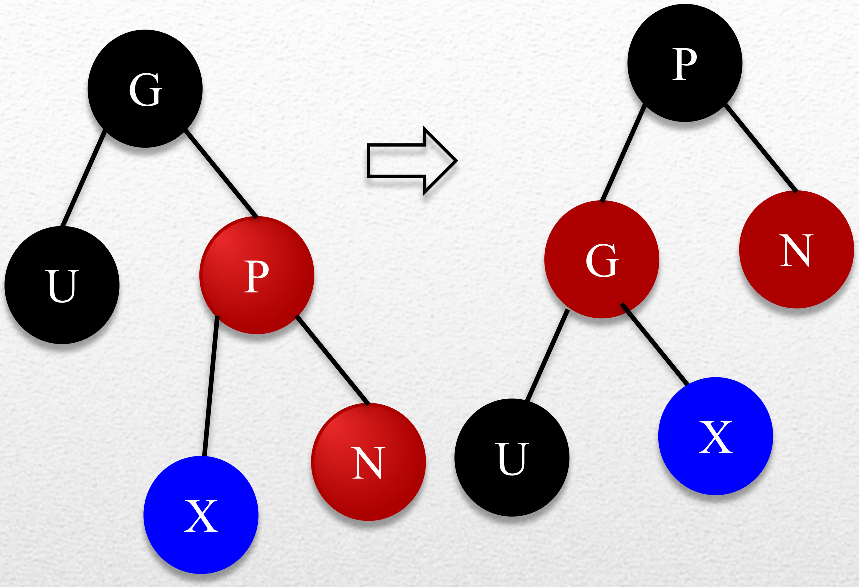
- Insertion
  - Find proper position of the node in the binary tree
  - Insert and color it red
  - If there is conflict then parent node is red
  - Fix conflict

- Case 1:
  - Parent Red, Uncle Red, Grandparent Black
  - Make Grandparent Red and Parent and Uncle Black



Insertion

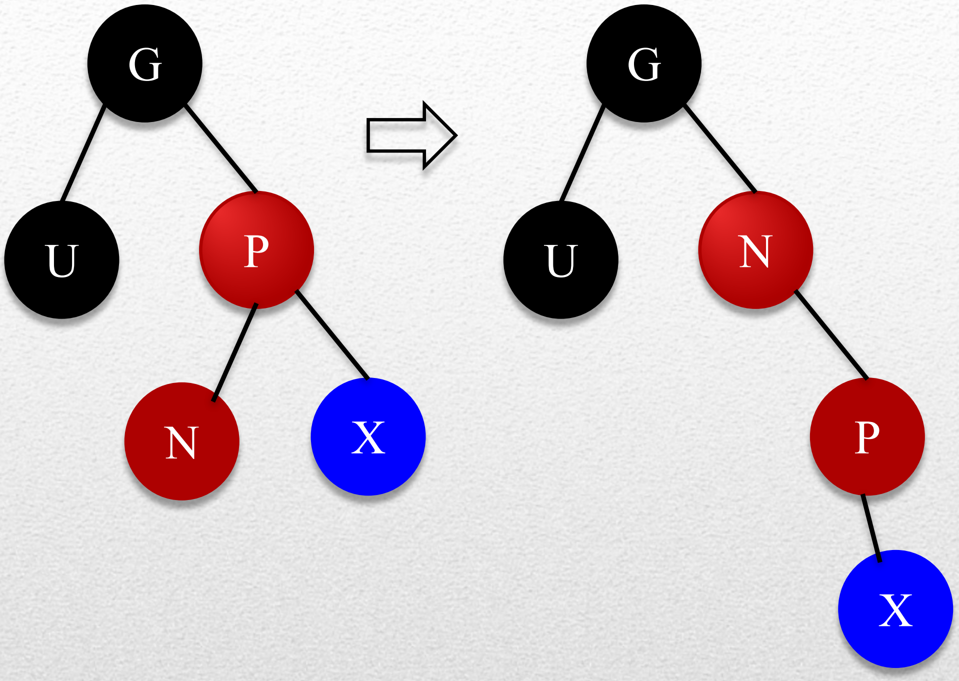
- Case 2a:
  - Parent Red, Uncle Black, Grandparent Black
    - Straight Line among grandparent, parent and child
    - If parent is left (right) child of grandparent, child is left(right) child of parent
  - Rotate such that parent is new parent, and grandparent is child
  - Exchange color of parent and grandparent



