# Árboles Binarios de Búsqueda Equilibrados (AVL)

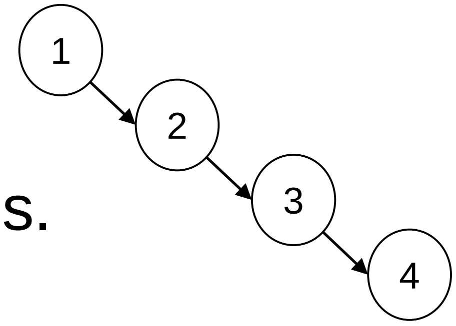
Programación II

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.

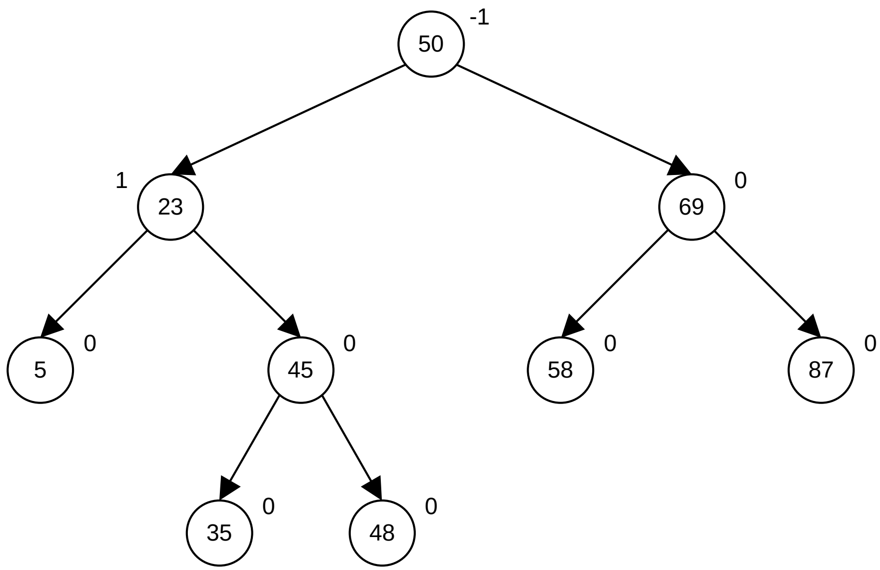


# Árboles Equilibrados AVL

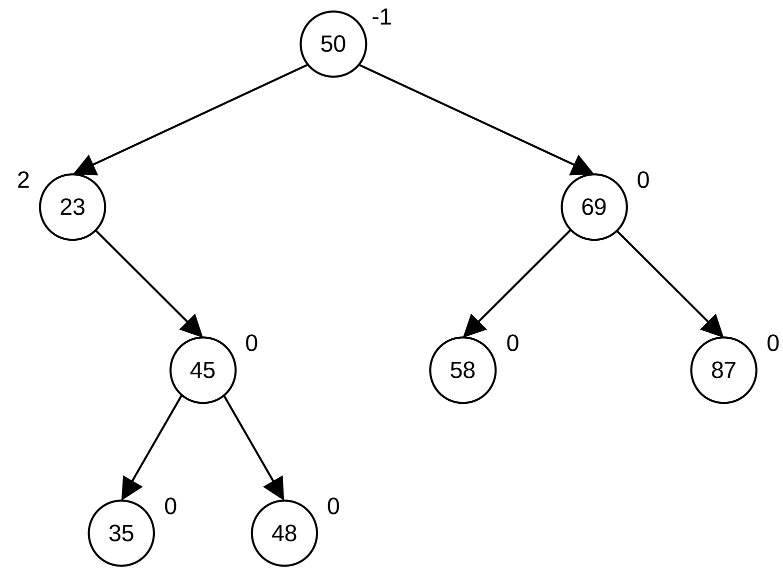
- Factor de equilibrio
  - El factor de equilibrio de un nodo se define como la altura de su subárbol derecho menos la altura de su subárbol izquierdo.
  - $Fe(N)=h_{NDch} - h_{NIzq}$
- Definición en base al factor de equilibrio
  - Un árbol AVL es un ABB en el que el factor de equilibrio de cada nodo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

# Árboles Equilibrados AVL

- Ejemplos




Árbol equilibrado




Árbol no equilibrado

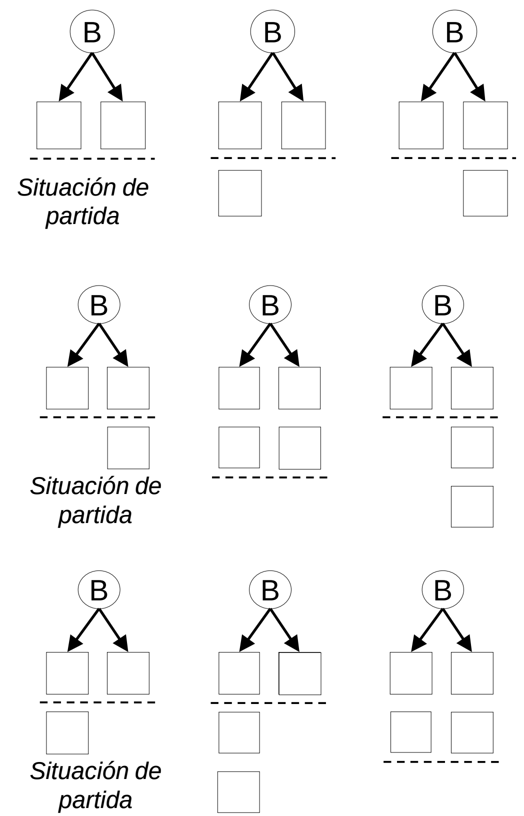
# Especificación del TAD AVL

- 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



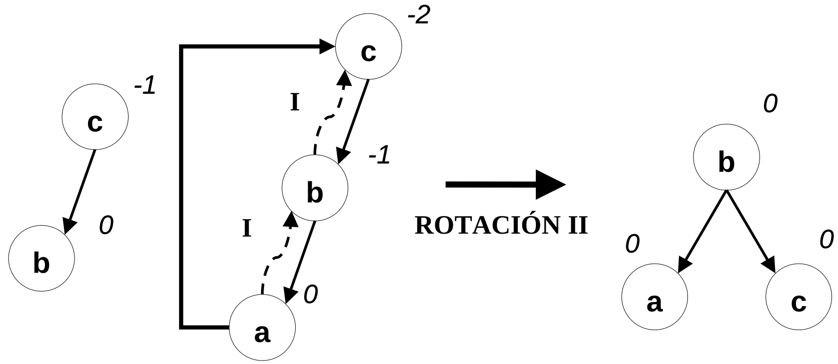
# Rotaciones para restaurar el equilibrio

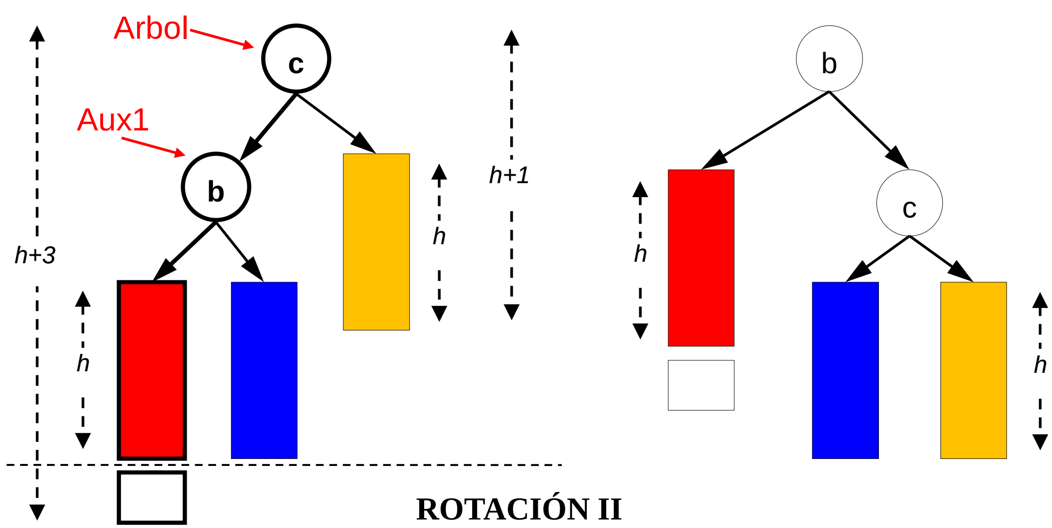
- Rotaciones simples
  - Son aquellas que involucran a dos nodos
  - Tenemos la rotación izquierda-izquierda (II) y la rotación derecha-derecha (DD)
- Rotaciones complejas
  - Son aquellas que involucran a tres nodos
  - Tenemos la rotación derecha-izquierda (DI) y la rotación izquierda-derecha (ID)

# Notación para implementar las rotaciones

- `Arbol` 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)

