

Definición de un árbol Rojinegro

- Árbol binario estricto (los nodos nulos se tienen en cuenta en la definición de las operaciones \Rightarrow todo nodo hoja es nulo)
- Cada nodo tiene estado rojo o negro
- Nodos hoja (nulos) son negros
- La raíz es negra (esta condición se impone para simplificar algunas operaciones)

Se cumplen las condiciones:

1. Un nodo **rojo** tiene dos hijos **negros**.

2. Todo **camino** de la raíz a cualquier hoja pasa por el mismo número de nodos **negros**.

Notación:



Nodo rojo



Nodo negro



Nodo rojo o negro (da igual)



Nodo y su subárbol asociado
(puede ser un nodo nulo)

La segunda condición se puede expresar de manera equivalente definiendo el concepto de **altura negra** de un nodo (de ahora en adelante la denotaremos por la letra **H**, para diferenciarla de la altura normal, **h**). La altura negra de un nodo es igual a la altura del nodo cuando sólo se tienen en cuenta los nodos negros del subárbol cuya raíz es el nodo.

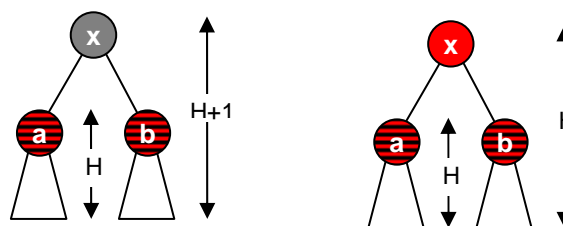
De manera más formal, si n es un nodo se puede calcular su altura negra así:

$$H(n) = \begin{cases} \text{máx}(H(n.\text{izdo}), H(n.\text{dcho})) + 1 & \text{si } n \text{ es negro} \\ \text{máx}(H(n.\text{izdo}), H(n.\text{dcho})) & \text{si } n \text{ es rojo} \end{cases}$$

La segunda condición de los árboles rojo-negro se puede expresar de la siguiente forma:

2. Para todo nodo interno (que no sea hoja, es decir nulo), la altura negra de su hijo izquierdo es igual a la altura negra de su hijo derecho.

Se debe cumplir, entonces:



Se cumplen las propiedades siguientes (se supone un árbol inicial que cumpla las condiciones):

- Cambiar un nodo de **rojo** a **negro** no afecta a la condición 1, pero afecta a la condición 2 (altura negra se incrementa en todos los nodos **ascendientes**).
- Cambiar un nodo de **negro** a **rojo** puede afectar a la condición 1 (si el padre o alguno de los hijos es rojo) y también a la condición 2 (altura negra se decreta en todos los nodos **ascendientes**).
- Si como resultado de una operación la **raíz** pasa a ser **rojo**, se puede cambiar a **negro** directamente sin afectar a las condiciones.
- **Borrar** un nodo **rojo** **no afecta** a las condiciones, pero borrar un nodo **negro** sí (altura negra decrece en los ascendientes).

Inserción de un nodo

La inserción de un nodo se realiza inicialmente igual que en un árbol binario de búsqueda. Al nuevo nodo se le da el color **rojo**. De ésta manera no se viola la segunda condición (altura negra), aunque se puede violar la primera (el padre del nodo insertado puede ser rojo).

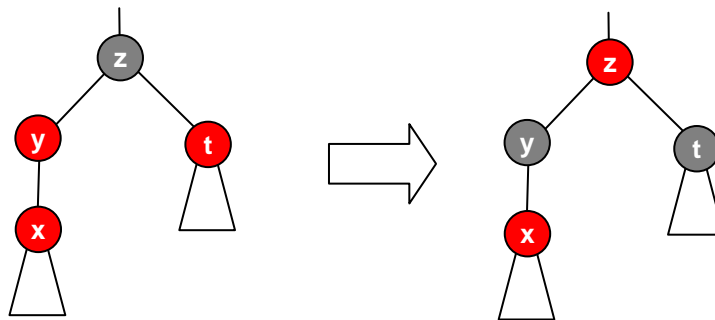
Caso Trivial: Si el padre del nodo insertado es **negro**, no se realiza ningún ajuste (el árbol es correcto).

