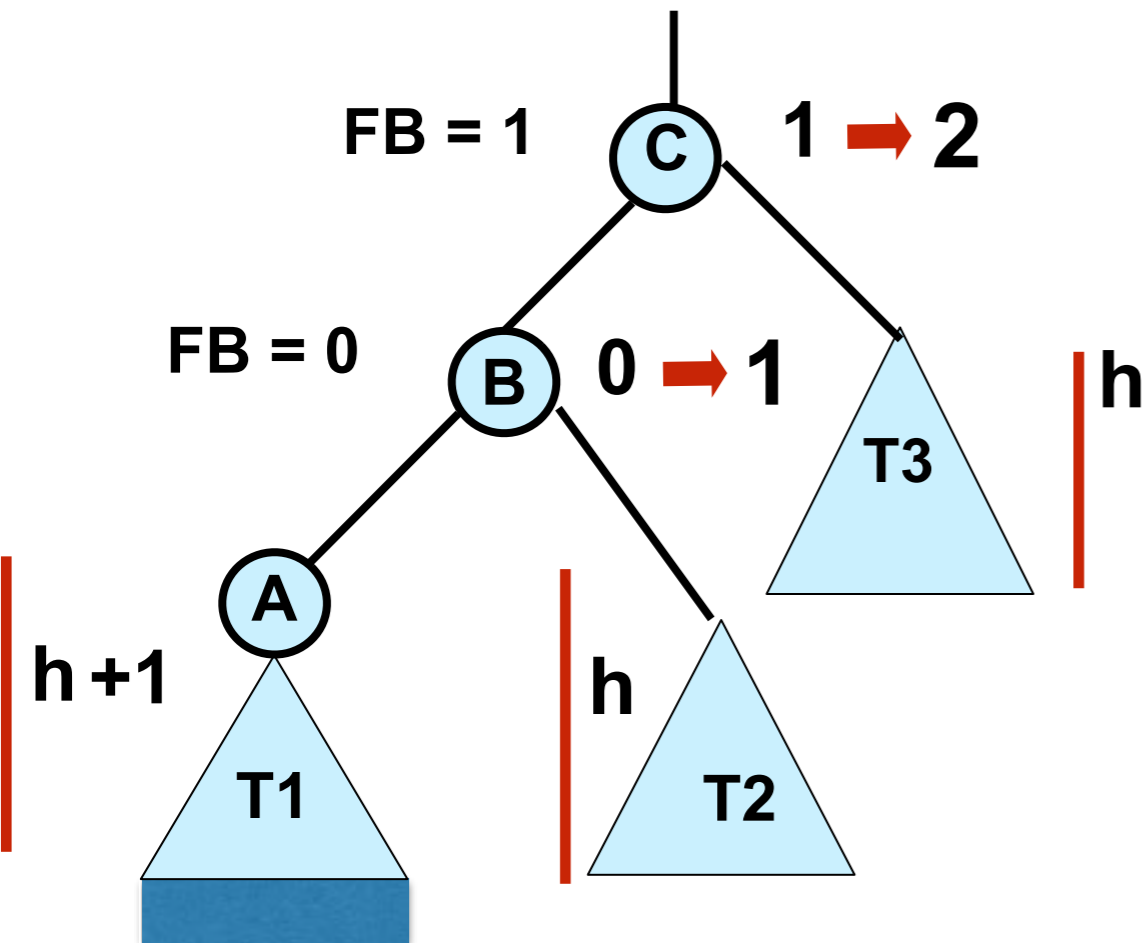
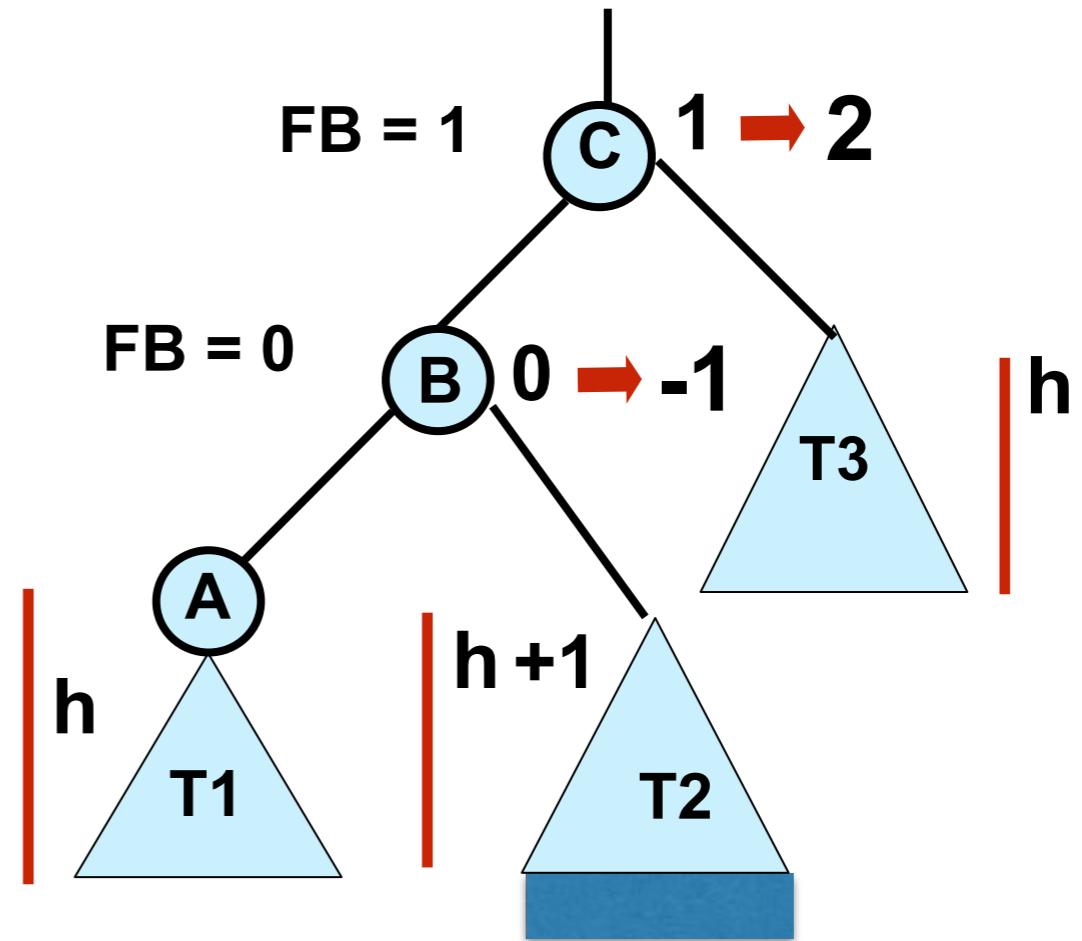


La cancellazione in un AVL

Casi di sbilanciamento inserimento



left- left



left- right

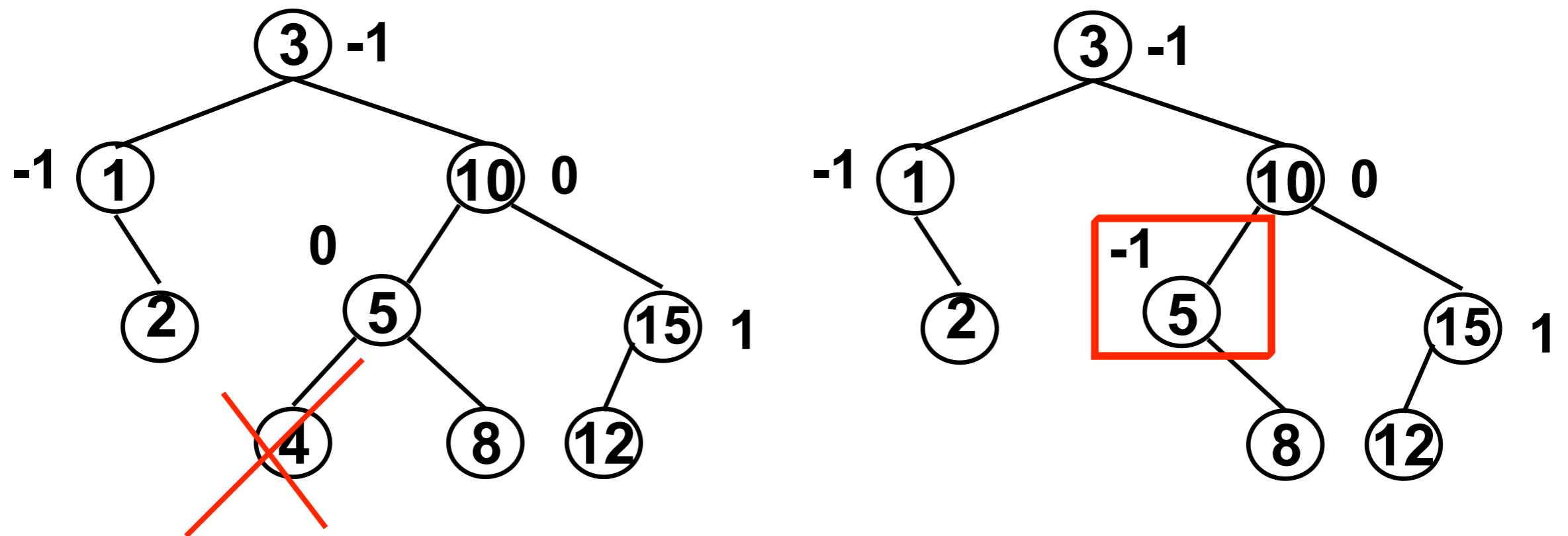
A questi si devono aggiungere i simmetrici: right-right, right-left

La cancellazione in un AVL

La cancellazione avviene come in un ABR, con la necessità di aggiornare i fattori di bilanciamento ed eventualmente ribilanciare l'albero.

Esempio:

Cancellazione di una foglia, 4

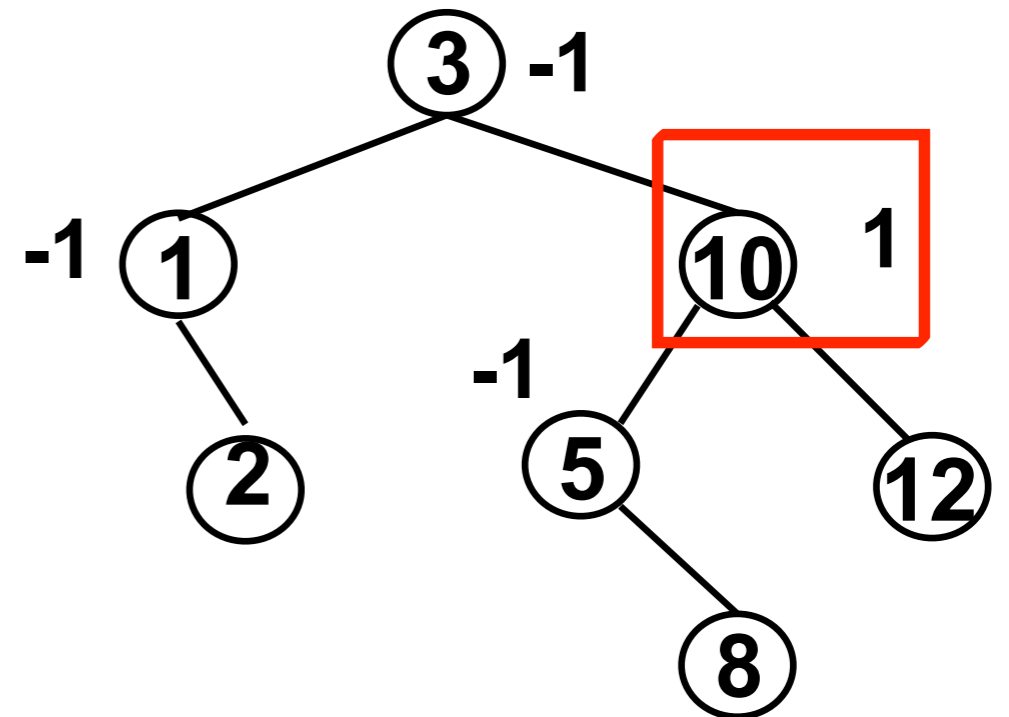
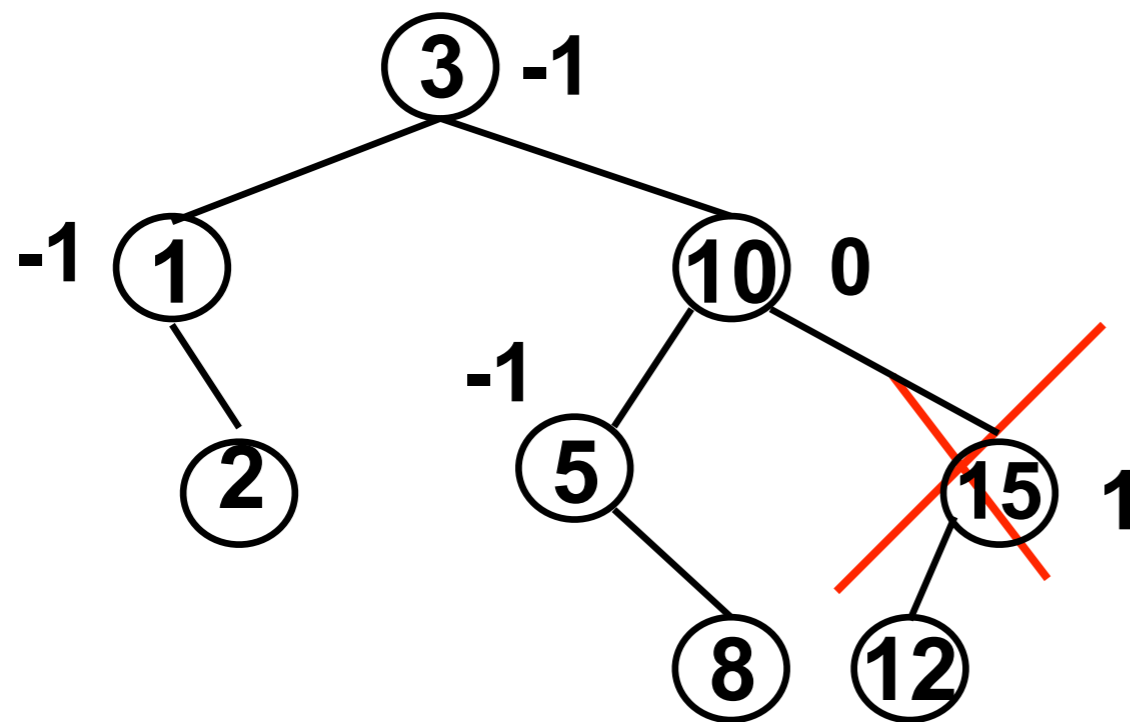


unico FB da aggiornare

La cancellazione in un AVL

Esempio:

Cancellazione di un nodo con un solo figlio, 15

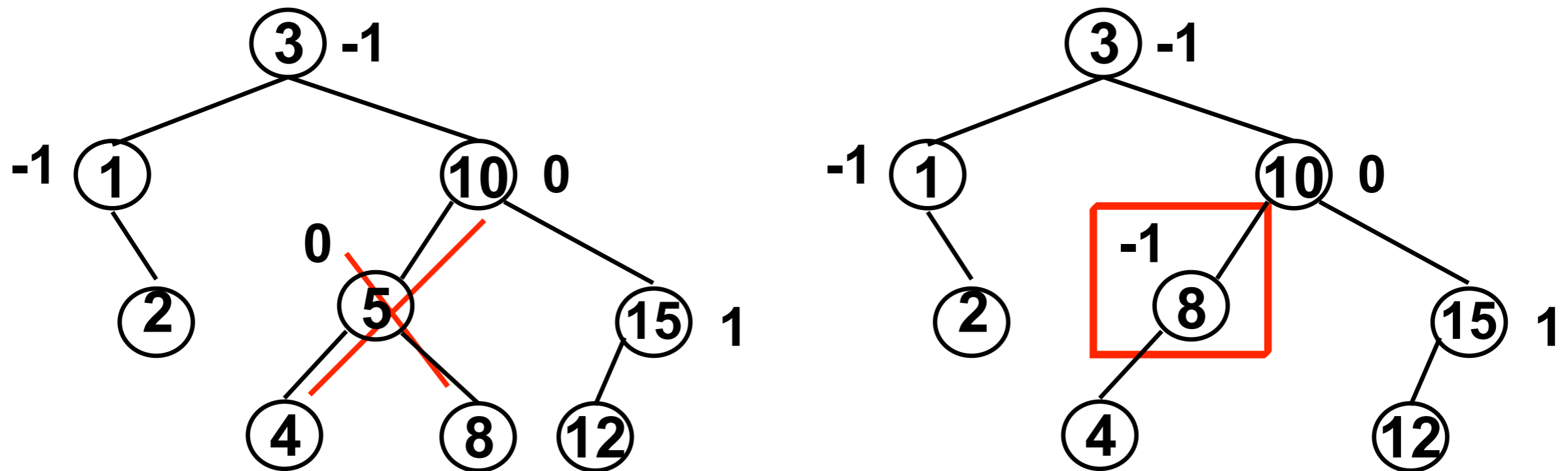


unico FB da aggiornare

La cancellazione in un AVL

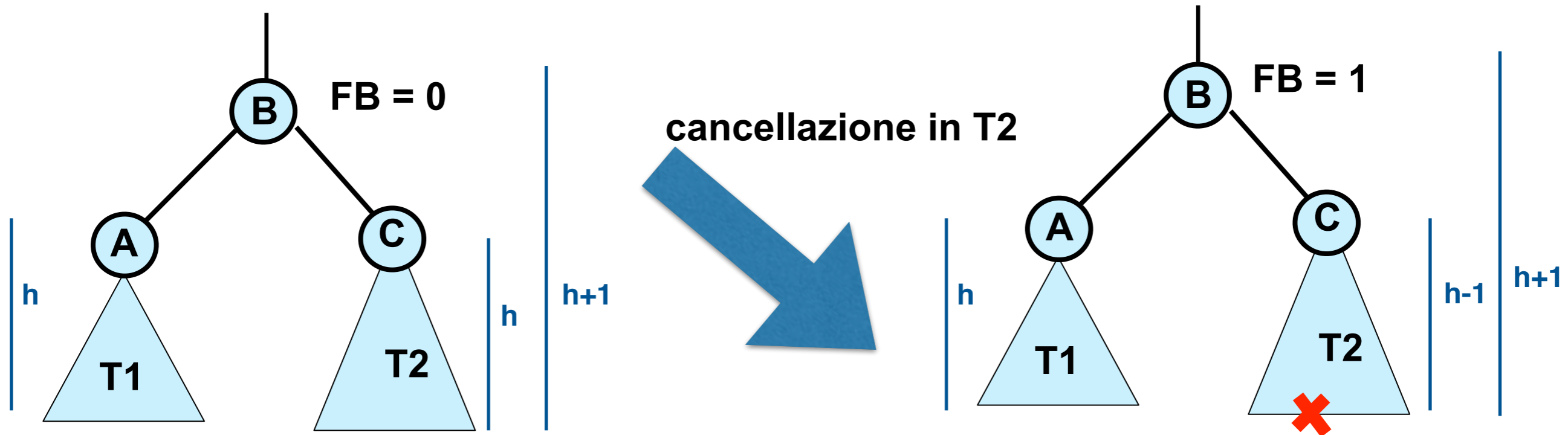
Esempio:

