Lecture 7

Instructor: Karteek Sreenivasaiah

4th September 2018

Delete procedure

The procedure to delete a node M at a high level:

- Case 1: M has two non-leaf children.
  - Replace (data of) M with the successor.
  - Splice out (delete) successor. This makes it case 2.

Delete procedure

The procedure to delete a node M at a high level:

- Case 1: M has two non-leaf children.
  - Replace (data of) M with the successor.
  - Splice out (delete) successor. This makes it case 2.
- Case 2: M has at most one non-leaf child. Call this C.
  - Trivial case: M is red.
  - Minor case: M is black but C is red.
  - Major case: M and C are both black.

Note: If M has both leaf children (NIL), then C is any one of the NIL nodes.

M has at most one non-leaf child C

Trivial Case

M is red:

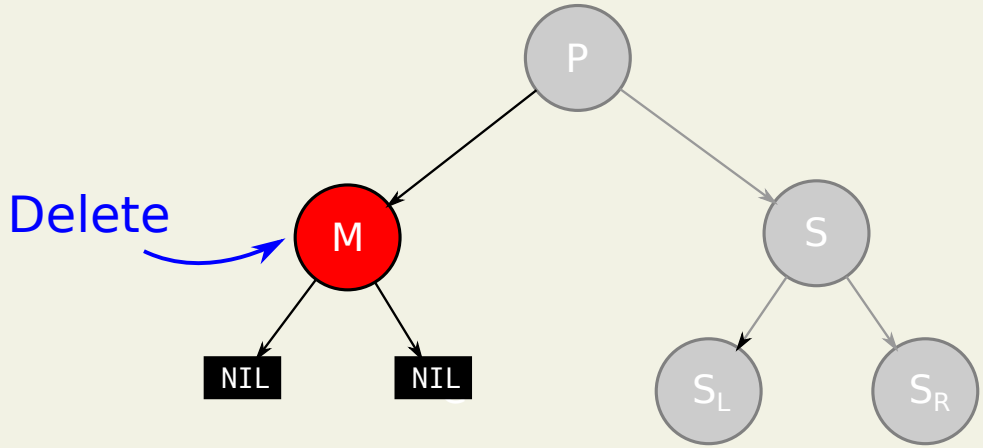
Resolution:

- Then simply replace M with C.

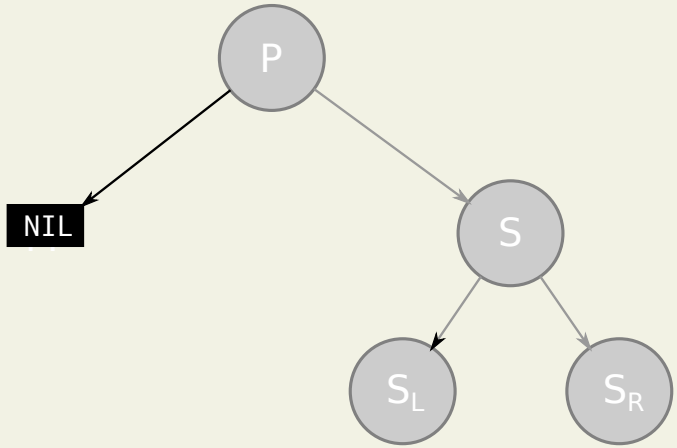
Note:

- M could not have been root.
- For all paths, the black height is not affected.

Trivial Case



Trivial Case



M has at most one non-leaf child C

Minor Case

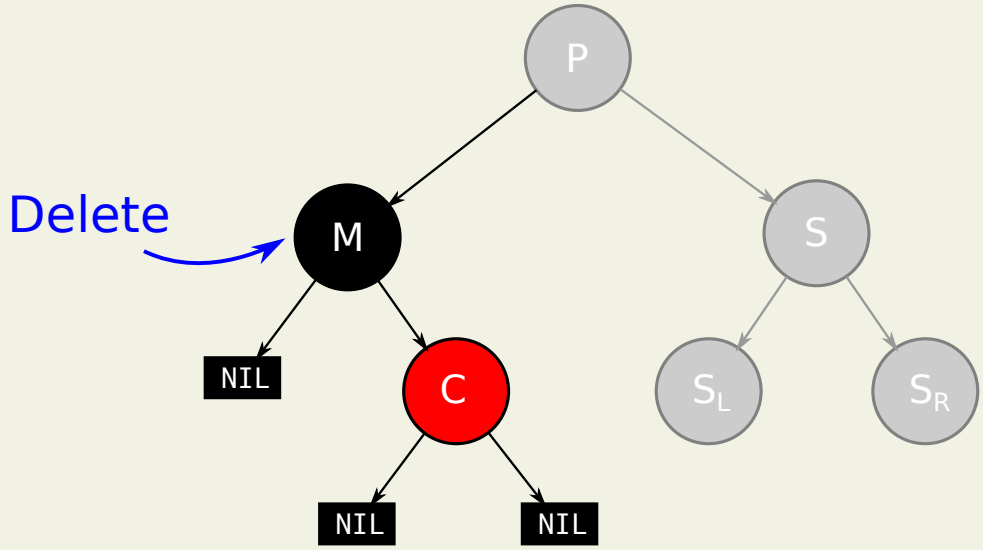
M is black and C is red.

