

# Árboles 2-3

Pablo Castro  
Algoritmos I-UNRC

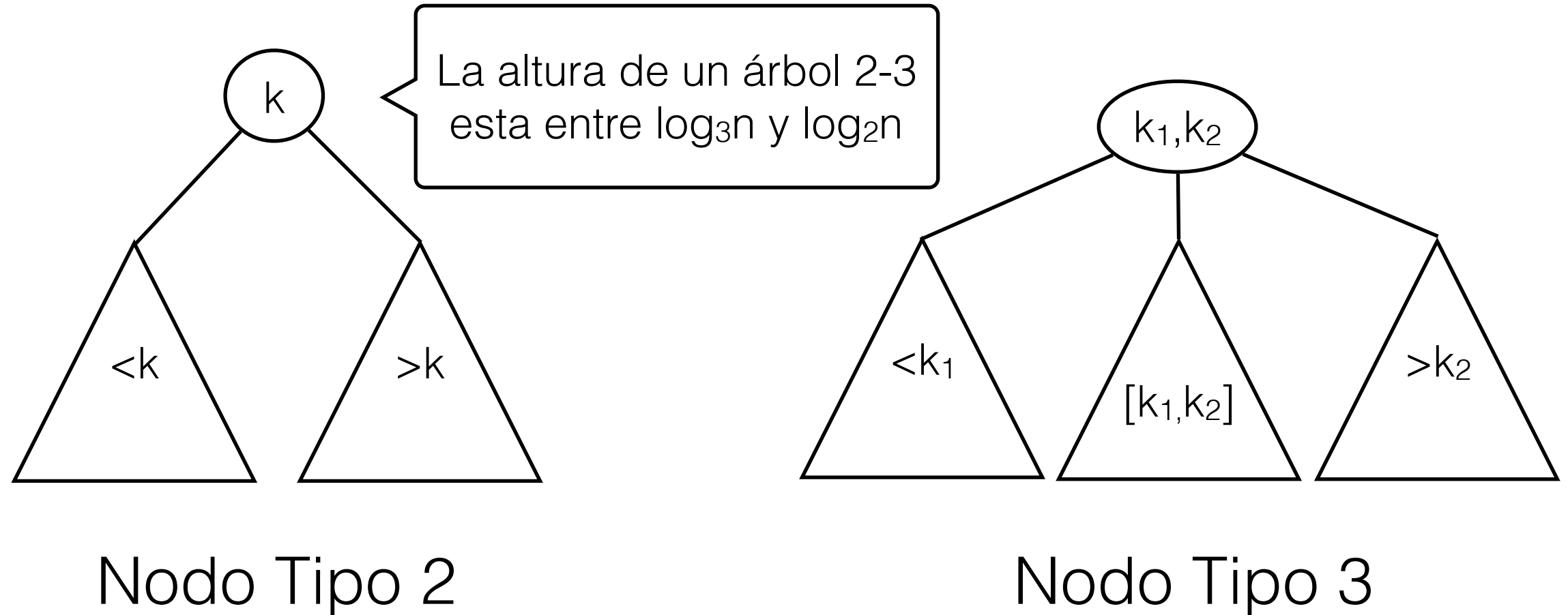
# Árboles 2-3

Los árboles 2-3 proveen otra forma de mantener los árboles balanceados, para hacer las operaciones eficientes.

- Tenemos dos tipos de nodos:
  - **Los nodos tipo 2**, que contienen un solo valor  $k$ , y dos subárboles. Los nodos del hijo izq. son menores que  $k$ , y los del hijo derecho son mayores.
  - **Los nodos de tipos 3**, que contienen dos valores  $k_1$ ,  $k_2$  y tres subárboles. Los nodos del hijo izq. son menores que  $k_1$ , los nodos del hijo der. son mayores que  $k_2$ . Y los nodos del subárbol del medio son mayores que  $k_1$  y menores que  $k_2$ .