Cancellazione di un nodo con DUE figli, 5



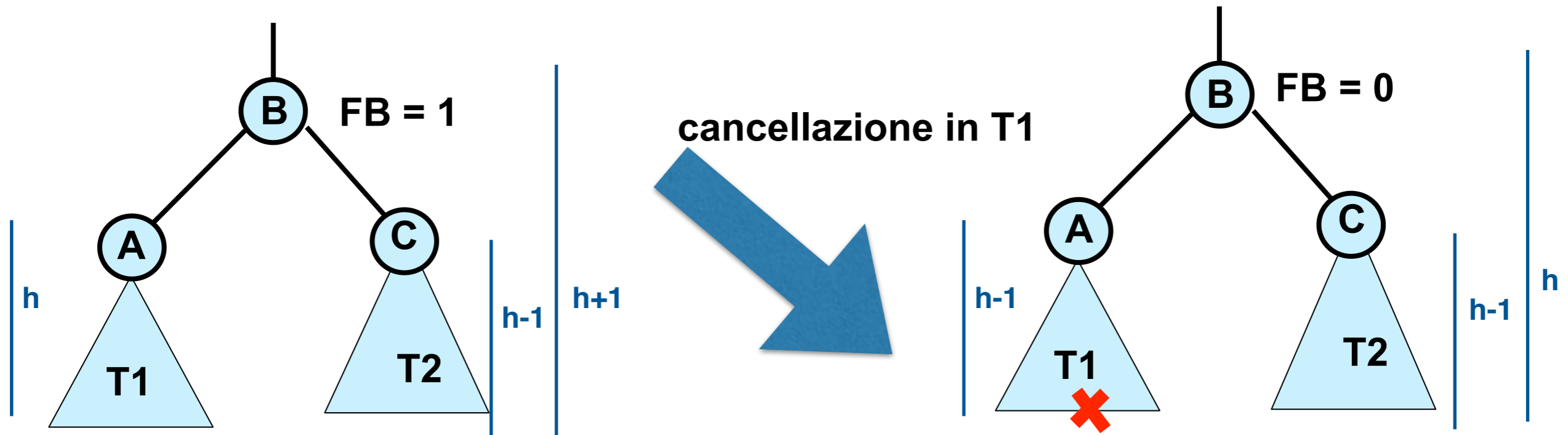
unico FB da aggiornare

La cancellazione in un AVL: aggiornamento FB



Se risalendo verso la radice il fattore di bilanciamento passa da 0 a 1, o da 0 a -1, si può fermare il processo di aggiornamento dei fattori di bilanciamento, perché vuol dire che la cancellazione non ha prodotto una diminuzione di altezza di tutto l'albero ma solo in un suo sotto albero.

La cancellazione in un AVL: aggiornamento FB

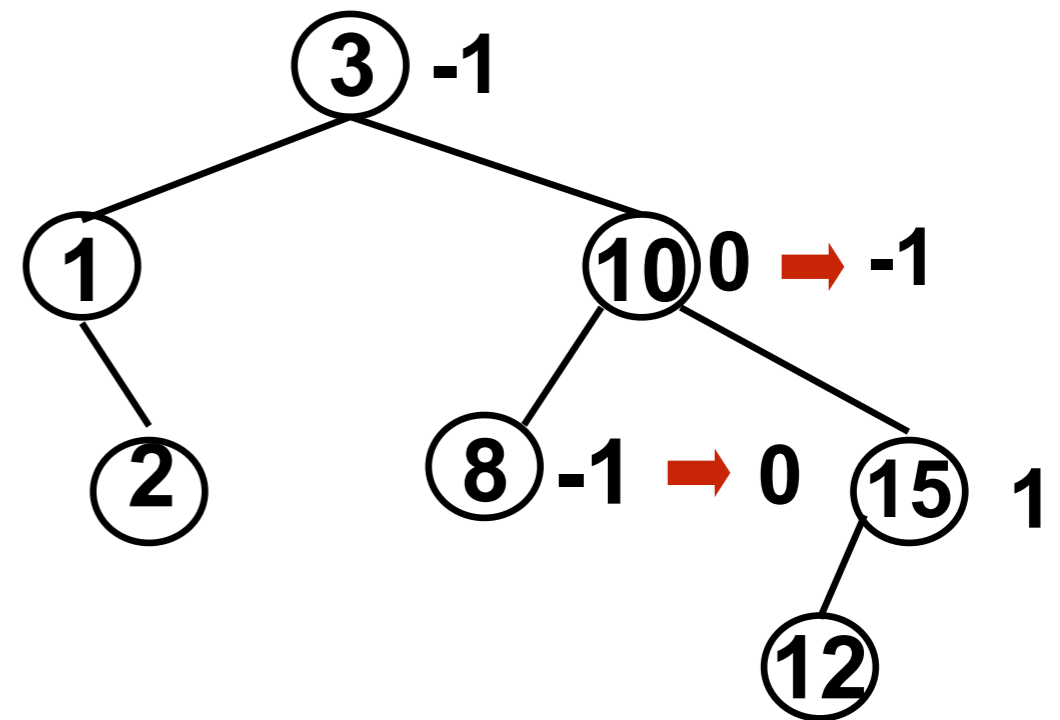
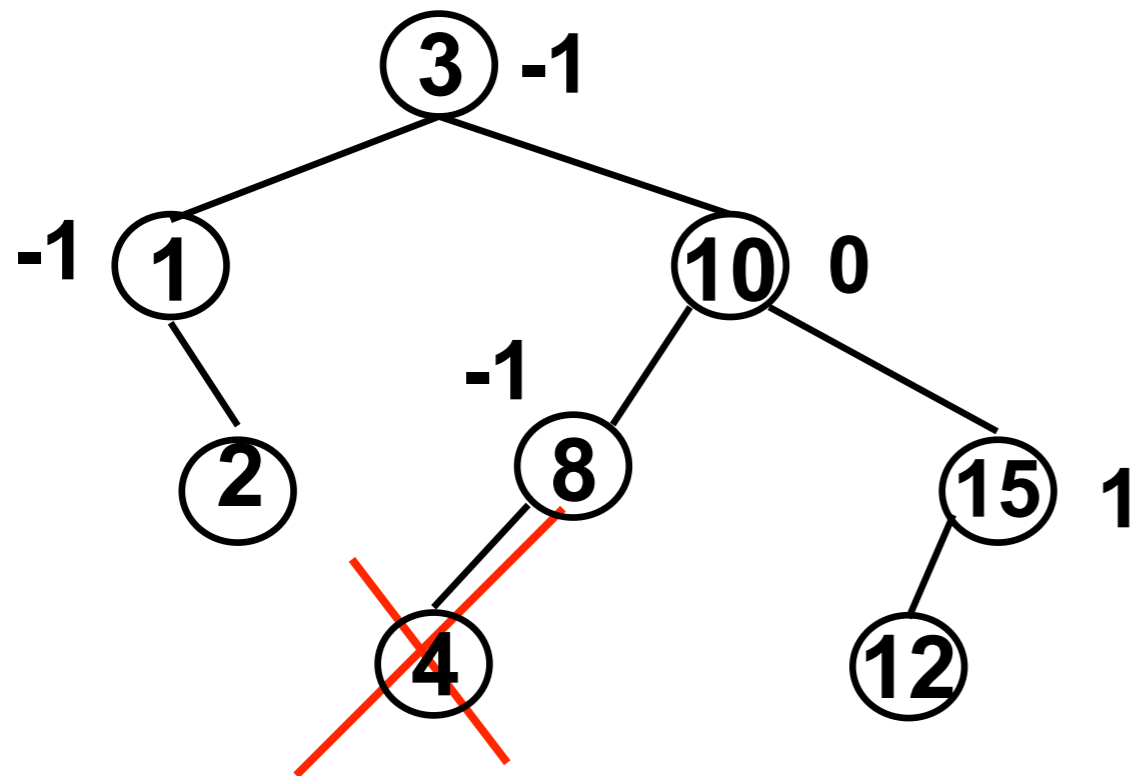


Se risalendo verso la radice il fattore di bilanciamento passa da 1 a 0, o da -1 a 0, non si può fermare a B il processo di aggiornamento dei fattori di bilanciamento, perché vuol dire che la cancellazione ha prodotto una diminuzione di altezza di tutto l'albero non solo in un suo sotto albero.

La cancellazione in un AVL

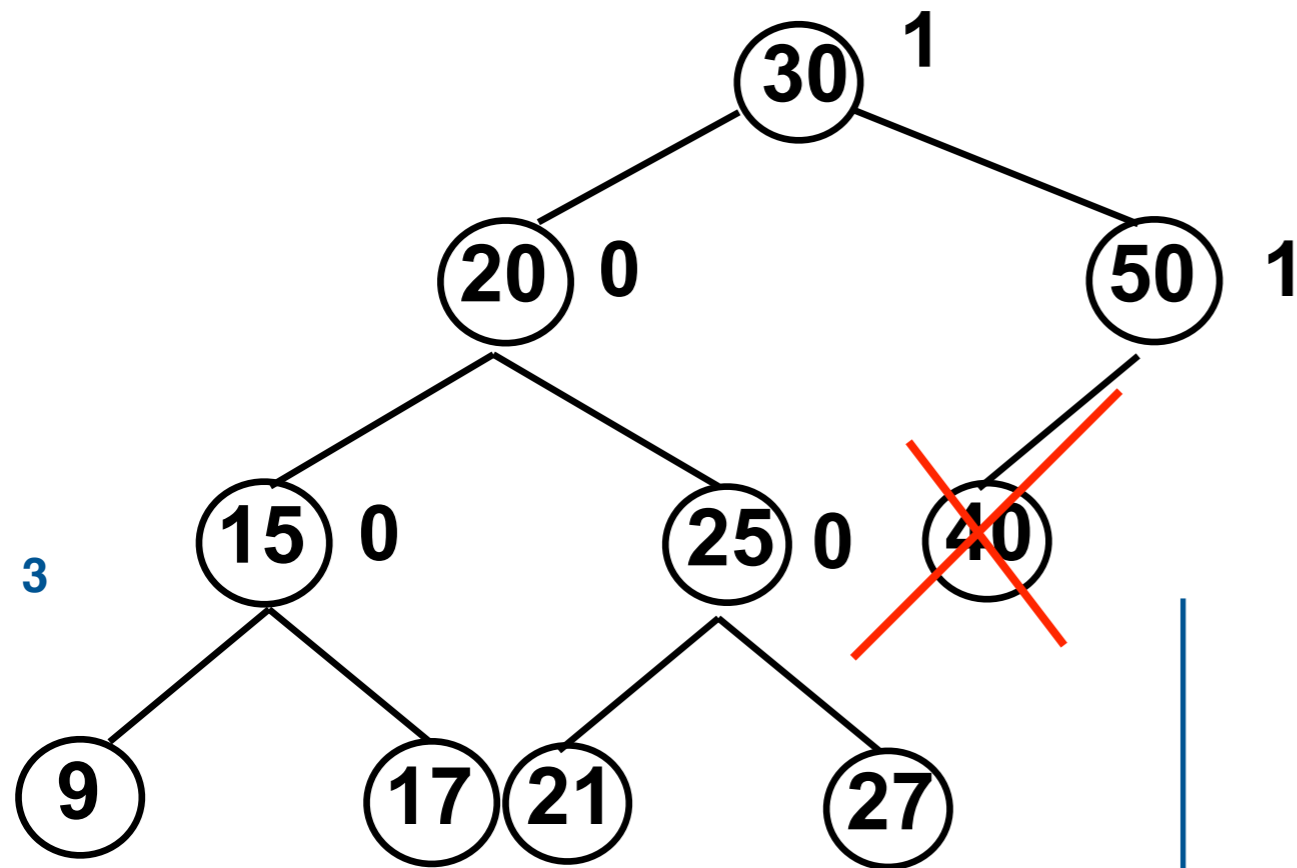
Esempio:

Cancellazione e aggiornamento di più FB

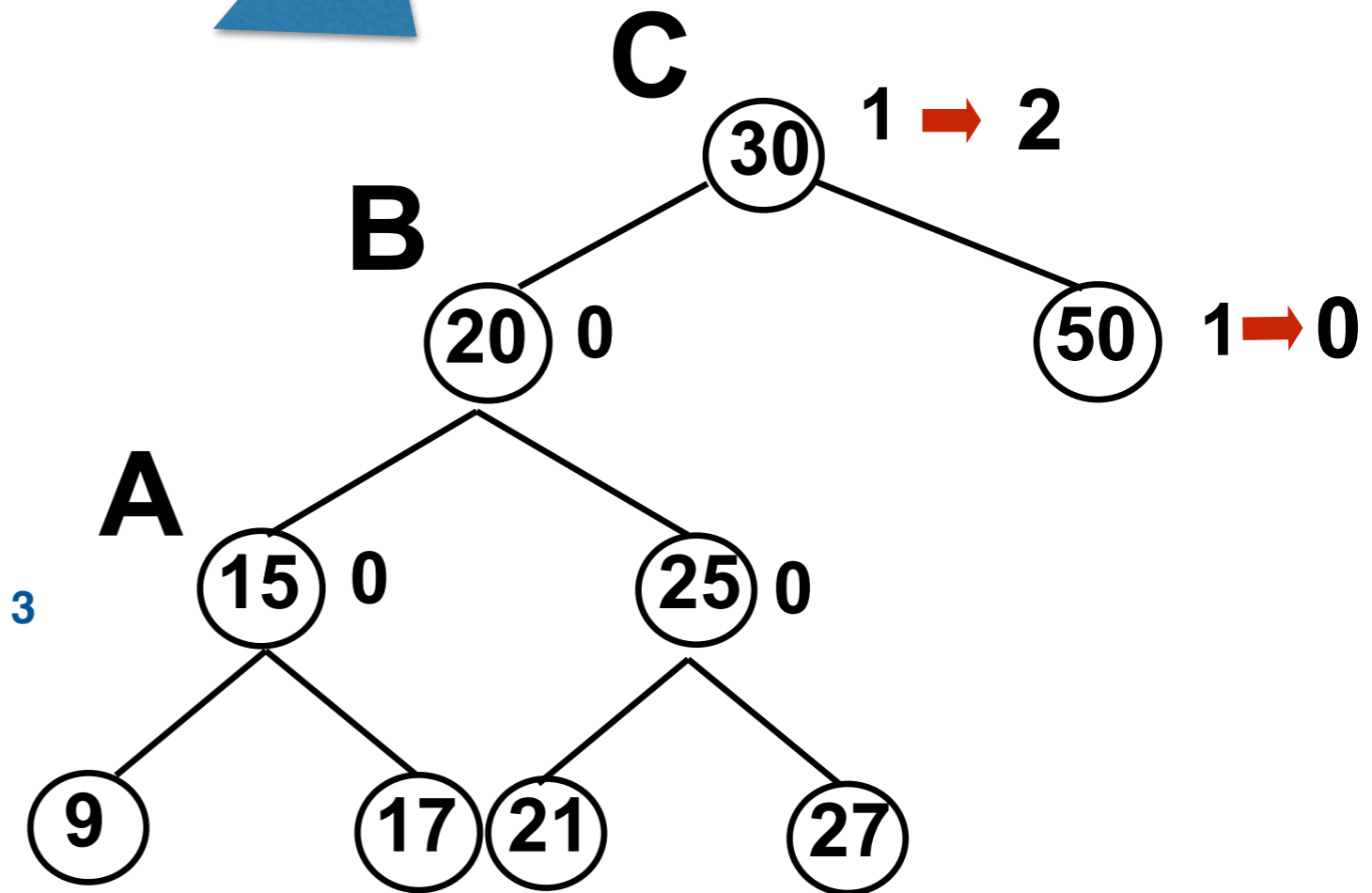
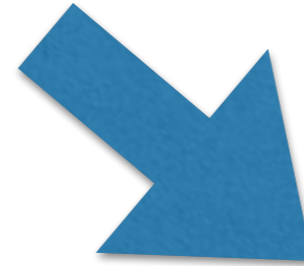