En caso contrario se entra en un bucle donde **x** representa el nodo que se está comprobando. **x** es un nodo **rojo**, y los casos se refieren a situaciones en que su padre existe y es también **rojo**: Cuando el padre no exista, simplemente se cambia el color de **x** (que es la raíz) a **negro** y se termina el bucle, y si el padre existe y es **negro** se termina directamente el bucle.

Dependiendo del caso, se realizarán una serie de operaciones y o bien continuará el bucle (**x** pasa a ser otro nodo) o bien se terminará. En cada caso se supone que el nodo que se comprueba (**x**) es **rojo**, su padre (**y**) existe y es **rojo**, y su abuelo (**z**) existe (no puede ser raíz un nodo rojo). El nodo hermano del padre se denomina tío (**t**).

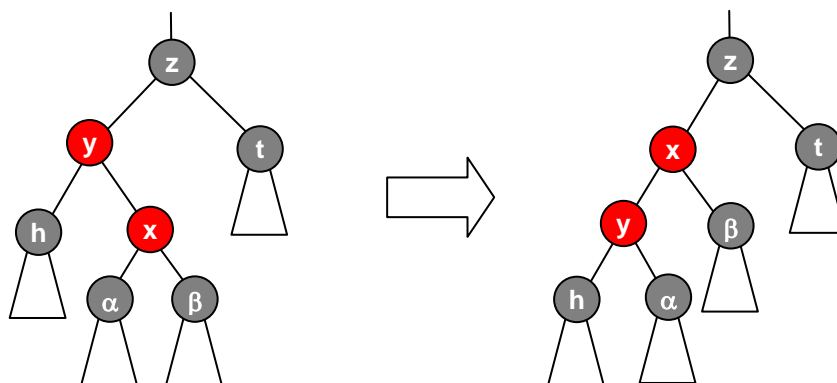
Atención: En lo que sigue se considera que el padre de **x** (nodo **y**) es un **hijo izquierdo**. Existen otros cinco casos no triviales cuando **y** sea un **hijo derecho**, que son totalmente **simétricos** respecto a los explicados.

Caso 1: Tío **rojo**, nodo **x** izquierdo o derecho.



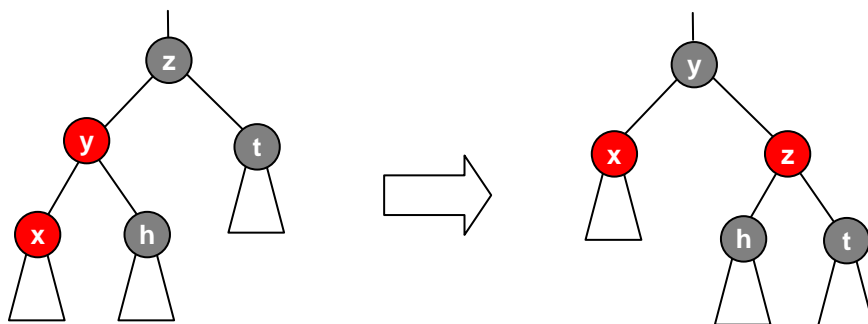
- **Solución:** Se cambia de color a los nodos **y**, **z** y **t**. Puede darse el problema de que el padre de **z** sea **rojo** y se viole de nuevo la primera condición
- **Iteración siguiente:** El nodo que se comprueba ahora es **z**.

Caso 2: Tío **negro**, nodo **x** es hijo derecho.



- **Solución:** Se efectúa una rotación **x-y**. Esto no resuelve el problema, pero hace que ahora el nodo que hay que comprobar sea un hijo izquierdo y por lo tanto se aplique el caso siguiente.
- **Iteración siguiente:** El nodo que se comprueba ahora es **y**. Se caerá en el Caso 3.

Caso 3: Tío negro, nodo x es hijo izquierdo.



- **Solución:** Se efectúa una rotación **z-y**, se cambia de color a **z** e **y**. Esto resuelve el problema.
- **Iteración siguiente:** El árbol cumple las condiciones. Se termina el bucle.