# Árboles 2-3

Gráficamente, los dos tipos de nodos pueden verse de la siguiente manera:



Además todas las hojas deben encontrarse en el mismo nivel

# Altura de un Árbol 2-3

Tenemos el siguiente teorema sobre los árboles 2-3:

Propiedad: La altura de un árbol 2-3 es  $O(\log n)$

Idea de la demostración:

$n$  es la cantidad de nodos

La cantidad de nodos de un árbol con altura  $h$  está acotada por los árboles con todos nodos 2 y todos nodos 3

$$2^h - 1 \leq n \wedge n \leq 3^h - 1$$

$\equiv$

$$h \leq \log_2(n + 1) \wedge \log_3(n + 1) \leq h$$

$\Rightarrow$

$$h \in O(\log n) \wedge h \in \Omega(\log n) \equiv h \in \Theta(\log n)$$

Para cualquier árbol  $k$ -ario full:

$$size(t) \leq \frac{k^{alt} - 1}{k - 1}$$

# Búsqueda

Se busca haciendo comparaciones con la raíz:

- Si la raíz es un nodo de tipo 2, procedemos igual que con los ABB's.
- Si la raíz es un nodo de tipo 3, tenemos que realizar dos comparaciones para saber si debemos buscar a la derecha, al medio o a la izquierda

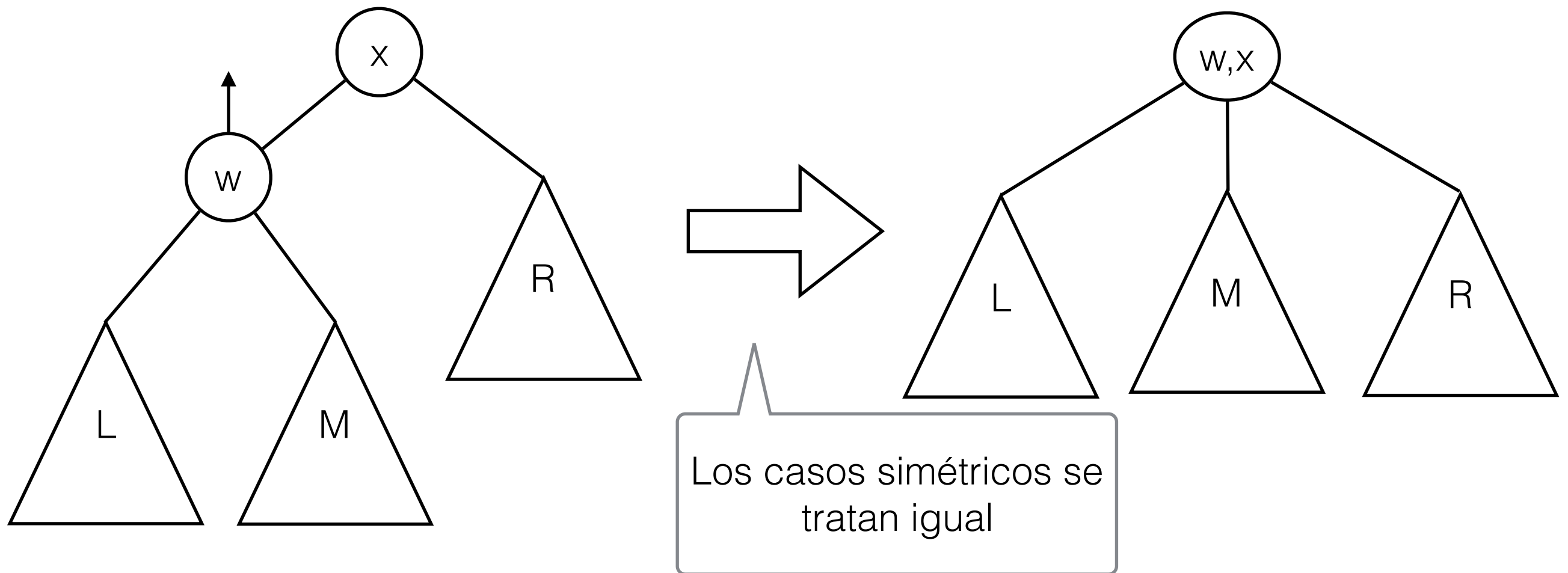
# Inserción

La inserción se realiza siempre en las hojas:

