

Árboles Binarios de Búsqueda Equilibrados (AVL)

Programación II

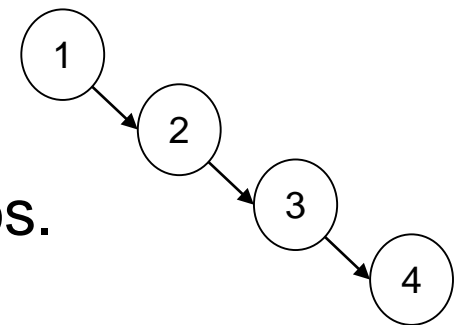
Facultade de Informática



UNIVERSIDADE DA CORUÑA

Árboles Equilibrados AVL

- Definición: Es un árbol binario de búsqueda en el que, para cada nodo, se cumple que la diferencia de altura de sus subárboles nunca es mayor que uno.
- Características
 - Se denominan AVL en honor a Adelson, Velskii y Landis, que fueron los primeros en proponer este TAD.
 - Hacen la búsqueda eficiente manteniendo una altura mínima evitando así los árboles degenerados.

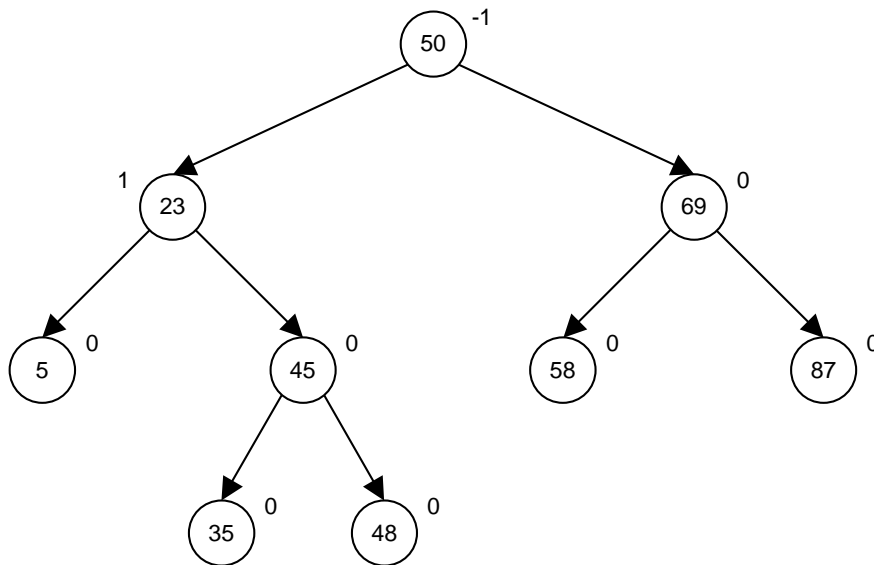


Árboles Equilibrados AVL

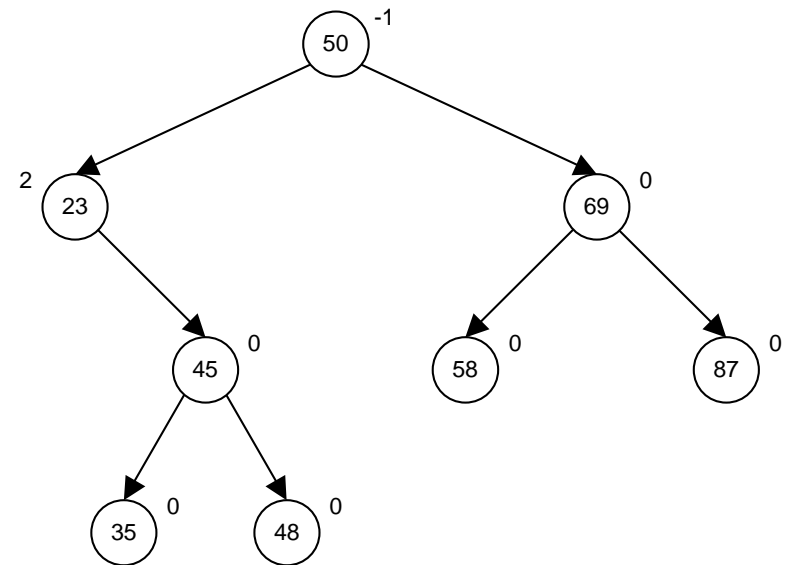
- Factor de equilibrio
 - El factor de equilibrio (*balance factor*) de un nodo se define como la altura de su subárbol derecho menos la altura de su subárbol izquierdo.
 - $bf(N) = h_{NDch} - h_{NIzq}$
- Definición en base al factor de equilibrio
 - Un árbol AVL es un ABB en el que el factor de equilibrio de cada nodo está en el intervalo $[-1, 1]$. Se dice entonces que el árbol está equilibrado.

Árboles Equilibrados AVL

- Ejemplos



Árbol equilibrado



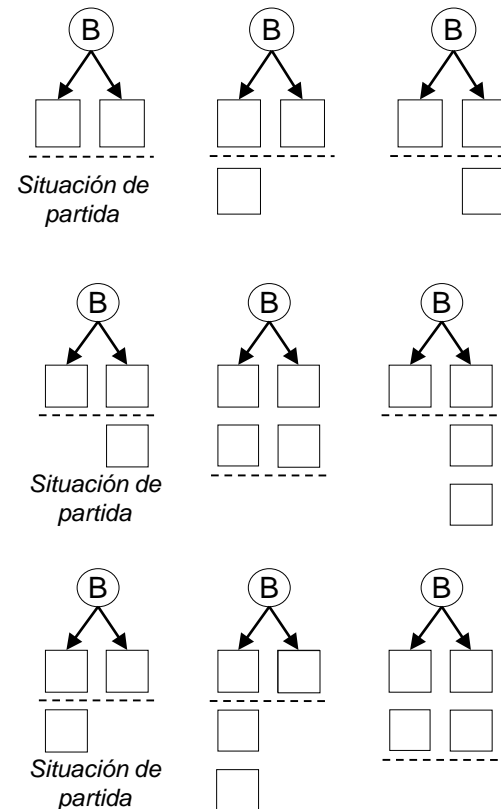
Árbol no equilibrado

Especificación del TAD AVL

- Es idéntica a la del árbol binario de búsqueda, sólo cambian internamente las funciones de inserción y borrado que, además de insertar y borrar, se encargan de mantener equilibrada la estructura del árbol.

Inserción en árboles AVL

- Si el árbol está perfectamente equilibrado una inserción no rompe el equilibrio
- Si el árbol no está perfectamente equilibrado una inserción puede romper el equilibrio o restituirlo



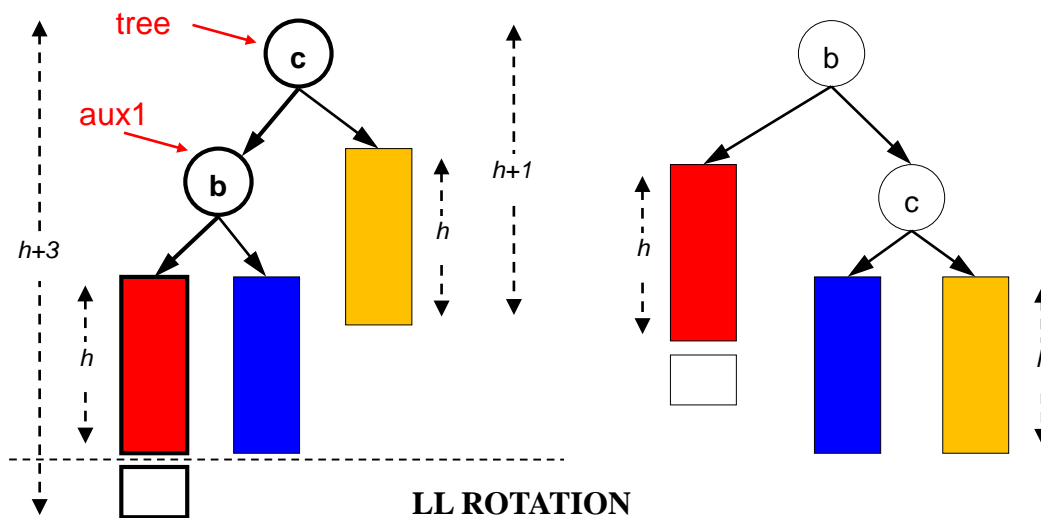
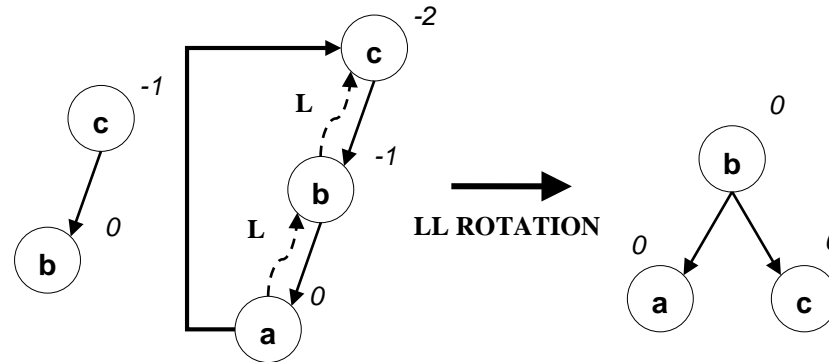
Rotaciones para restaurar el equilibrio

- Rotaciones simples
 - Son aquellas que involucran a dos nodos
 - Tenemos la rotación left-left (LL) y la rotación right-right (RR)
- Rotaciones complejas
 - Son aquellas que involucran a tres nodos
 - Tenemos la rotación right-left (RL) y la rotación left-right (LR)