FB da aggiornare

Esempio cancellazione con sbilanciamento

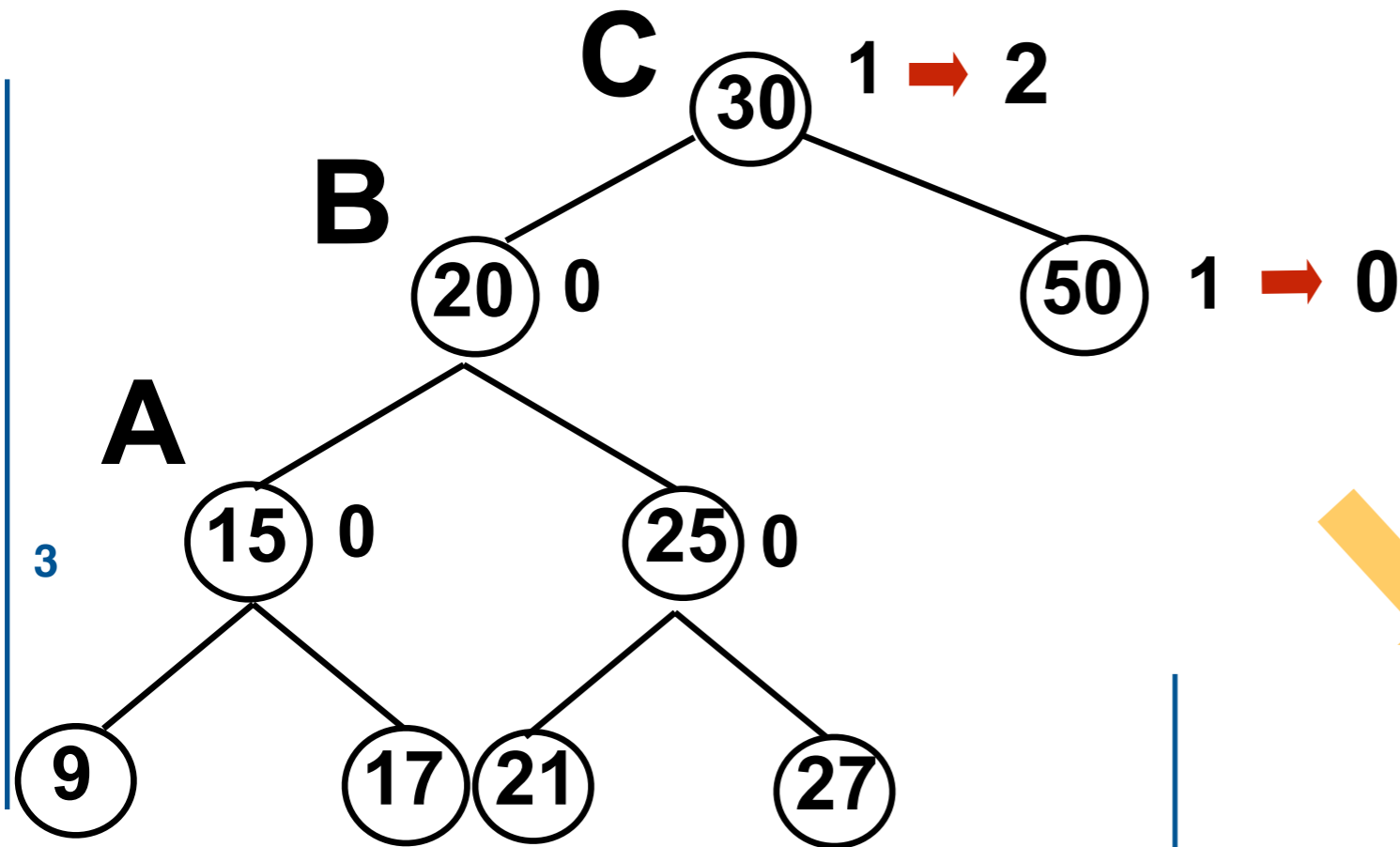


Cancellazione di 40

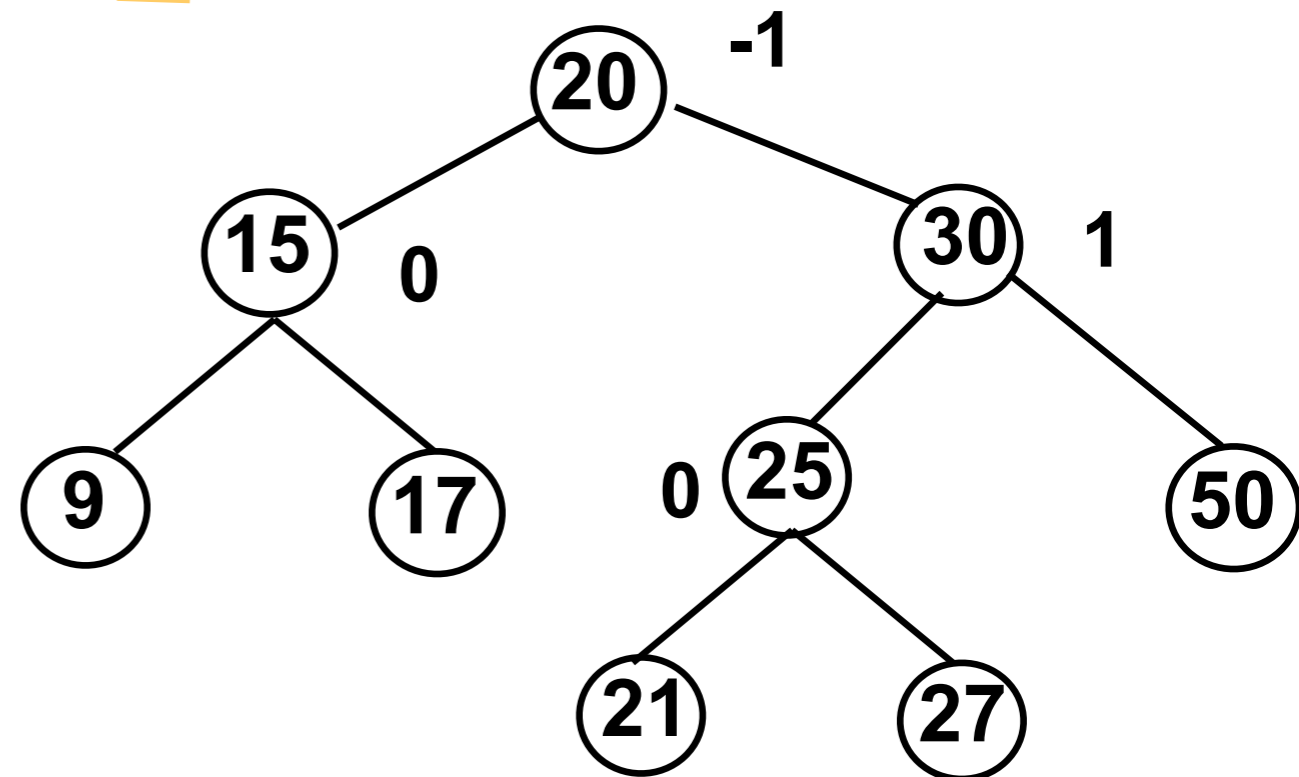


Risalendo da destra da 50 ,
passiamo a un nodo con FB
= 2, il cui figlio sinistro ha
FB = 0 questo identifica il
caso left left1.

Il ribilanciamento



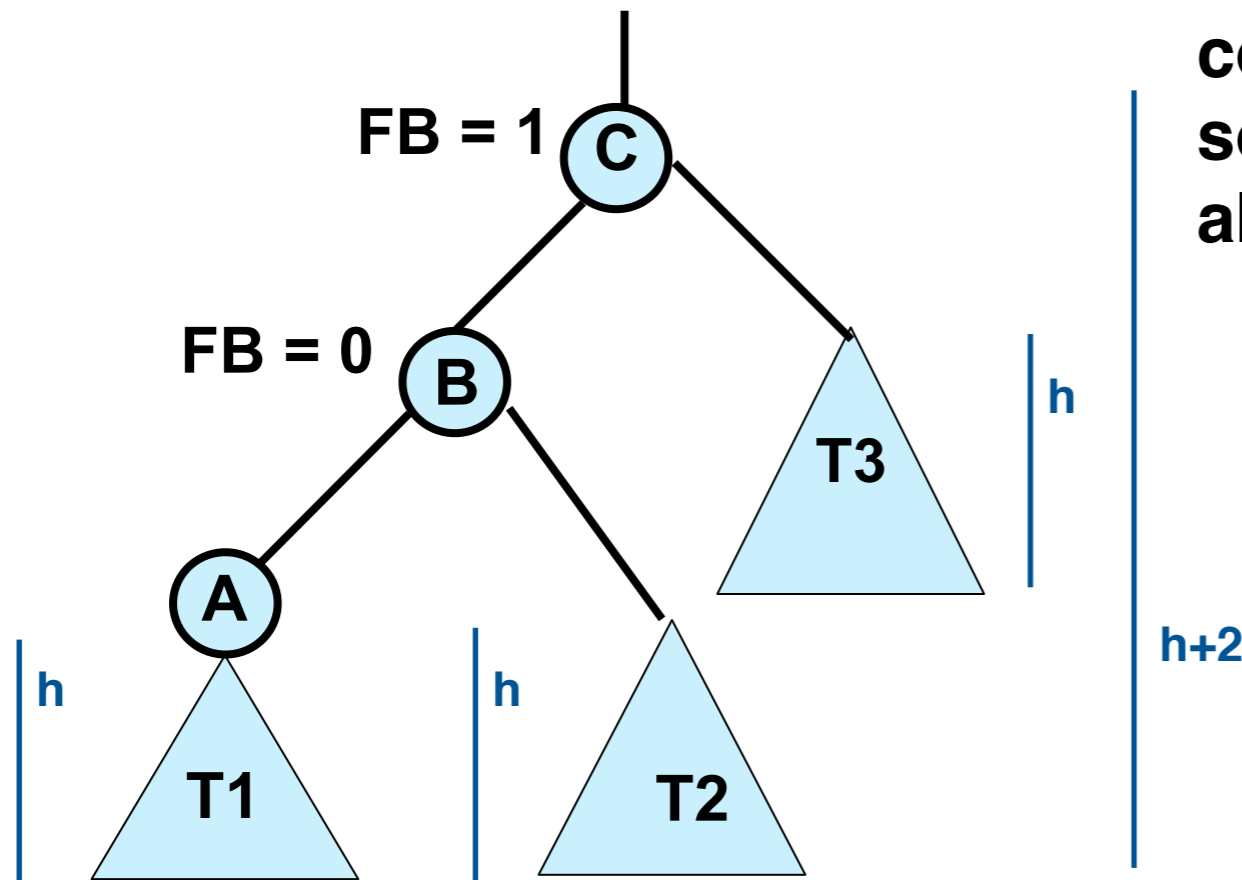
Rotazione a
destra su C



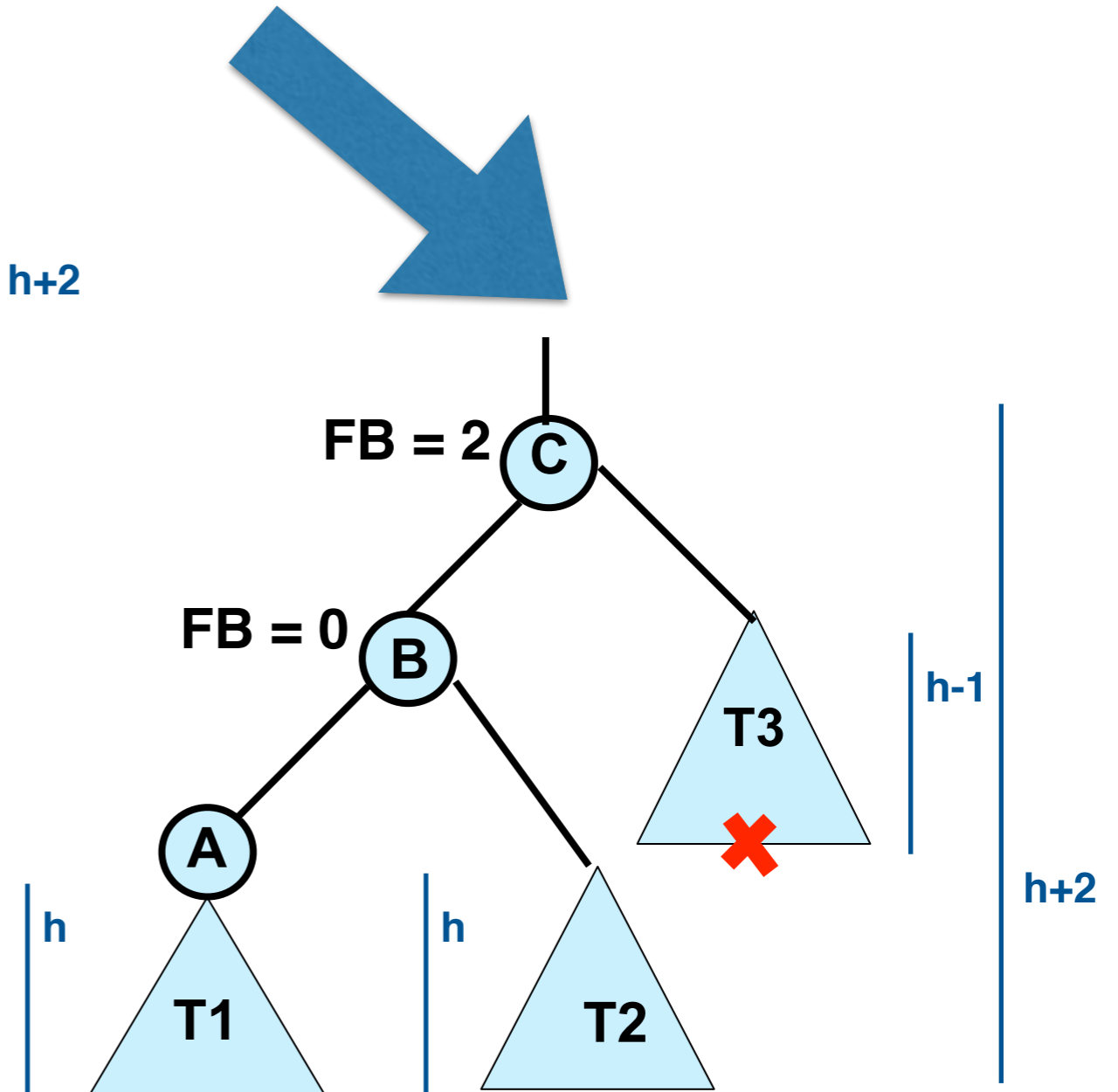
L'altezza dell'albero radicato in 30 prima della cancellazione è la stessa dell'albero ottenuto dopo la cancellazione e la rotazione. Quindi se l'albero radicato in 30 fosse stato un sotto albero, non sarebbe stato necessario andare a controllare i fattori di bilanciamento degli antenati di 20, nuova radice del sotto albero.

La cancellazione: caso left-left 1

Si chiama LEFT LEFT perché corrisponde ad un inserimento nel sotto albero sinistro del sotto albero sinistro, T1.

