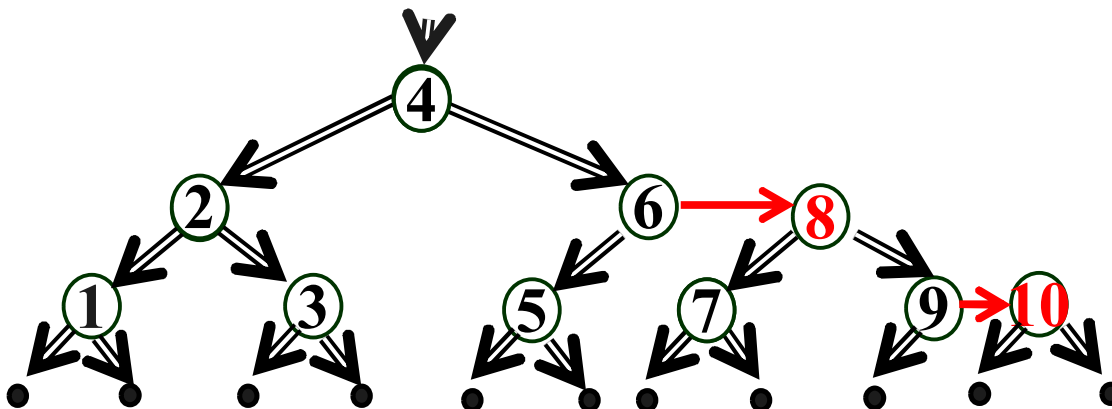


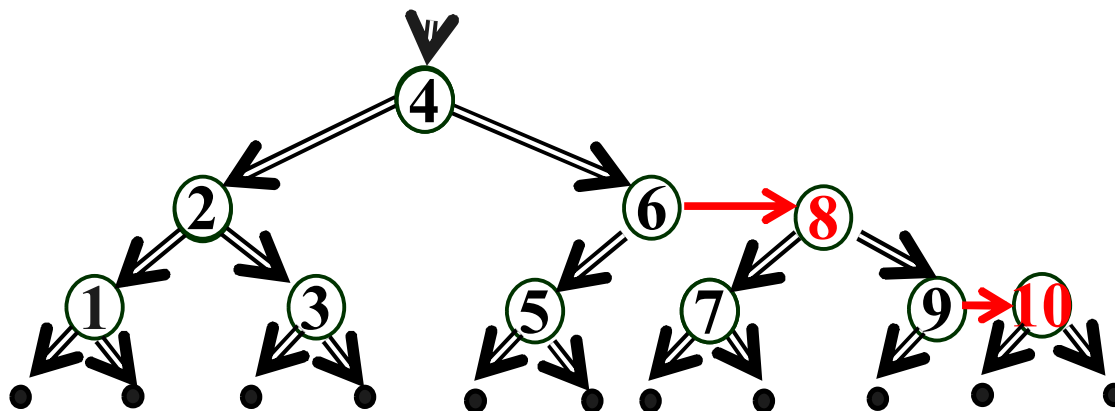
# 红黑树的删除操作

- 在一棵红黑树中删除值为 $u$ 的结点：
  - (1) 在红黑树中查找 $u$ ，然后删除 $u$ 。
  - (2) 删除一个结点后，可能黑高度的限制不满足——要调整
- 删除操作比插入操作要复杂
  1. 逻辑删除——~~所删结点的值消失~~，
  2. 物理（结构）删除——~~结点结构空间消失~~



# 红黑树的删除操作

- 删除值为 $u$ 的结点:
  - 若 $u$ 的孩子均为外部结点，删除 $u$ ，若 $u$ 为黑色----调整
  - 若 $u$ 只有一个**非**外部结点的孩子，则其一定为黑结点，其孩子一定为红色，用其孩子代替它（颜色除外）
  - 若 $u$ 存在2个**非**外部结点的孩子，用其**右子树**的最小值 $min$ 代替 $u$ （不改变 $u$ 所在结点的颜色），删 $min$ ；调整以维护红黑树的黑高度