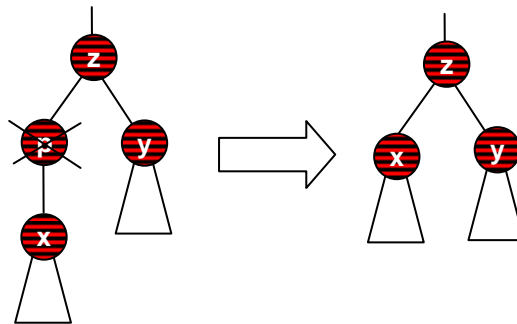
Borrado de un nodo

La operación se lleva a cabo de la misma manera que en los árboles binarios de búsqueda:

- Se busca el nodo a borrar (como los nodos hojas son nodos nulos, en caso de existir el nodo a borrar debe tener siempre dos hijos).
- Si es un nodo con dos hijos **no nulos**, se busca el mayor nodo (el más a la derecha) de su subárbol izquierdo (lo llamamos x), se intercambia sus datos con el nodo a borrar y se pasa a borrar el nodo x . Como es el nodo más a la derecha, su hijo derecho será nulo.

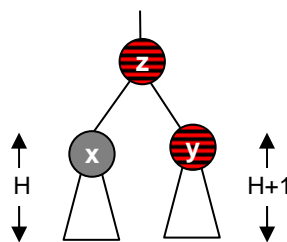
Como consecuencia de lo anterior, la operación de borrado siempre afecta a un nodo **con un hijo nulo** (o los dos, en ese caso a uno cualquiera de ellos le consideramos como un nodo normal y al otro como nulo).

El esquema de un borrado será entonces:



p es el nodo que se borra, x es el hijo no nulo o uno cualquiera de los hijos si ambos son nulos, z es el padre del nodo borrado e y es el nodo hermano del nodo borrado. Para ajustar el árbol tras el borrado es necesario conocer los nodos x y z . El nodo x puede ser nulo (al igual que y), pero el nodo z debe existir (es decir, el caso en que se tenga que borrar el nodo raíz se trata como un caso especial – simplemente se elimina la raíz y si el nuevo nodo raíz es de color rojo se cambia su color a negro).

La operación de ajuste (reestructuración) consistirá en una comprobación de los casos triviales (ver a continuación). Si no se está en un caso trivial se entra en un bucle, en cada iteración x y z apuntan a nodos del árbol con la siguiente estructura:



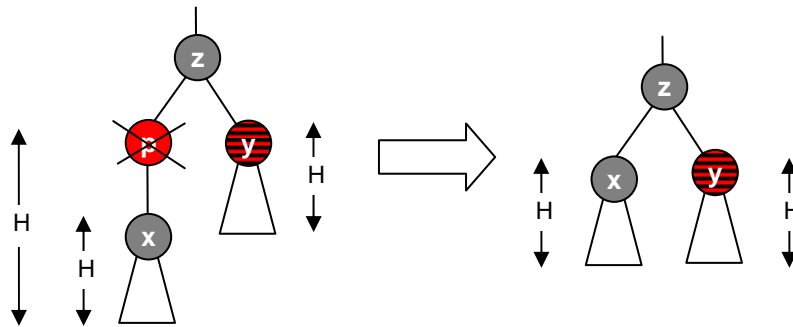
Los nodos z e y no son nulos y el nodo x puede serlo. Se comprueba en cuál de los cinco casos no triviales estamos, se efectúa la operación adecuada y se decide si se continúa a la siguiente iteración.

Atención: En lo que sigue se considera que x es un **hijo izquierdo**. Existen otros cinco casos no triviales cuando x sea un **hijo derecho**, que son totalmente **simétricos** respecto a los explicados.

Casos triviales

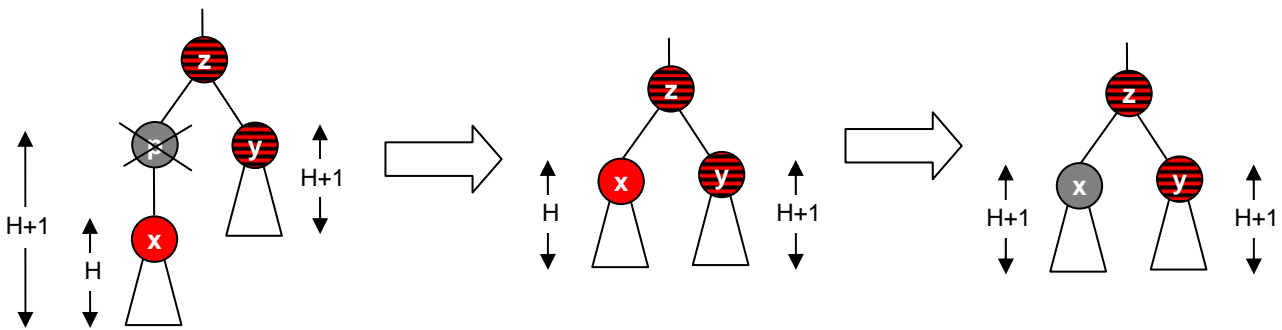
Nodo borrado es rojo:

El árbol sigue siendo rojo-negro, por lo que no hay que realizar ningún ajuste. En lo que sigue consideramos que el nodo borrado es **negro**.



El hijo, x , es rojo:

En este caso se incumple la condición de que los hijos de z tengan la misma altura negra. Cambiando el color de x a **negro** se restablece la condición.



En lo que sigue, entonces, consideraremos que el nodo borrado y su hijo son **negros**. Para detectar los casos es necesario fijarse en el color del nuevo padre (z) y, sobre todo, en el del hermano (y). En un caso incluso hay que tener en cuenta el color de los sobrinos (los hijos de y).

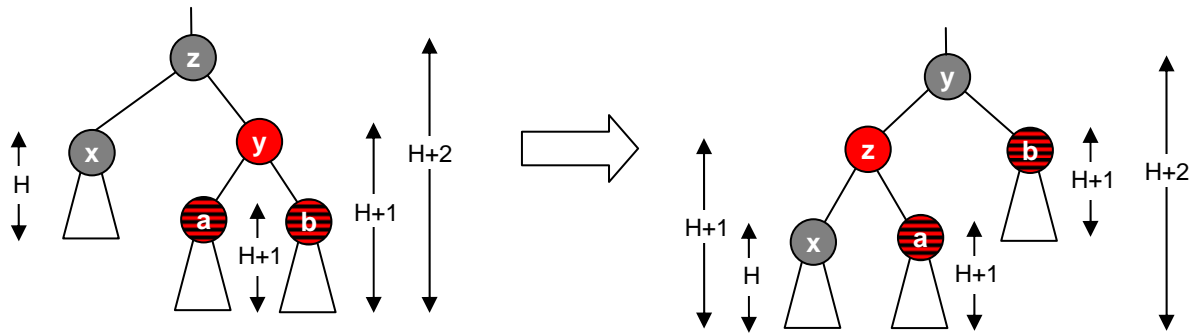
Se debe tener presente que tanto x como y pueden ser nulos (serían nodos **negros**) y que si un nodo es **rojo** entonces obligatoriamente no es nulo y por lo tanto tiene hijos.

Casos imposibles:

Hermano **negro** nulo:

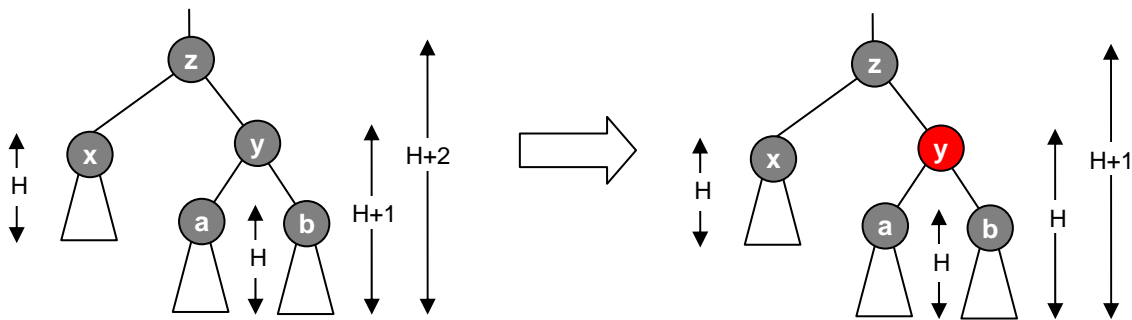
Si el hermano de x es **negro y nulo** tiene altura negra 1. Por lo tanto x debería tener (tras el borrado) una altura una unidad menor, es decir 0. Pero x es un nodo **negro**, y aunque sea nulo tendría altura 1. Por lo tanto, no se puede dar que x sea negro y esté desequilibrado respecto a un hermano nulo: El hermano debe existir.