- Se busca el lugar donde insertar por comparaciones,
- Si se tiene que insertar en una hoja tipo 2, se agrega el valor al nodo, y se lo convierte a un nodo tipo 3.
- Si se tiene que agregar en una hoja tipo 3, se parte el nodo en dos, dejando el valor del medio como raíz y los dos restantes como hijos izq. y der.
- La raíz de este nuevo subárbol se va desplazando hacia arriba, hasta que se acomoda.

# Inserción

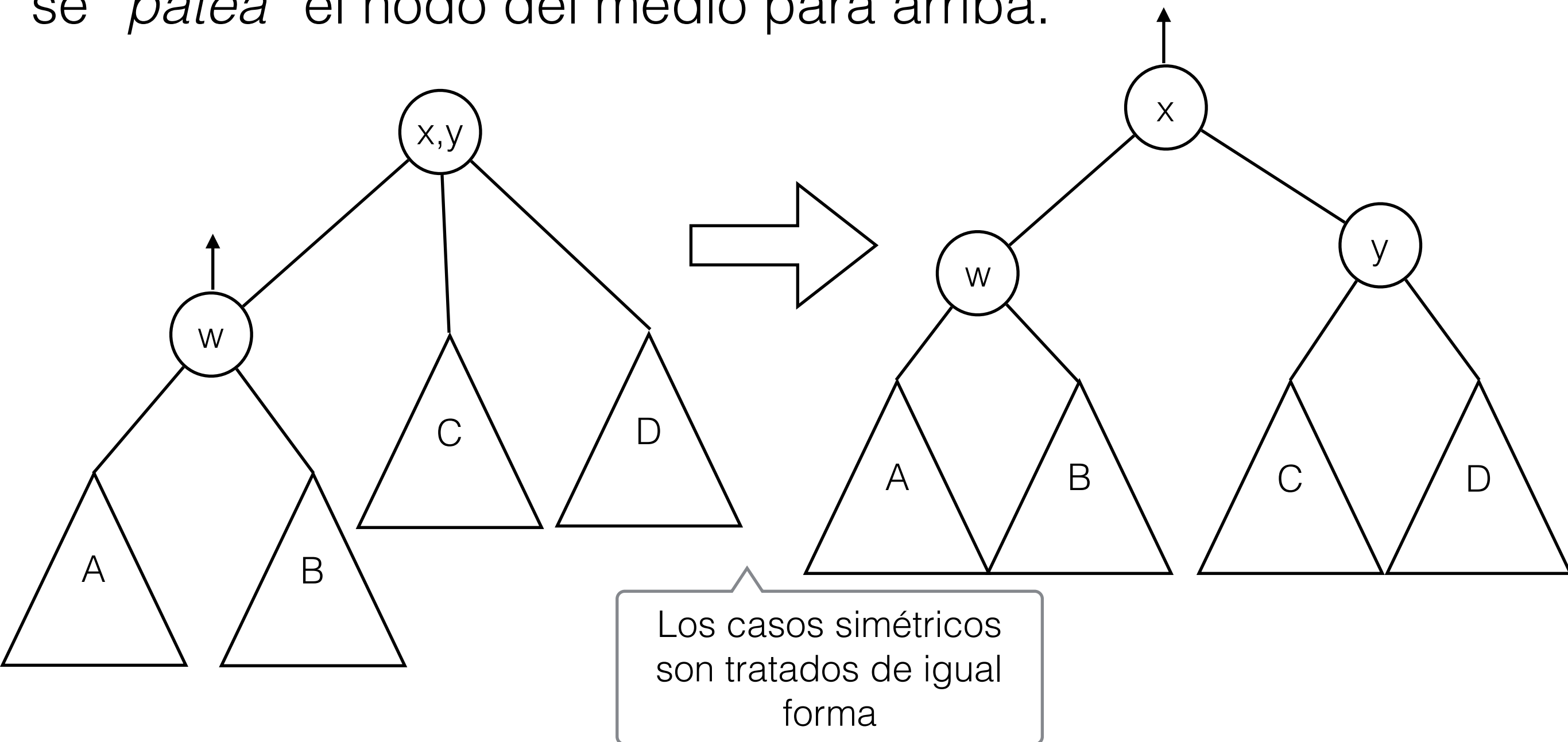
Cuando se va llevando un valor hacía arriba, hay varios casos:



Si el padre del nodo que se “*patea*” arriba es nodo 2, se lo acomoda como en la figura.

# Inserción

Si el padre es un nodo tipo 3, se forman dos subárboles y se “*patea*” el nodo del medio para arriba.





# Notar que...

- En el peor caso el nodo nuevo que se va llevando hacia arriba se acomoda finalmente en la raíz, en este caso la altura aumenta por uno.
- En todos los otros casos la altura del árbol vuelve a ser la misma que el original.
- La inserción es  $O(\log n)$  debido a que la altura es  $O(\log n)$ .

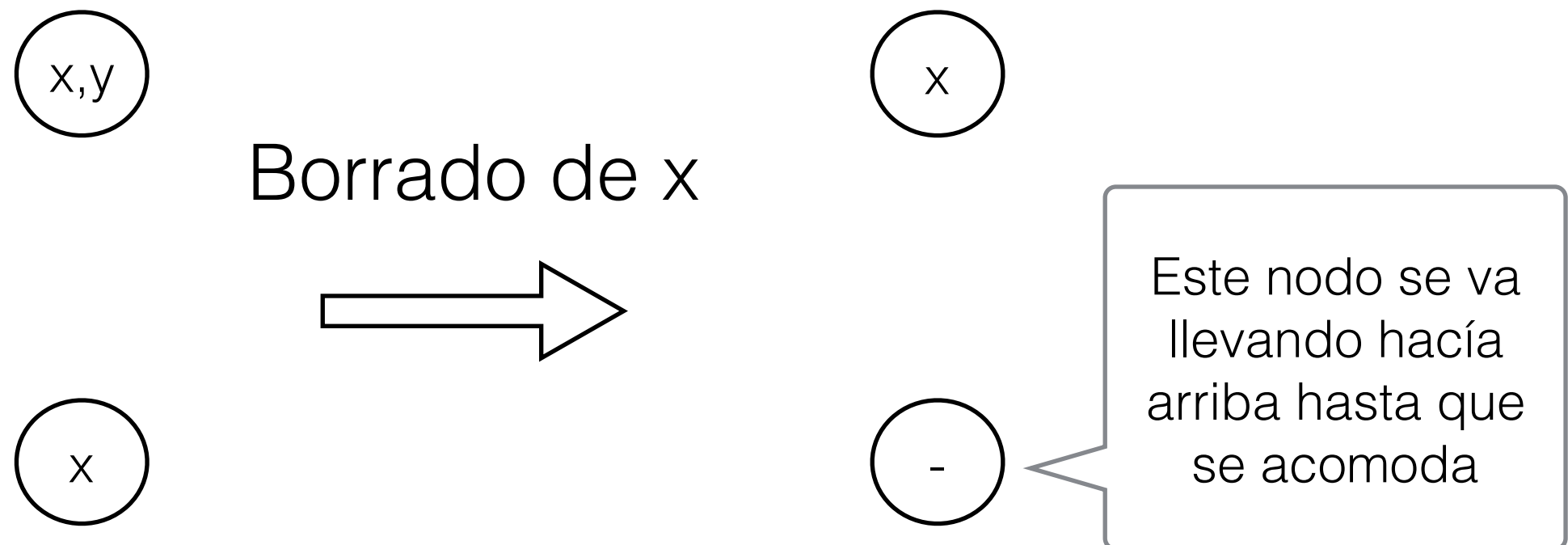
# Borrado

El borrado es similar pero se utiliza un nodo vacío:

- Se busca como los ABBs, se borra el nodo deseado, y se lo reemplaza con el más grande de la izq. o el más chico de la derecha.
- El lugar donde estaba el nodo que se sacó se llena con un elemento ficticio.
- Este elemento ficticio (o vacío) se lo va llevando hacia arriba hasta que desaparece.

