

Red-Black Trees II

Red-Black Properties

A binary search tree is a red-black tree if it satisfies the following red-black properties:

1. Every node is either red or black
2. The root is black
3. Every leaf (Nil) is black
4. If a node is red, then both its children are black
5. For each node, all paths from the node to descendant leaves contain the same number of black nodes

RB-Deletion

RB-Delete also first runs the regular Deletion algorithm, and then fixes violations of the RB properties

RB-Delete(T, z)

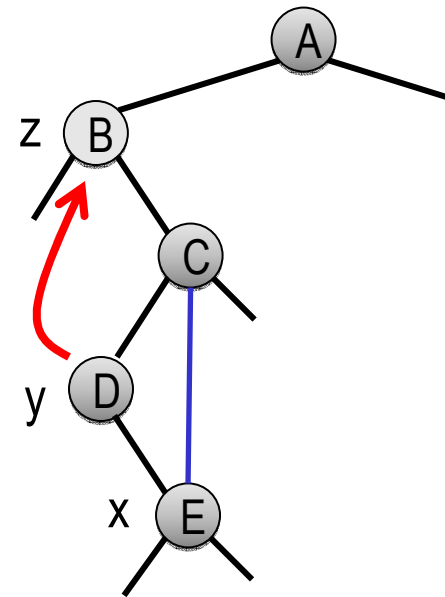
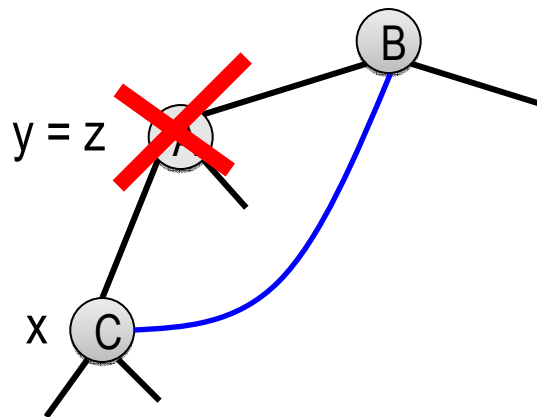
```

if left[z]=Nil[T] or right[z]=Nil[T] then set y:=z
    else set y:=Tree-Successor(z)
if left[y]≠Nil[T] then set x:=left[y]
    else set x:=right[y]
set parent[x]:=parent[y]
if parent[y]=Nil[T] then set root[T]:=x
    else if y=left[parent[y]]:=x then left[parent[y]]:=x
        else right[parent[y]]:=x

if y≠z then do
    set key[z]:=key[y]
    copy y's data into z
if color[y]=BLACK then  RB-Delete-FixUp(T,x)
return y

```

Deletion: Cases



RB-properties violations:

- if y is red, then no violations
- if y is black, and is the root, then prop 2 may be violated
- prop. 4 is violated if the parent of y and x are both red
- prop. 5 is violated, as all paths containing y now contain fewer black nodes