Notación para implementar las rotaciones

- `tree` apunta al nodo no equilibrado
- `aux1` apunta al segundo nodo implicado en la rotación
- `aux2` apunta al tercer nodo implicado en la rotación (si es una rotación compuesta)

LL Rotation



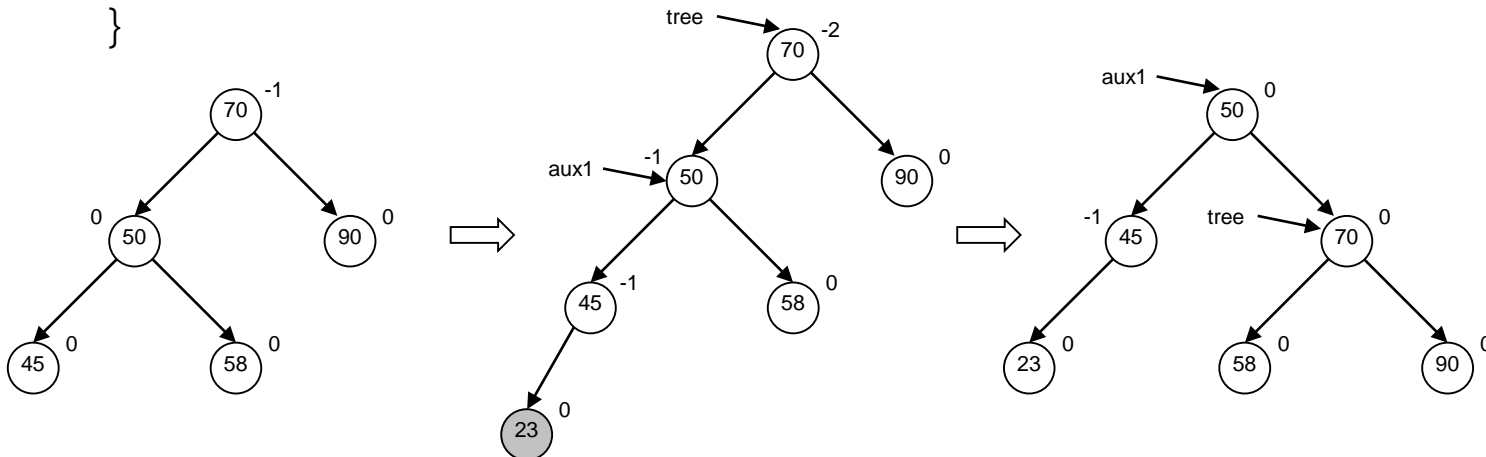
LL Rotation

- Cambios en bf

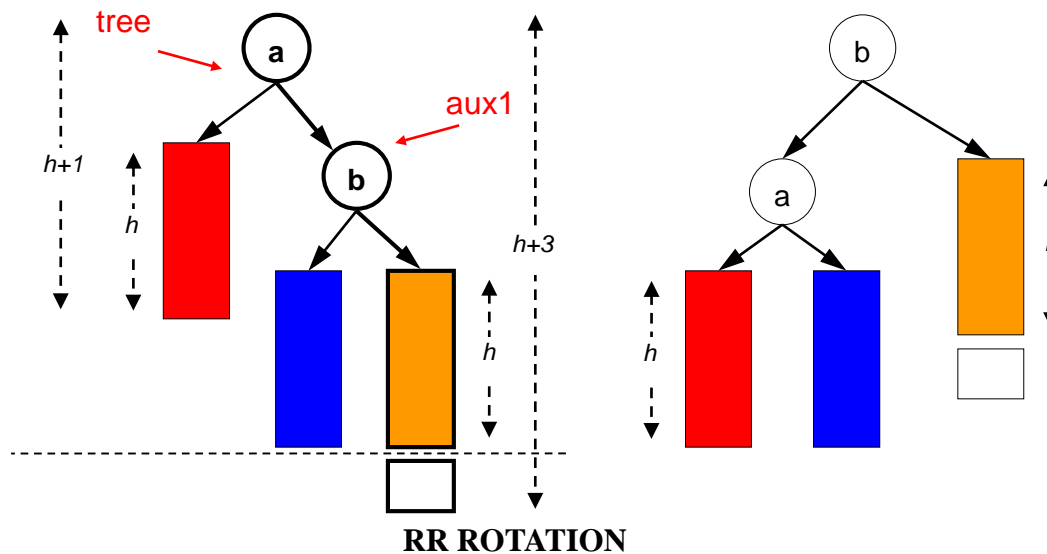
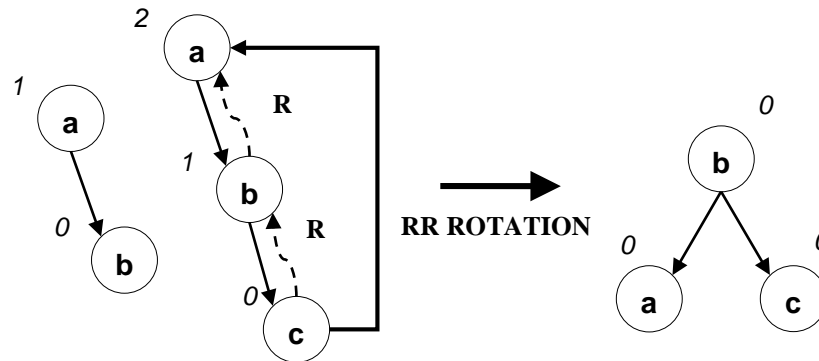
```
if (aux1->bf == 0) {  
    (*tree)->bf = -1;  
    aux1->bf = 1;  
}  
else { // actually: aux1->bf == -1  
    (*tree)->bf = 0;  
    aux1->bf = 0;  
}
```

- Cambios en punteros

```
(*tree)->left = aux1->right;  
aux1->right = *tree;  
*tree = aux1;
```



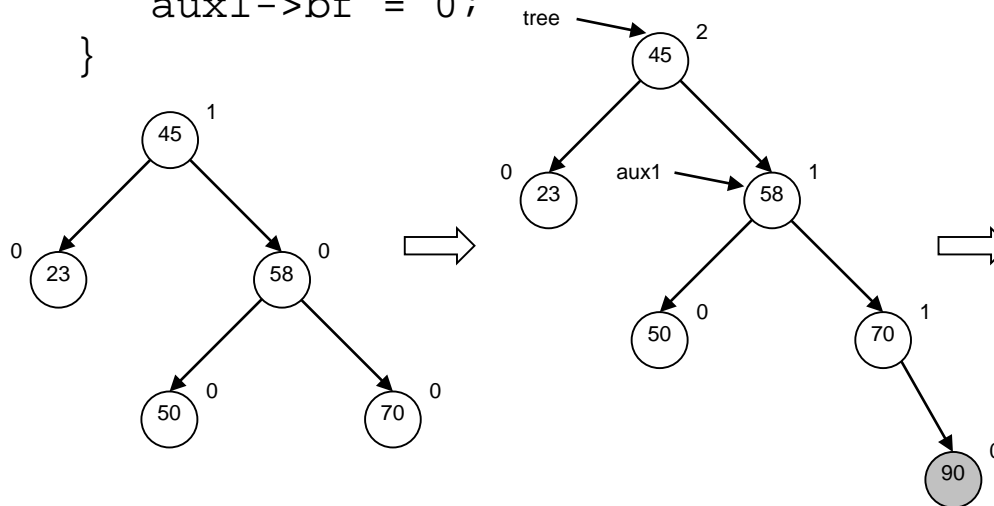
RR Rotation



RR Rotation

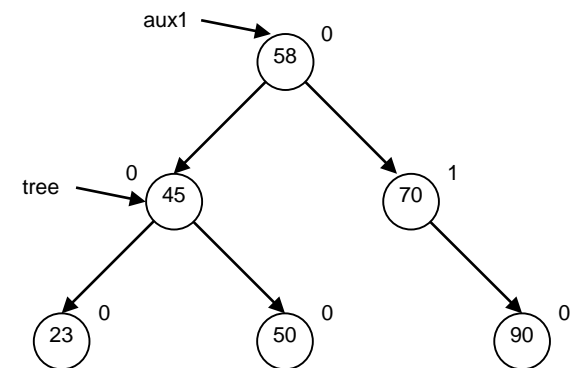
- Cambios en bf

```
if (aux1->bf == 0) {  
    (*tree)->bf = 1;  
    aux1->bf = -1;  
}  
else { // actually: aux1->bf == 1  
    (*tree)->bf = 0;  
    aux1->bf = 0;  
}
```