# Rotación II

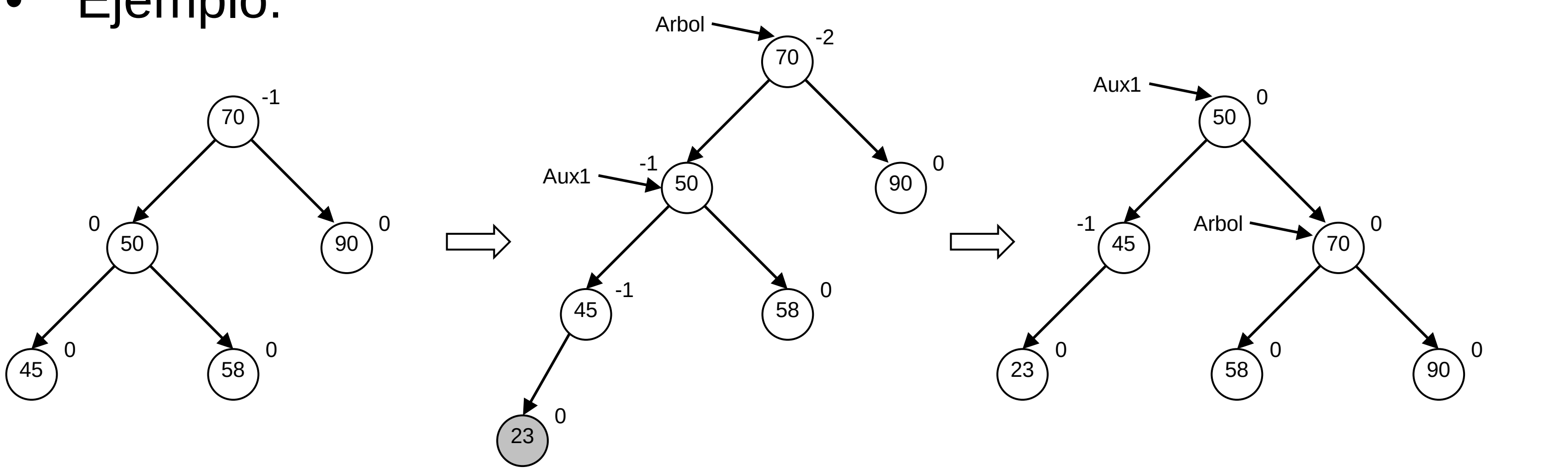

c
-1
b
0
c
-2
I
b
-1
I
a
0
ROTACIÓN II
b
0
a
0
c
0


Arbol
c
Aux1
b
h+3
h
h
h+1
b
c
h
h
ROTACIÓN II

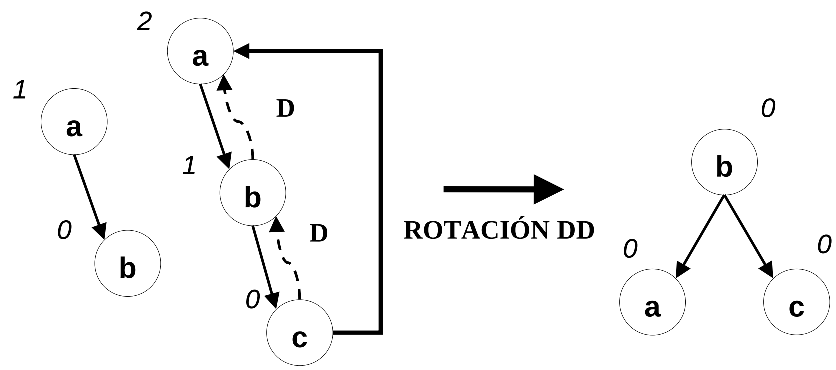
# Rotación II

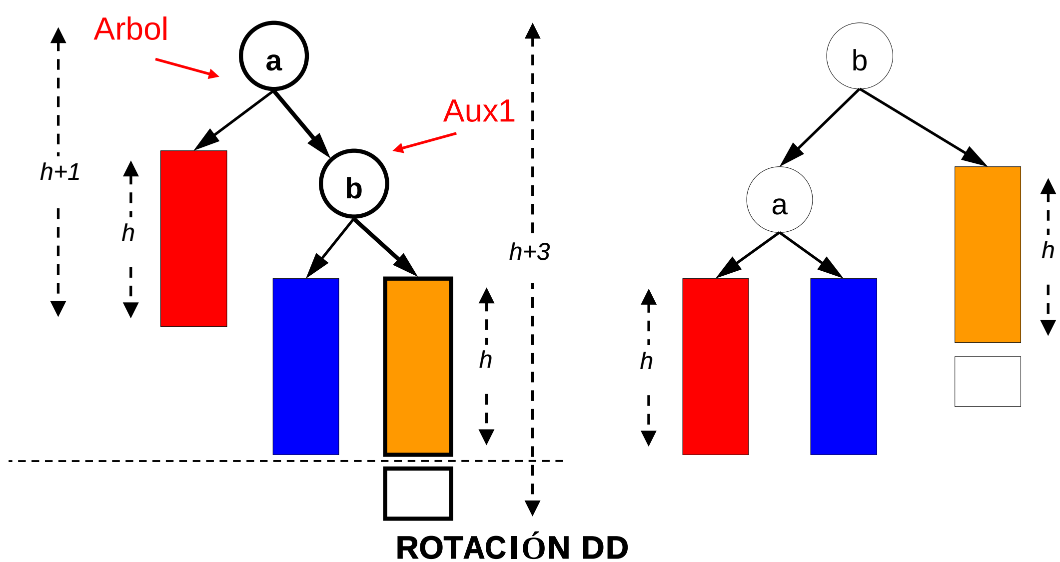
- Cambios a realizar en los punteros
  - `Arbol^.izdo:= Aux1^.dcho`
  - `Aux1^. dcho := Arbol`
  - `Arbol:= Aux1`
- Ejemplo:

- Cambios en los Fe
  - `SI Aux1^.Fe = 0` ⇒ `Arbol^.Fe:= -1 y Aux1^.Fe:= 1`
  - `SI Aux1^.Fe <> 0` ⇒ `Arbol^.Fe:= 0 y Aux1^.Fe:= 0`