Resolution:

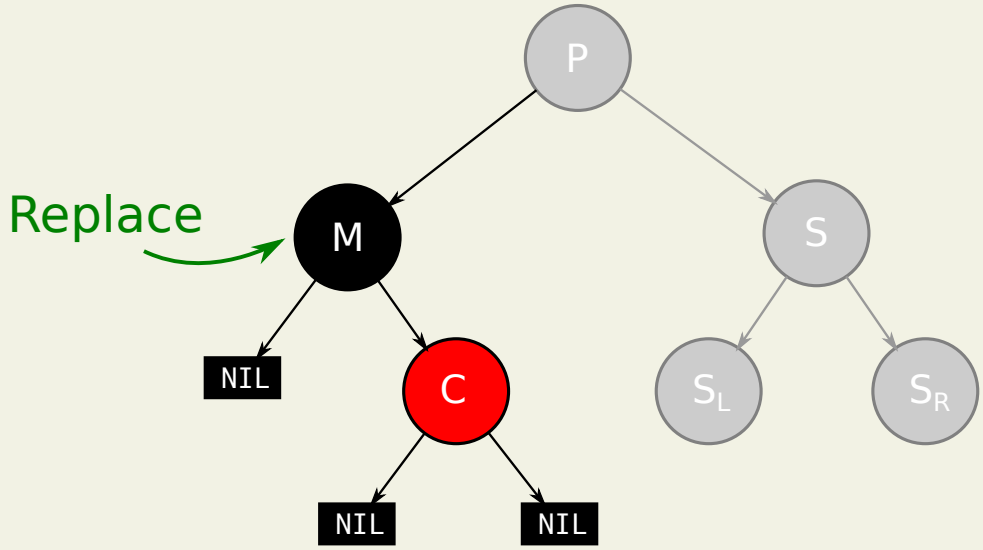
- Replace (splice) M with C.
- Place a "black token" on C.
- Safely discard black token by coloring C black.

The token indicates that the node contributes an extra black to the black count.