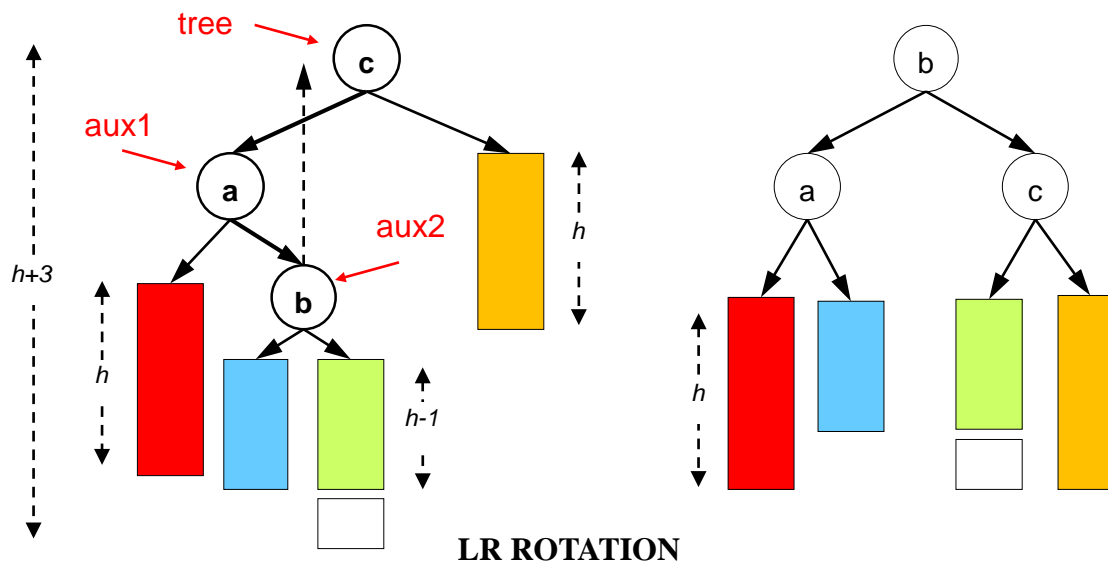
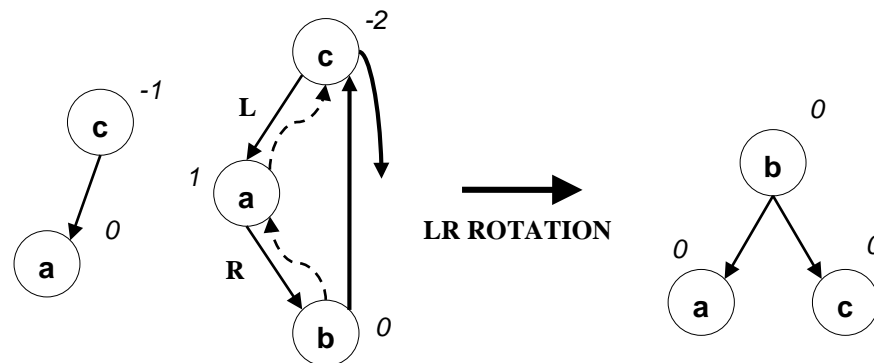


- Cambios en punteros

```
(*tree)->right = aux1->left;  
aux1->left = *tree;  
*tree = aux1;
```



LR Rotation



LR Rotation

- Cambios en los punteros

1. `aux2 = aux1->right`

3. `aux1->right = aux2->left`

5. `aux2->left = aux1`

2. `(*tree)->left = aux2->right`

4. `aux2->right = (*tree)`

6. `(*tree) = aux2`

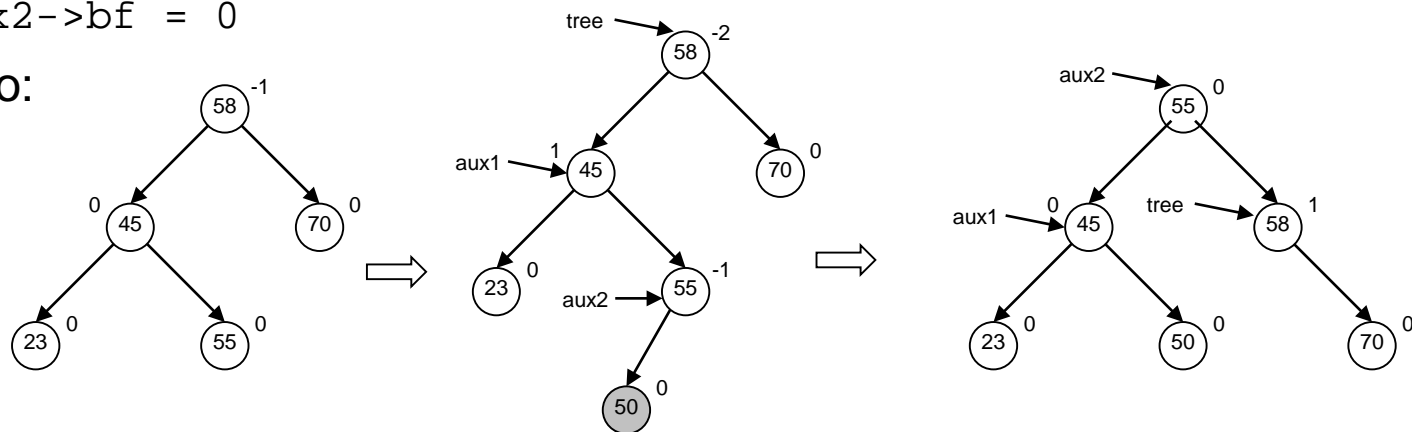
- Cambios en los bf

- SI `aux2->bf = 1` \Rightarrow `aux1->bf = -1` SINO `aux1->bf = 0`

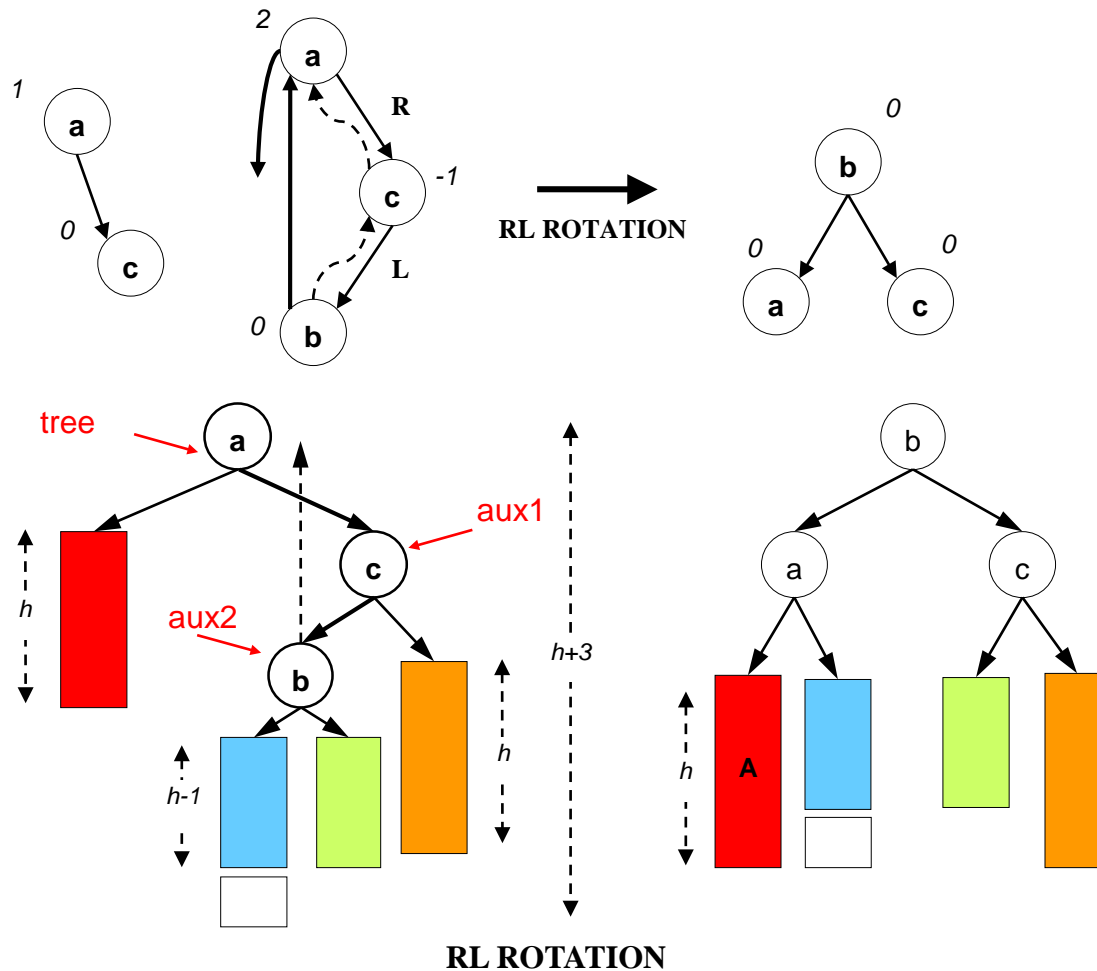
- SI `aux2->bf = -1` \Rightarrow `(*tree)->bf = 1` SINO `(*tree)->bf = 0`

- `aux2->bf = 0`

- Ejemplo:



RL Rotation



RL Rotation

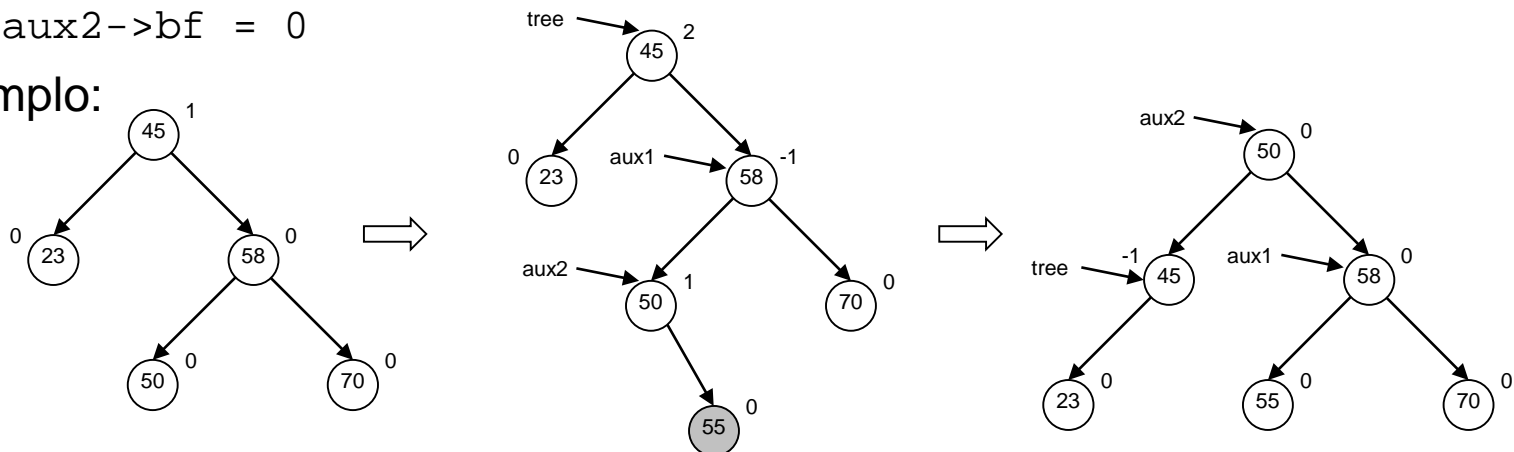
- Cambios a realizar en los punteros

1. `aux2 = aux1->left`
2. `(*tree)->right = aux2->left`
3. `aux1->left = aux2->right`
4. `aux2->left = (*tree)`
5. `aux2->right = aux1`
6. `(*tree) = aux2`