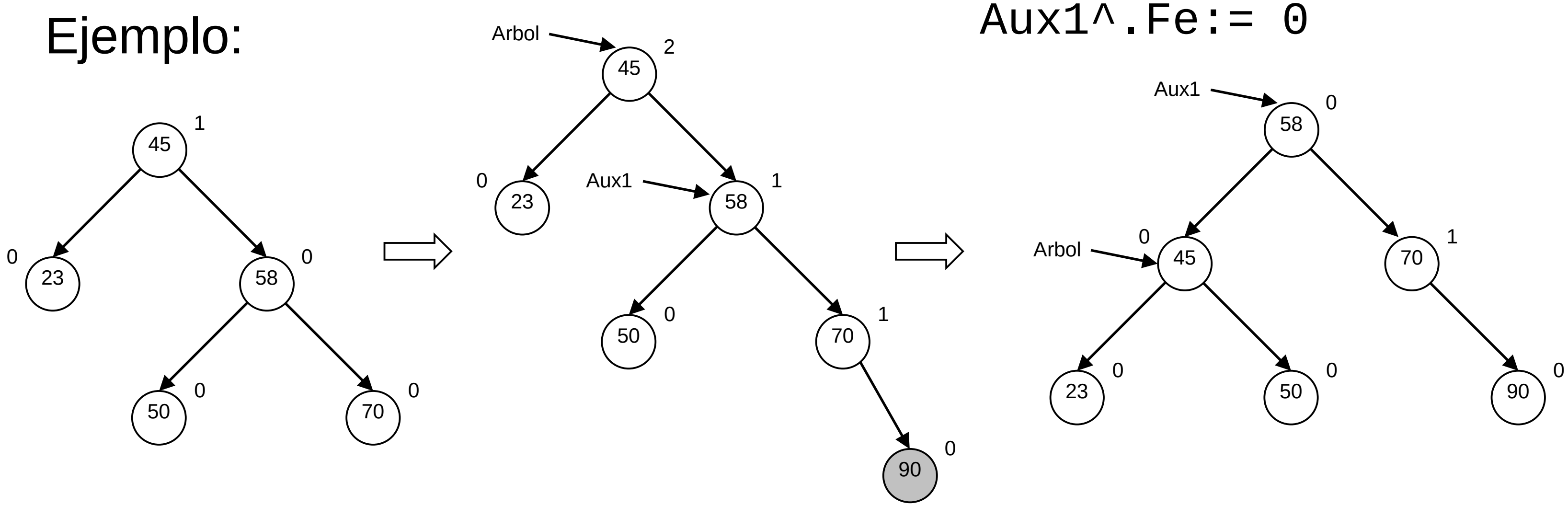
# Rotación DD


2
1
a
a
D
1
b
0
b
D
0
c
ROTACIÓN DD
0
b
0
0
a
c


Arbol
a
Aux1
h+1
h
b
h+3
h
b
a
h
h
ROTACIÓN DD

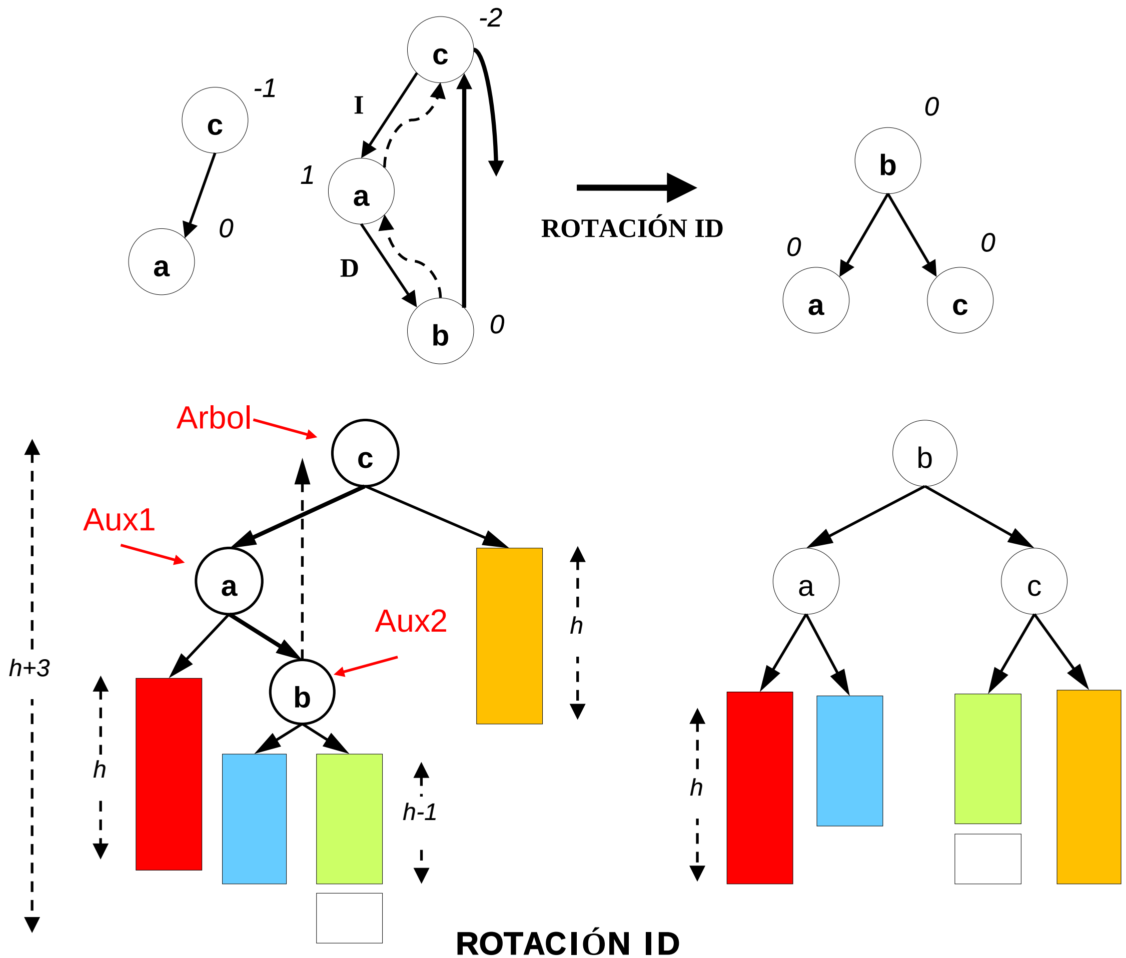
# Rotación DD

- Cambios a realizar en los punteros
  - `Arbol^.dcho:= Aux1^.izdo`
  - `Aux1^.izdo := Arbol`
  - `Arbol:= Aux1`
- Ejemplo:



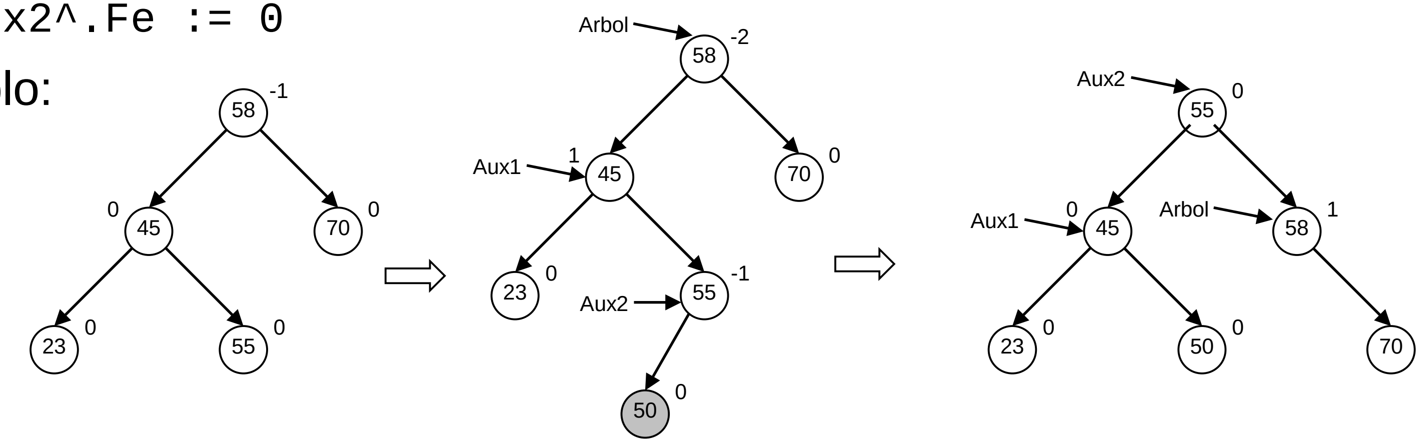

- Cambios en los Fe
  - `SI Aux1^.Fe = 0` ⇒ `Arbol^.Fe:= 1 y Aux1^.Fe:= -1`
  - `SI Aux1^.Fe <> 0` ⇒ `Arbol^.Fe:= 0 y Aux1^.Fe:= 0`

# Rotación ID

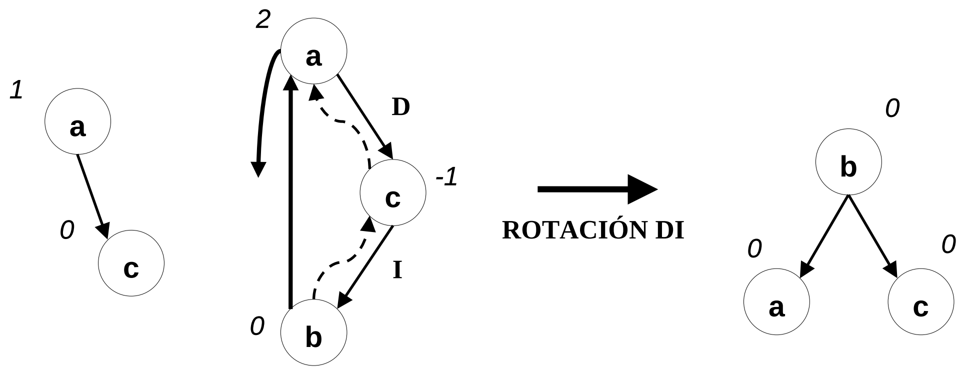

c
-1
a
0
c
-2
I
1
a
D
b
0
ROTACIÓN ID
b
0
0
a
0
c
Arbol
c
Aux1
a
Aux2
b
h+3
h
h-1
h
b
a
c
h
ROTACIÓN ID

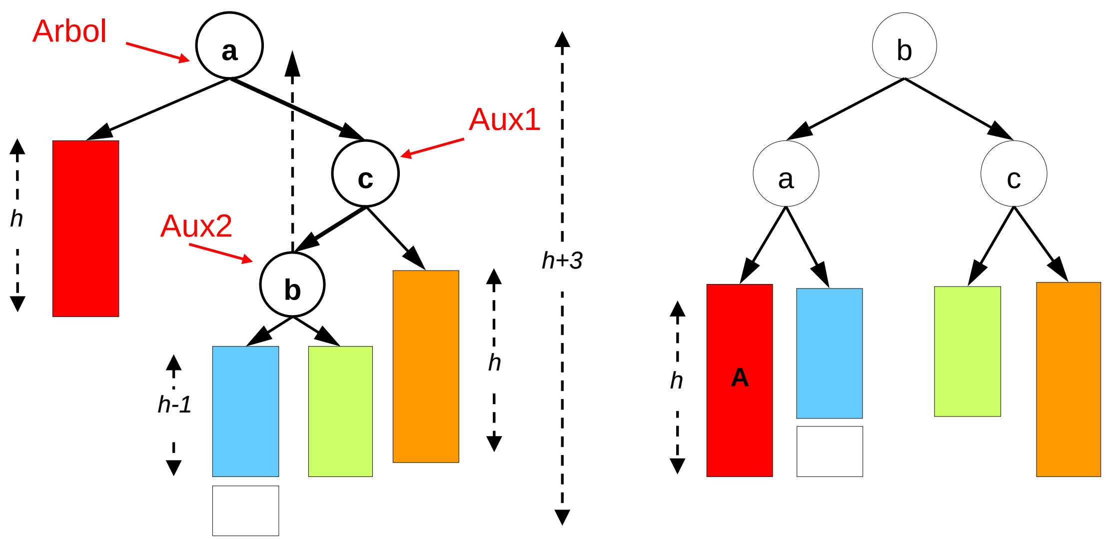
# Rotación ID

- Cambios a realizar en los punteros
  1. `Aux2:= Aux1^.dcho`
  2. `Arbol^.izdo:= Aux2^.dcho`
  3. `Aux1^.dcho:= Aux2^.izdo`
  4. `Aux2^.dcho := Arbol`
  5. `Aux2^.izdo := Aux1`
  6. `Arbol := Aux2`
- Cambios en los Fe
  - `SI Aux2^.Fe = 1` ⇒ `Aux1^.Fe := -1 SINO Aux1^.Fe := 0`
  - `SI Aux2^.Fe = -1` ⇒ `Arbol^.Fe := 1 SINO Arbol^.Fe := 0`
  - `Aux2^.Fe := 0`
- Ejemplo:



# Rotación DI

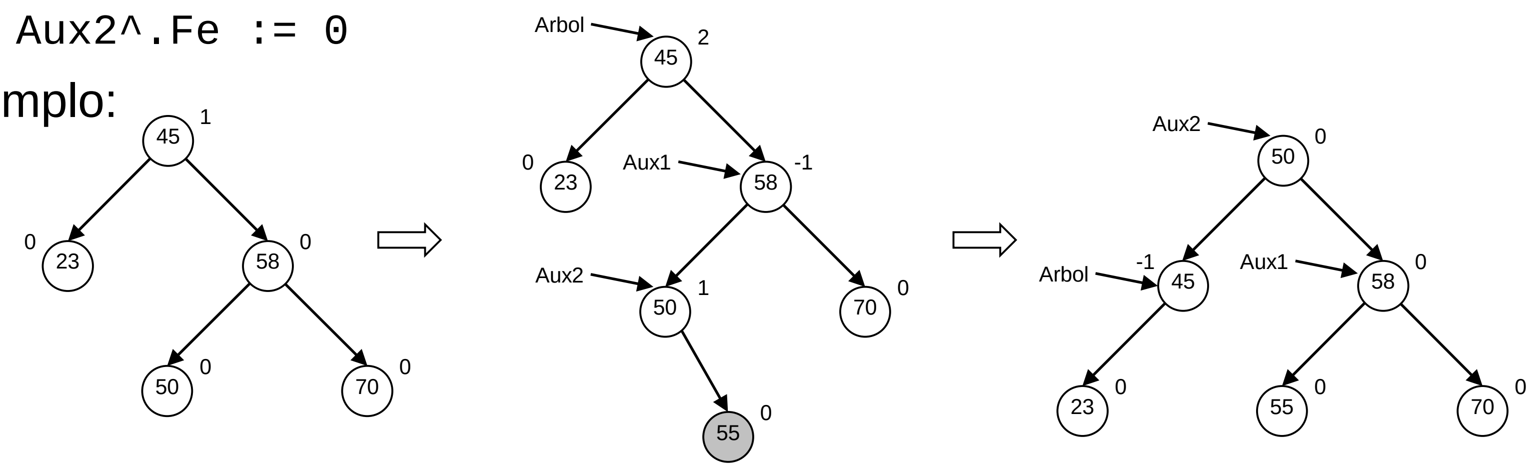



ROTACIÓN DI

# Rotación DI

- Cambios a realizar en los punteros
  1. `Aux2:= Aux1^.izdo`
  2. `Arbol^.dcho := Aux2^.izdo`
  3. `Aux1^.izdo := Aux2^.dcho`
  4. `Aux2^.izdo := Arbol`
  5. `Aux2^.dcho := Aux1`
  6. `Arbol := Aux2`
- Cambios en los Fe
  - `SI Aux2^.Fe = 1` ⇒ `Arbol^.Fe := -1 SINO Arbol^.Fe := 0`
  - `SI Aux2^.Fe = -1` ⇒ `Aux1^.Fe := 1 SINO Aux1^.Fe := 0`
  - `Aux2^.Fe := 0`
- Ejemplo:

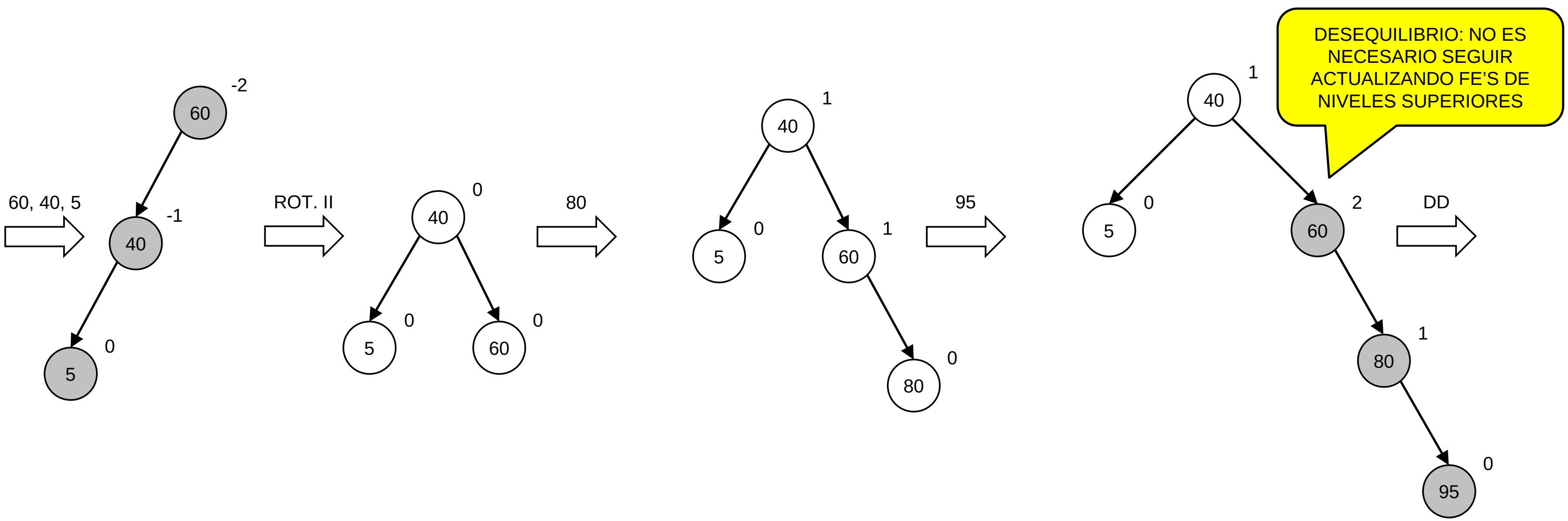


# Inserción en Árboles AVL: Pseudocódigo

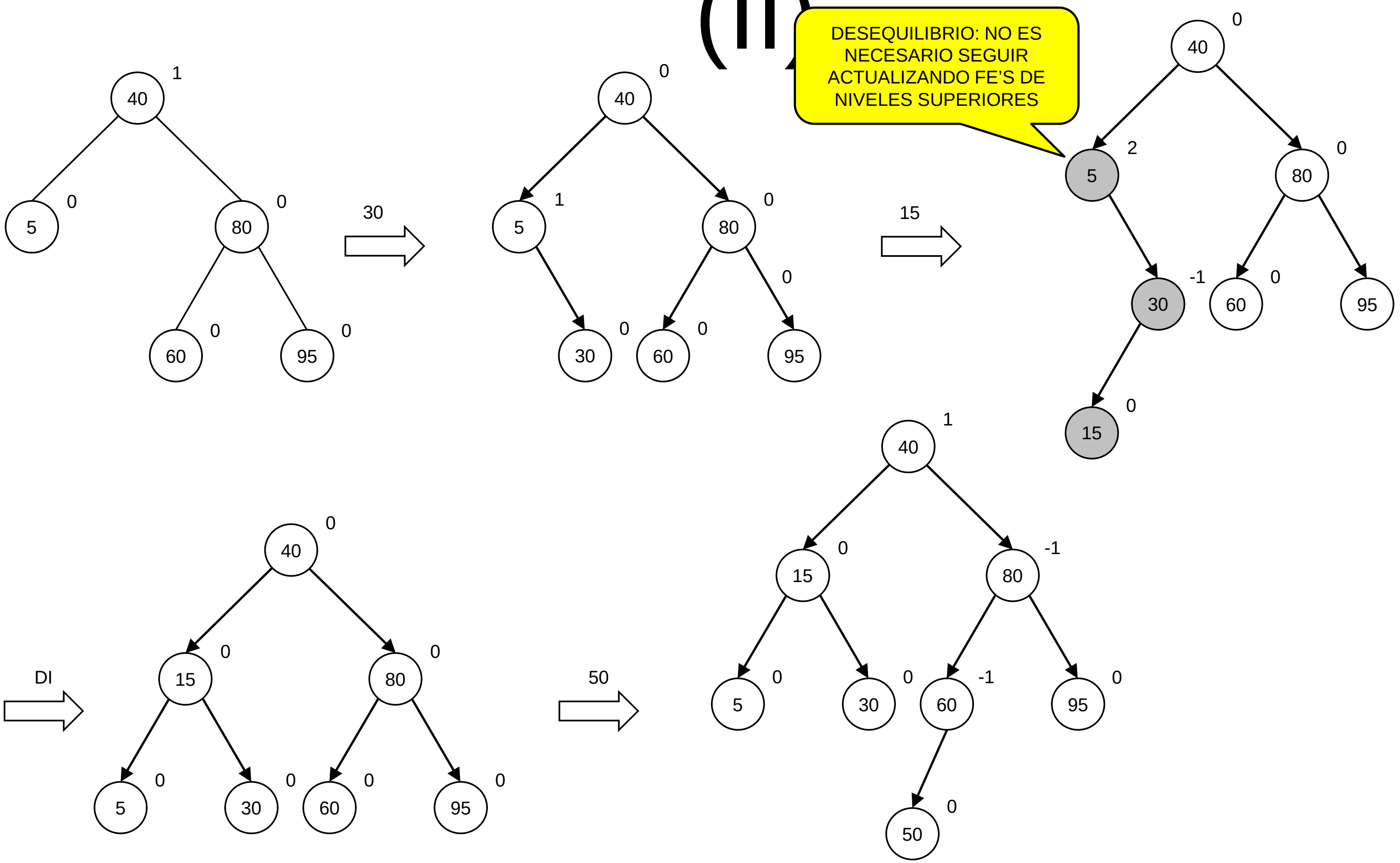
- Insertar como una hoja el nuevo nodo (proceso similar a ABB).
- El factor de equilibrio del nuevo nodo hoja es 0.
- Recorrer en sentido inverso el camino de búsqueda desde el lugar de inserción (camino de búsqueda). En cada nodo visitado recalcular el factor de equilibrio. El proceso termina cuando 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 `Arbol` (el nodo desequilibrado), `Aux1` y `Aux2` (los dos nodos siguientes a `Arbol` en el camino de búsqueda)
  - Según el FE 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)