Insertion

Insertion

- Case 2b:
  - Parent Red, Uncle Black, Grandparent Black
    - Zigzag among grandparent, parent and child
    - If parent is left (right) child of grandparent, child is right(left) child of parent
  - Rotate such that child is new parent, and parent is child
  - Becomes Case 2a

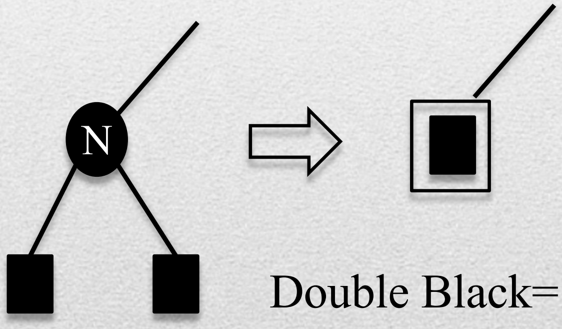


Deletion

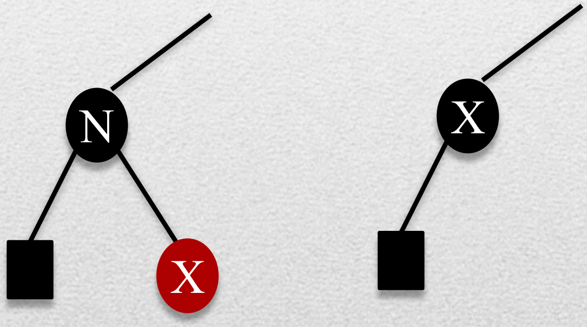
- Deletion
  - If red node deleted –nothing further to be done
  - If black node deleted—there is imbalance

Deletion

- As per binary tree deletion rules, node to be deleted is either leaf or has one child
- In terms of R-B tree two situations can occur
- We only need to fix when double black occurs



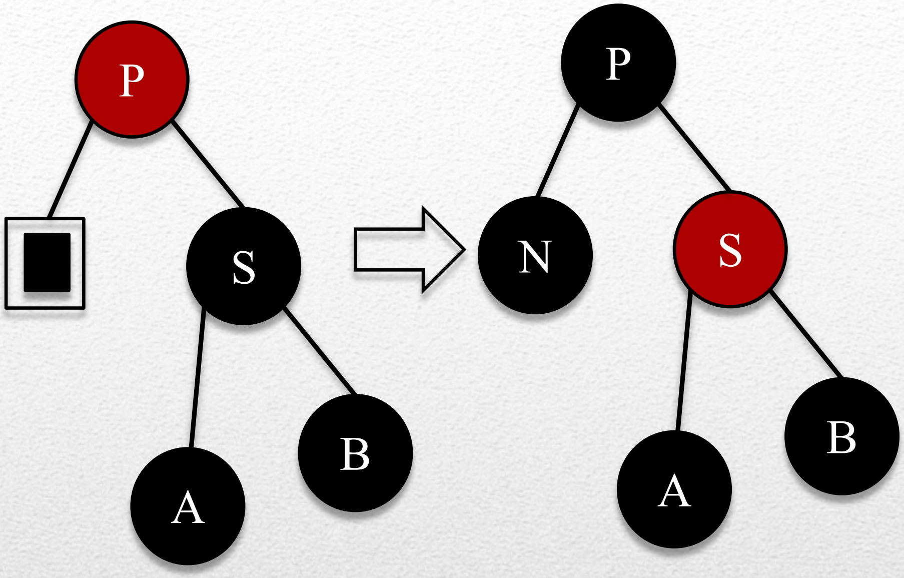
Double Black= Mismatch of black height by 1