Fixing the Properties

If property 2 is violated, we just color x black

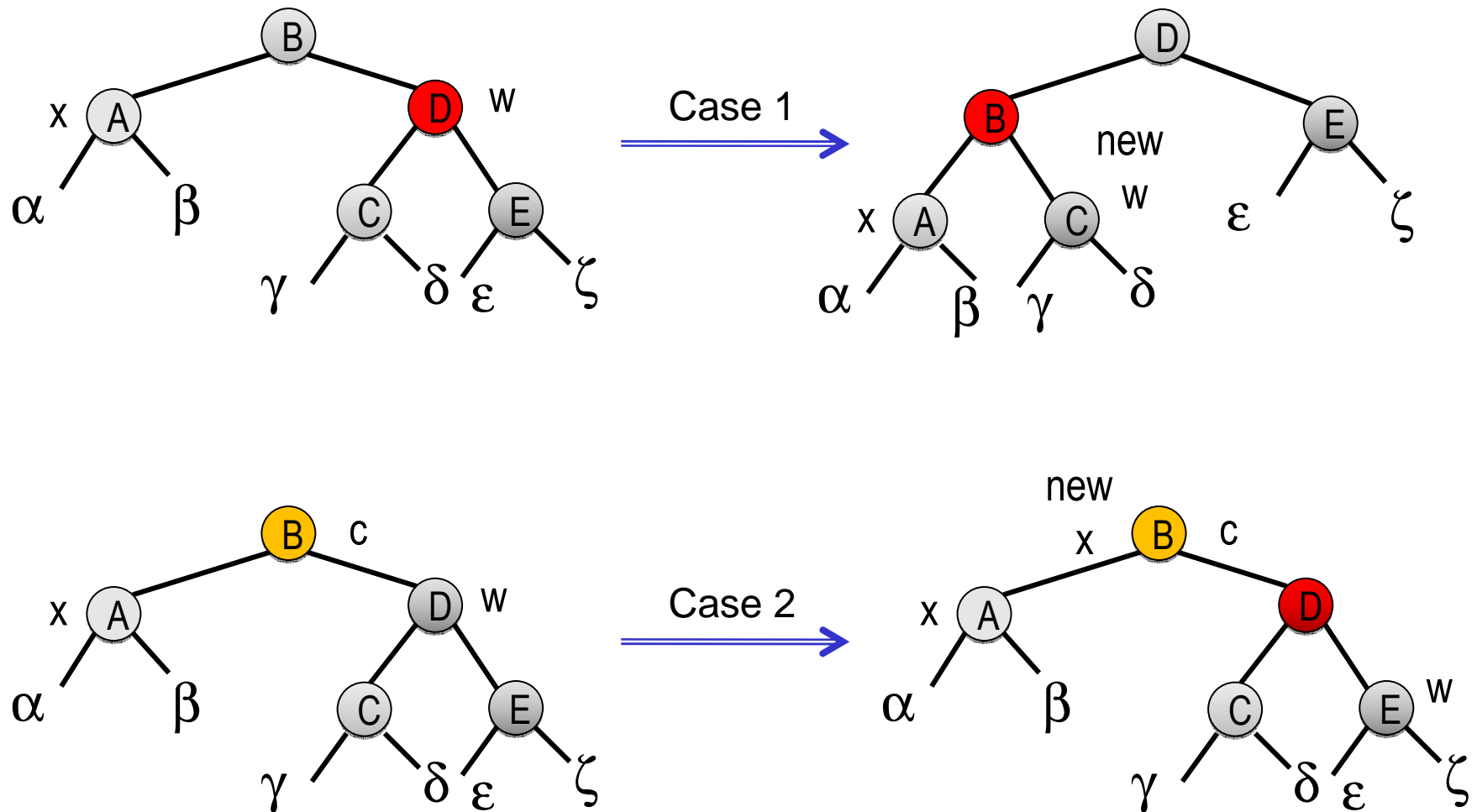
To restore property 5 we put an extra black color onto x

It is now either red-black, if its true color is red (black is fictitious)
or double black, if its true color is black