Caso 1: Hermano **rojo**, padre **negro**



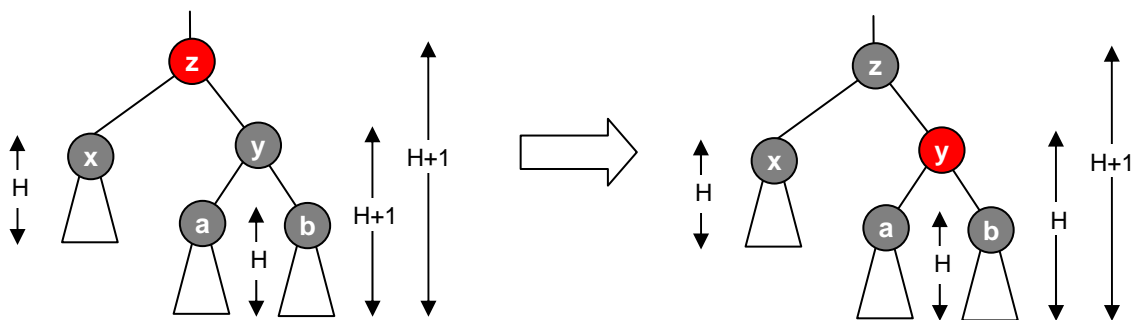
- **Solución:** Se hace una rotación padre-hermano y se cambian sus colores. El nodo **x** pasa a tener como nuevo hermano al nodo **a**, sigue teniendo una altura menor en uno que la de su hermano, pero ahora su padre es **rojo** y por lo tanto caera en alguno de los siguientes casos (dependiendo del color de **a**): Caso 3, 4 o 5.
- **Iteración siguiente:** Se vuelve a comprobar con los mismo nodos **x** y **z**, se garantiza que no se cae en éste caso.

Caso 2: (Hermano **negro** no nulo, sobrinos **negros**, padre **negro**)



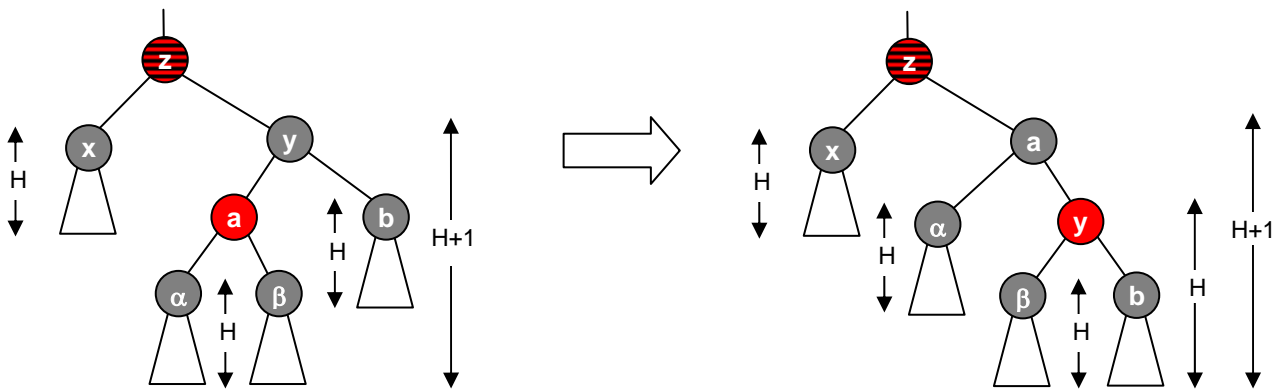
- **Solución:** Se cambia el color del hermano (**y**) a **rojo**. Con ello los nodos **x** e **y** pasan a tener la misma altura negra. El problema es que la altura de **z** ha disminuido.
- **Iteración siguiente:** Se vuelve a comprobar, ahora el nodo llamado **x** es el nodo **z** y el nodo llamado **z** es el padre de **z**. **Nota:** Si **z** es el raíz el arbol cumple todas las condiciones y se termina el bucle.