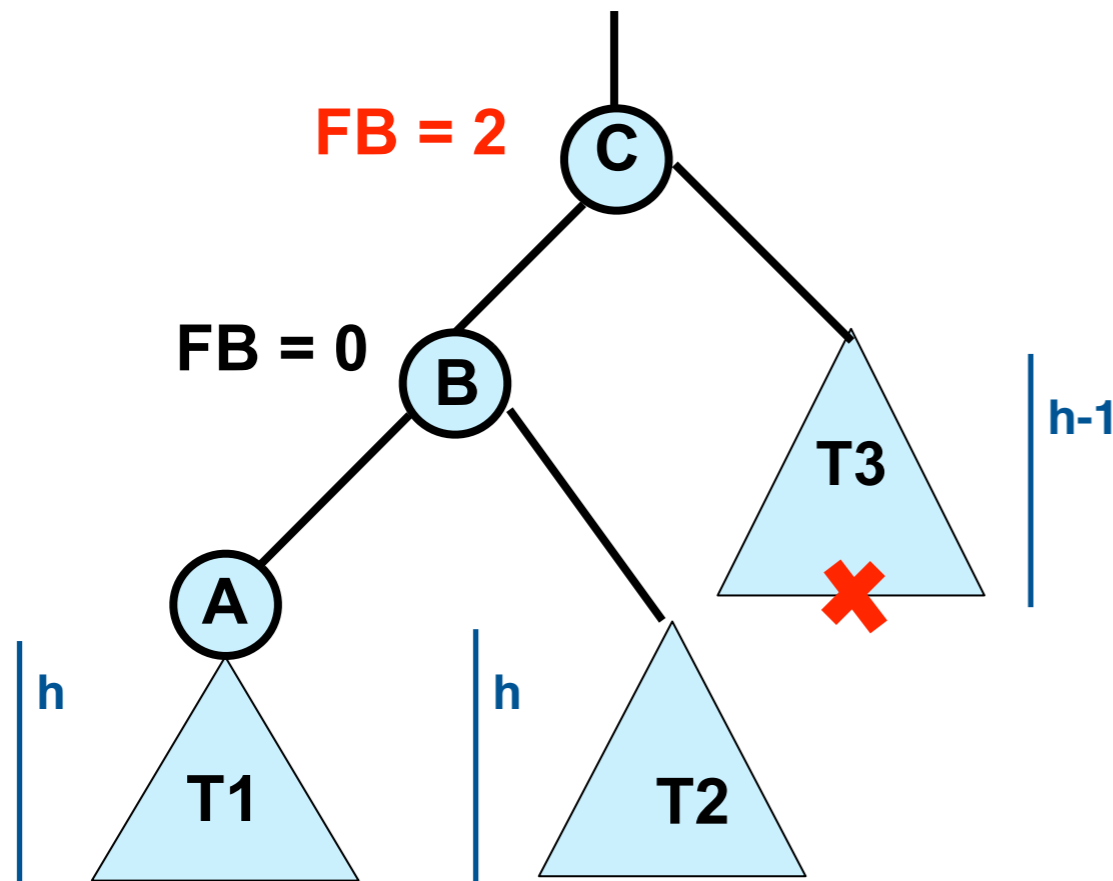


cancellazione in T3

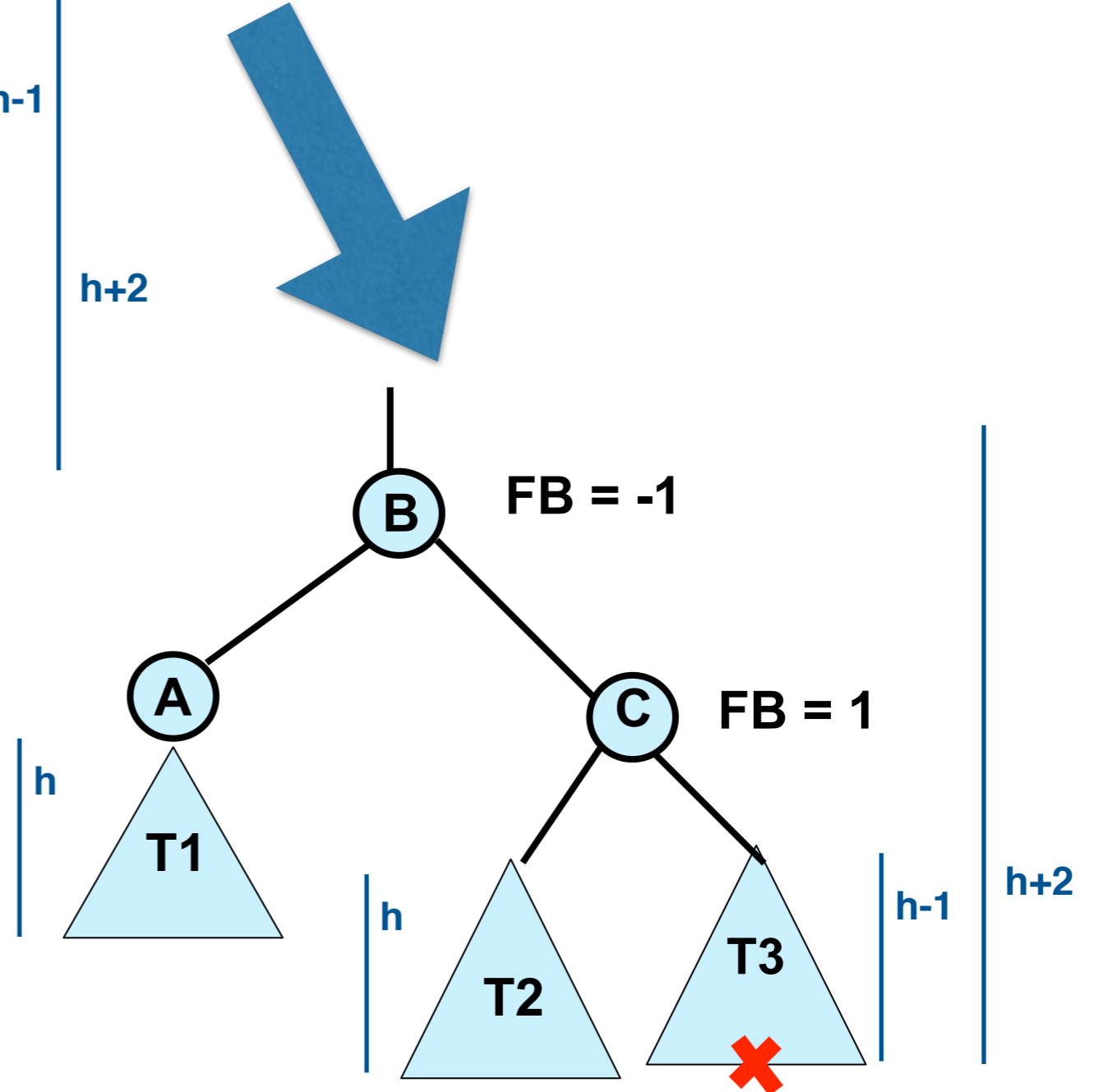


La cancellazione in T3 ne provoca una diminuzione di altezza, quindi nel nodo C il fattore di bilanciamento diventa illegale, anche se l'altezza del sotto albero radicato in C non cambia. Risalendo verso la radice da destra si incontra un nodo con $FB = 2$, mentre il suo figlio sinistro ha $FB = 0$, questo identifica il caso left-left1.

CASO LEFT LEFT 1- rotazione



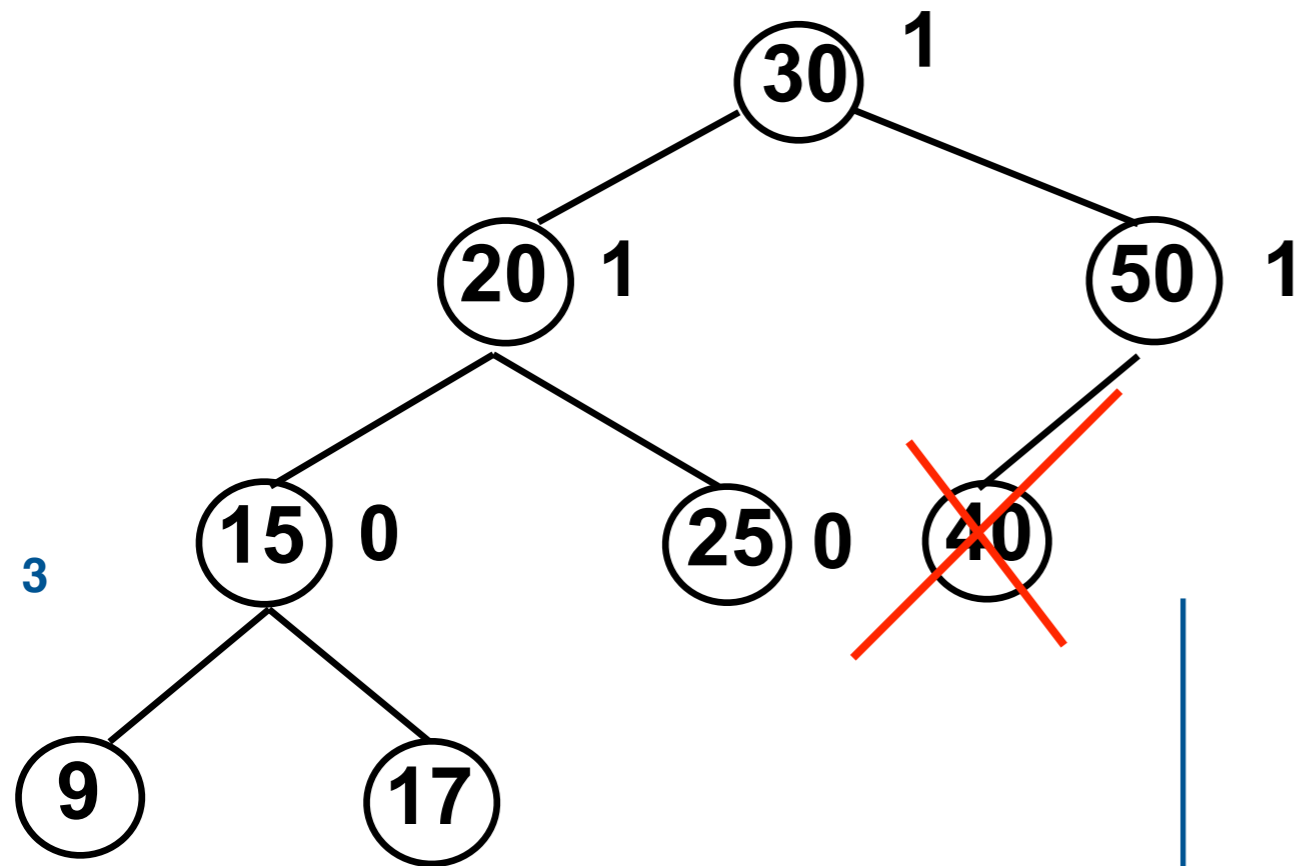
rotazione a destra su C



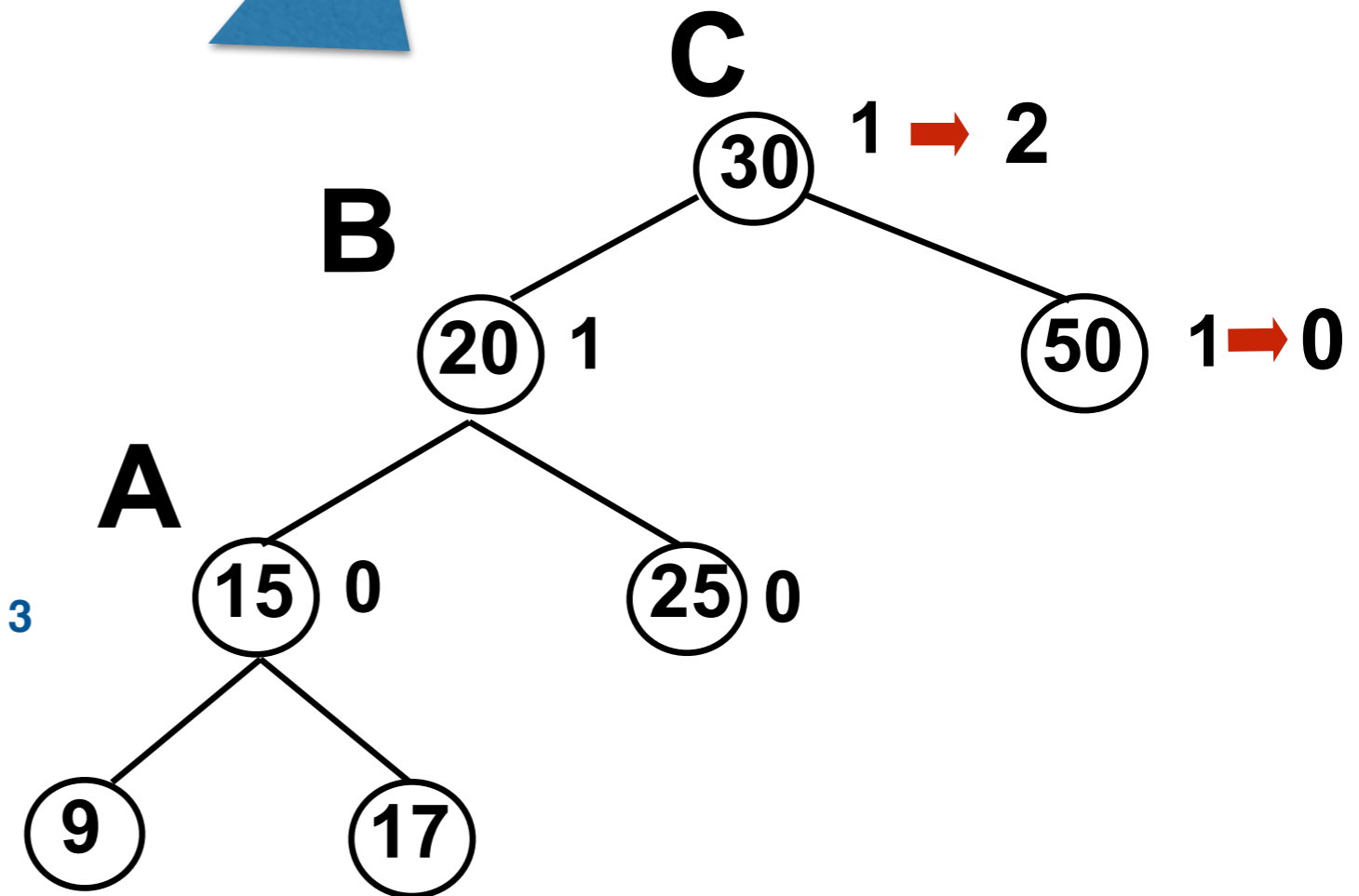
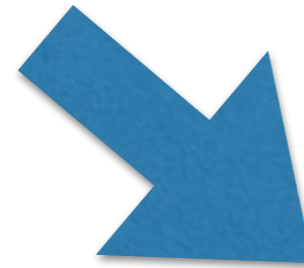
L'altezza del sottoalbero radicato in B, che rimpiazza quello radicato in C, non è cambiata. Quindi non sono necessarie altre operazioni.

$FB = 1$ di C diventa -1 per B.

Esempio cancellazione con sbilanciamento

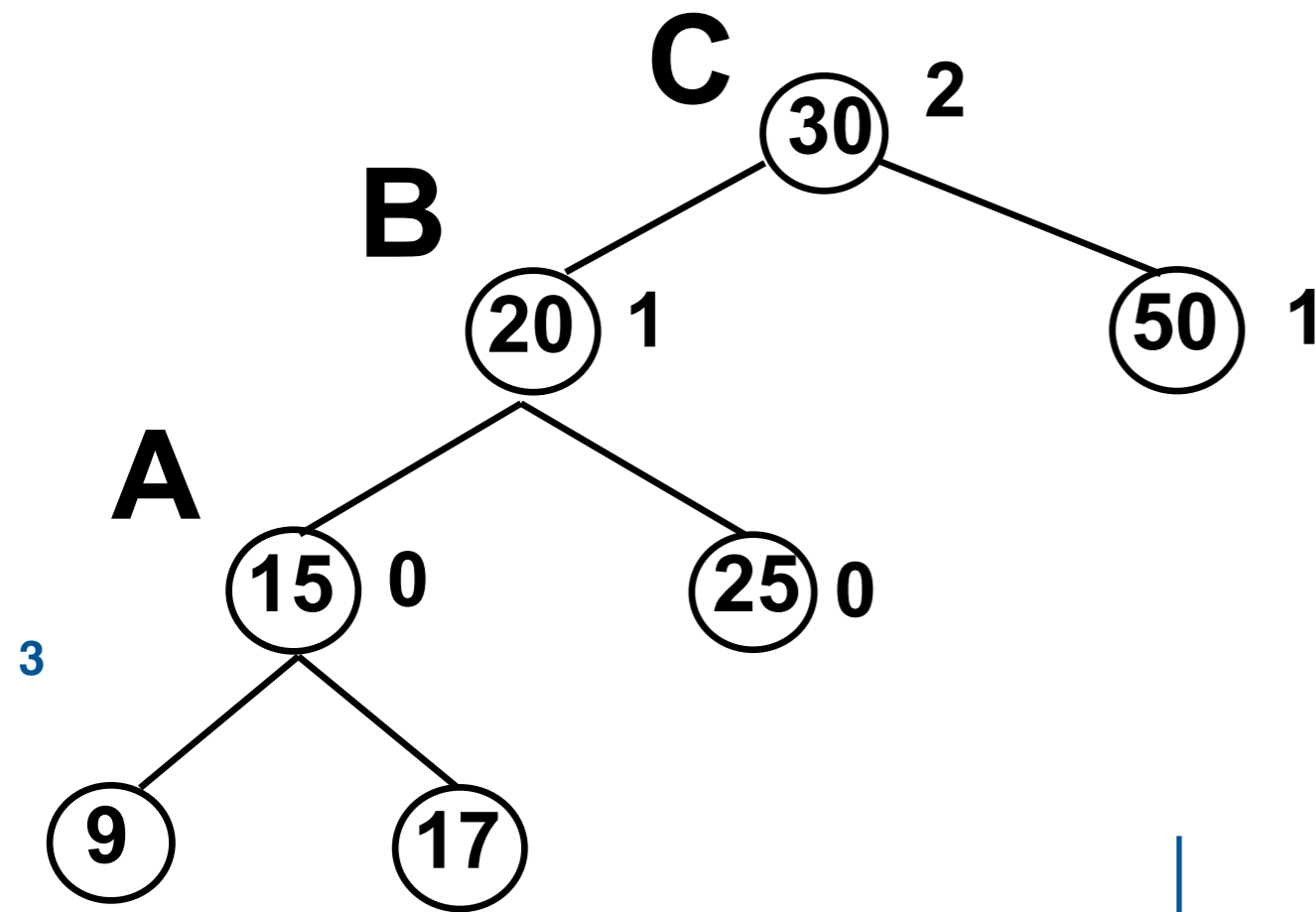


Cancellazione di 40

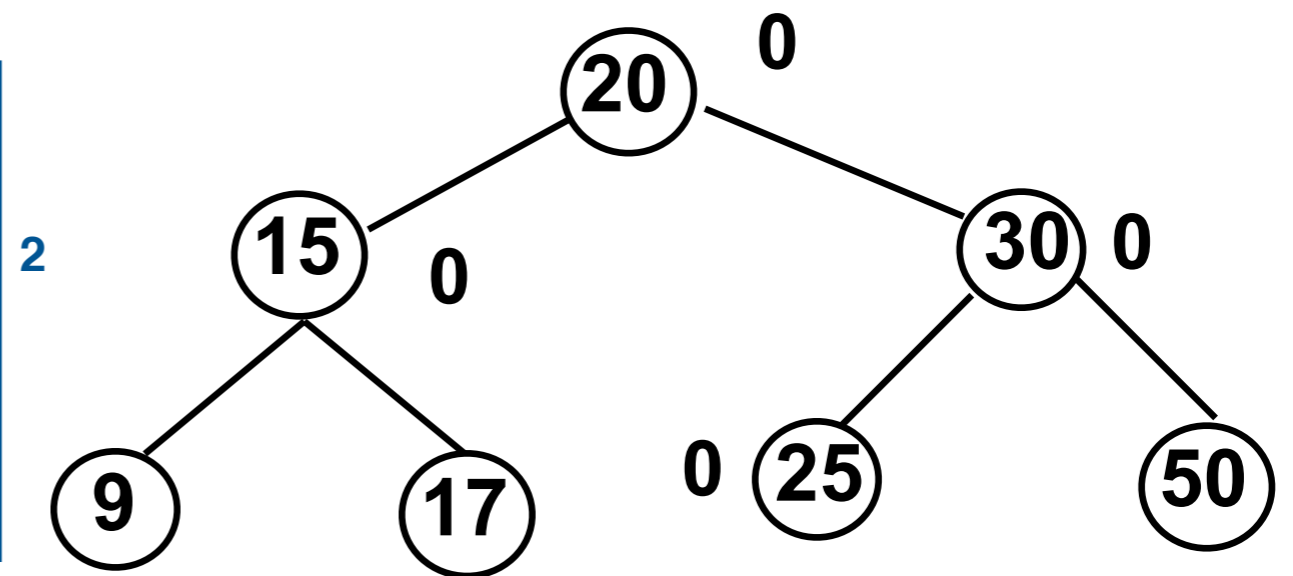


Risalendo da 50, passiamo a un nodo con $FB = 2$, il cui figlio sinistro ha $FB = 1$, questo identifica il caso left left2.