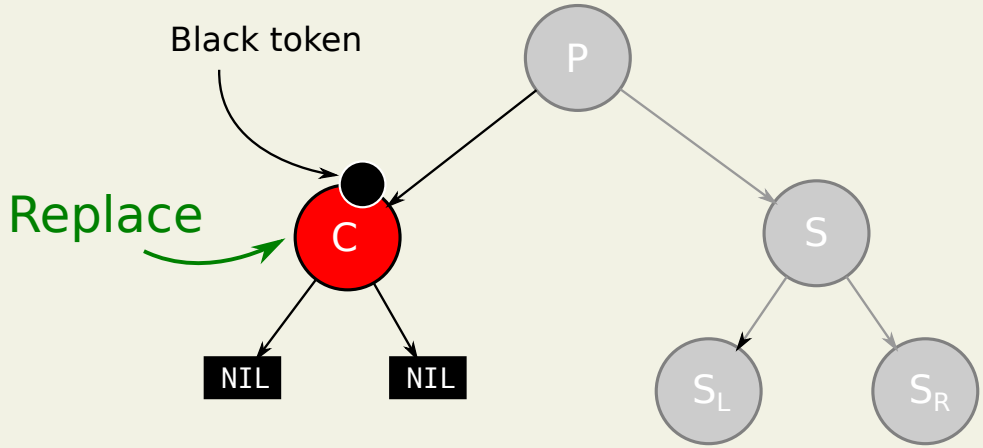
Minor Case



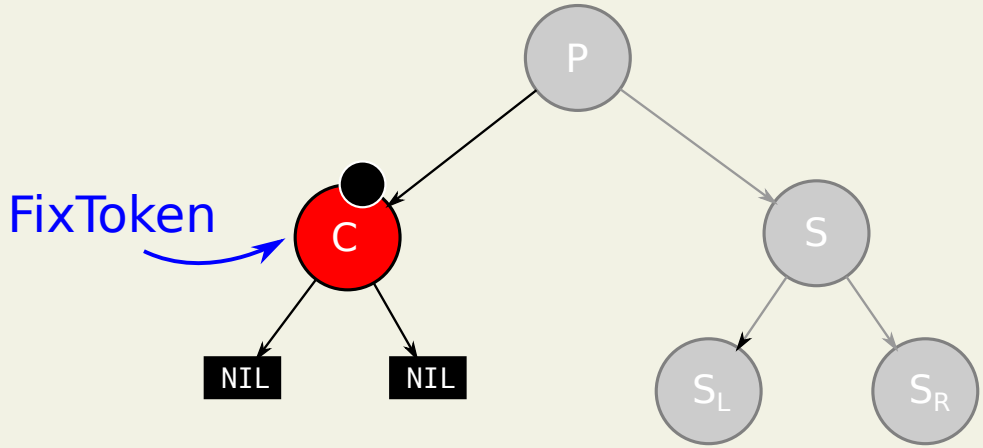
Minor Case



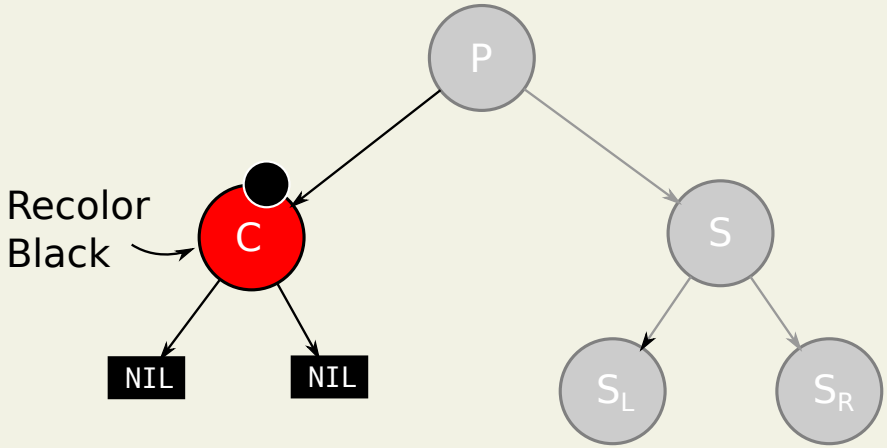
Minor Case



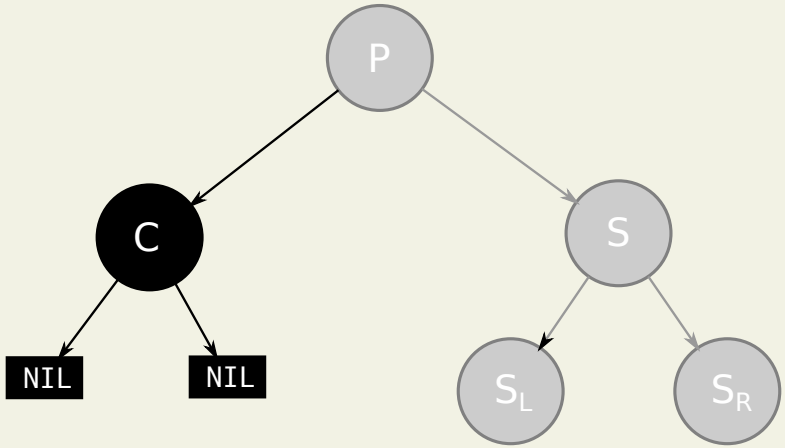
Minor Case



Minor Case



Minor Case



M has at most one non-leaf child C

Major Case

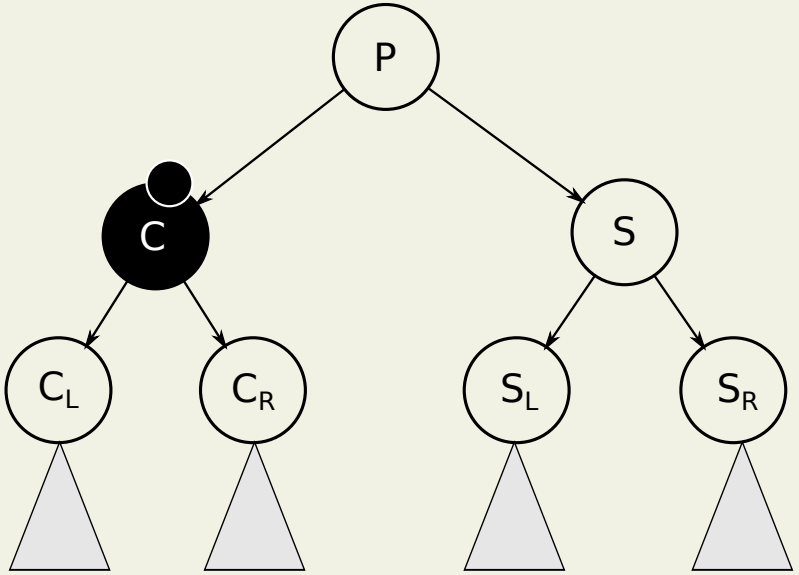
M is black and C is black.

Resolution:

- Replace (splice) M with C.
- Place a black token on C.
- Four cases to safely remove the token placed on C.

Major case - notation

Before replacement of M by C



Removing the token from C

Four cases based on the sibling S of C:

1. S is red.
2. S is black and has both children colored black.
3. S is black and has left child red and right child black.
4. S is black and has right child red.

Removing the token from C

Case 1

Sibling S of C is red.

Since S is red:

- It must have both black children.
- The parent P of S must be black.

Removing the token from C

Case 1

Sibling S of C is red.

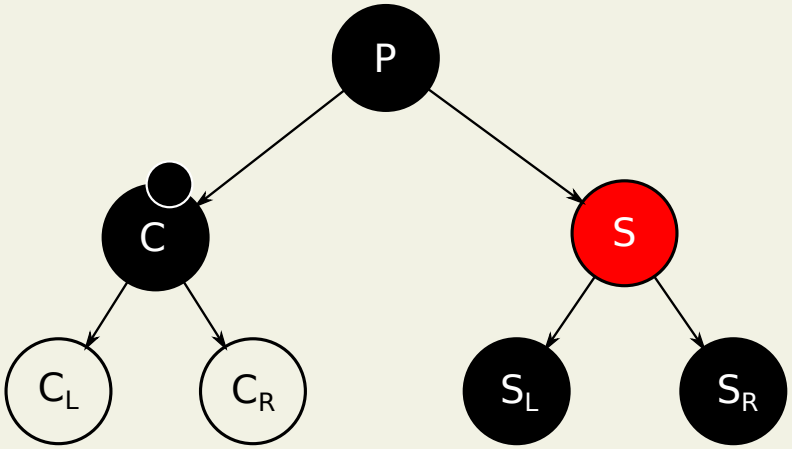
Since S is red:

- It must have both black children.
- The parent P of S must be black.

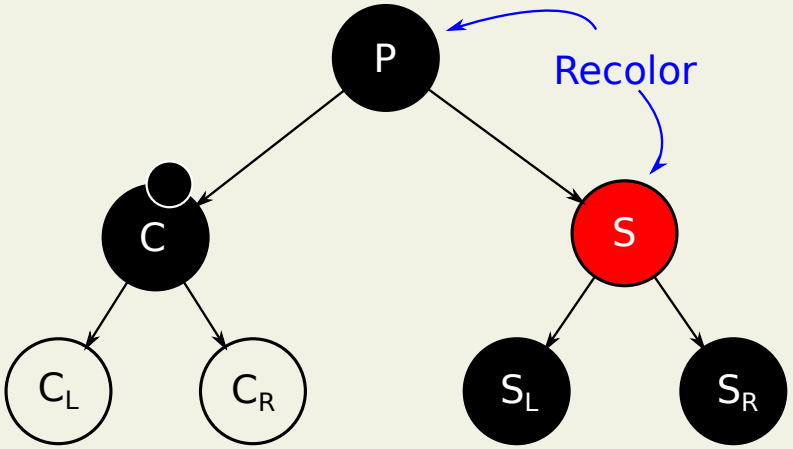
Resolution:

- Recolor S black and its parent red.
- Rotate at parent. (left rotate if M was left child)
- This is now case 2.