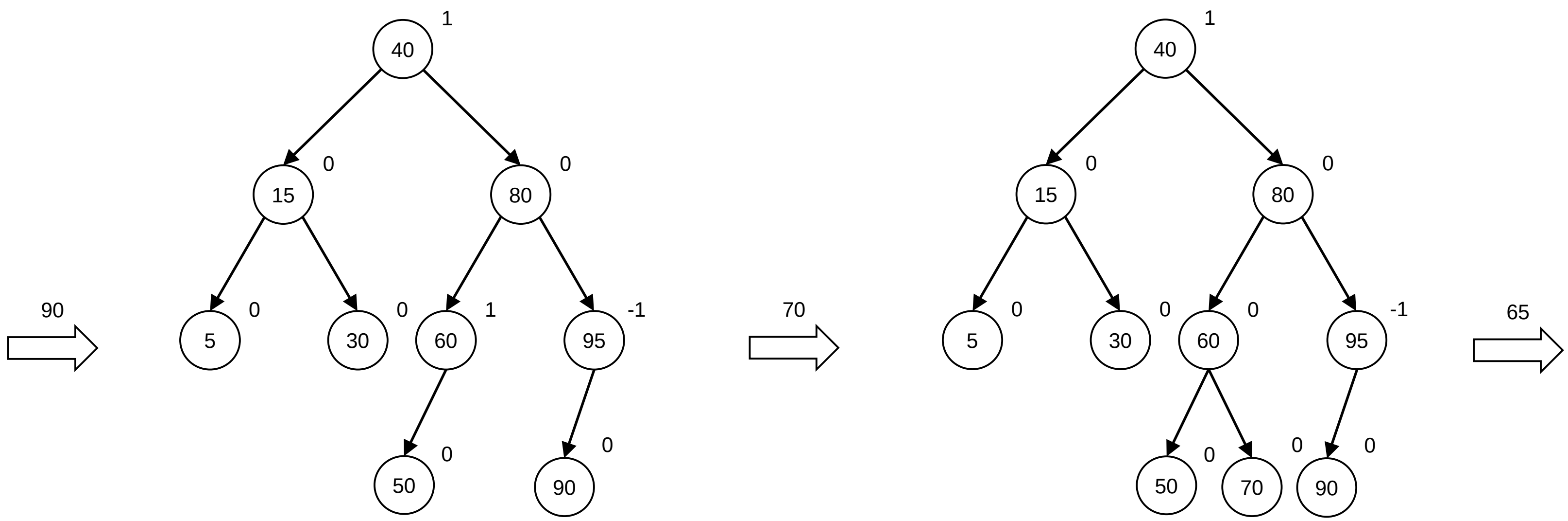
- Insertar 60, 40, 5, 80, 95, 30, 15, 50, 90, 70, 65, 10, 25, 35, 38



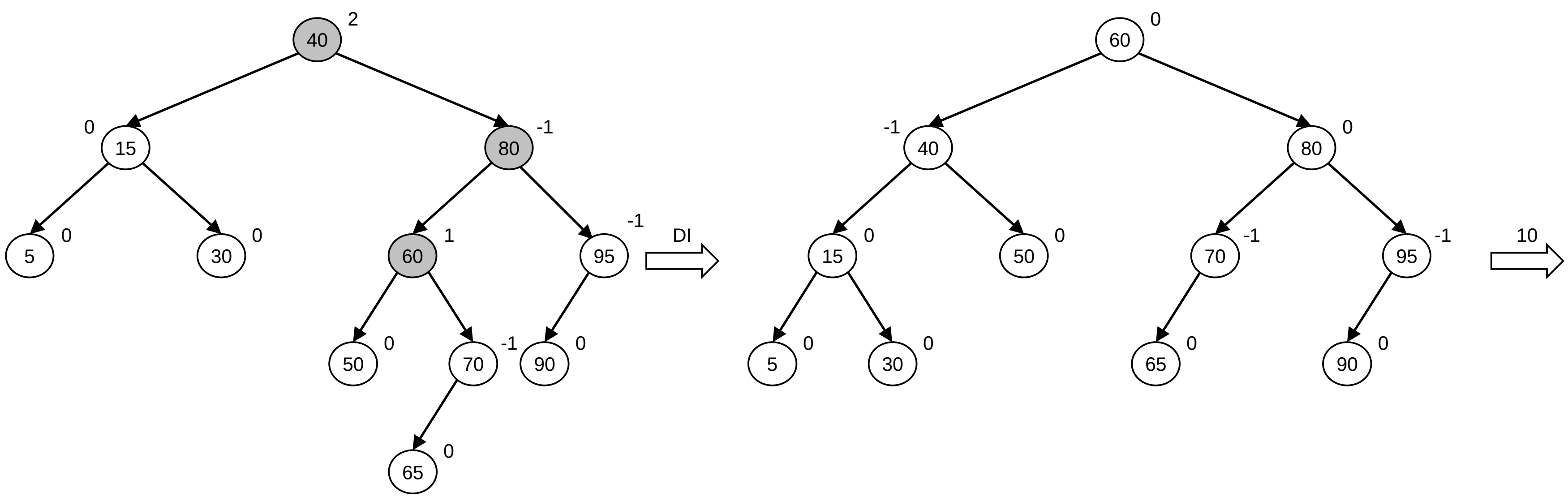
# Ejemplo de Inserción en AVL (II)



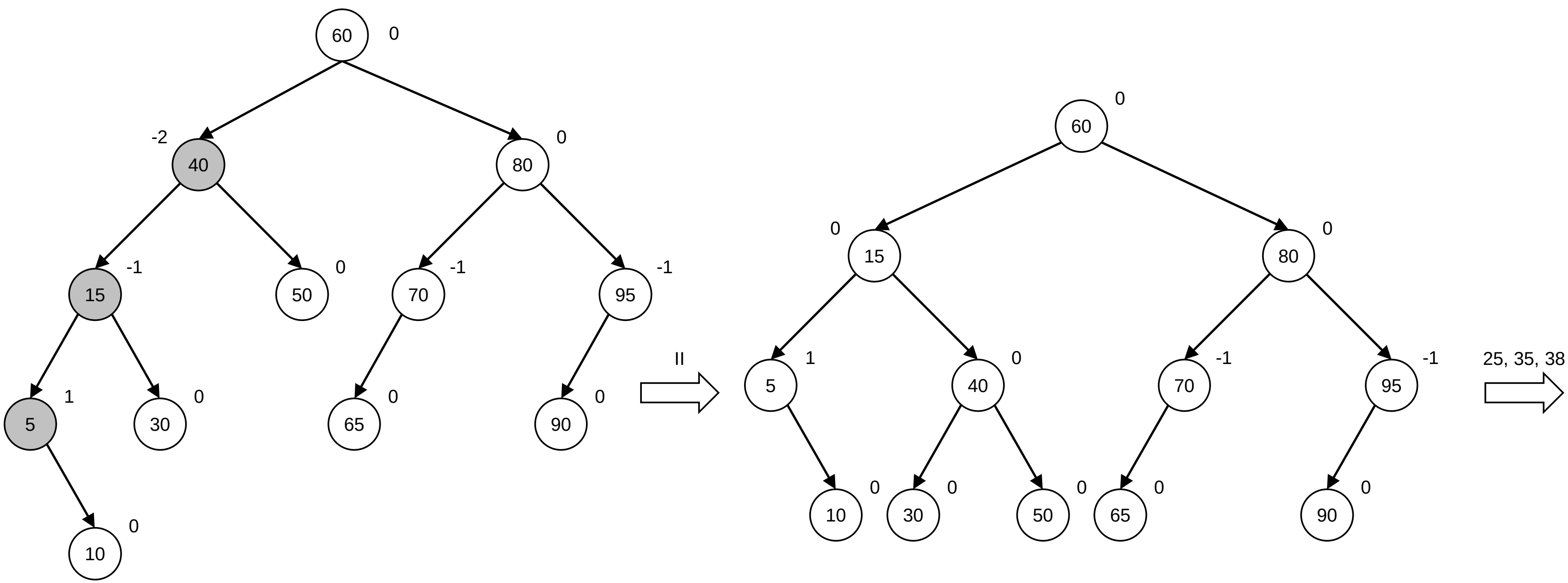
# Ejemplo de Inserción en AVL (II)



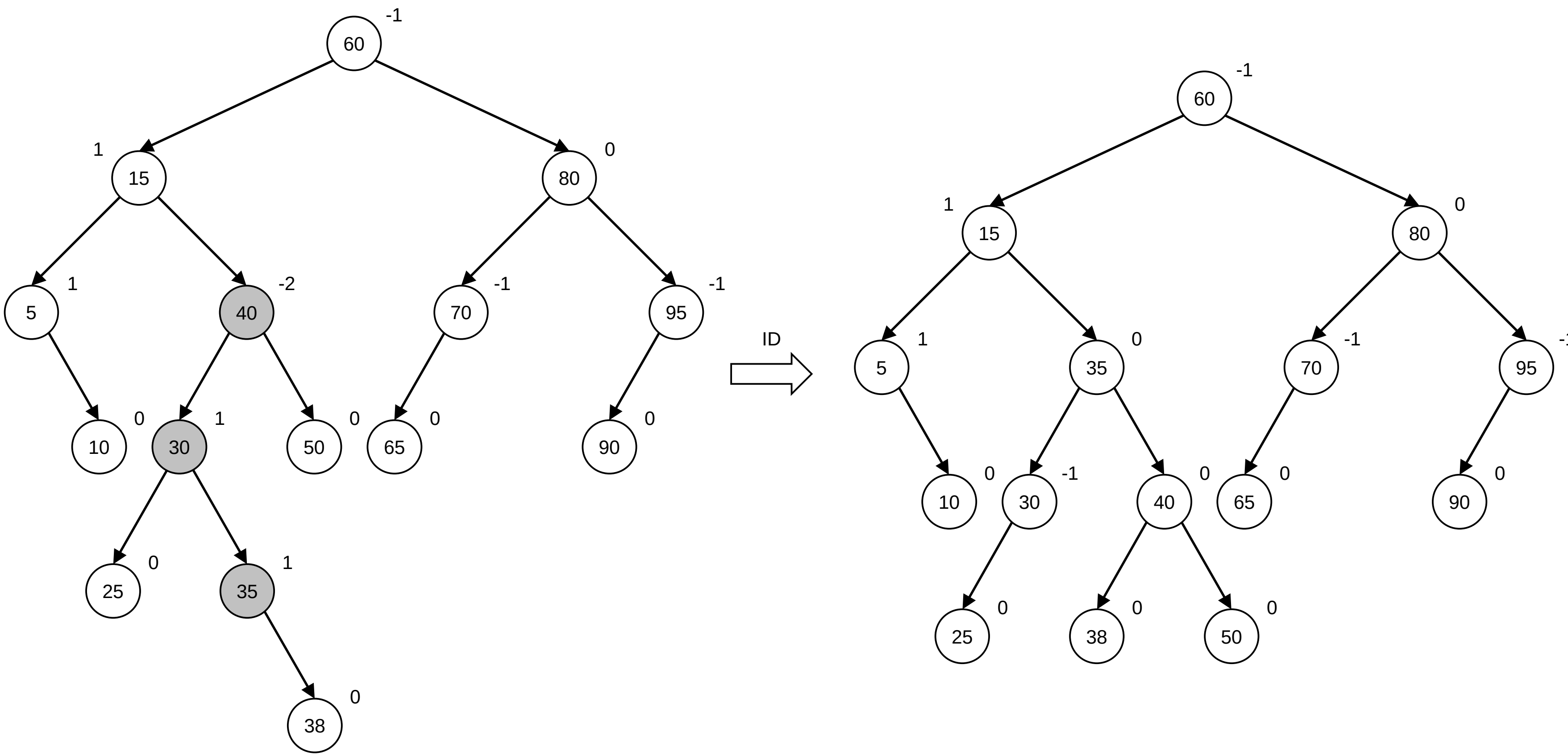
# Ejemplo de Inserción en AVL (III)