Il ribilanciamento

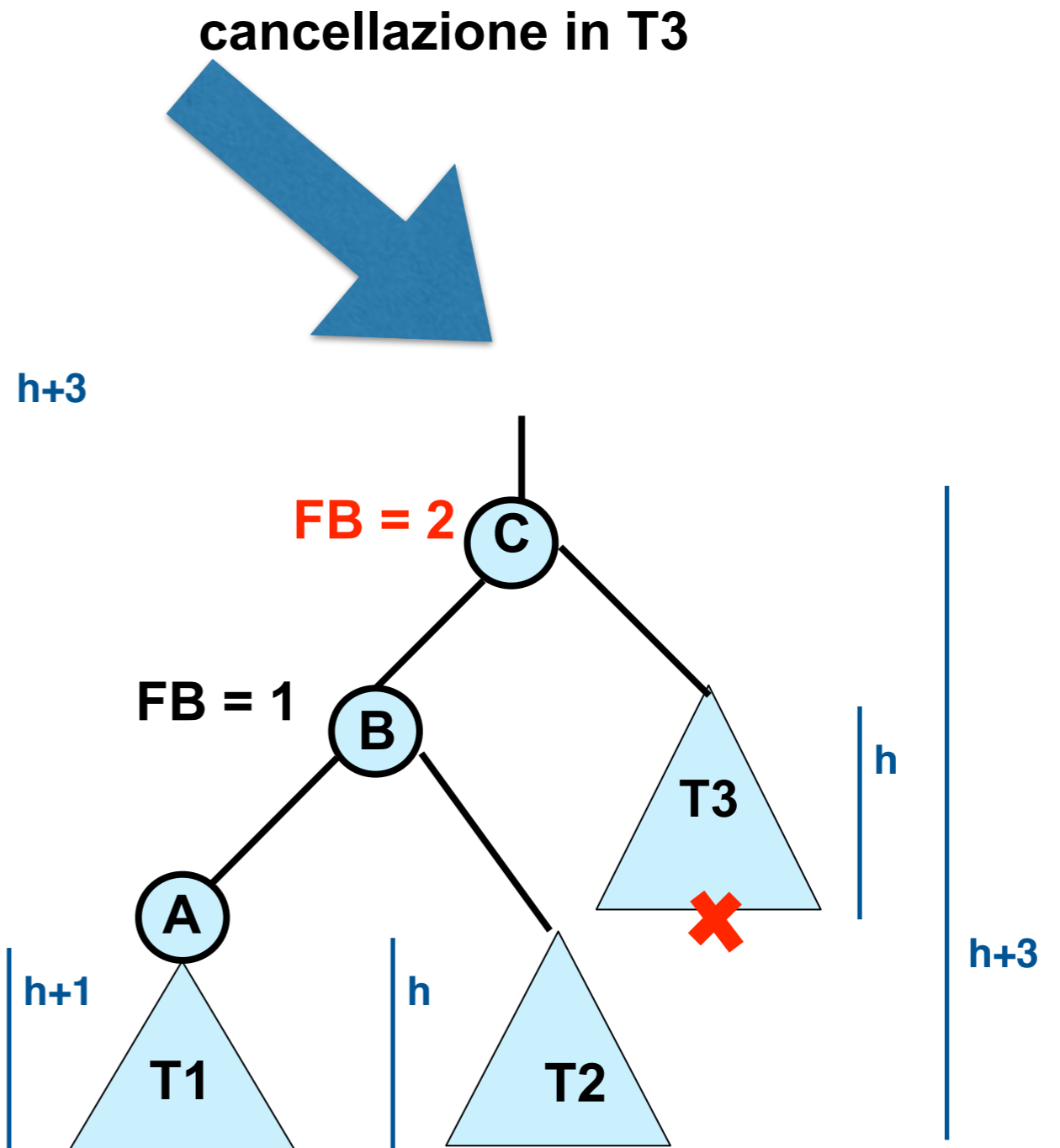
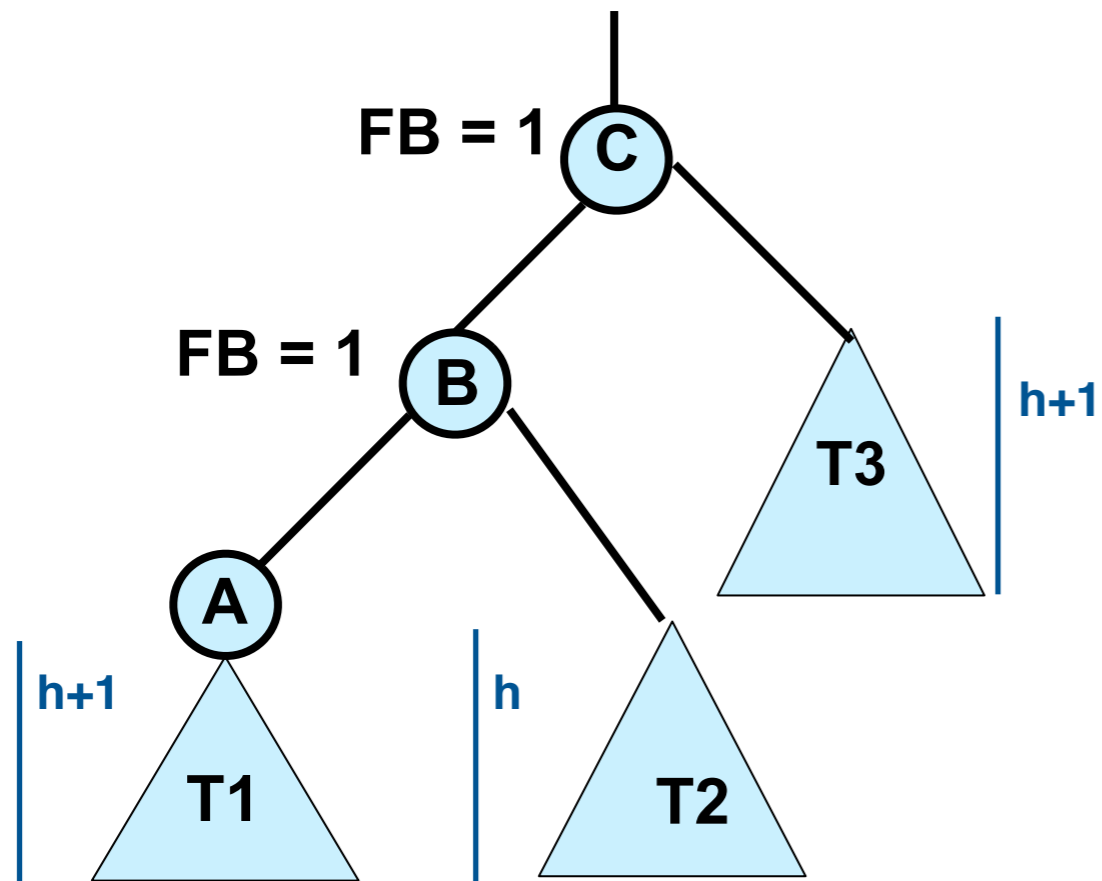


Rotazione a
destra su A



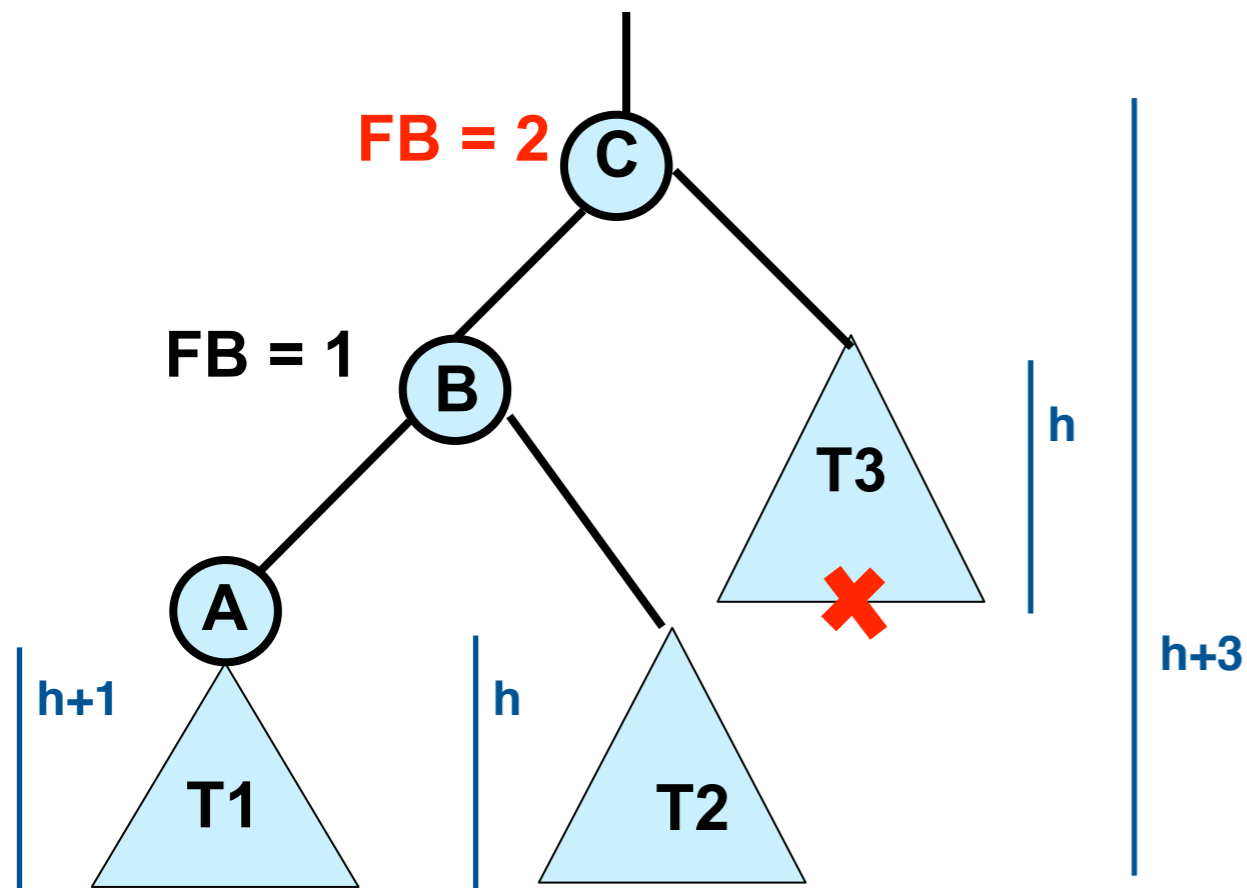
L'altezza dell'albero radicato in B è diminuita di 1 rispetto a quella dell'albero radicato in C che rimpiazziamo. Quindi se C non fosse stata la radice, avremmo dovuto proseguire nella modifica dei FB ed eventualmente eseguire altre rotazioni.

CASO LEFT LEFT 2



Se risalendo verso la radice da destra si incontra un nodo con $FB = 2$, il cui figlio sinistro ha $FB = 1$, ci si trova nel caso left-left2.

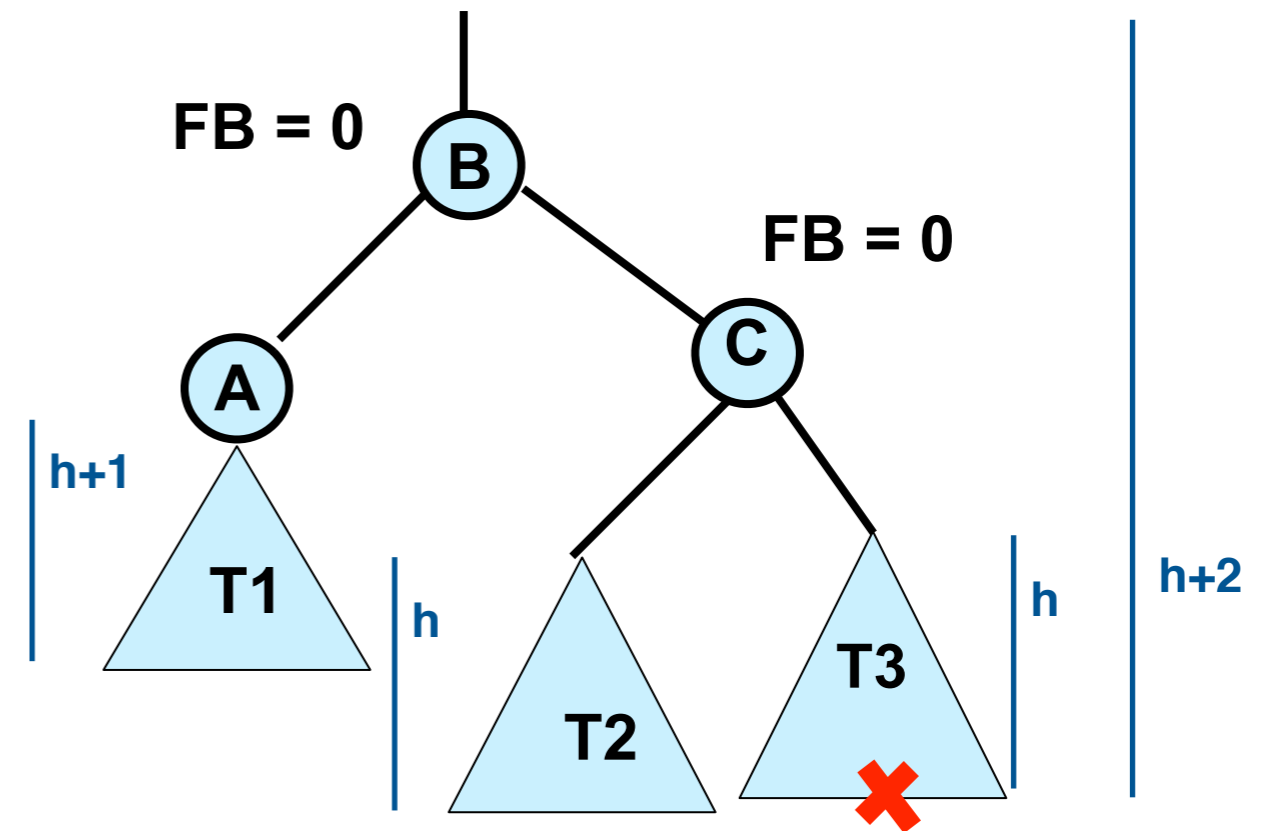
CASO LEFT LEFT 2



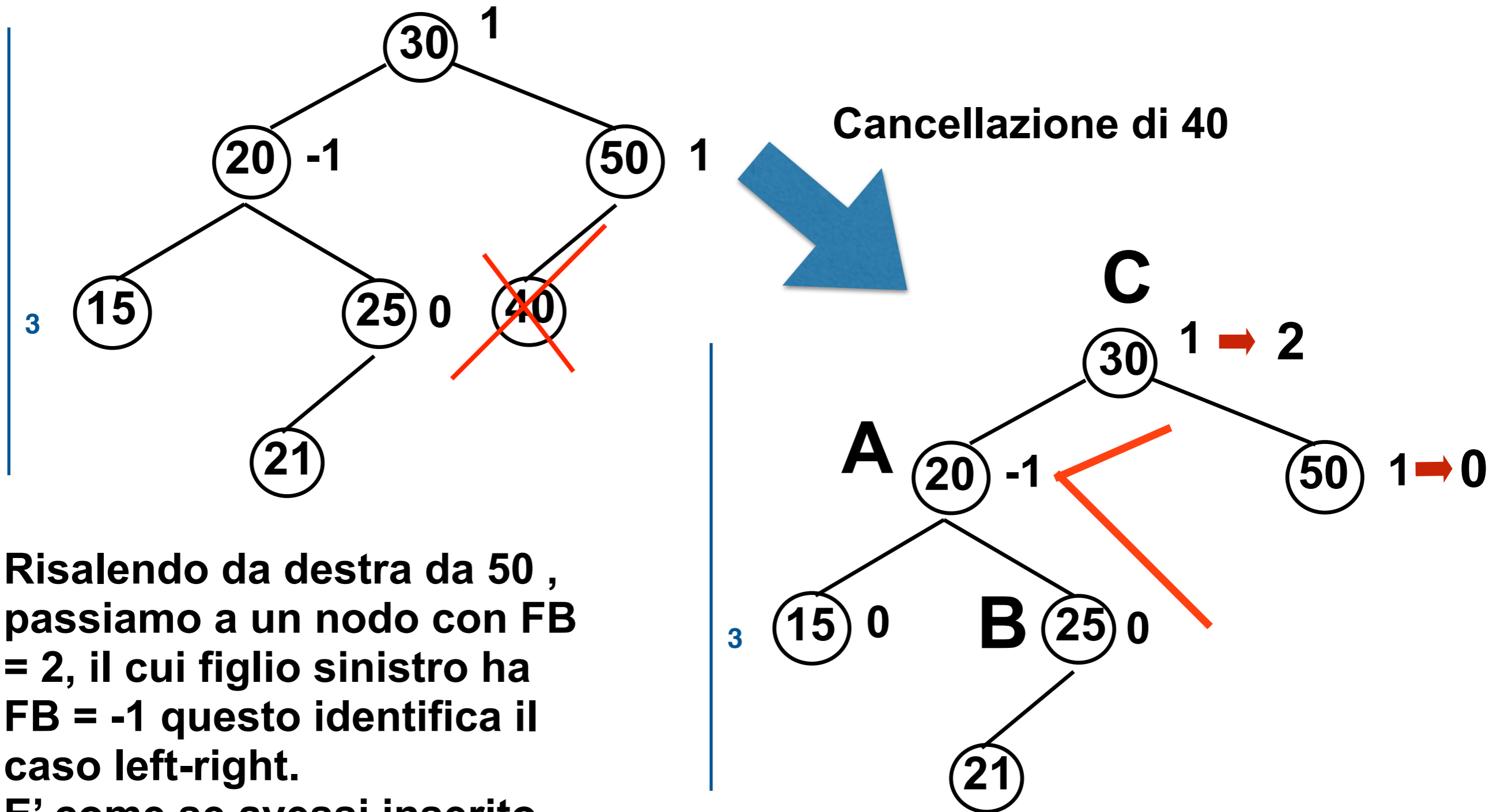
rotazione a destra su C



La cancellazione e la rotazione hanno prodotto una diminuzione di altezza del sotto albero radicato in B, che ha sostituito quello radicato in C. Bisogna risalire al padre di B, se non è la radice, per modificare i fattori di bilanciamento ed eventualmente ribilanciare con altre rotazioni.