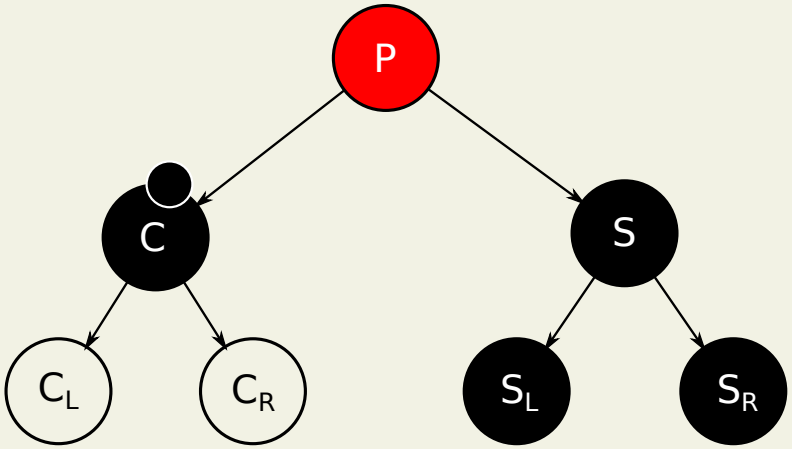
Removing the token from C



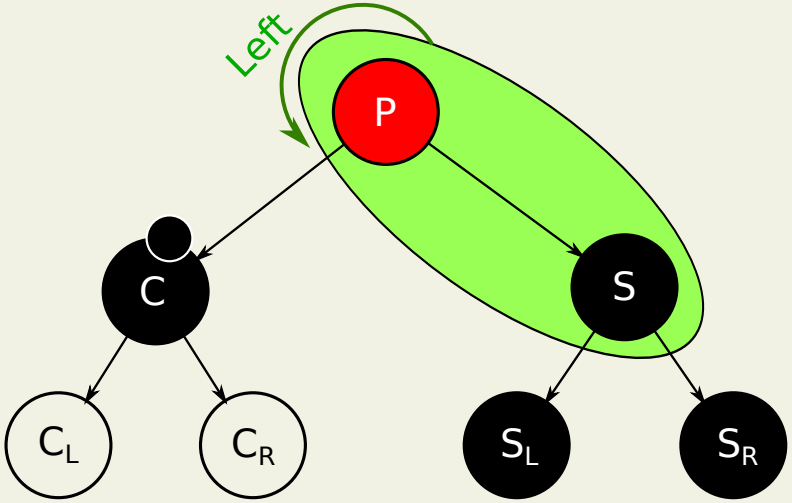
Removing the token from C



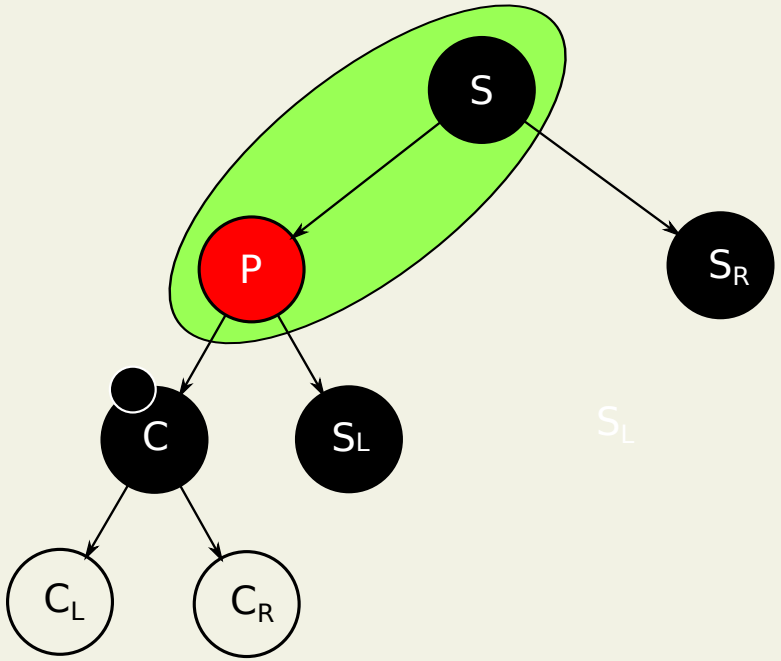
Removing the token from C



Removing the token from C



Removing the token from C



Case 2

Sibling S is black and has both black children.

Removing the token from C

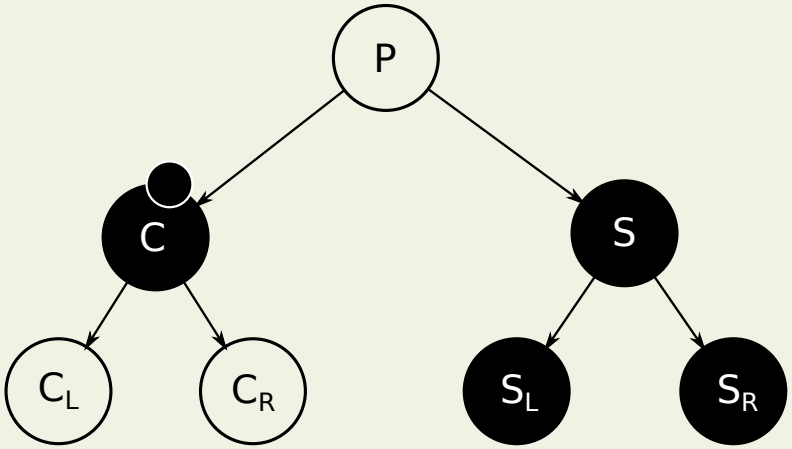
Case 2

Sibling S is black and has both black children.