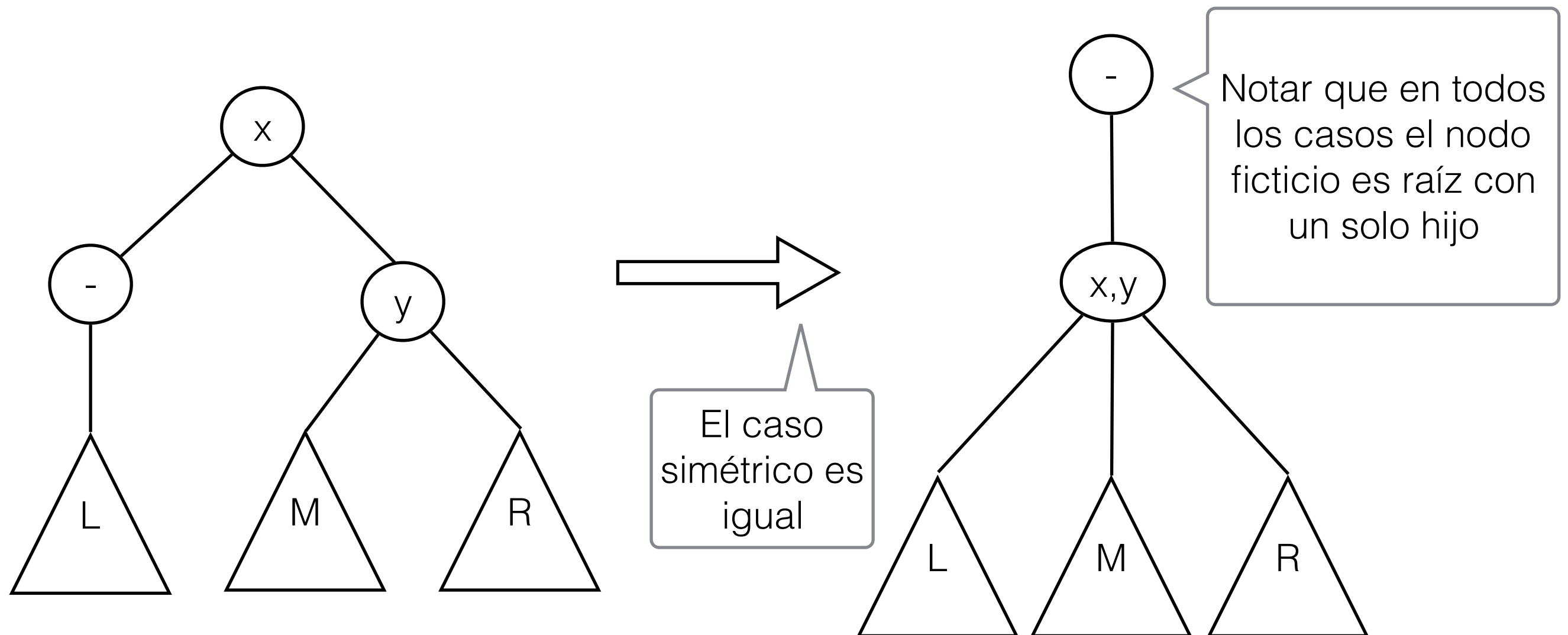
# Borrado-Casos

En el caso de una hoja, si es un nodo tipo 3 se lo transforma a tipo 2, sino se forma un nodo tipo 2 vacío y se lo lleva hacia arriba.