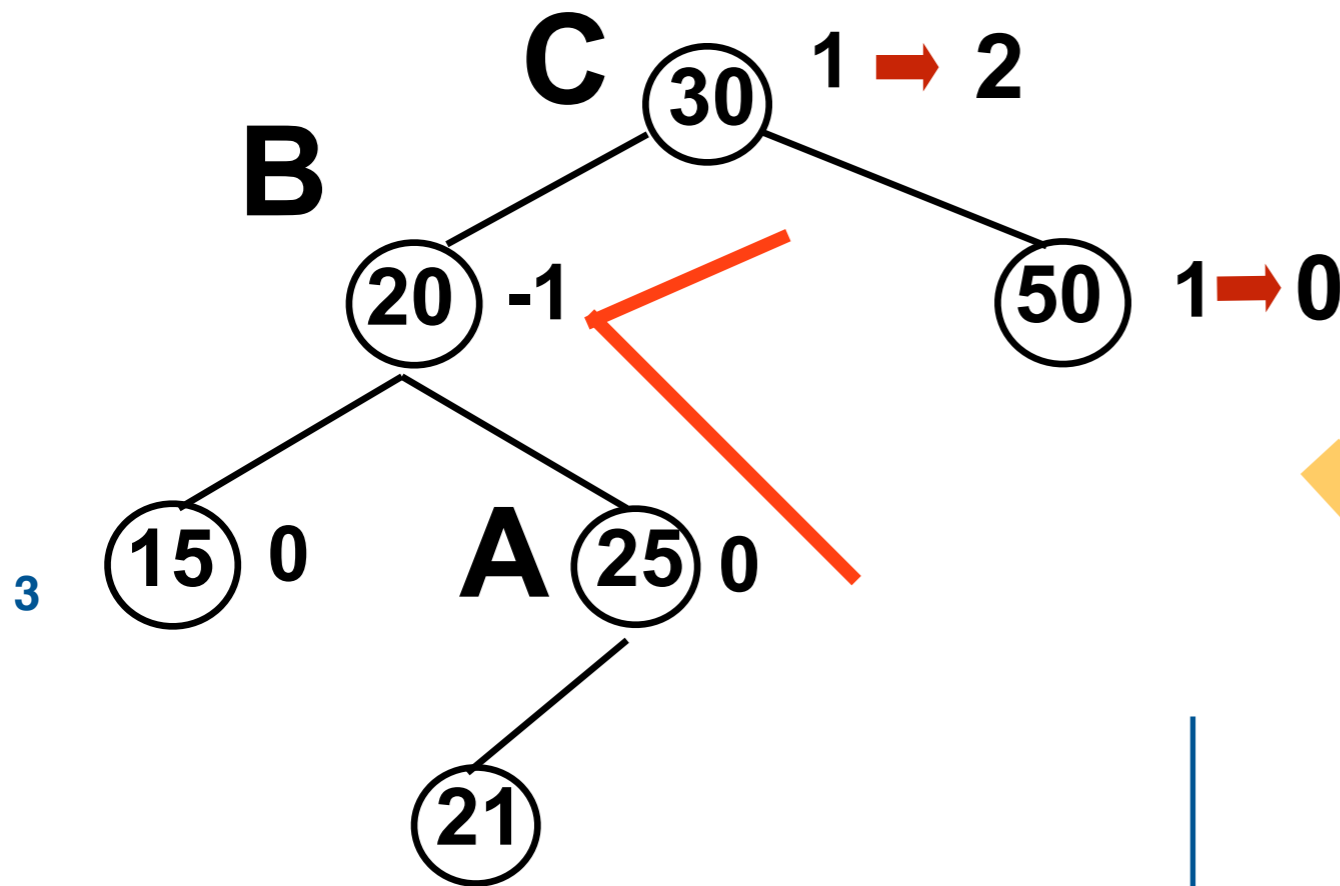


Esempio cancellazione con sbilanciamento

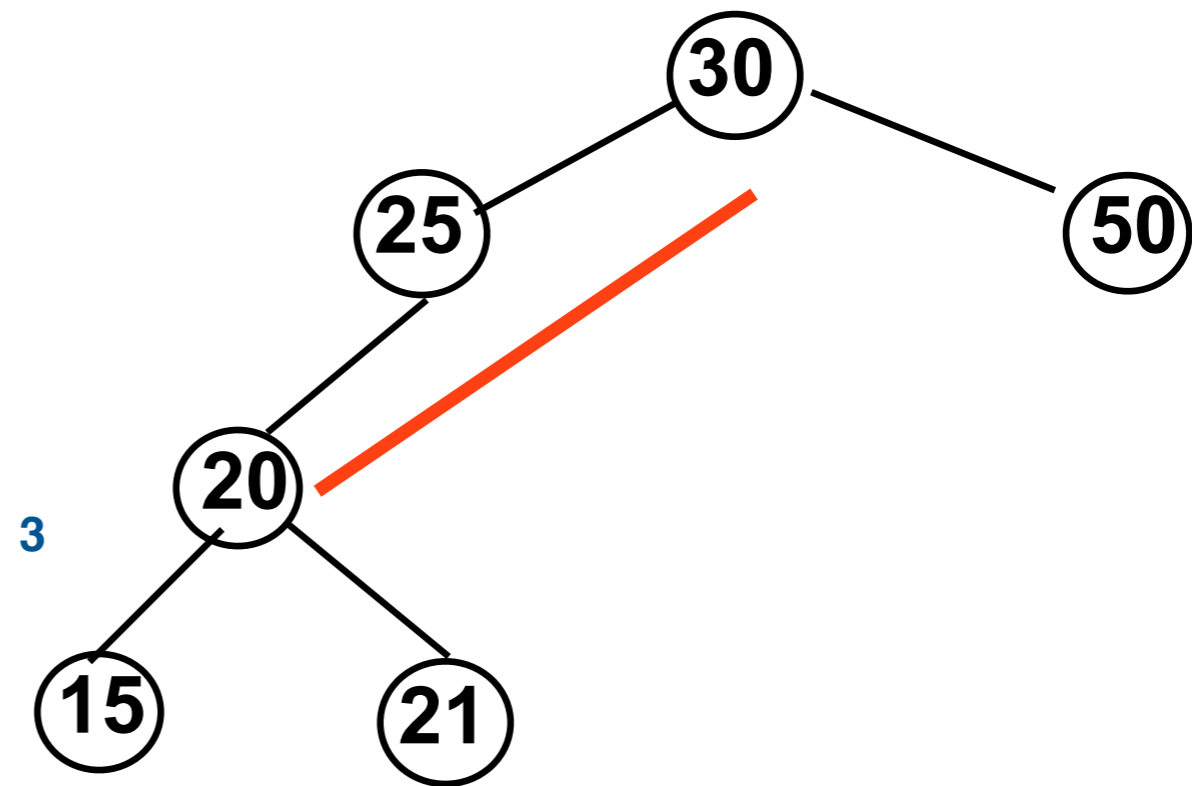


Risalendo da destra da 50 ,
passiamo a un nodo con FB
= 2, il cui figlio sinistro ha
FB = -1 questo identifica il
caso left-right.
E' come se avessi inserito
un nodo nel sottoalbero
radicato in B, aumentandone
l'altezza

Il ribilanciamento: prima rotazione

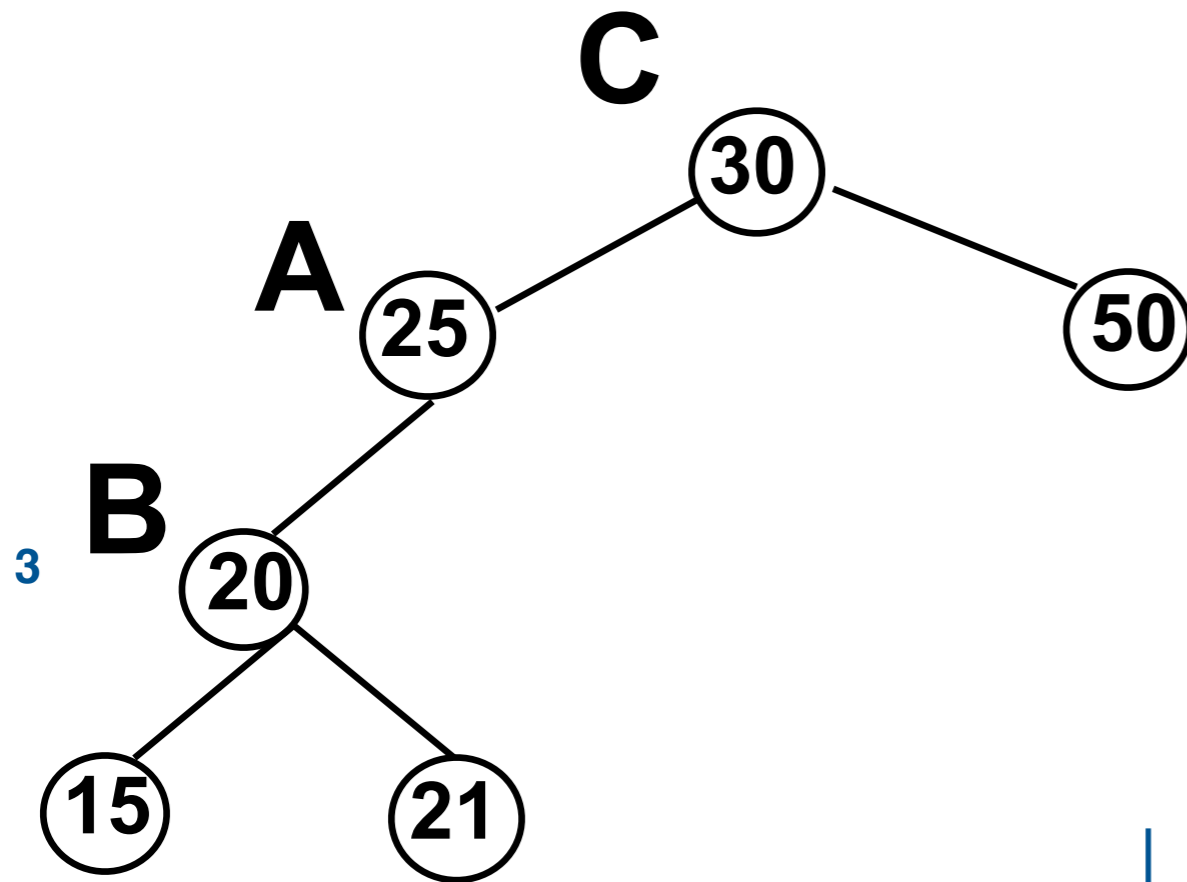


Rotazione a sinistra su B



La rotazione a sinistra, sul figlio sinistro del nodo con fattore di bilanciamento 2, riallinea i nodi, per poter ribilanciare l'albero con un'altra rotazione

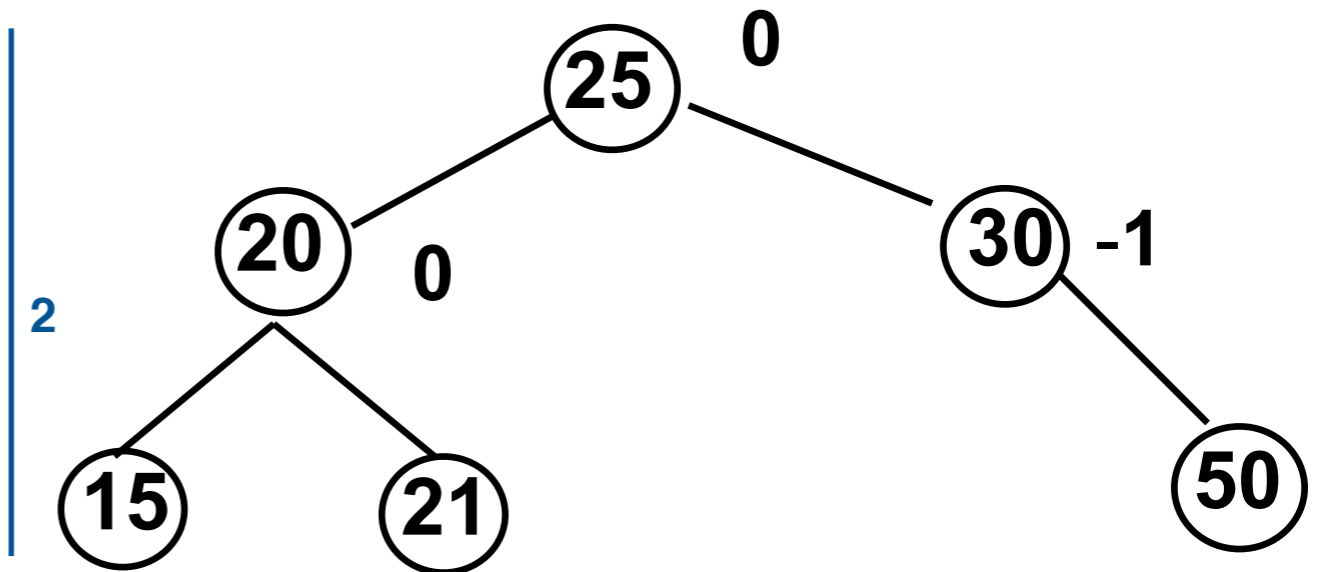
Il ribilanciamento: seconda rotazione



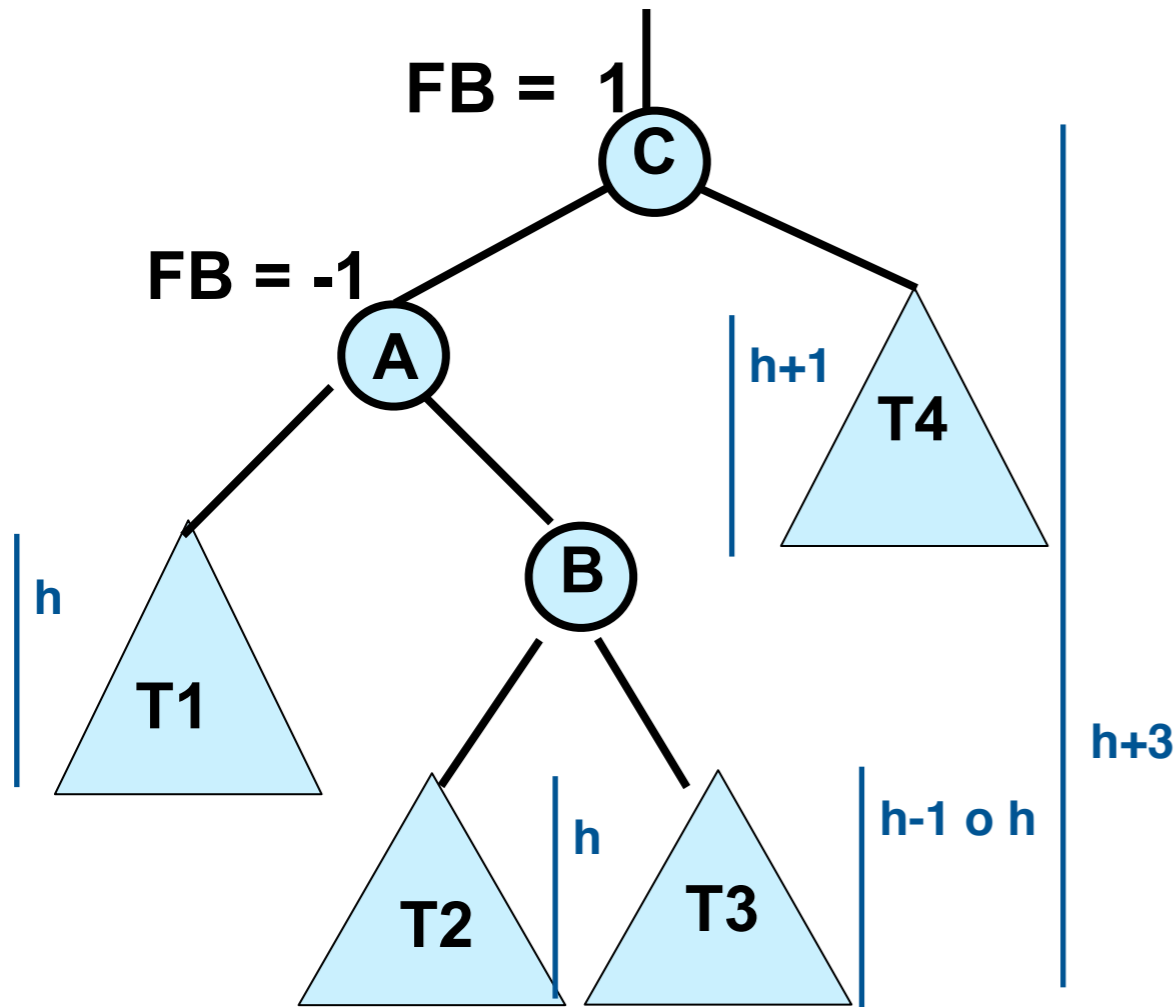
Rotazione a
destra su C



La rotazione a destra, sul nodo con fattore di bilanciamento 2, ribilancia l'albero, ma l'altezza dell'albero radicato in A, che rimpiazza l'albero radicato in C, prima della cancellazione, è diminuita di uno!