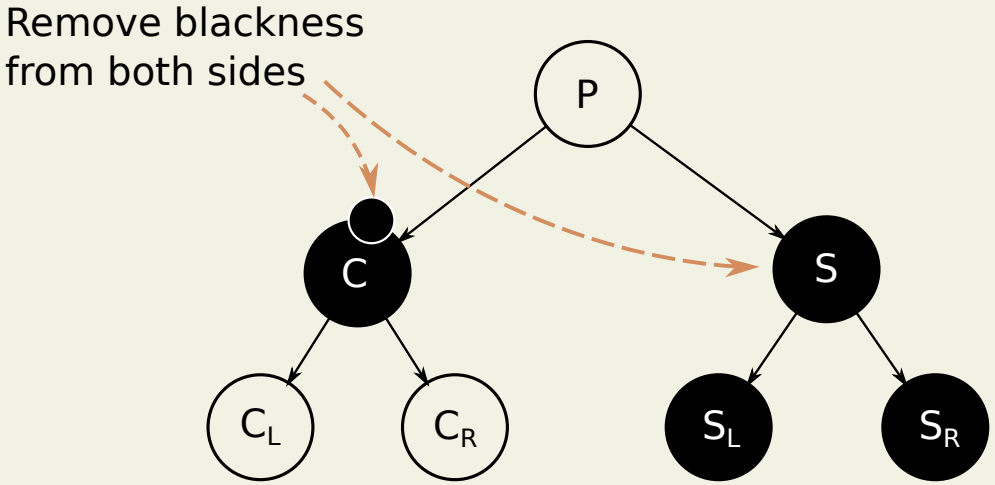
Resolution:

- Remove a black from both C and S.
- Paste token on the parent P.

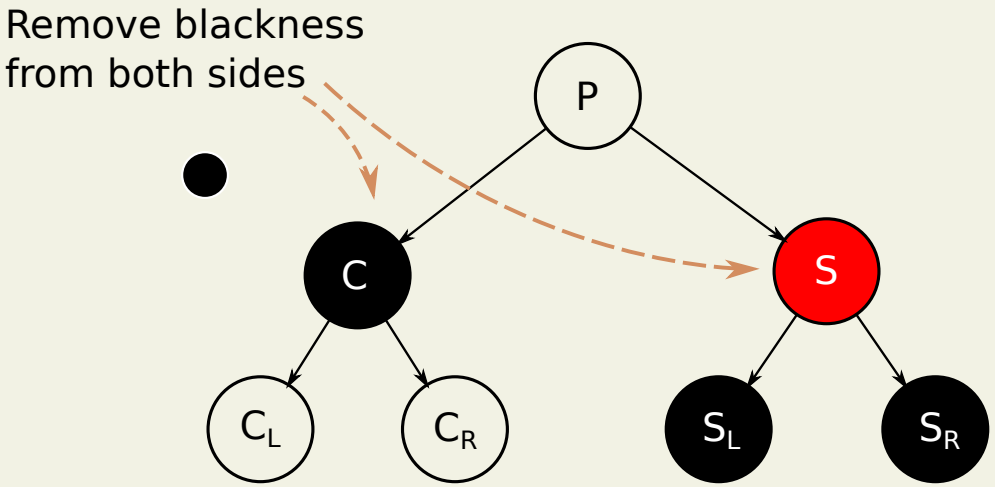
Removing the token from C



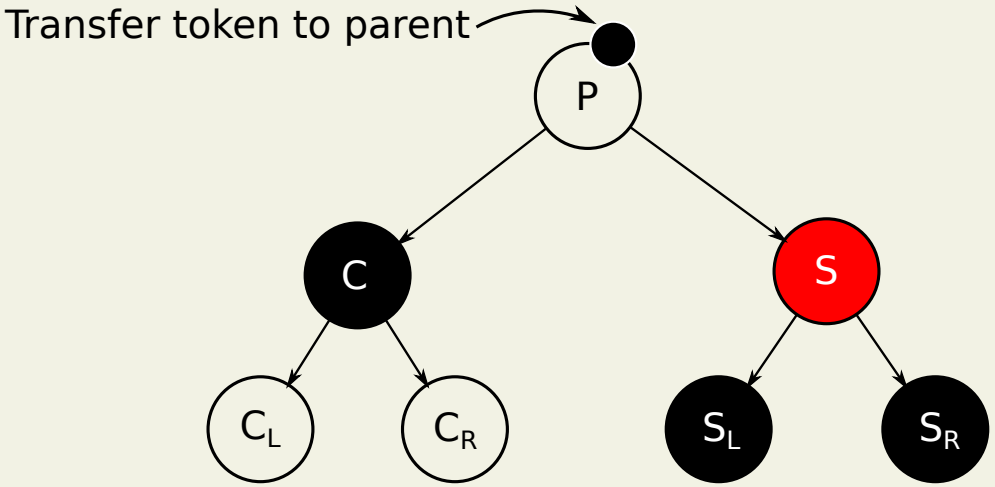
Removing the token from C



Removing the token from C



Removing the token from C



Removing the token from C

Case 3

Sibling S is black.

Left child S_L is red.

S_R is black.