# Ejemplo de Inserción en AVL (IV)

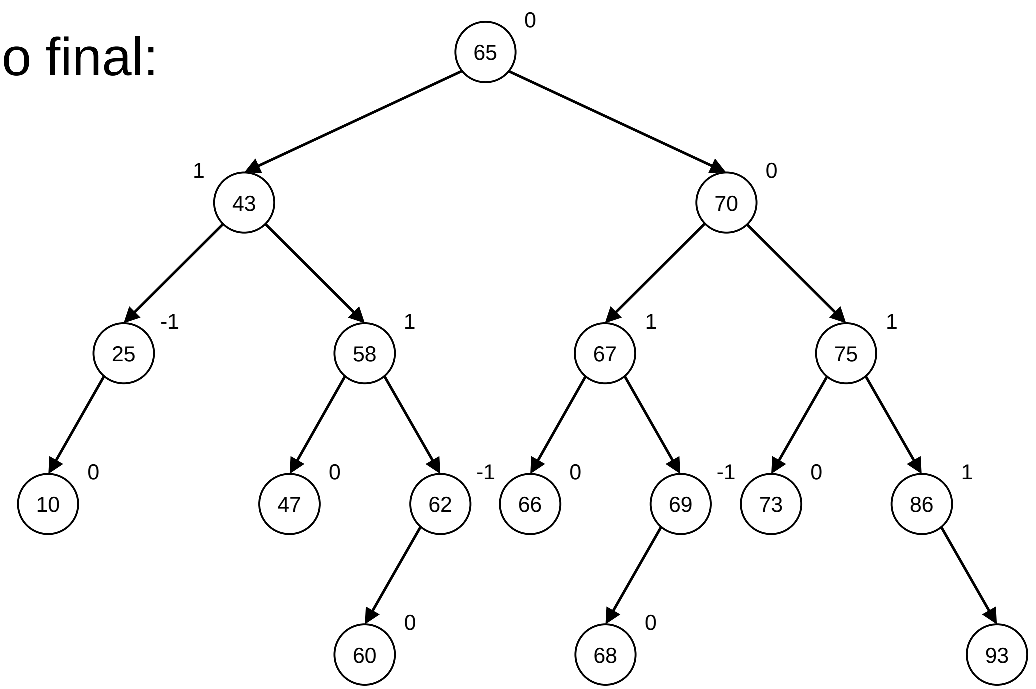


# Ejemplo de Inserción en AVL (V)



# 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:



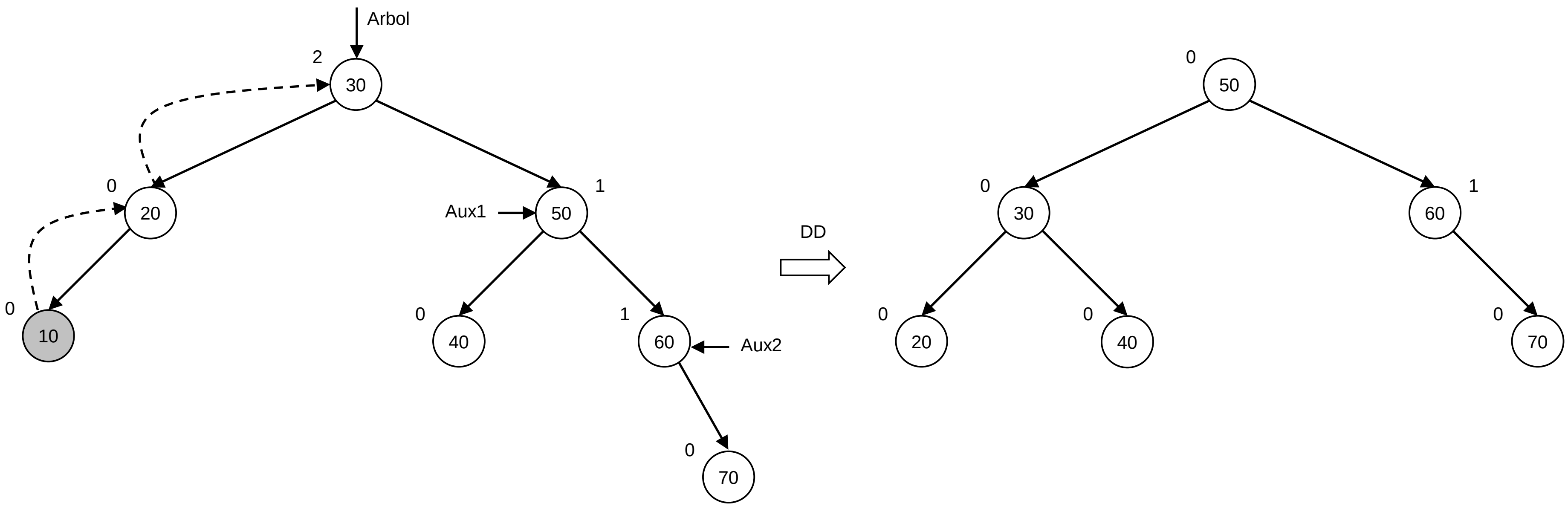
# Borrado en Árboles AVL: Pseudocódigo

- Se borra el nodo como en un ABB.
- A continuación es necesario recalcular los Fe
  - Se regresa por el camino de búsqueda recalculando los nuevos Fe 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 Fe 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

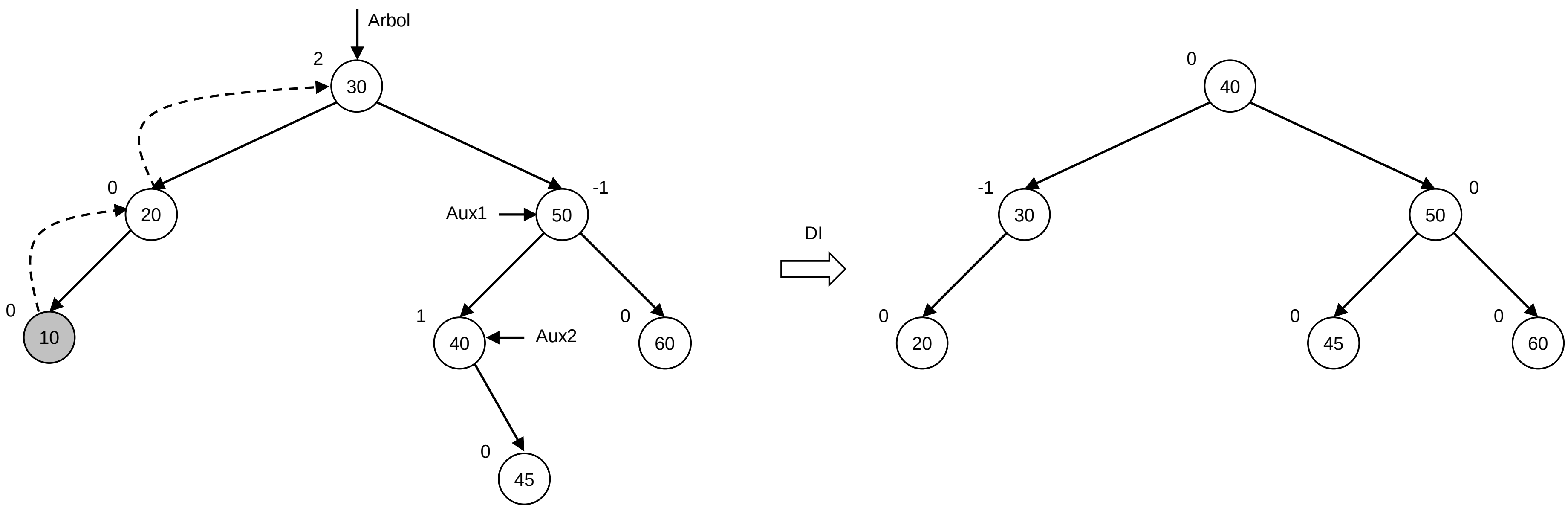
- ¿Cómo determinar los nodos implicados en una rotación?
  - El primer nodo (el apuntado por el puntero `Arbol`) es siempre el nodo desequilibrado (igual que en la inserción). El proceso para determinar el resto de nodos ya no es idéntico a la inserción.
  - El siguiente nodo involucrado (`Aux1`) es el nodo que está bajo el nodo `Arbol` 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 (II o DD)