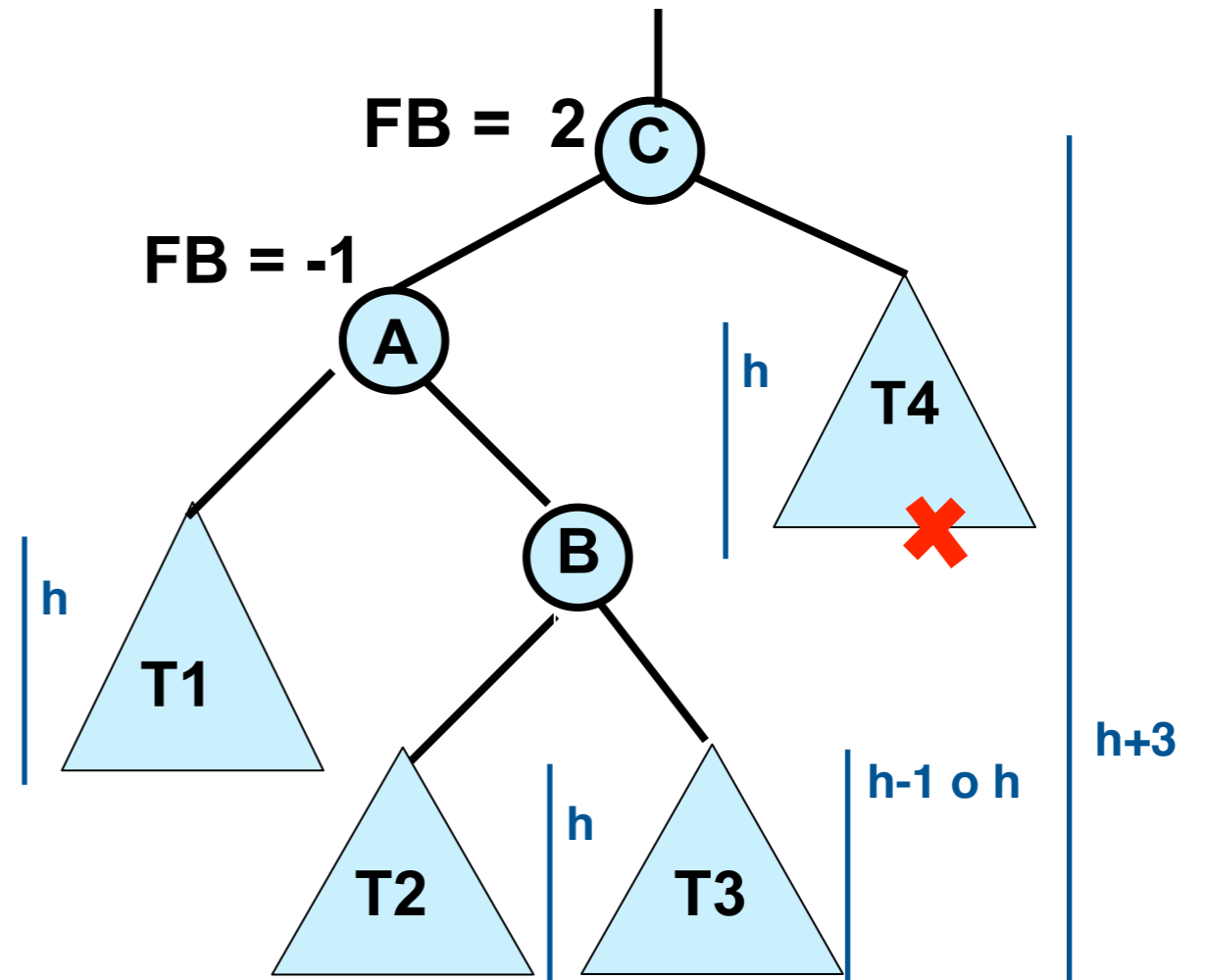
CASO LEFT-RIGHT



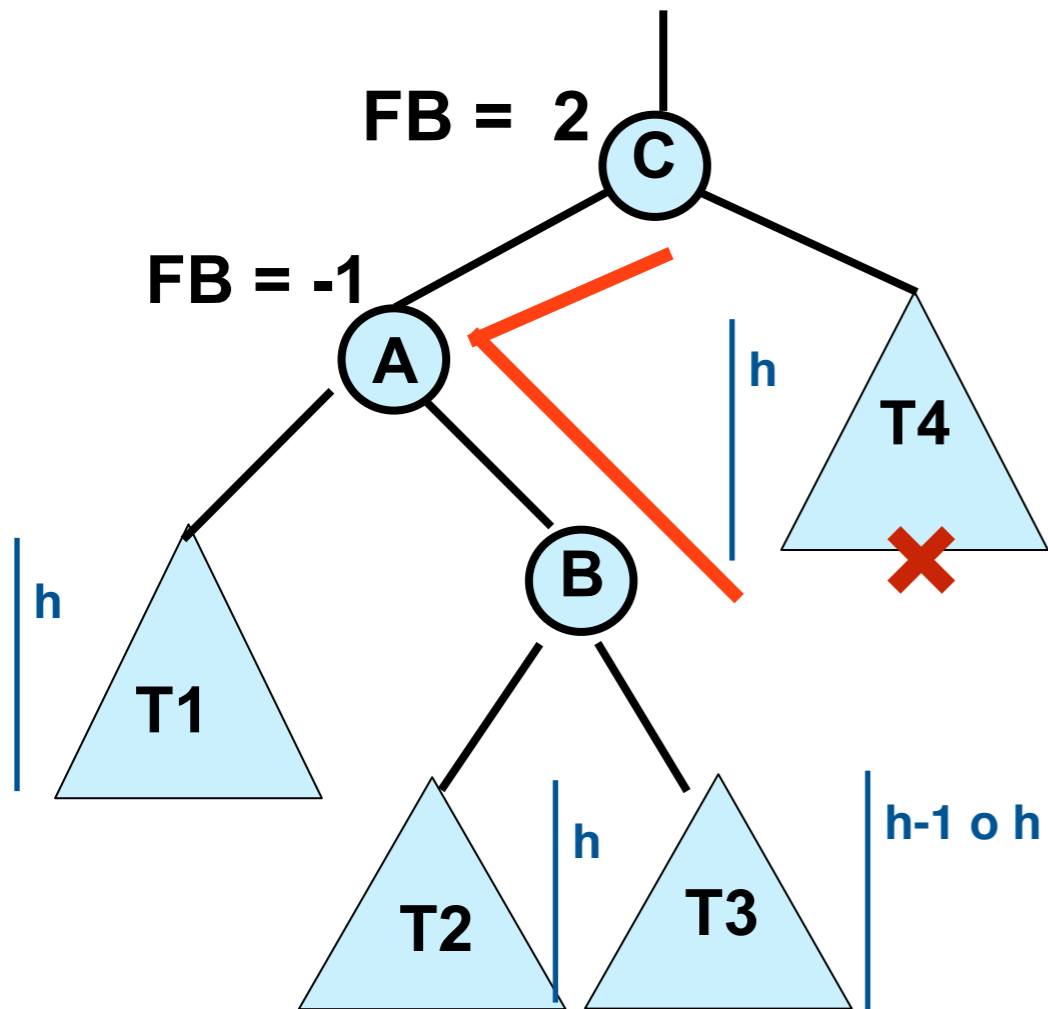
cancellazione in T4



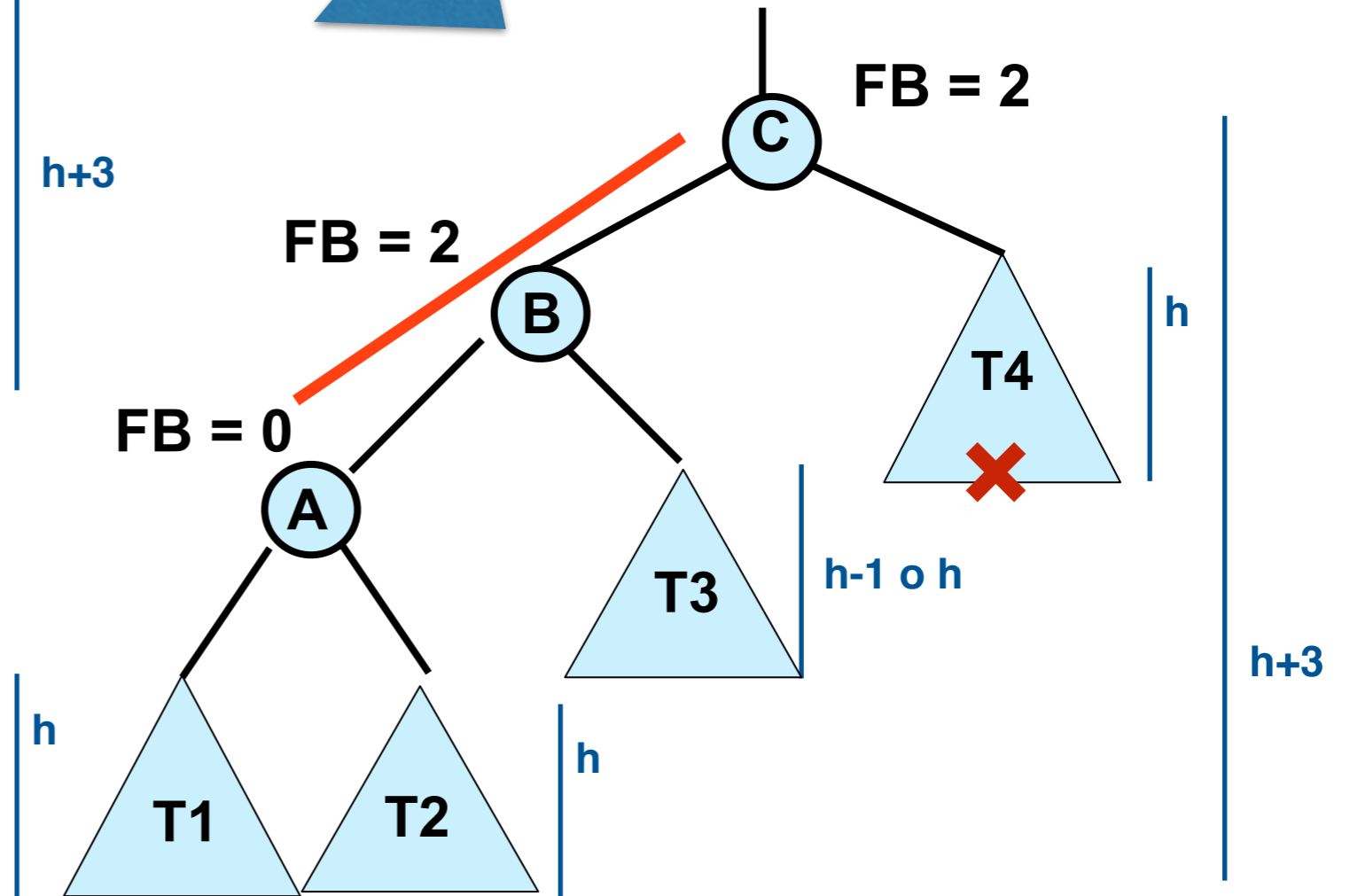
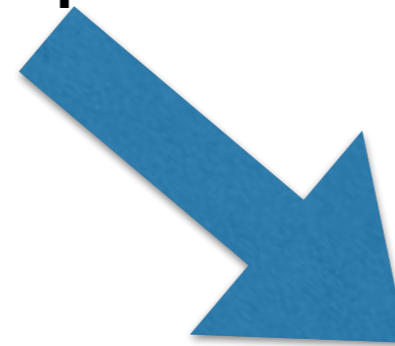
Questo caso corrisponde al caso LEFT-RIGHT dell'inserimento nel sotto albero radicato in **B**. Risalendo dal nodo cancellato verso la radice da destra si incontra un nodo con $FB = 2$, mentre il suo figlio sinistro ha $FB = -1$, questo identifica il caso left-right.