Removing the token from C

Case 3

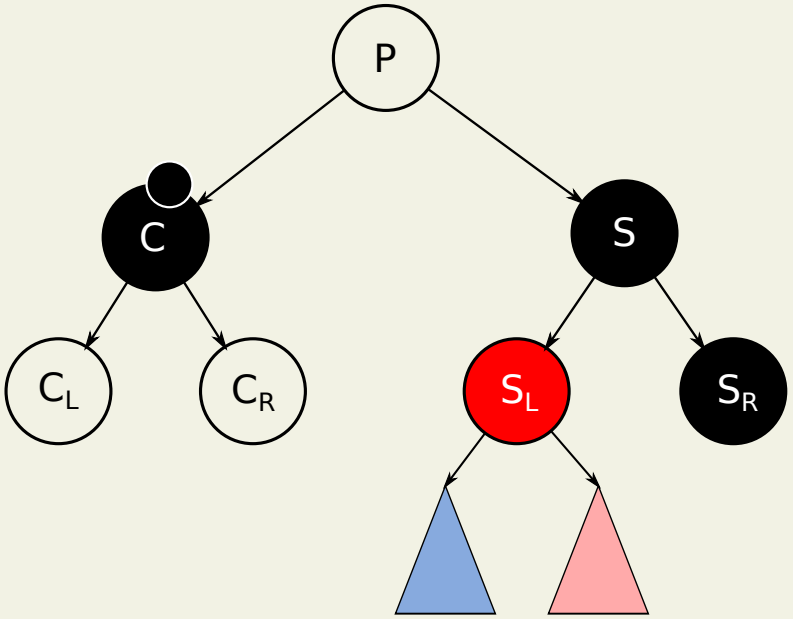
Sibling S is black.
Left child S_L is red.
S_R is black.

Resolution:

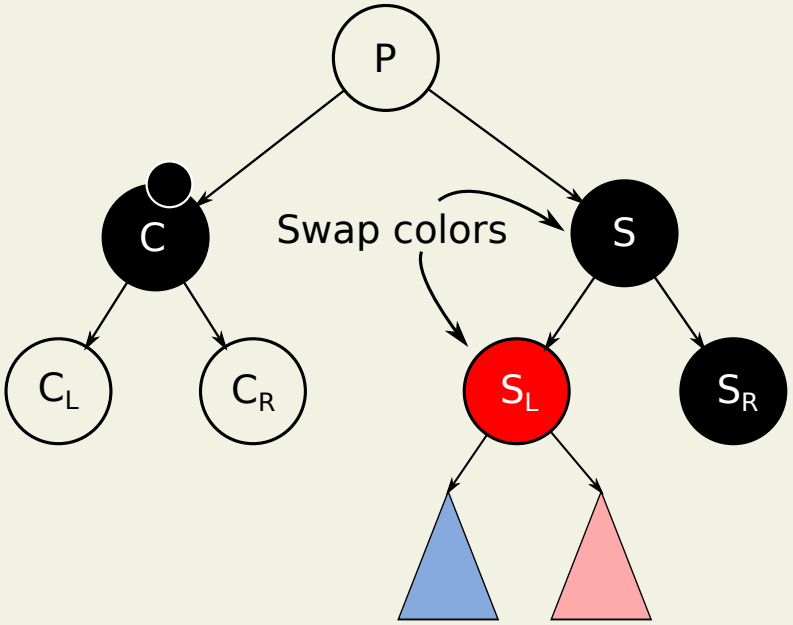
- Swap colors of S_L and S.
- Rotate right at S.

This gives us case 4.