Replace N with X and color X black

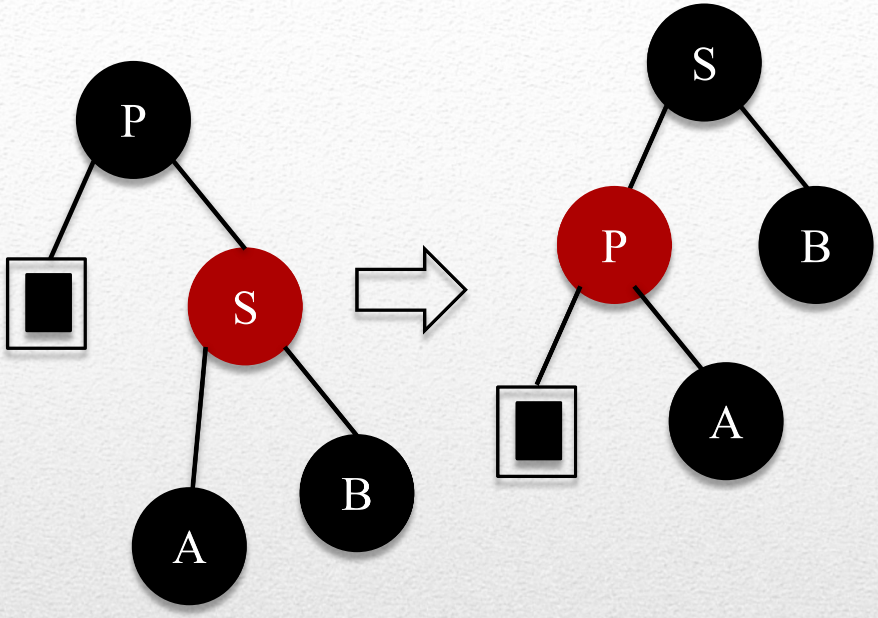
Deletion

- Case 1: Parent red
- Sibling children black
- Change color of sibling and parent

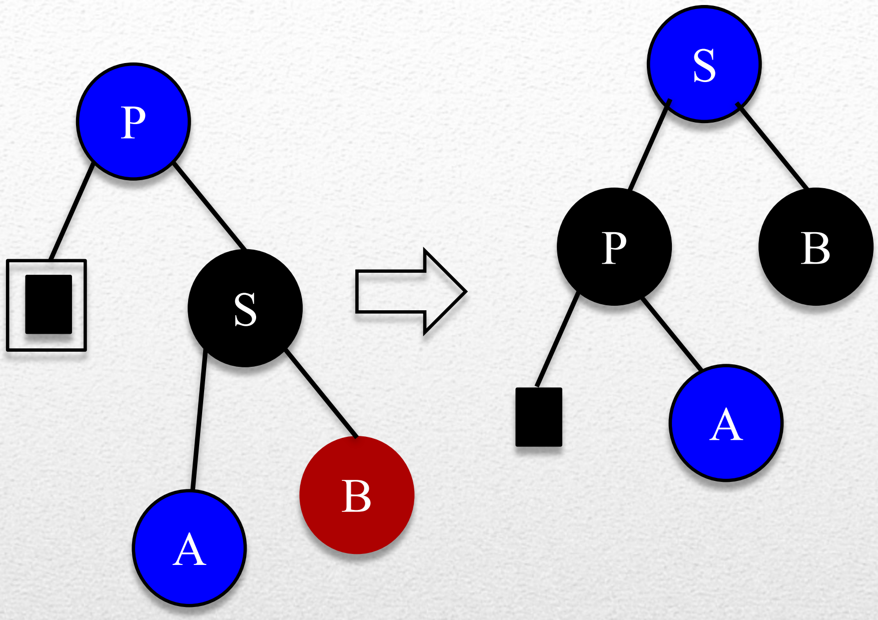


Deletion

- Case 2
  - Parent Black
  - Sibling Red
- Rotate such that sibling is parent, and parent is child of sibling
- Exchange color of parent and sibling
- Now it is Case 1



- Case 3a
  - Sibling is black
  - If sibling is left (right) child of grand parent then left (right) child of sibling is red (Straight line)