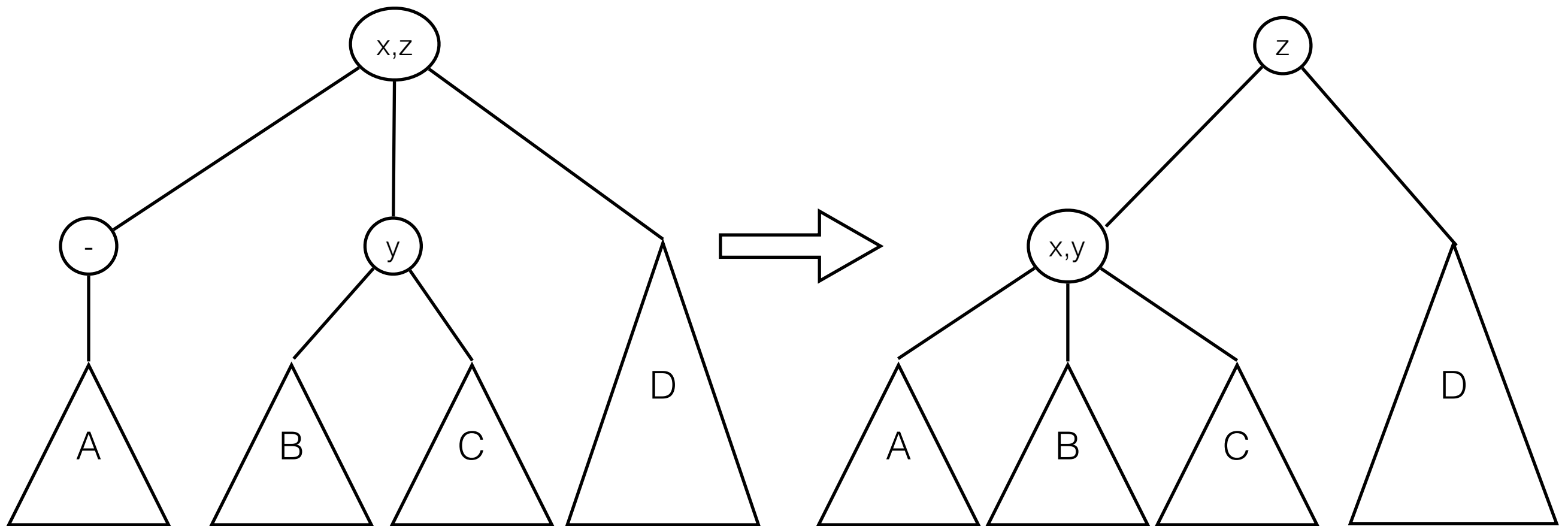
# Borrado

En el caso que el nodo ficticio tenga como padre un nodo tipo 2 y hermano de tipo 2:



# Borrado

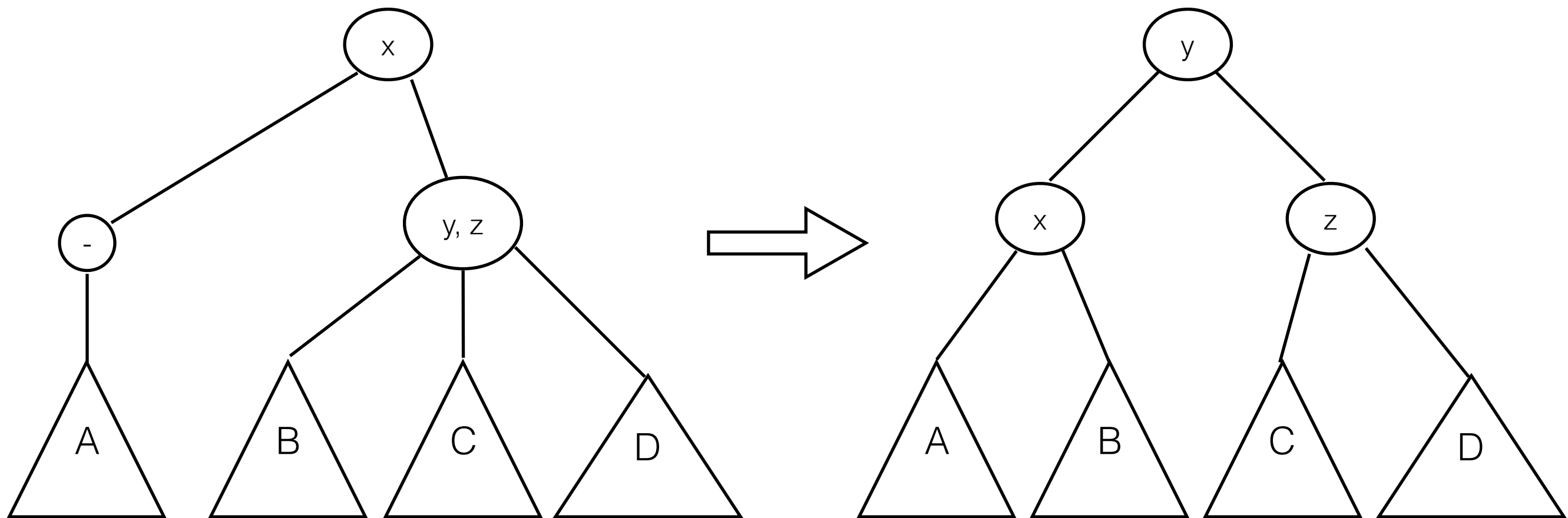
En el caso que el nodo ficticio tenga como padre un nodo tipo 3 y hermano de tipo 2