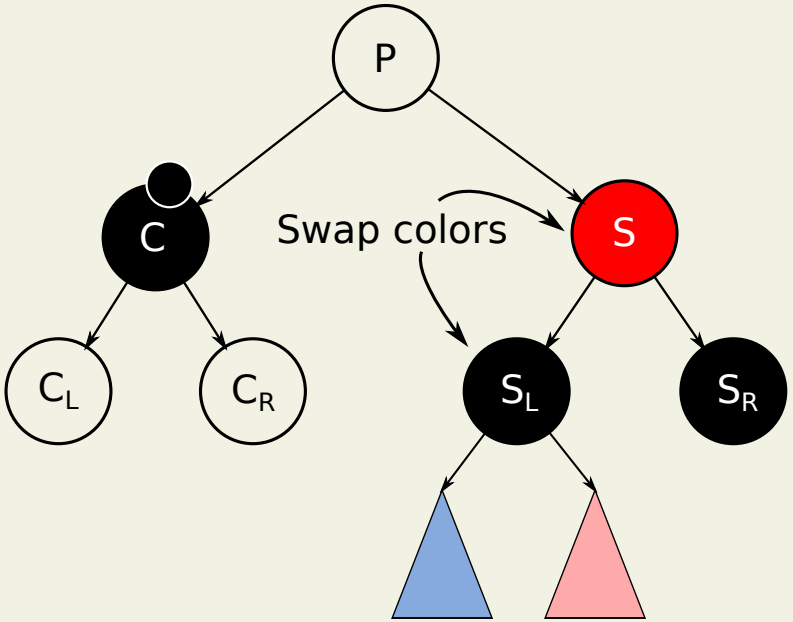
Removing the token from C



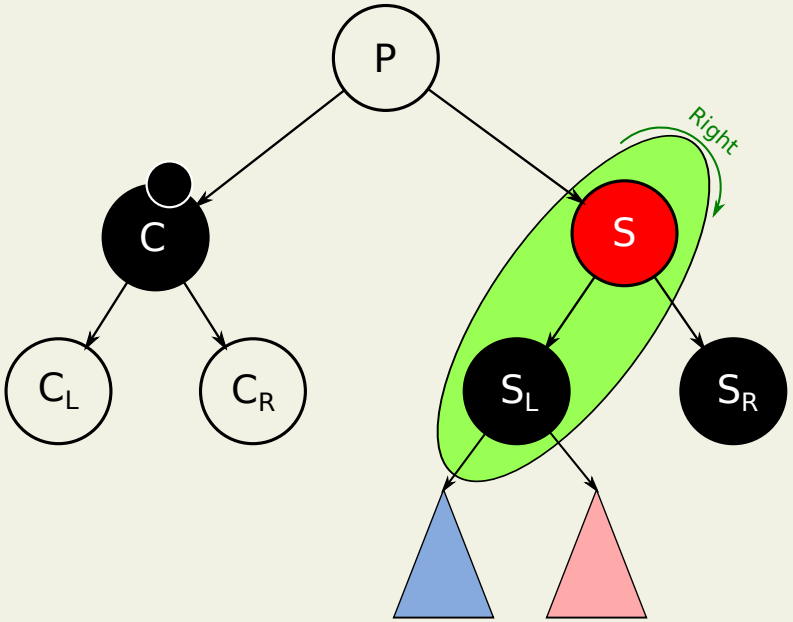
Removing the token from C



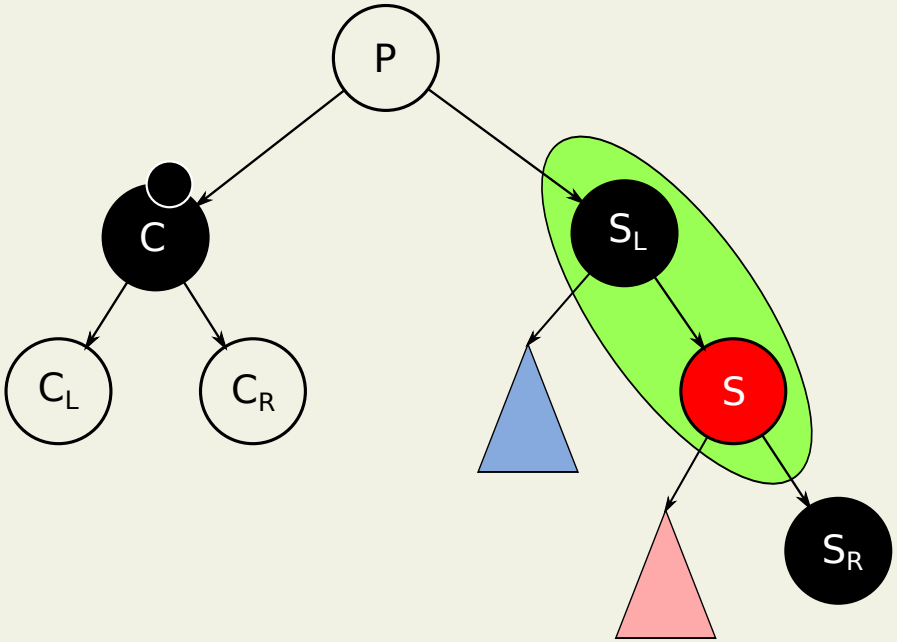
Removing the token from C



Removing the token from C



Removing the token from C



Removing the token from C

Case 4

Sibling S is black.

Right child S_R is red.

Removing the token from C

Case 4

Sibling S is black.

Right child S_R is red.

CLRS:

"*By making some color changes and performing a left rotation on p[x], we can remove the extra black on x*"

Removing the token from C

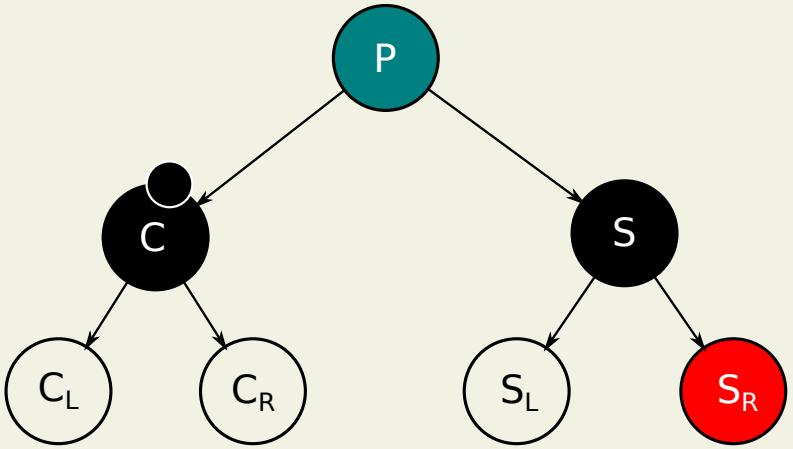
Case 4

Sibling S is black.
Right child S_R is red.

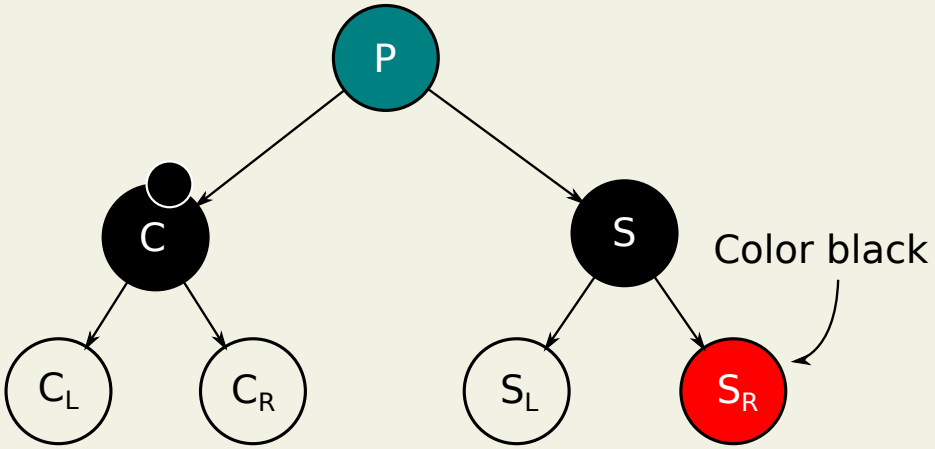
Resolution:

- Color S_R black.
- S inherits the color of parent P.
- Color P black.
- Rotate left at P.
- Set token on root.