# 红黑树的删除----删 $u$

- 删结点 $u$ 为红结点----不破坏RB的定义, 不做任何调整
- 删结点 $u$ 为黑结点----要调整
- 删结点 $u$ :
  - $p$ -- $u$ 的双亲结点
  - $s$ -- $u$ 的兄弟结点
  - $r$ -- $s$ 的右孩子结点
  - $l$ -- $s$ 的左孩子结点
- 根据删结点 $u$ 为父结点的左孩子、删结点 $u$ 为父结点的右孩子分两种情况进行调整



# 红黑树的删除----删 $u$

---

□ 调整分2种情况考虑:

➤ 结点 $u$ 为其父结点 $p$ 的左孩子

①  $p, s, l, r$ 均为黑色

②  $l$ 为红色

③  $p$ 为红色

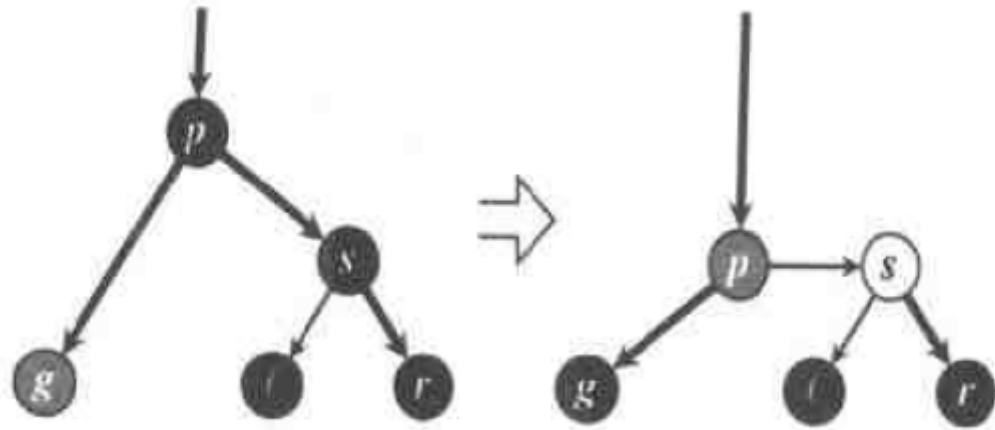
④  $r$ 为红色

⑤  $s$ 为红色

➤ 结点 $u$ 为其父结点 $p$ 的右孩子

u为p的左孩子-Case 1-- $p, s, r, l$ 都是黑色的

$g$ 代表灰色被删结点，需要调整