# Rotación DD. Ejemplo*

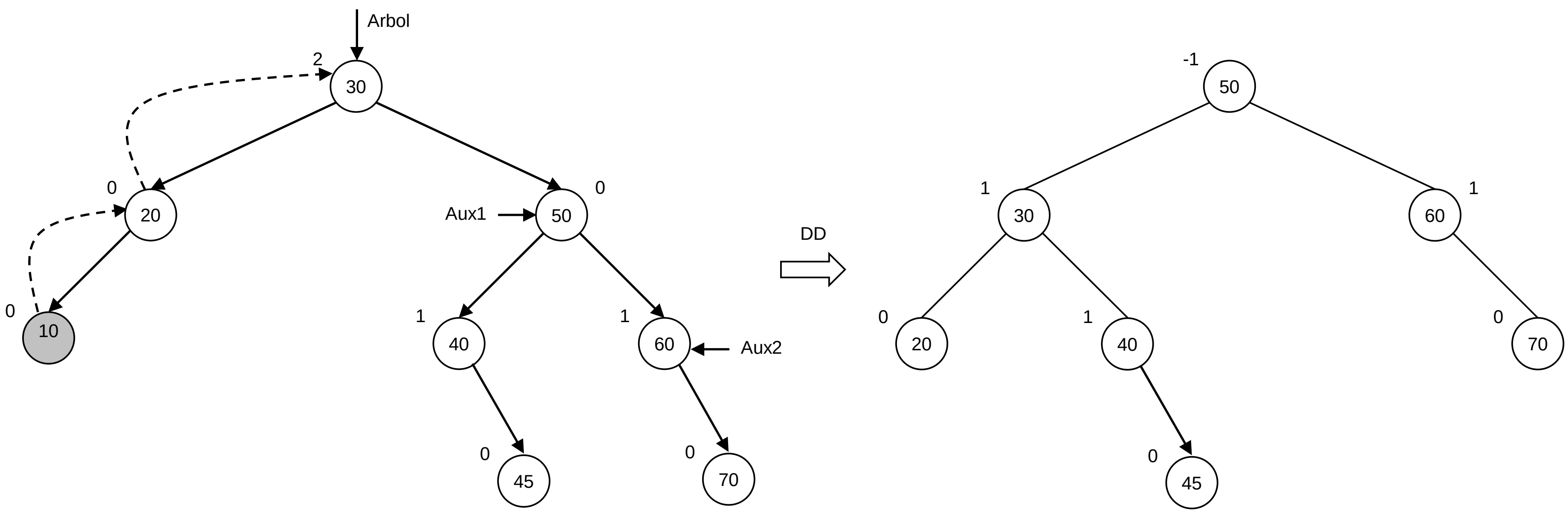



* Se elimina la clave sombreada

# Rotación DI. Ejemplo



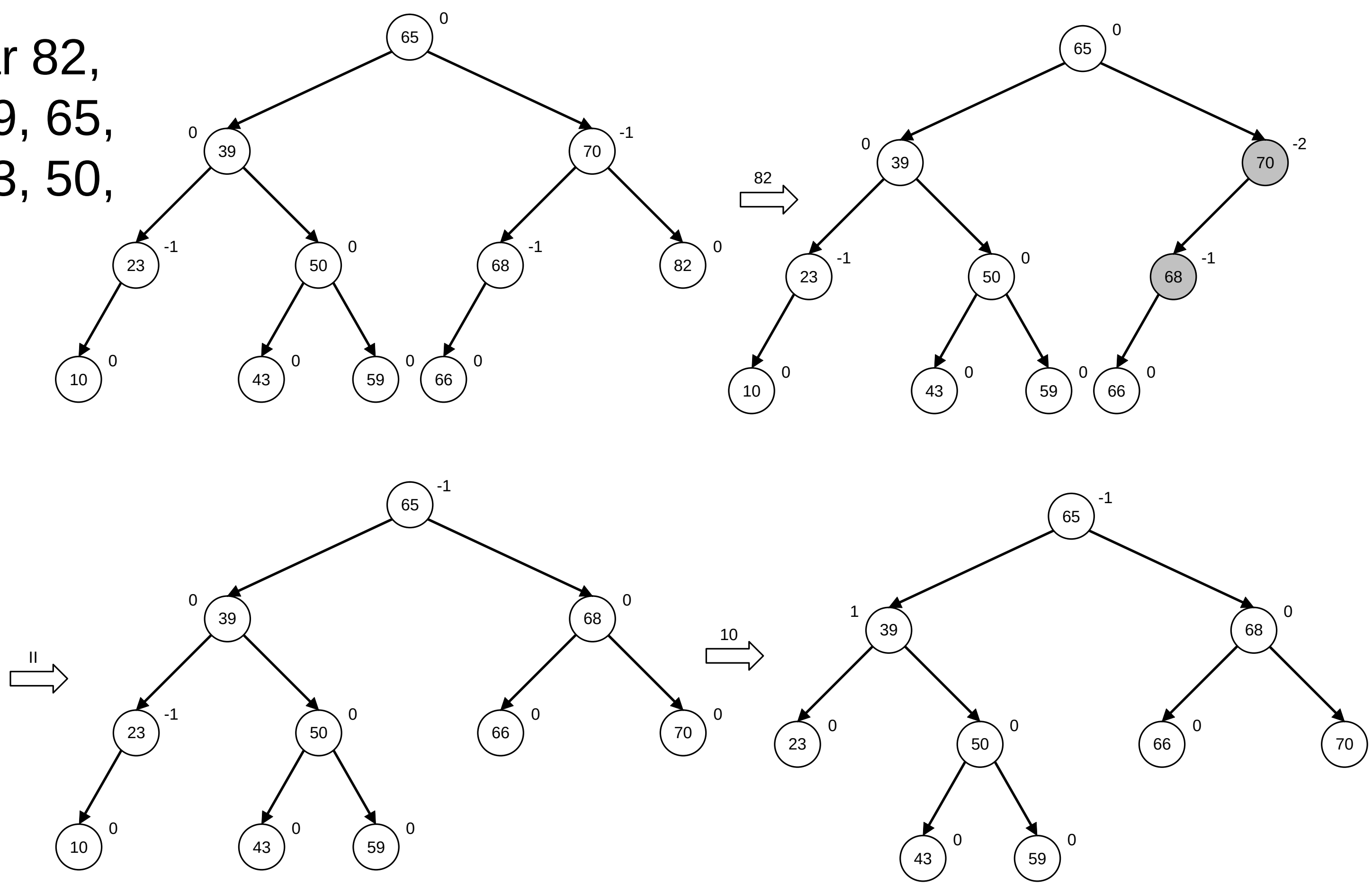
# Rotación DD o DI*. Ejemplo



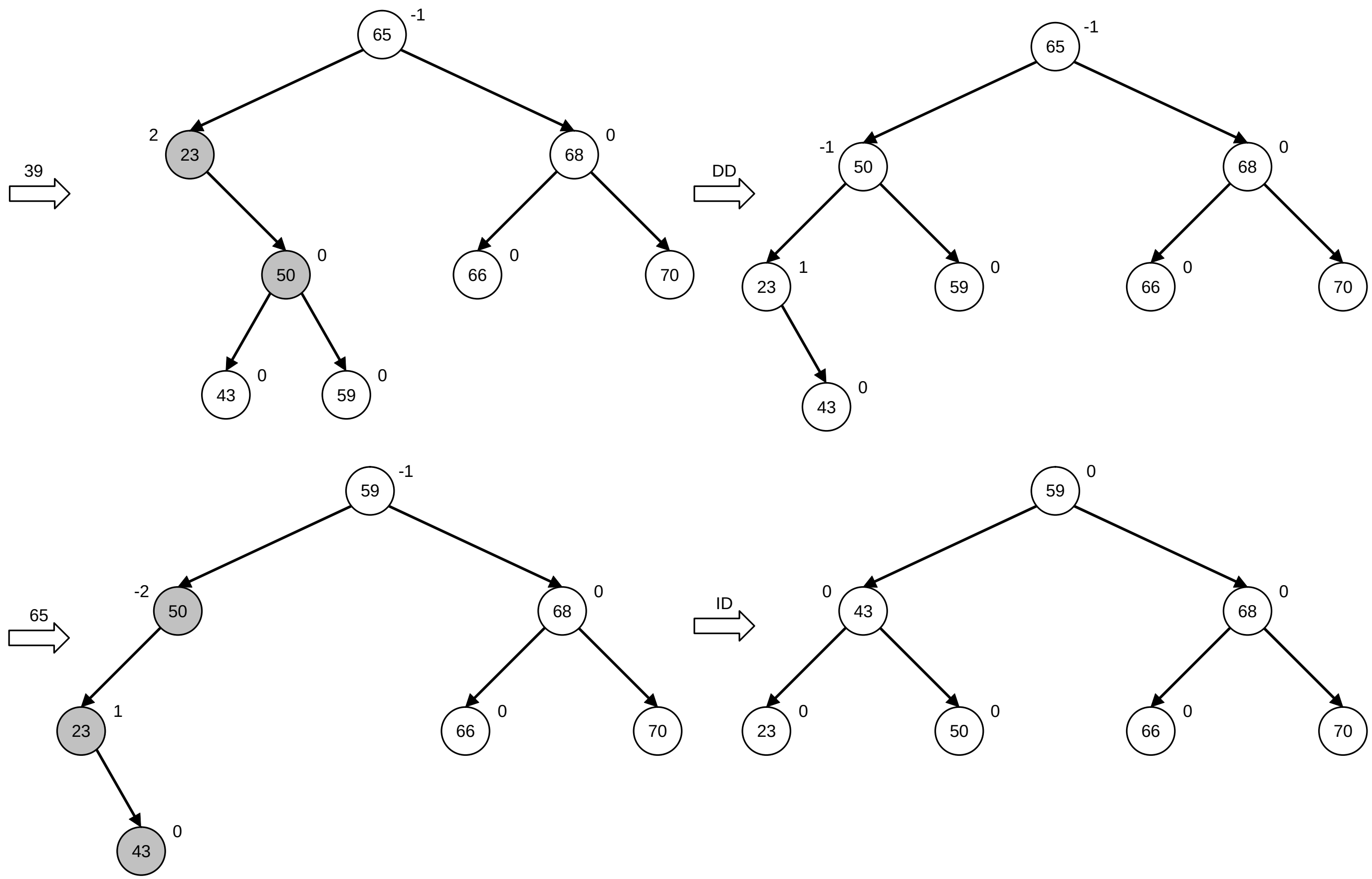

* Caso especial que sólo se da en el borrado.