- Cambios en los bf

- SI `aux2->bf = 1` \Rightarrow `(*tree)->bf = -1` SINO `(*tree)->bf = 0`
- SI `aux2->bf = -1` \Rightarrow `aux1->bf = 1` SINO `aux1->bf = 0`
- `aux2->bf = 0`

- Ejemplo:

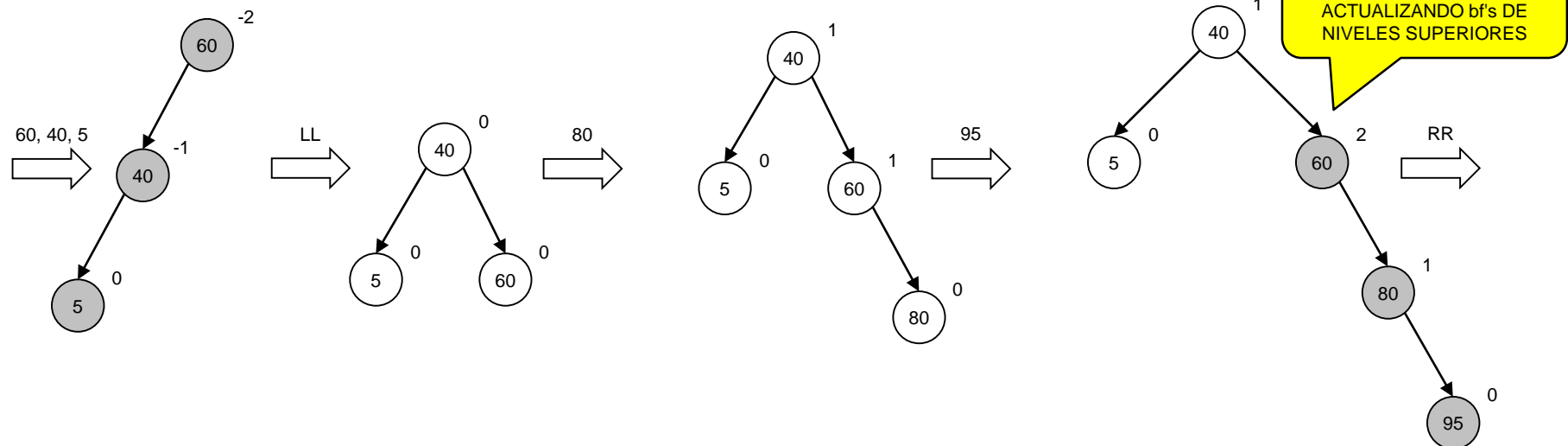


Inserción en Árboles AVL: Pseudocódigo

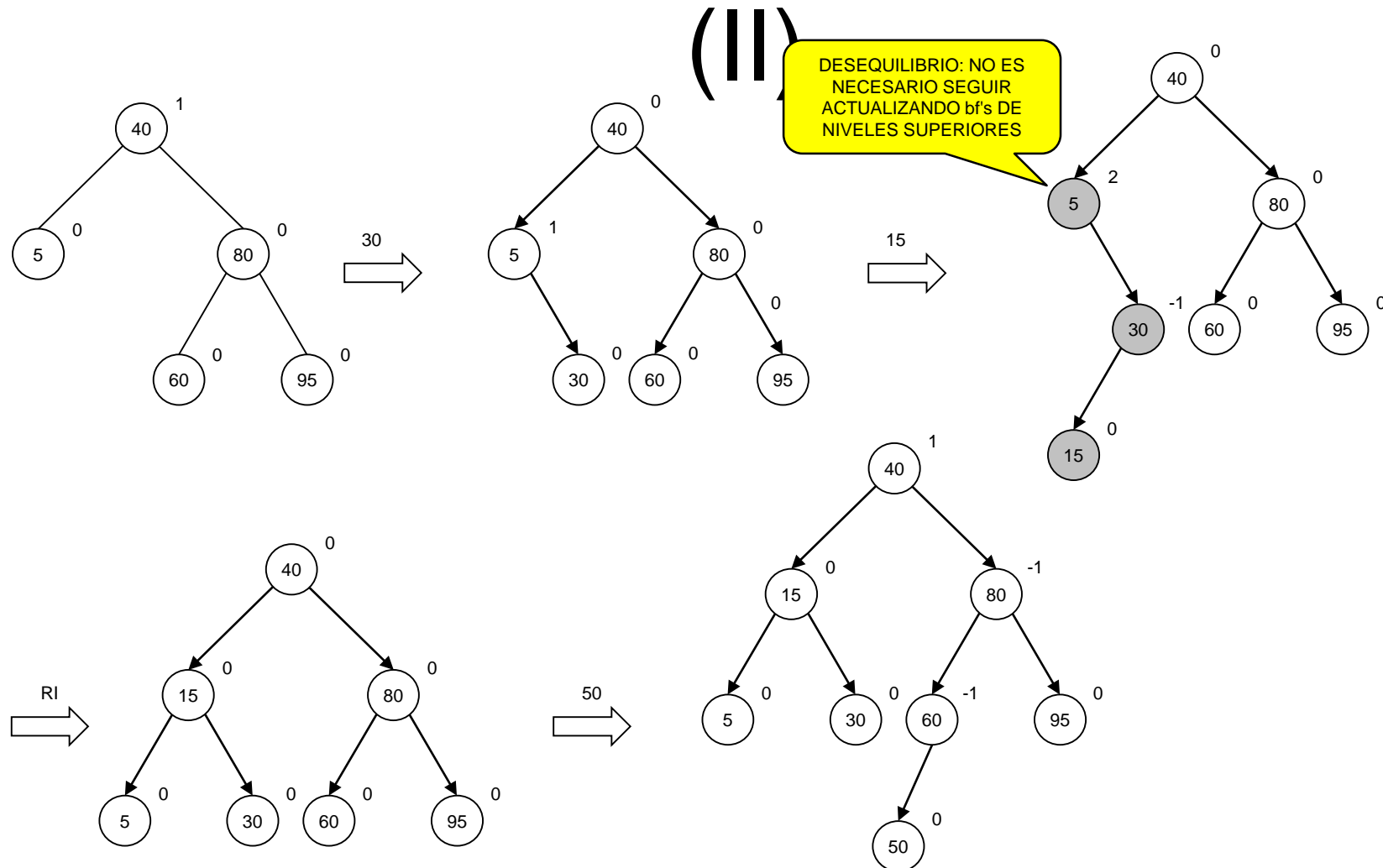
- Insertar como una hoja el nuevo nodo (proceso similar a ABB).
- El factor de equilibrio del nuevo nodo hoja es 0.
- Recorrer en sentido inverso el camino de búsqueda desde el lugar de inserción (camino de búsqueda). En cada nodo visitado recalcular el factor de equilibrio. El proceso termina cuando se encuentra un nodo desequilibrado o hasta alcanzar un nodo cuya altura no ha sido modificada.
 - Si hay un nodo desequilibrado se determina el valor de los punteros que hemos denominado `tree` (el nodo desequilibrado), `aux1` y `aux2` (los dos nodos siguientes a `tree` en el camino de búsqueda)
 - Según el `bf` de estos nodos se determina la rotación a realizar
 - Se realiza la rotación y se acaba el proceso (en las inserciones una única rotación garantiza que el árbol este equilibrado)

Ejemplo de Inserción en AVL (I)

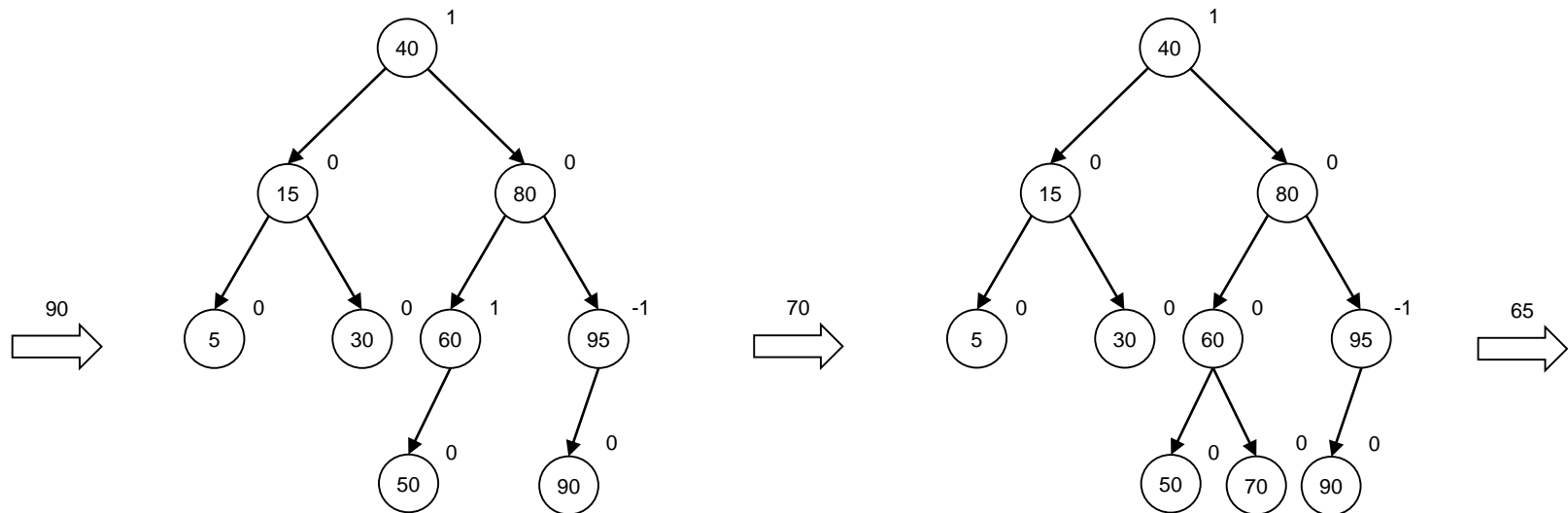
- Insertar 60, 40, 5, 80, 95, 30, 15, 50, 90, 70, 65, 10, 25, 35, 38