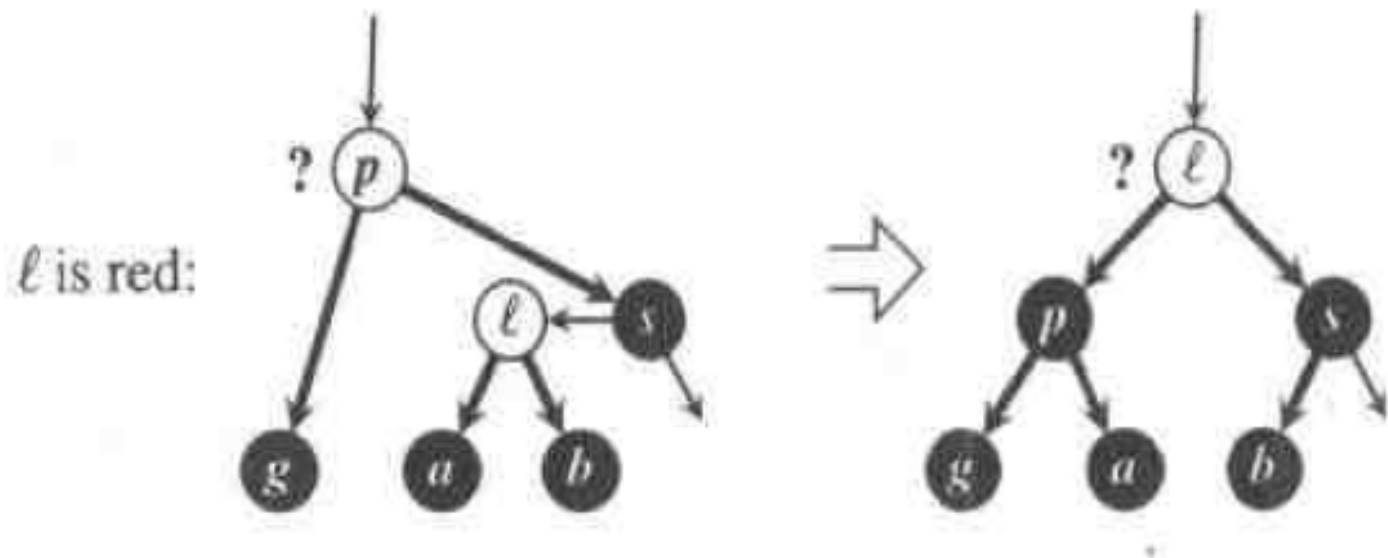


$s$ 变红结点，被删结点变黑

$p$ 结点黑高度少一，继续调整

若 $p$ 为根结点，则结束调整

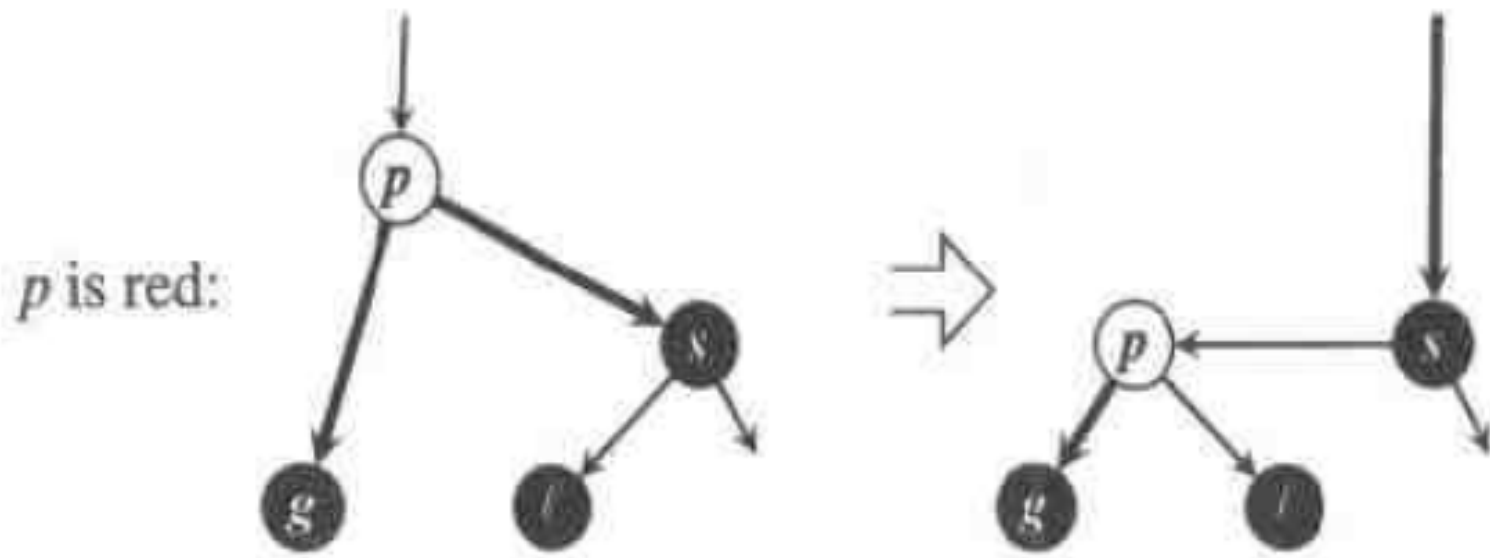
## $u$ 为 $p$ 的左孩子-Case 2-- $l$ 是红色的



先右后左双旋转

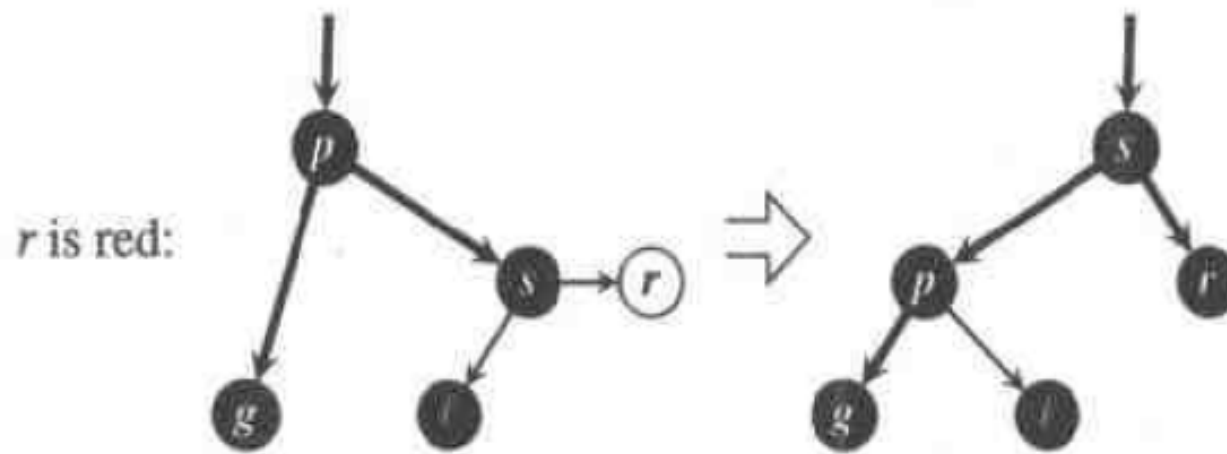