Los demás casos son tratados de la misma forma

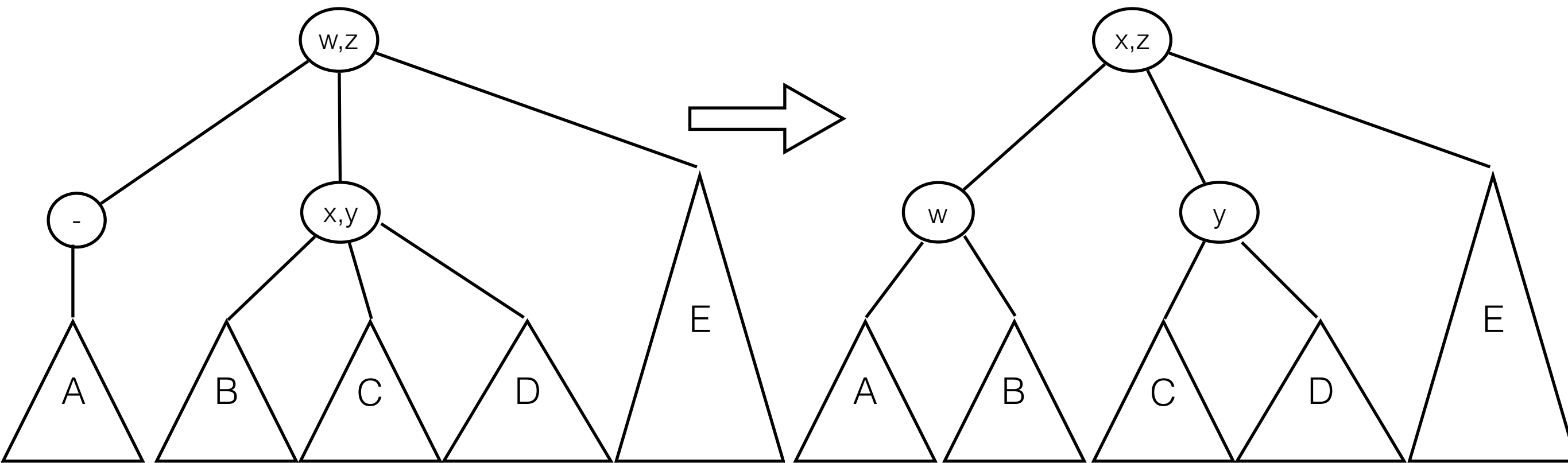
# Borrado

En el caso que el nodo ficticio tenga como padre un nodo tipo 2 y hermano de tipo 3



# Borrado

En el caso que el nodo ficticio tenga como padre un nodo tipo 3 y hermano de tipo 3



# Ejemplo de Inserción