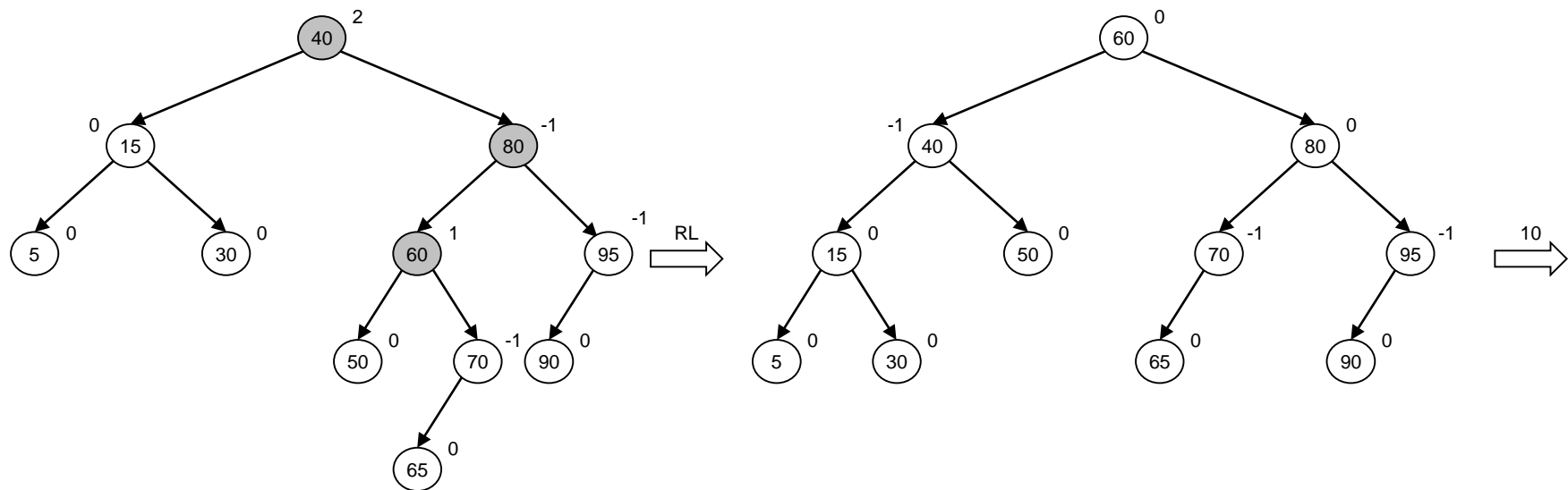
Ejemplo de Inserción en AVL



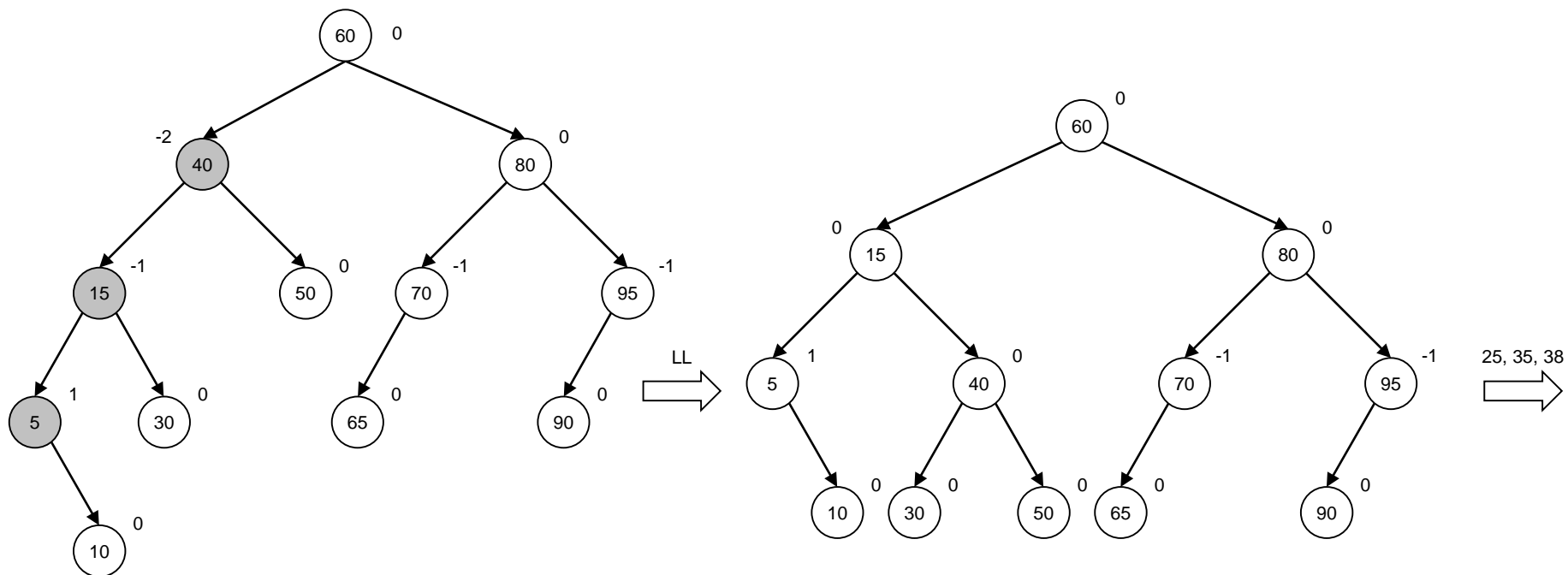
Ejemplo de Inserción en AVL (II)



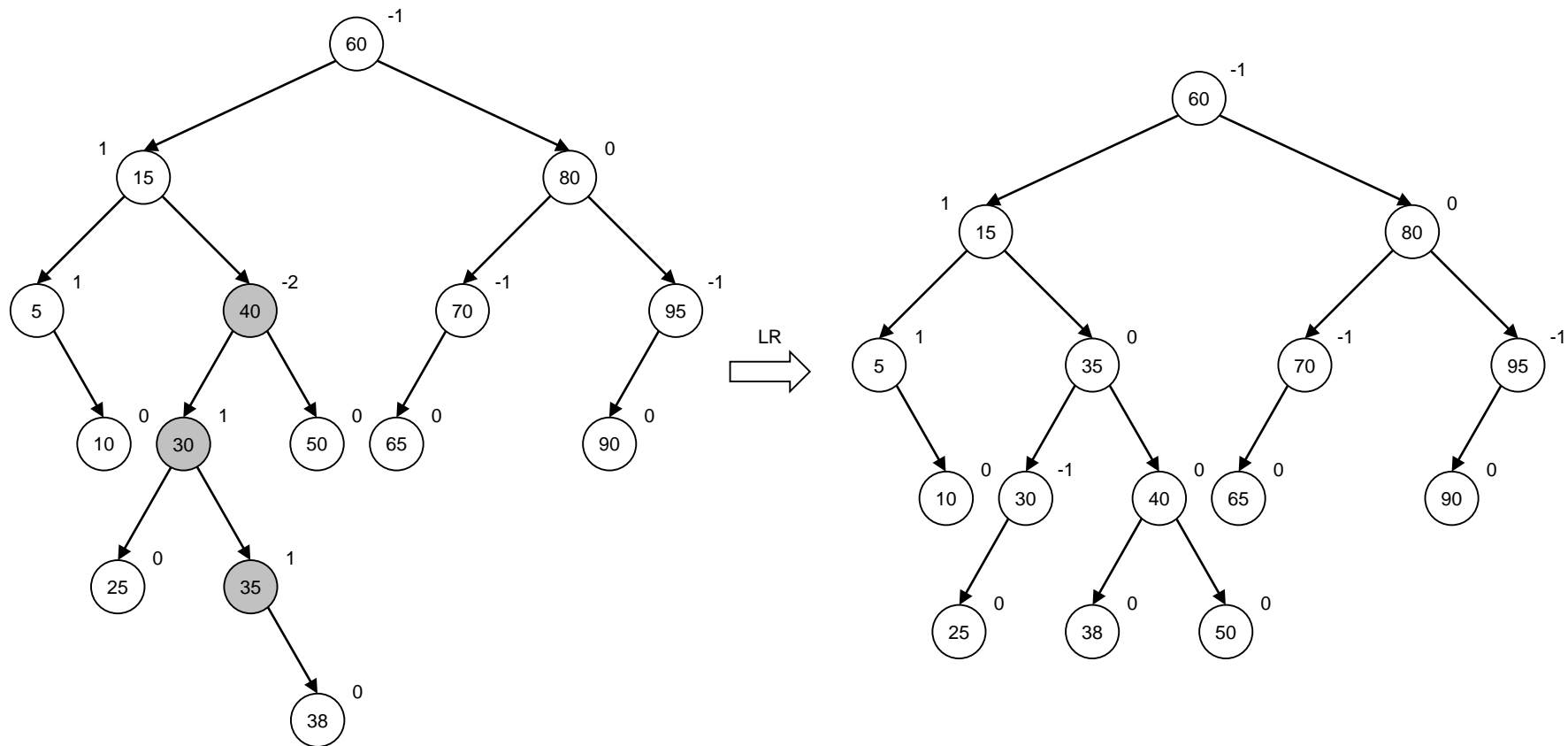
Ejemplo de Inserción en AVL (III)