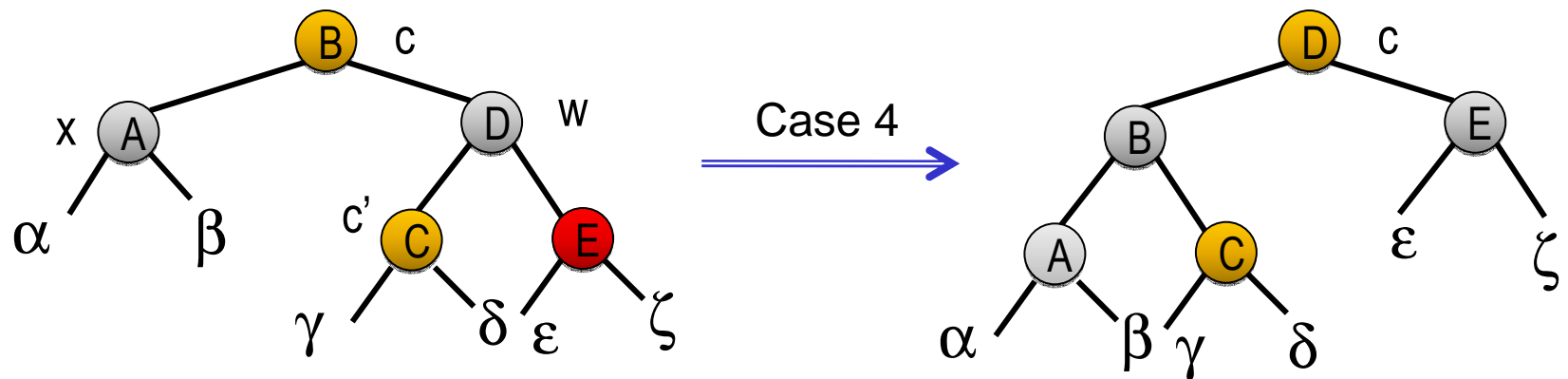
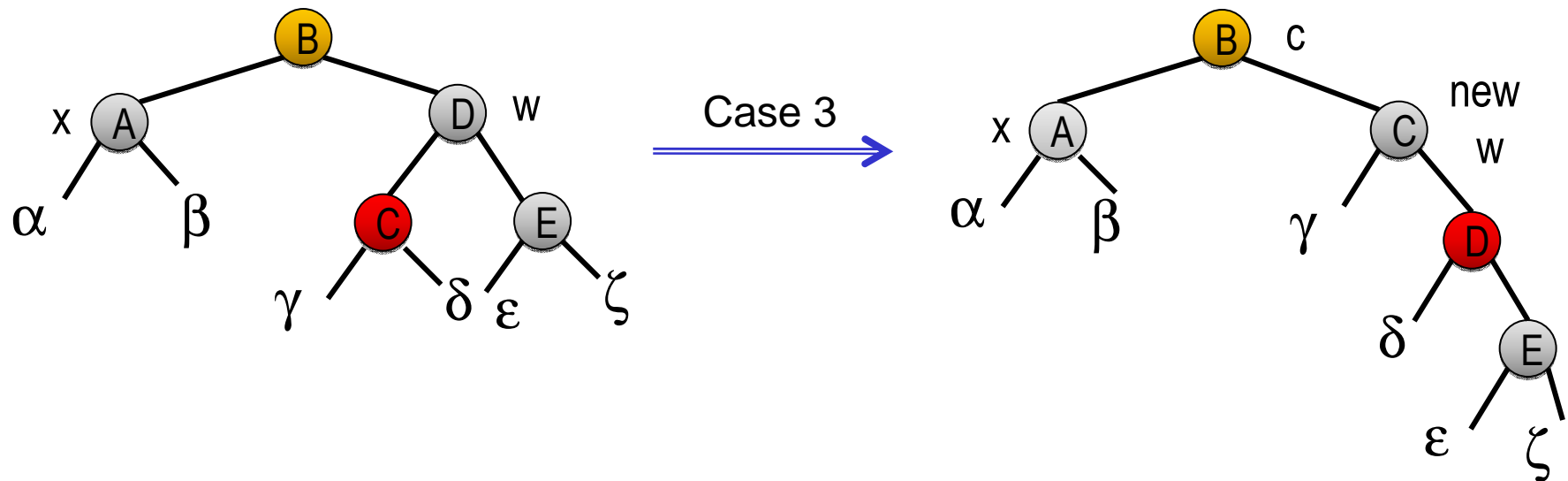
If x is red-black, we color x black

Otherwise we need to find a node to transfer the extra black token
from x

Deletion: Pictures



Deletion: Pictures



new $x = \text{root}[T]$

RB-Delete-FixUp

RB-Delete-FixUp(T,x)

while $x \neq \text{root}[T]$ and $\text{color}[x] = \text{BLACK}$ do

 if $x = \text{left}[\text{parent}[x]]$ then do

 set $w := \text{right}[\text{parent}[x]]$

 if $\text{color}[w] = \text{RED}$ then do

 set $\text{color}[w] := \text{BLACK}$ $\text{color}[\text{parent}[x]] := \text{RED}$

 Left-Rotate(T, $\text{parent}[x]$)

 set $w := \text{right}[\text{parent}[x]]$

 if $\text{color}[\text{left}[w]] = \text{BLACK}$ and $\text{color}[\text{right}[w]] = \text{BLACK}$

 then

 set $\text{color}[w] := \text{RED}$ $x := \text{parent}[x]$

Case 1

Case 2

RB-Delete-FixUp

RB-Delete-FixUp(T,x)

Case 3 { else if color[right[w]]=BLACK then do
 set color[left[w]]:=BLACK color[w]:=RED
 Right-Rotate(T,w) set w:=right[parent[x]]
Case 4 { set color[w]:=color[parent[x]]
 set color[parent[x]]:=BLACK
 set color[right[w]]:=BLACK
 Left-Rotate(T,parent[x]) set x:=root[T]
else (same as then flipping left and right)
set color[x]:=BLACK