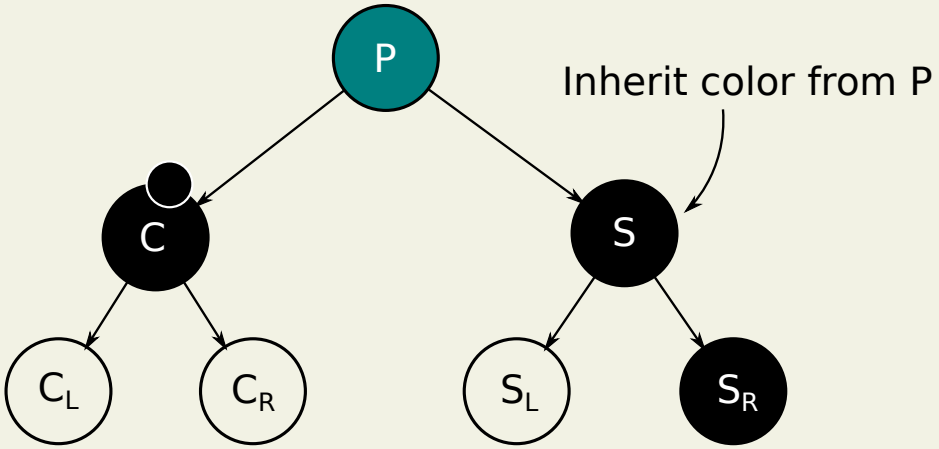
Removing the token from C



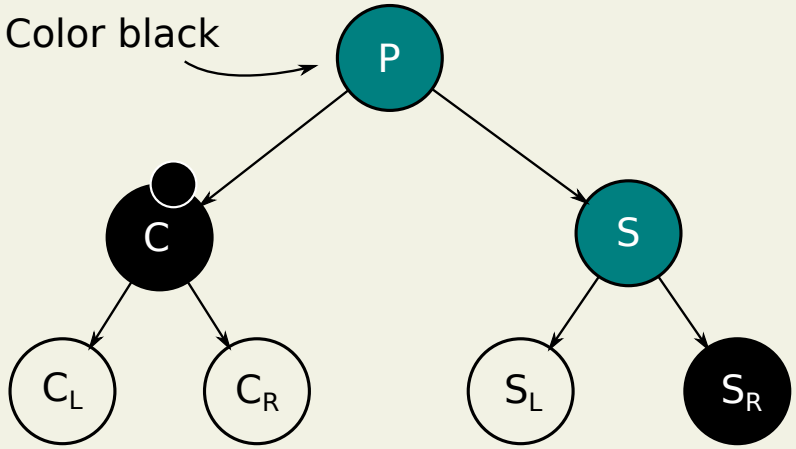
Removing the token from C



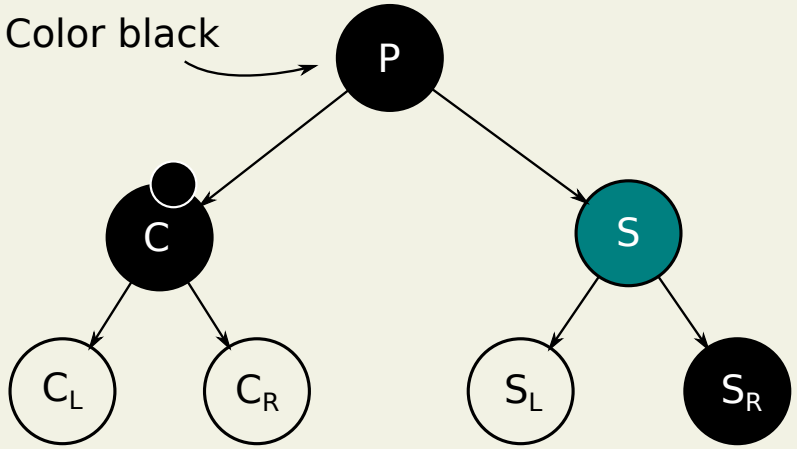
Removing the token from C



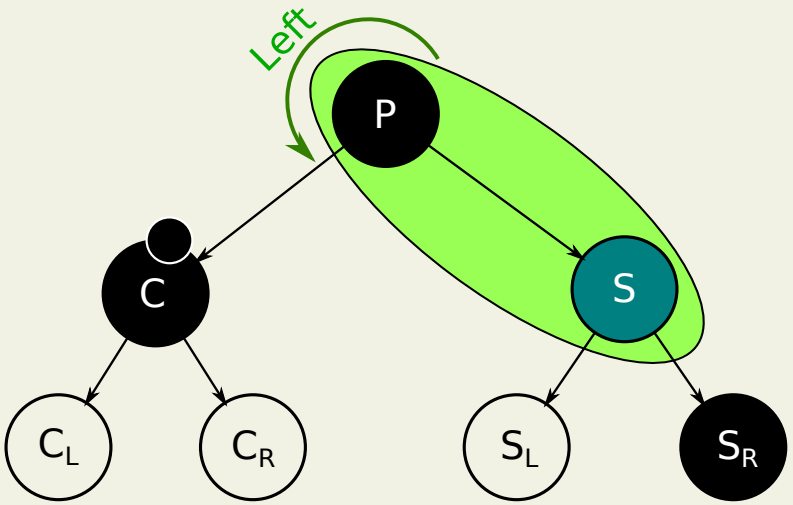
Removing the token from C



Removing the token from C



Removing the token from C



Removing the token from C