Ejemplo de Inserción en AVL (IV)

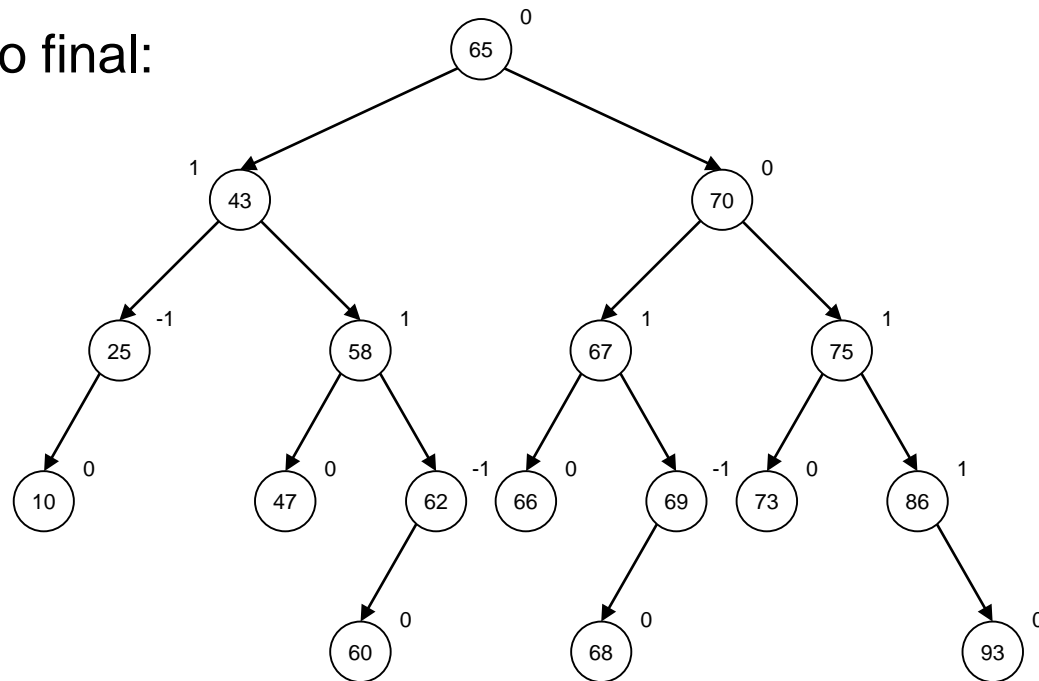


Ejemplo de Inserción en AVL (V)



Ejemplo de Inserción en AVL (VI)

- Insertar 58, 43, 75, 86, 65, 70, 67, 73, 93, 69, 25, 66, 68, 47, 62, 10, 60
- Resultado final:



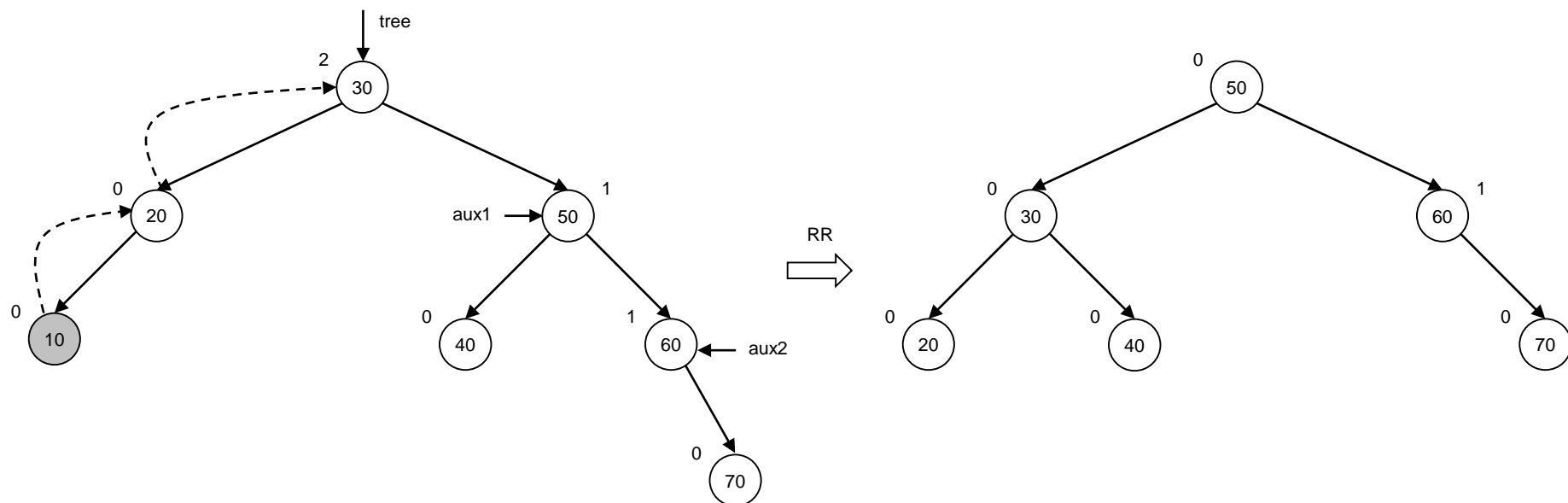
Borrado en Árboles AVL: Pseudocódigo

- Se borra el nodo como en un ABB.
- A continuación es necesario recalcular los bf
 - Se regresa por el camino de búsqueda recalculando los nuevos bf hasta llegar a la raíz. En este caso, el camino de búsqueda es el camino hacia la clave que hay que eliminar más el camino hacia la clave con la que se sustituye (en caso de tener dos descendientes)
 - Si algún nodo tiene un bf no equilibrado debe realizarse una rotación para equilibrar
 - Al contrario que en las inserciones una rotación no garantiza que el árbol haya quedado equilibrado. Si la altura del nodo que queda como raíz local después de la rotación ha disminuido pueden aparecer más desequilibrios en la estructura del árbol

Borrado en Árboles AVL: Pseudocódigo

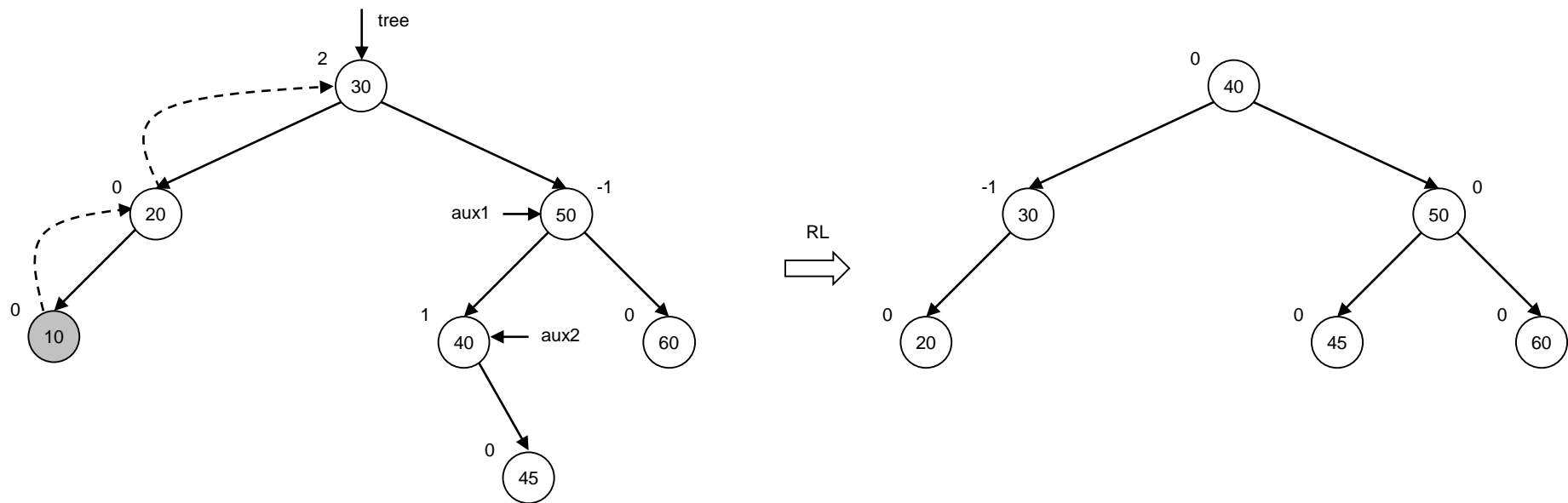
- ¿Cómo determinar los nodos implicados en una rotación?
 - El primer nodo (el apuntado por el puntero `tree`) es siempre el nodo desequilibrado (igual que en la inserción). El proceso para determinar el resto de nodos ya no es idéntico a la inserción.
 - El siguiente nodo involucrado (`aux1`) es el nodo que está bajo el nodo `tree` y por la rama contraria a la que se ha producido el borrado (al borrar por una rama es como si insertáramos en la otra).
 - El nodo `aux2` será la raíz del subárbol de `aux1` que tenga más altura, en caso de empate se buscará el nodo que resulta en una rotación simple (LL o RR)