# Ejemplo de Borrado en AVL (I)

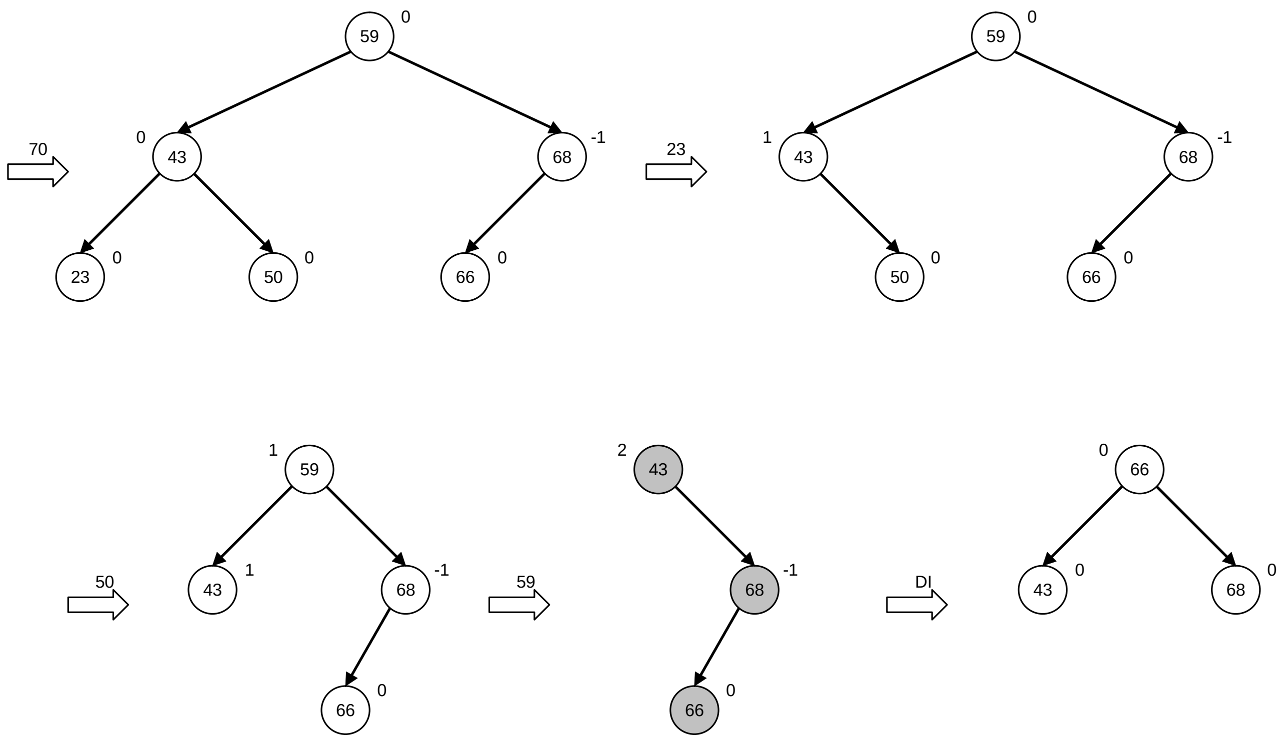
- Borrar 82, 10, 39, 65, 70, 23, 50, 59



# Ejemplo de Borrado en AVL (I)

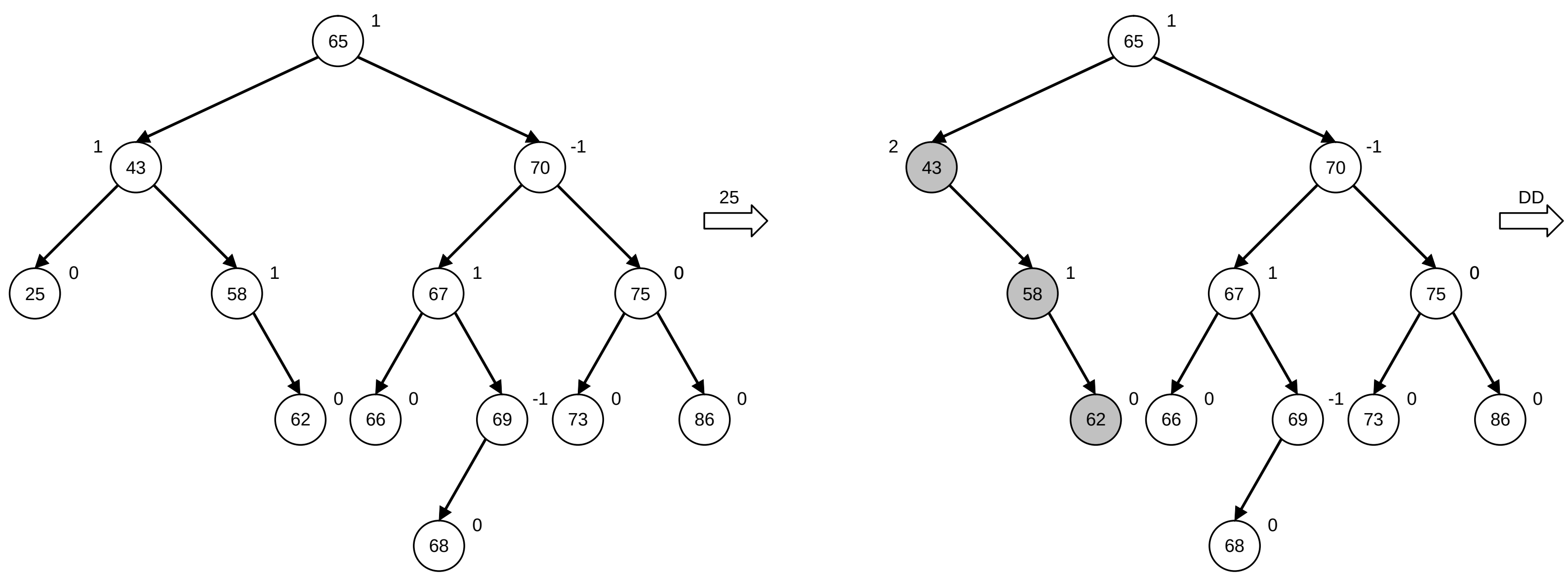


# Ejemplo de Borrado en AVL (I)



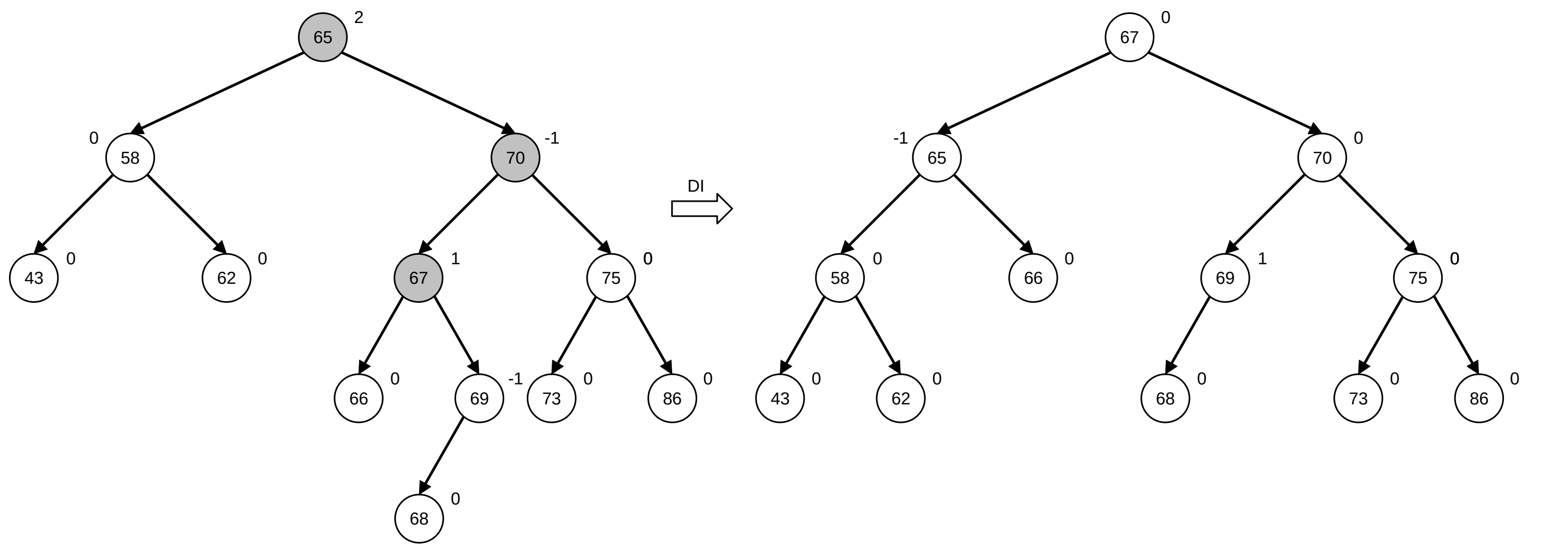
# Ejemplo de Borrado en AVL (II)

- 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