$l$ 取 $p$ 的颜色,  $p$ 变为黑色

u为p的左孩子-Case 3--p是红色的



左单旋转

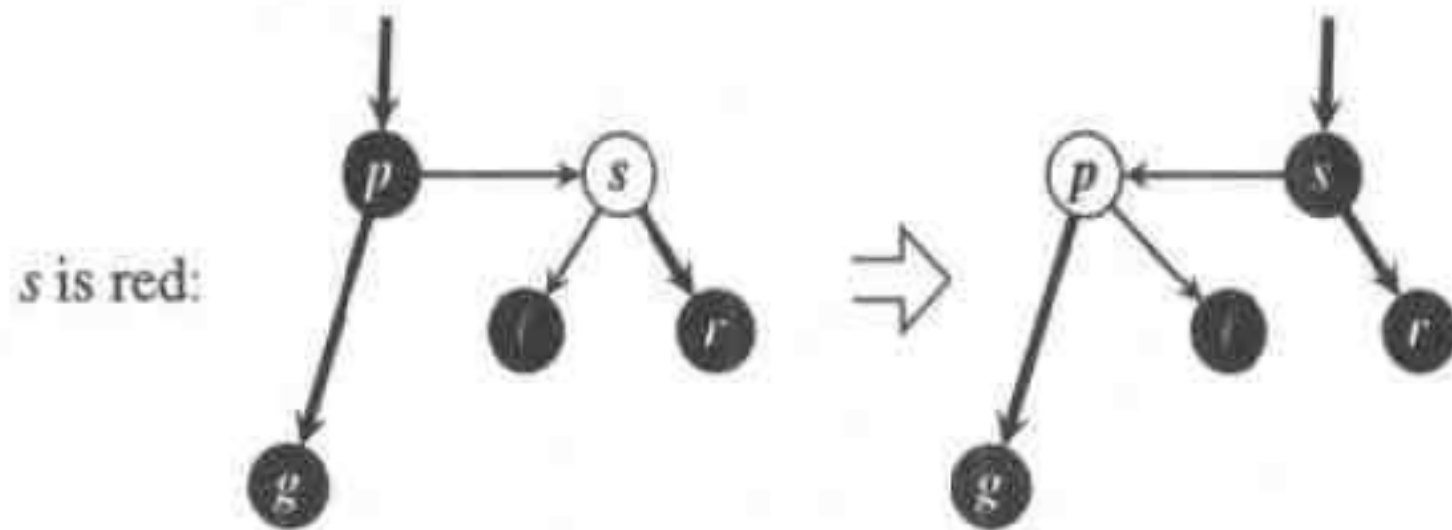
  $u$  为  $p$  的左孩子-Case 4-- $r$  是红色的



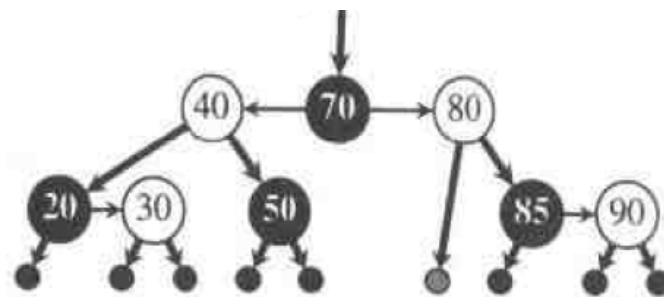
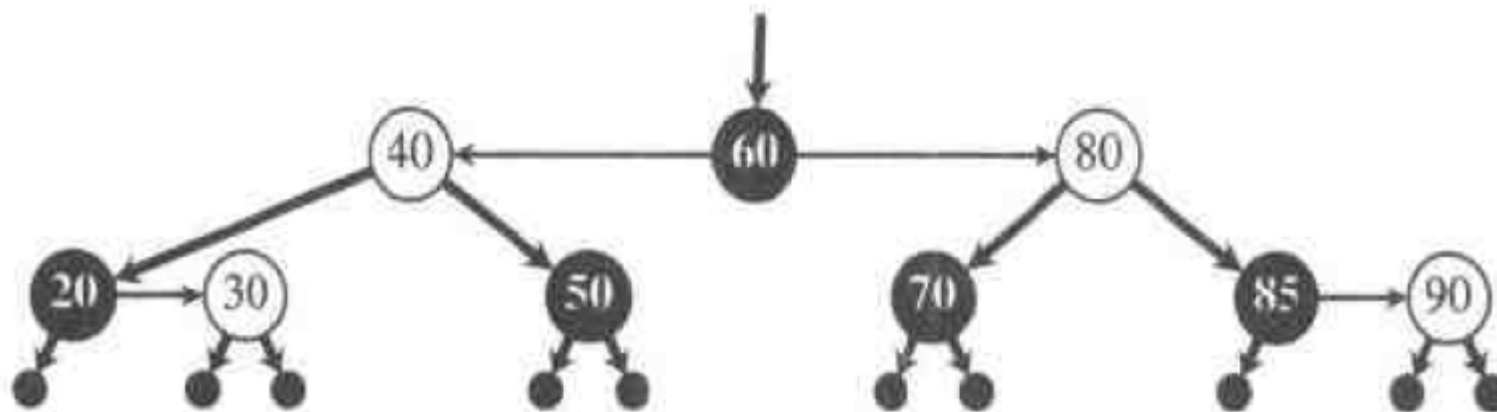
左单旋转  
 $r$  变黑