Caso 3: (Hermano **negro** no nulo, sobrinos **negros**, padre **rojo**)



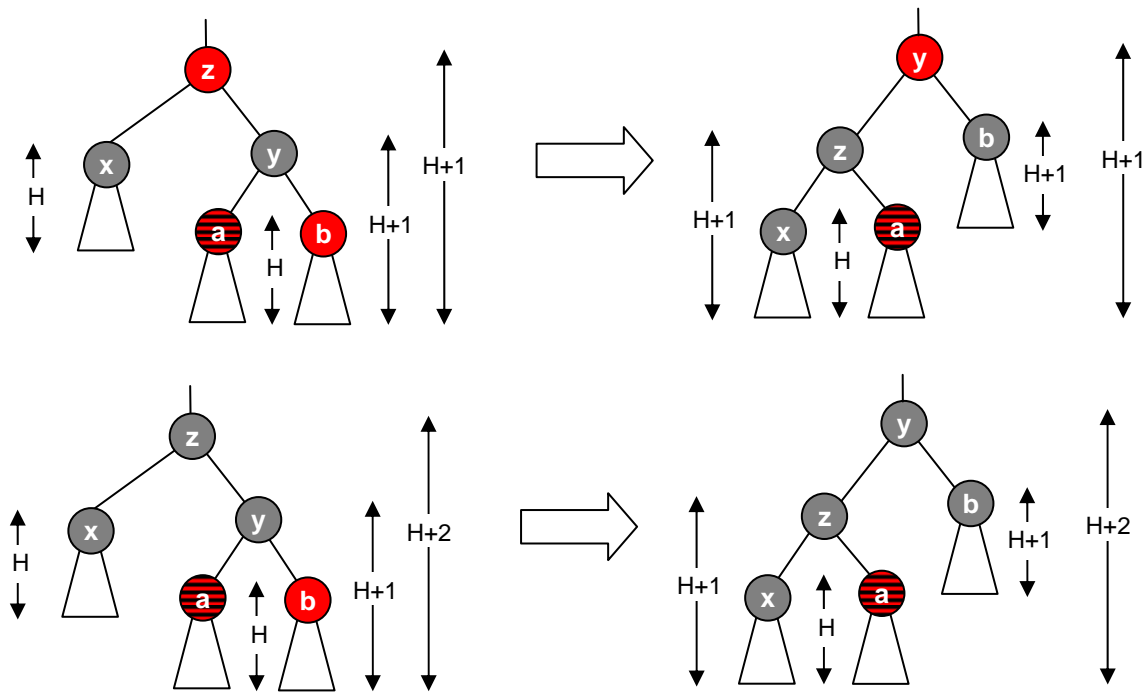
- **Solución:** Se cambia el color del hermano (**y**) a **rojo** y el del padre (**z**) a **negro**. Con ello los nodos **x** e **y** pasan a tener la misma altura negra. La altura de **z** sigue siendo la misma.
- **Iteración siguiente:** El arbol ya cumple todas las condiciones. Se termina el bucle.

Caso 4: (Hermano **negro** no nulo, sobrinos **rojo/negro**, padre **cualquier color**)



- **Solución:** Se realiza una rotación hermano-sobrino izquierdo y se cambian sus colores. Los hijos del sobrino izquierdo existen (aunque pueden ser nulos) y son negros, ya que el sobrino izquierdo es **rojo**. El nodo **x** pasa a tener como hermano al nodo **a** y sigue teniendo una altura negra menor en uno que la de su hermano.
- **Iteración siguiente:** Se vuelve a comprobar con los mismo nodos **x** y **z**, se caera en el caso 5 ya que el hermano sigue siendo negro y los sobrinos son **negro** y **rojo**.

Caso 5: (Hermano **negro** no nulo, sobrinos **cualquiera/rojo**, padre **cualquier color**)



- **Solución:** Se realiza una rotación padre-hermano y se cambia el color de la siguiente forma: El padre (**z**) pasa a ser **negro**, el hermano (**y**) toma el mismo color que el que tenía originalmente **z**, el sobrino derecho pasa de **rojo** a **negro** (este sobrino debía existir ya que era rojo).
- **Iteración siguiente:** El arbol ya cumple todas las condiciones. Se termina el ajuste.

Consideraciones: En ningún caso se cambia el color de **x**, por lo que puede perfectamente ser un nodo nulo.