Rotación RR. Ejemplo*

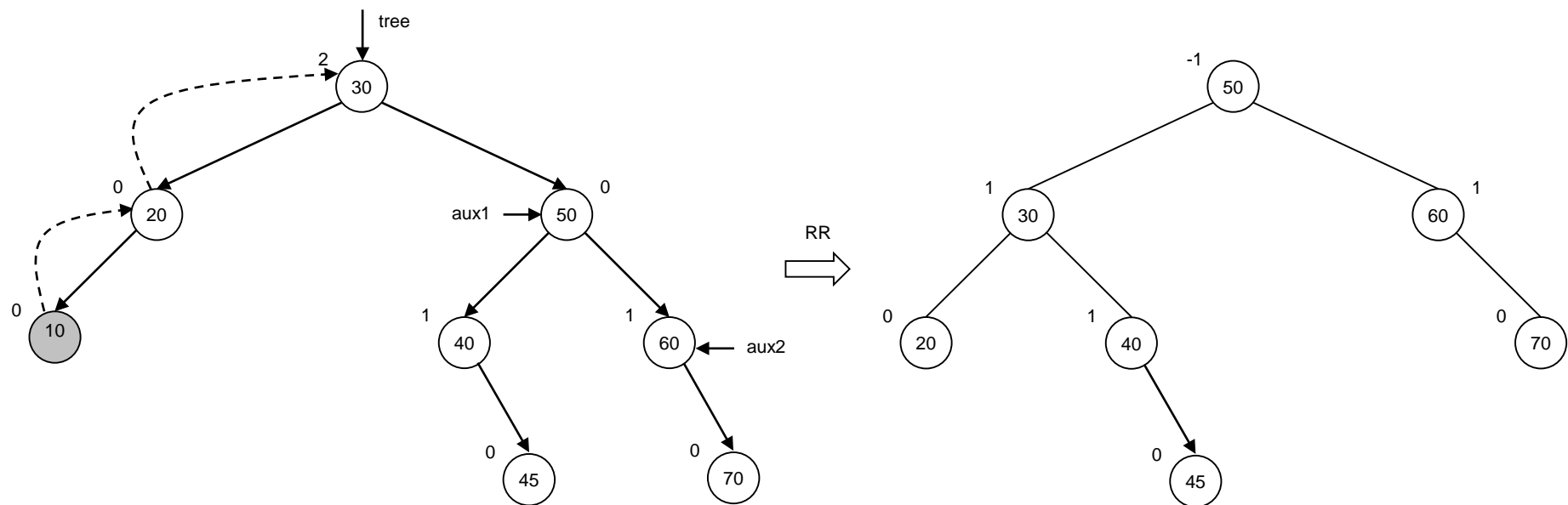


* Se elimina la clave sombreada

Rotación RL. Ejemplo



Rotación RR o RL*. Ejemplo

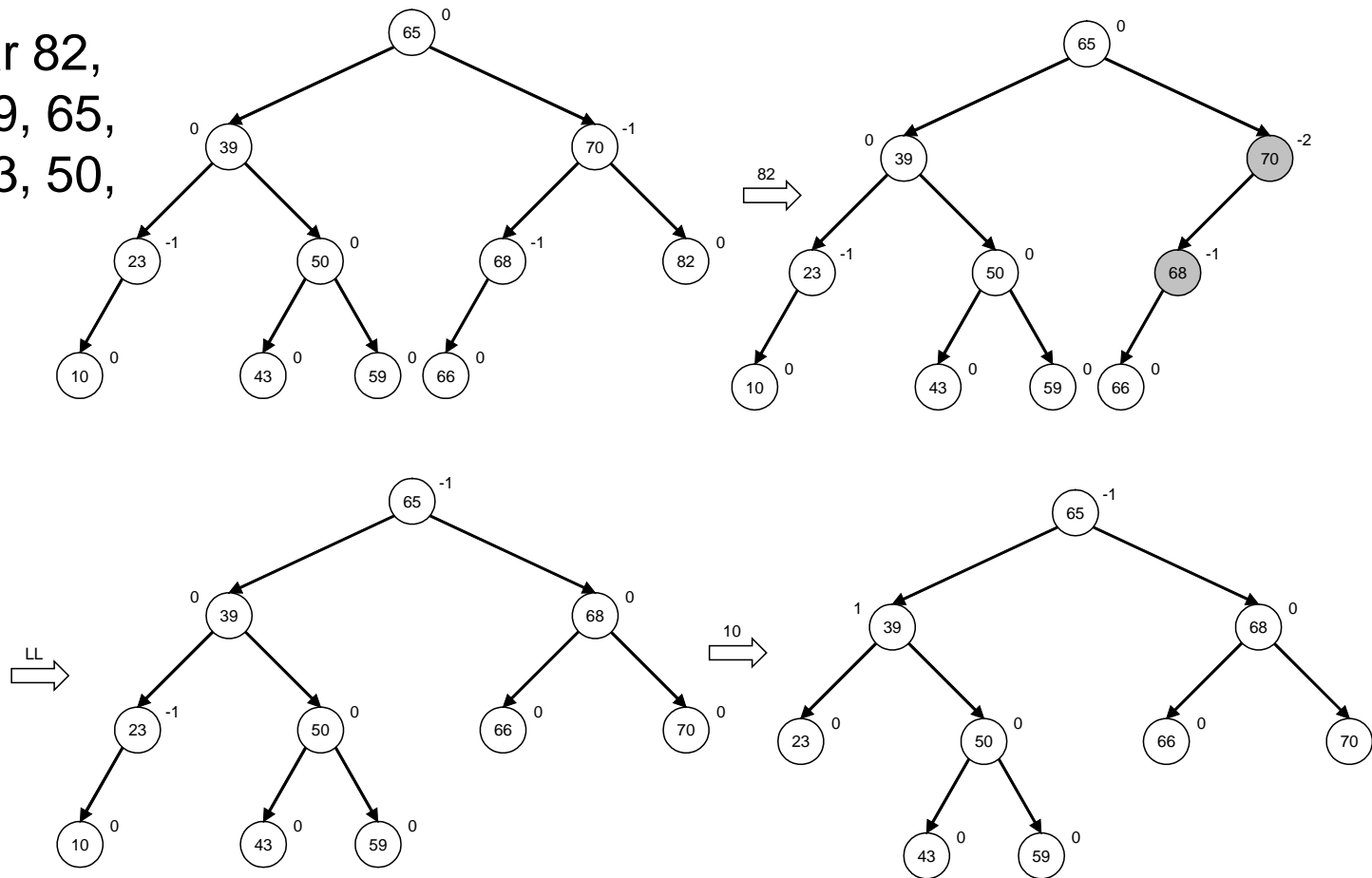


* Caso especial que sólo se da en el borrado.

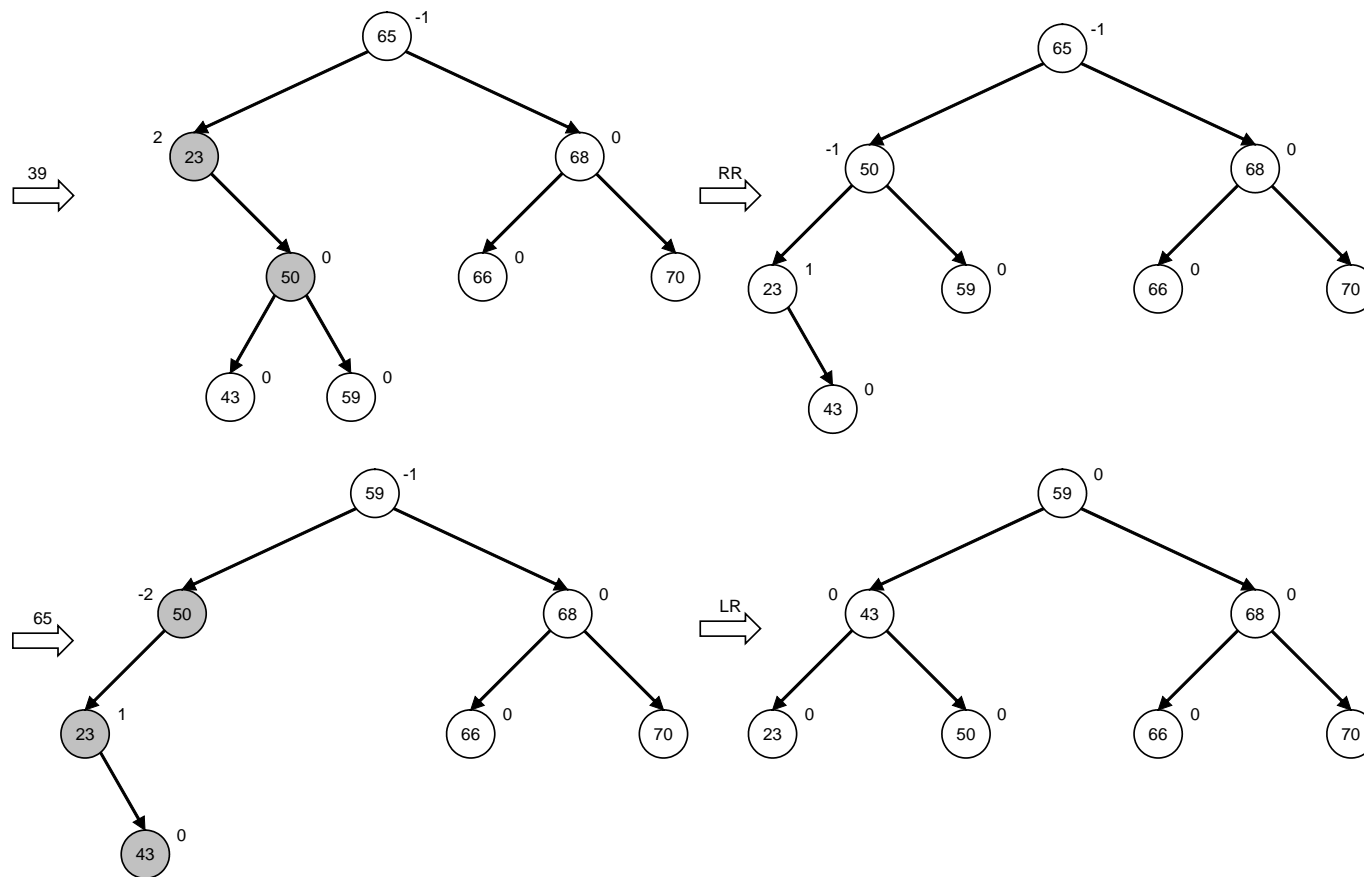
Ejemplo de Borrado en AVL

(I)