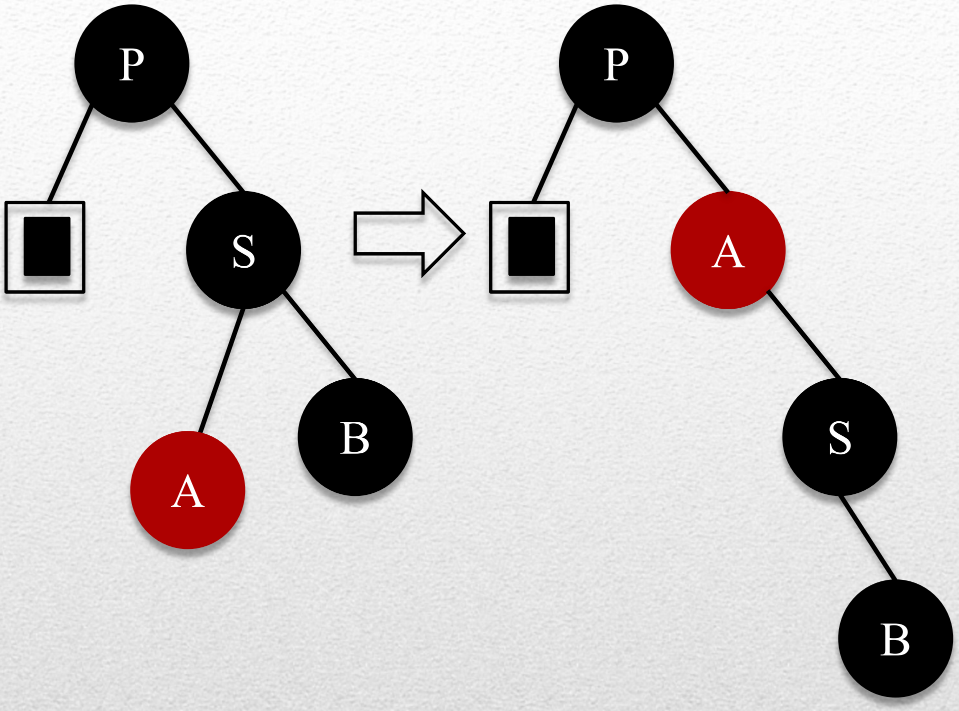


- Rotate to make sibling parent of grand parent.
- Color left child of sibling black

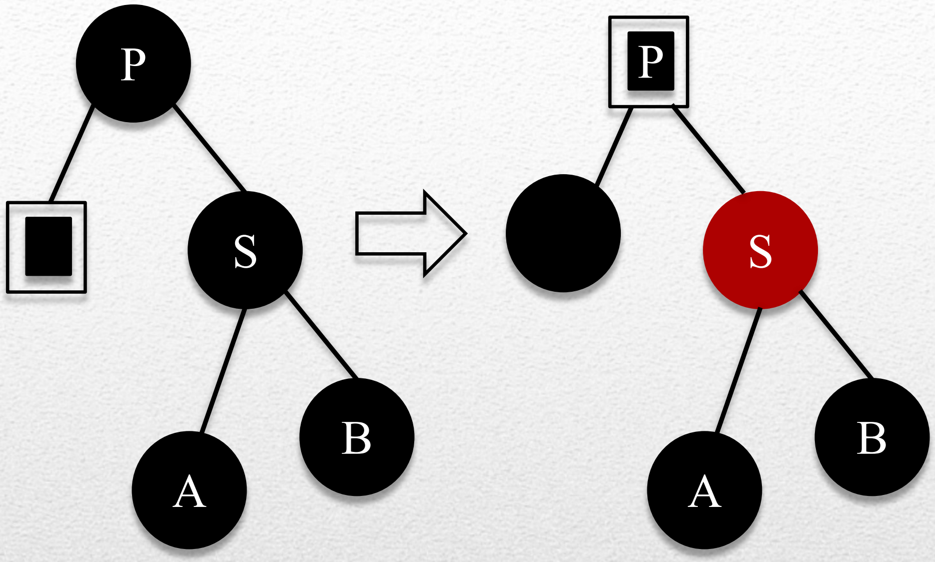
Deletion

Deletion

- Case 3b
  - Sibling is black
  - If sibling is left (right) child of grand parent then right(left) child of sibling is red (zig-zag line)
- (i) Rotate to make sibling and its child to obtain straight line
- It now becomes Case 2
- (ii) If after rotation tree is unbalanced, then switch color of A and S
- It now becomes Case 3a



- Case 3c
  - Sibling is black
  - Parent black
  - Both children of sibling are black



- Make sibling red, and push the double node up
- Transforms to one of the other cases

Deletion