u为p的左孩子-Case 5--s是红色的

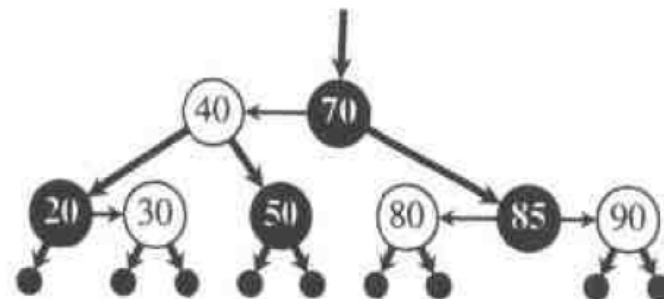


左单旋转

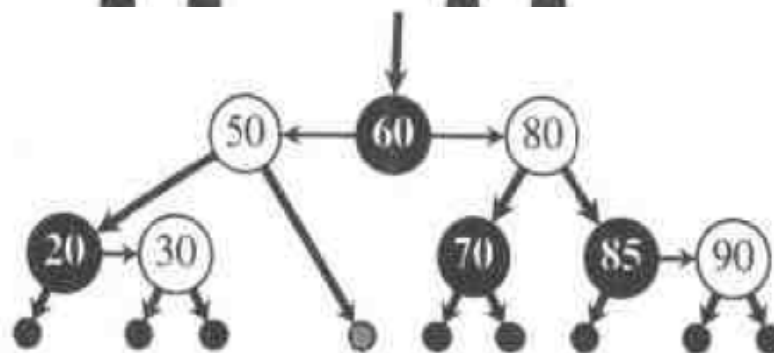
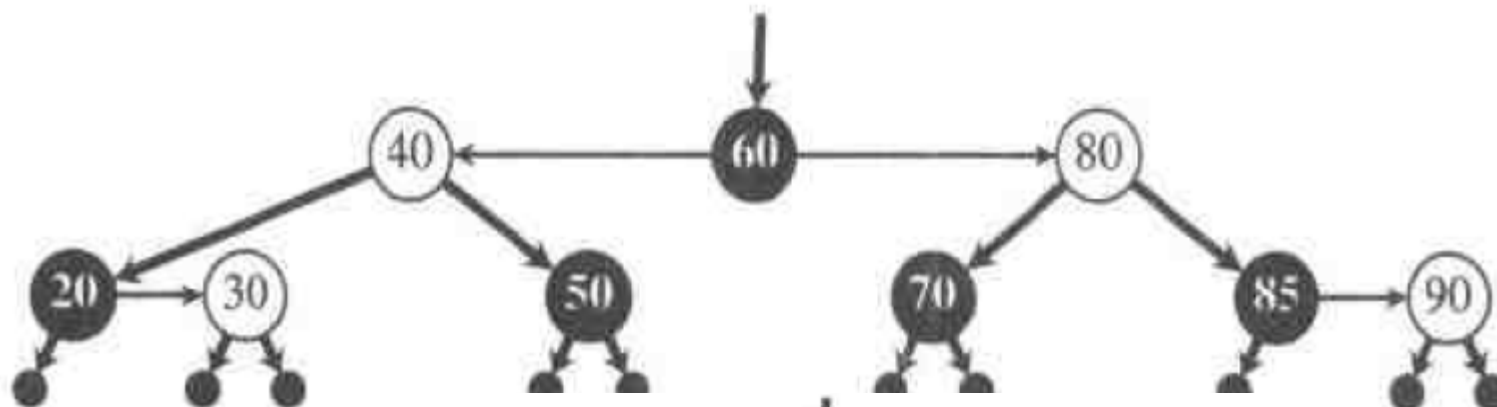


删60

*p*是红色的



左单旋转

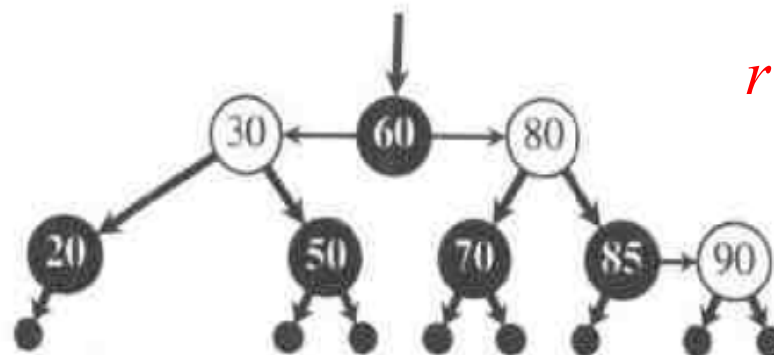


删40

$r$ 是红色的



先左后右双旋转



$r$ 取 $p$ 的颜色,  $p$ 变黑