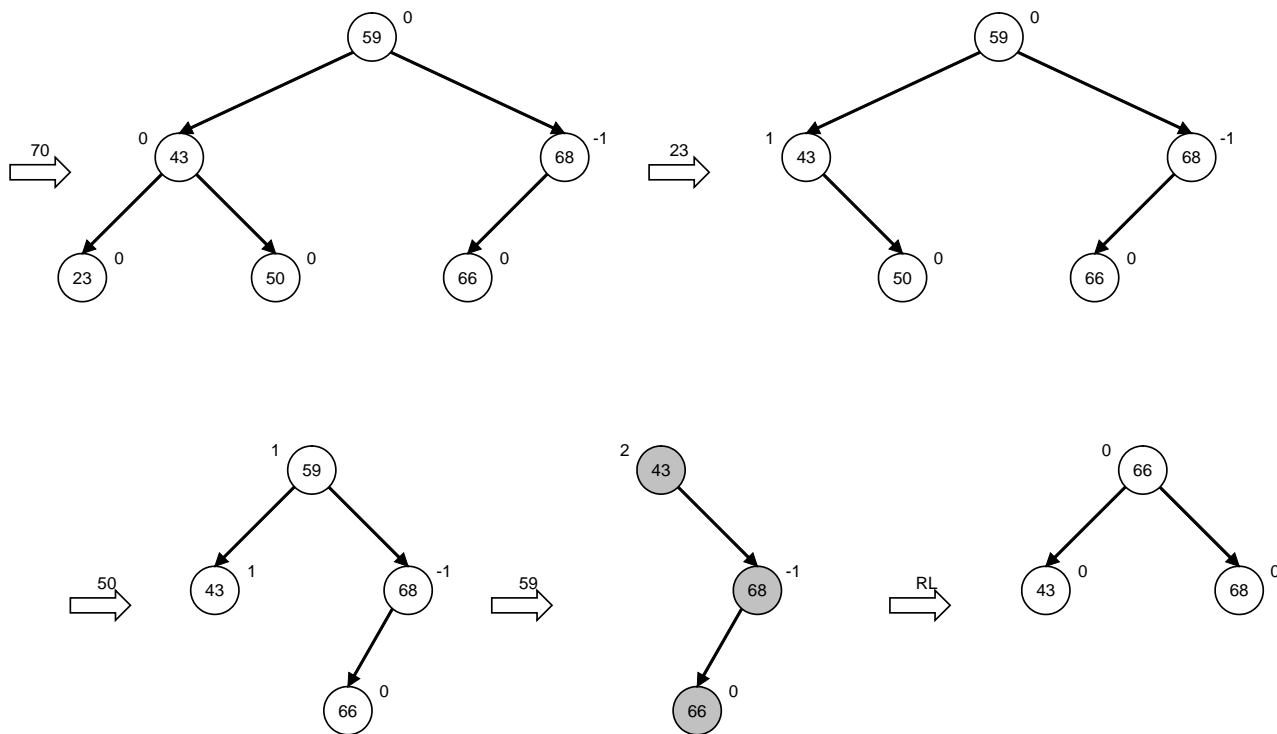
- Borrar 82,
10, 39, 65,
70, 23, 50,
59



Ejemplo de Borrado en AVL (I)

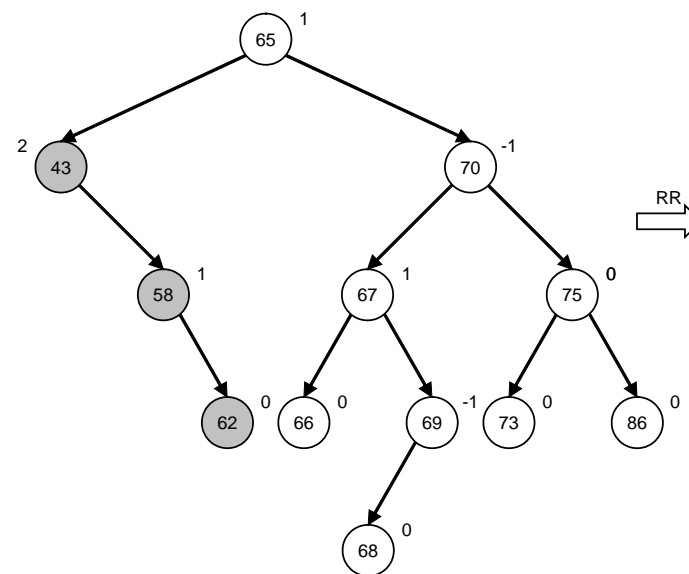
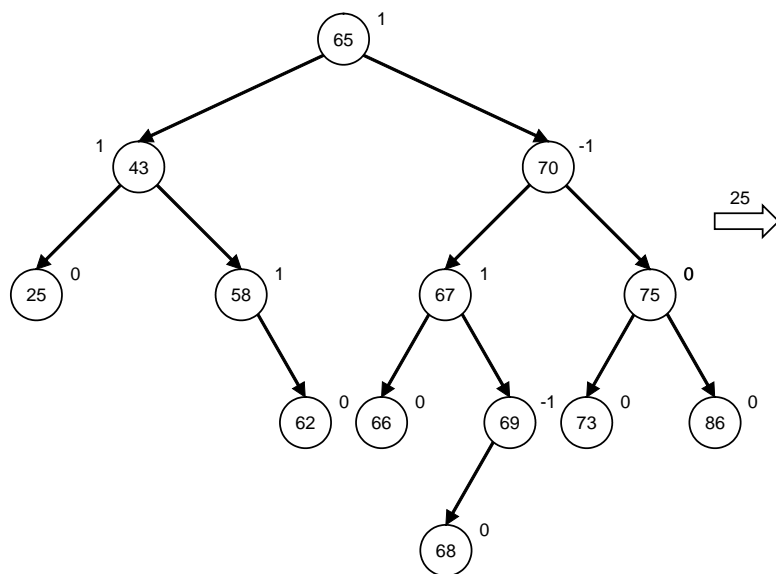


Ejemplo de Borrado en AVL (I)



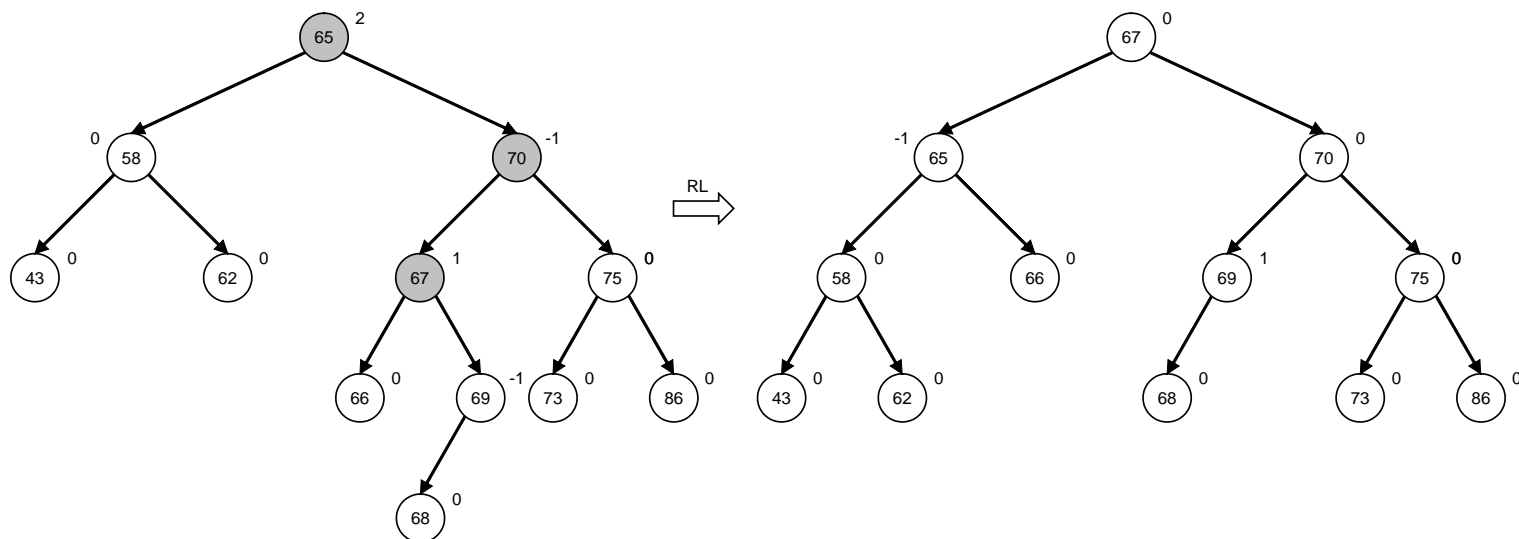
Ejemplo de Borrado en AVL (II)

- Ejemplo con rotaciones consecutivas. Borrar 25



Ejemplo de Borrado en AVL (II)

- Borrar 25 (cont.)



Ejemplo de Borrado en AVL (III)

- Borrar 25, 75, 66, 65, 62, 10, 43, 47