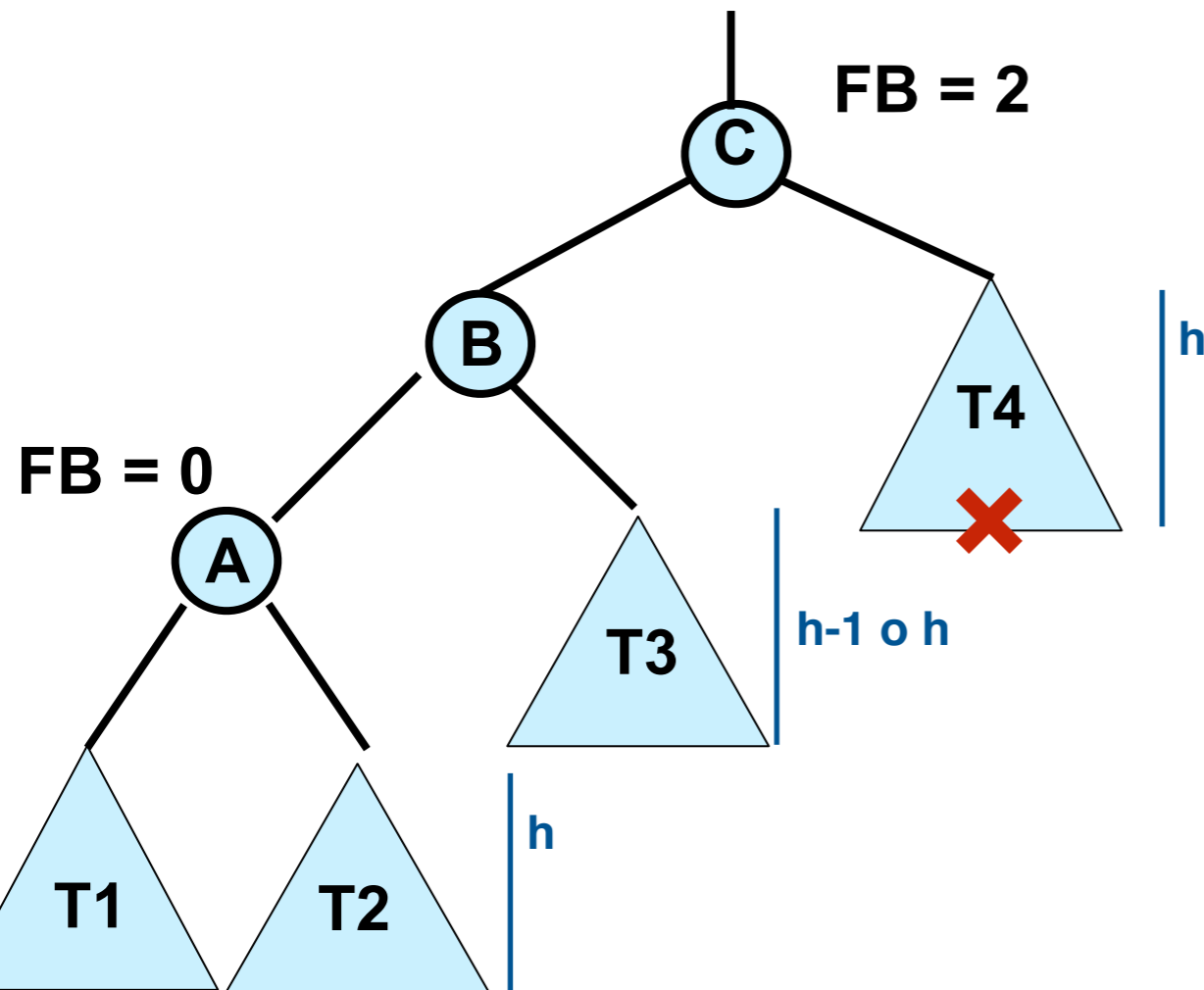
CASO LEFT-RIGHT



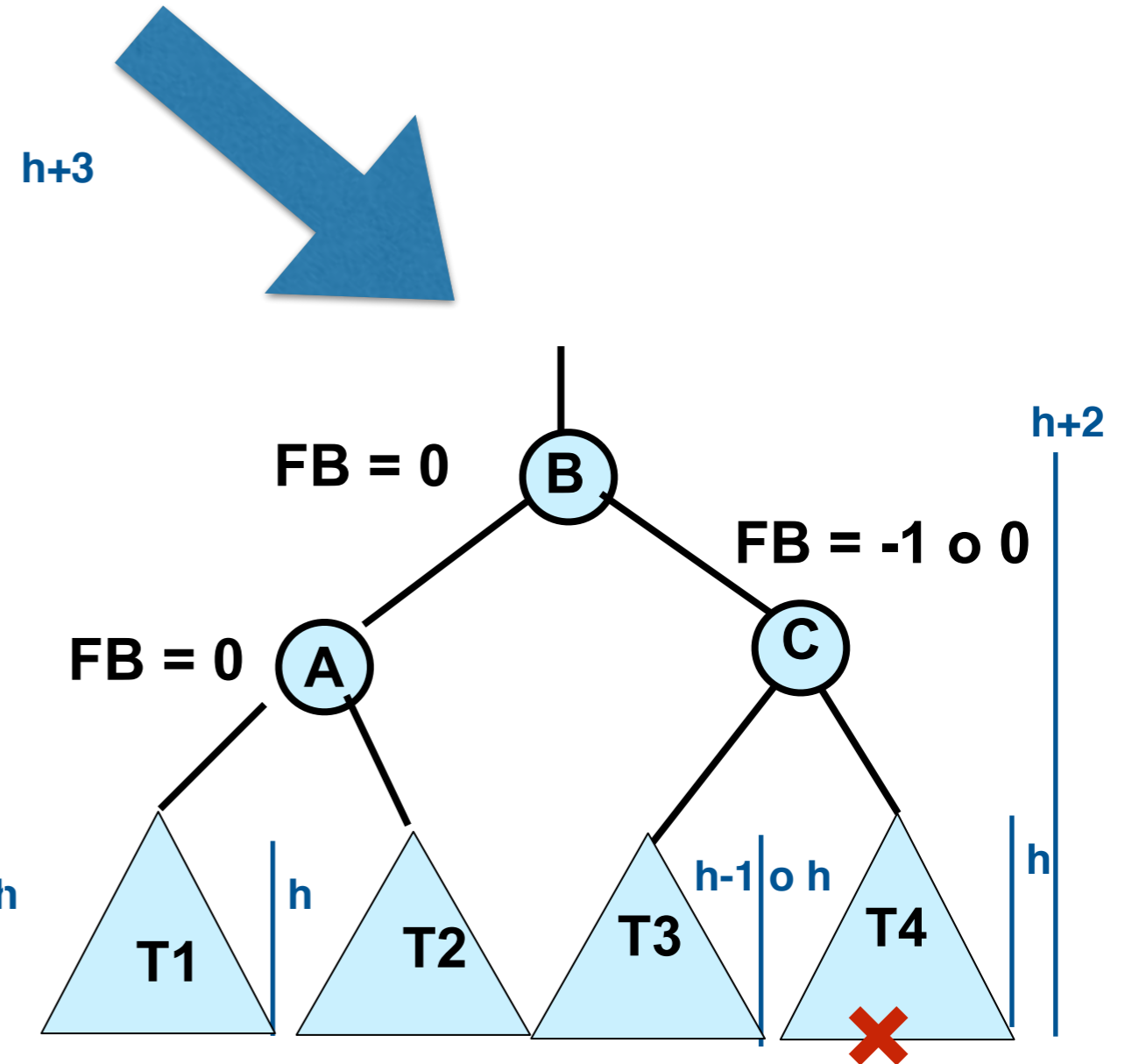
prima rotazione: a sinistra su A



CASO LEFT-RIGHT

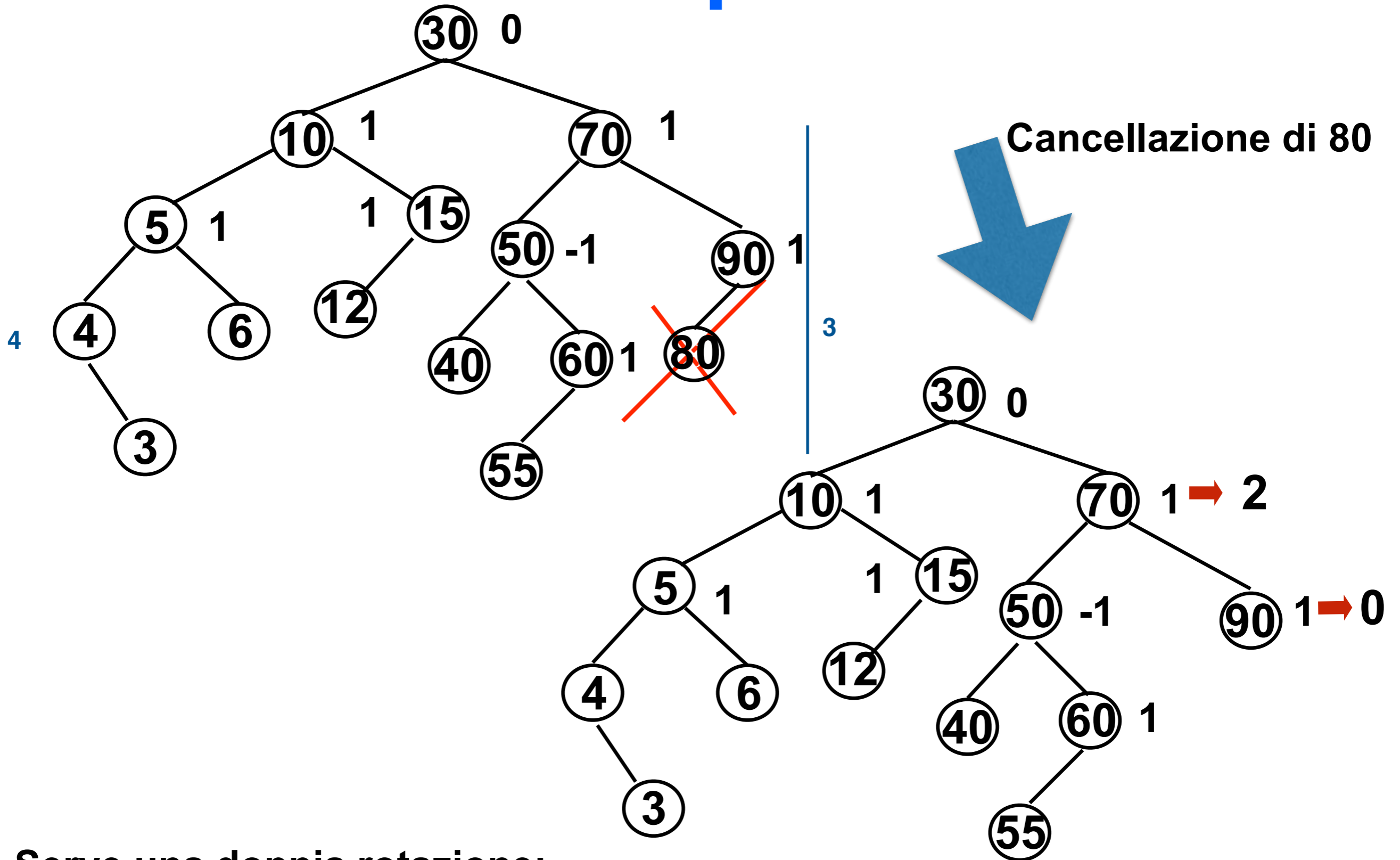


seconda rotazione: a destra su C



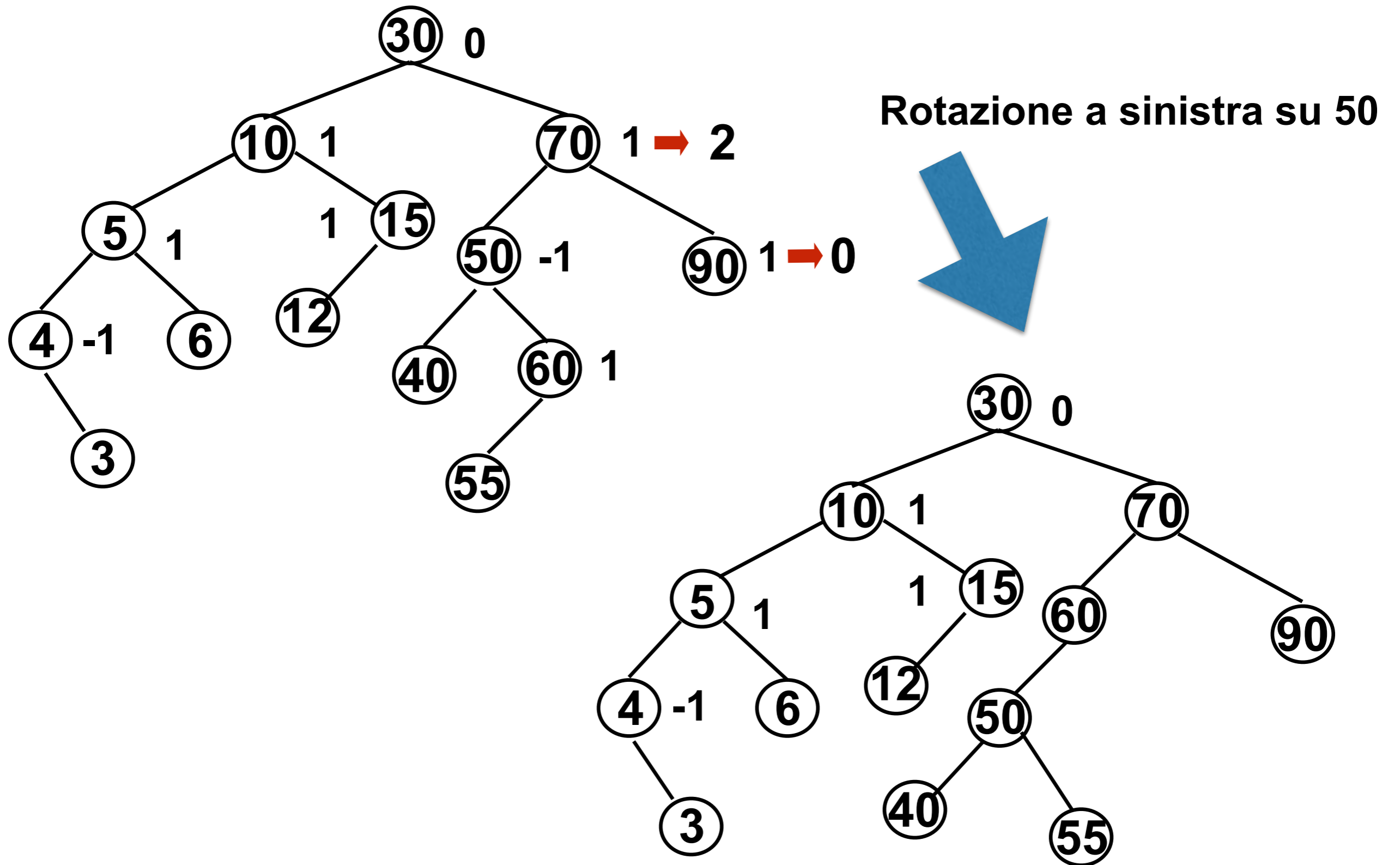
Dopo la seconda rotazione l'altezza di B è **minore** di quella del nodo C prima dell'inserimento, quindi è necessario risalire ancora verso la radice per aggiornare i fattori di bilanciamento ed eventualmente, con ulteriori rotazioni, ribilanciare l'albero!

Esempio

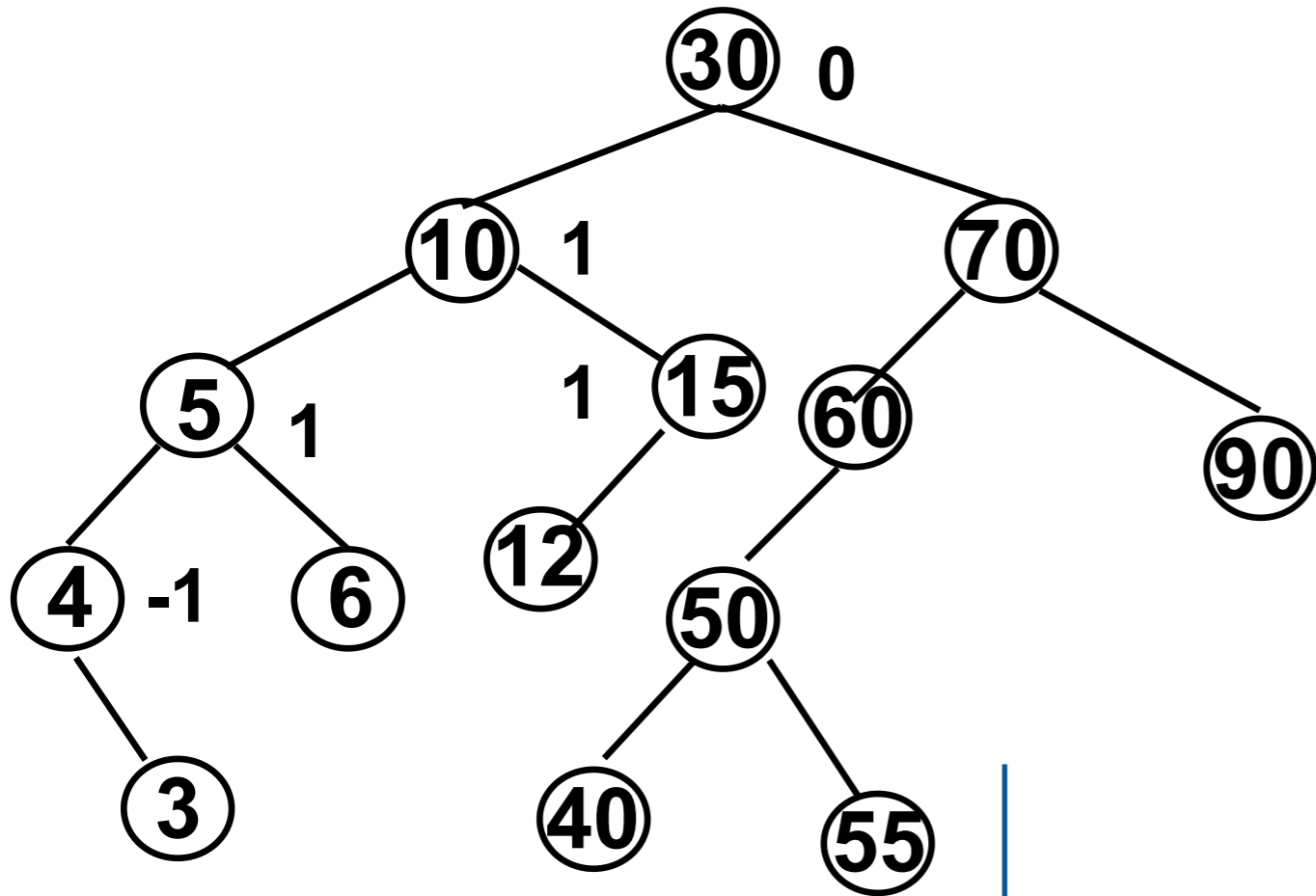


Serve una doppia rotazione:
il nodo con $FB = 2$ ha il figlio sinistro con $FB = -1$

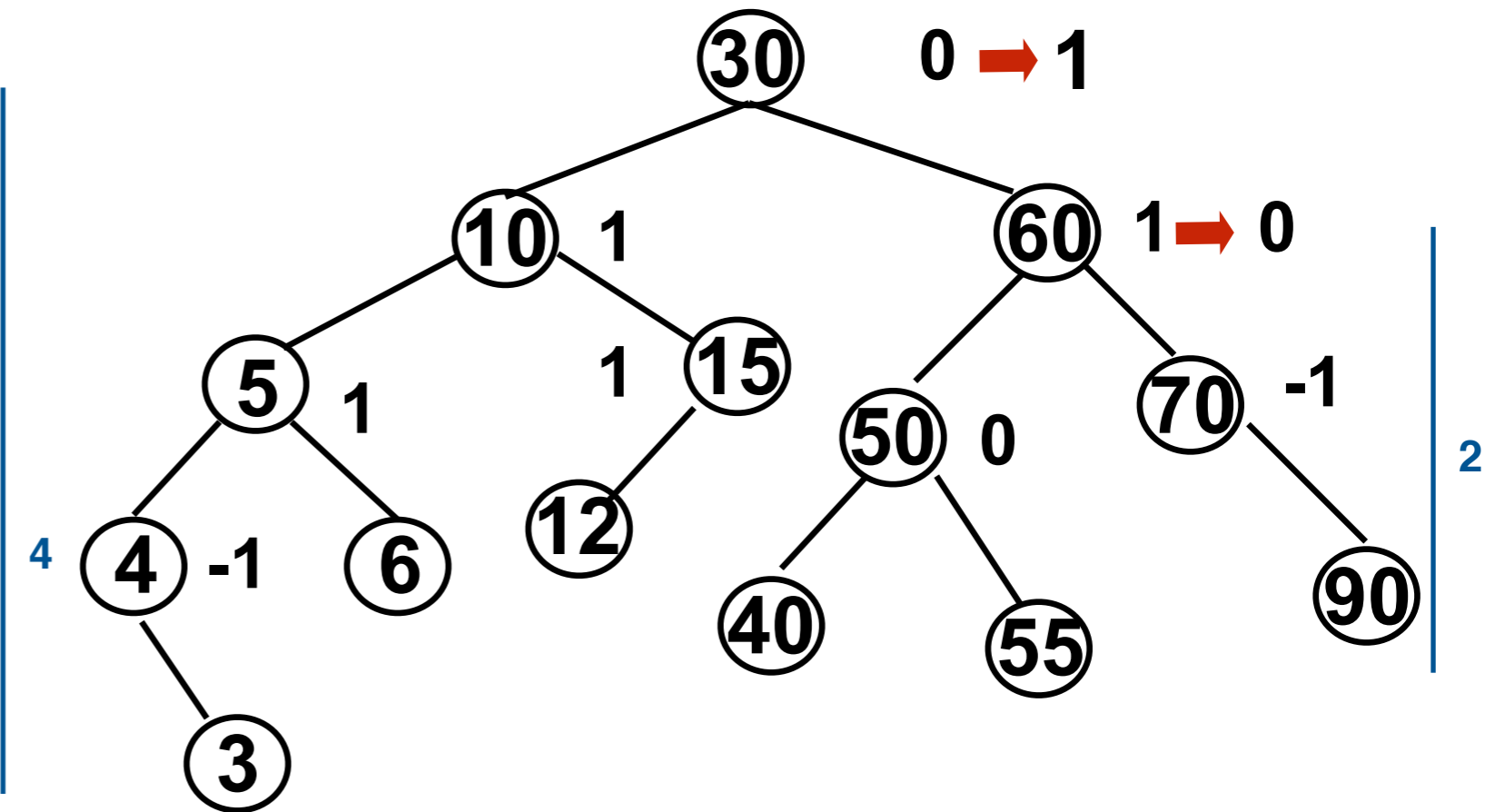
Esempio



Esempio



Rotazione a destra su 70



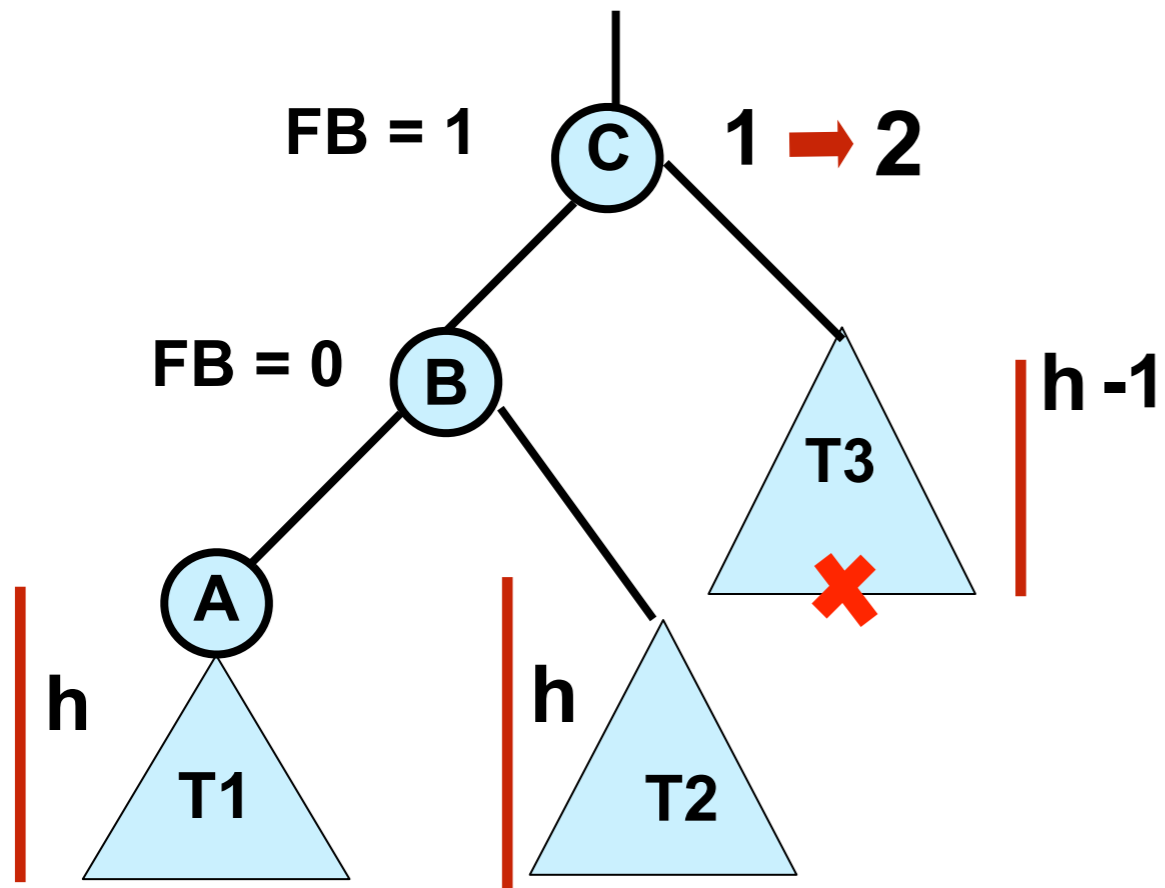
Conclusione

La cancellazione prevede quindi 6 casi. 4 casi di singola rotazione e 2 di doppia, perché ai casi considerati si devono aggiungere i simmetrici: right-right1, right-right2, right-left

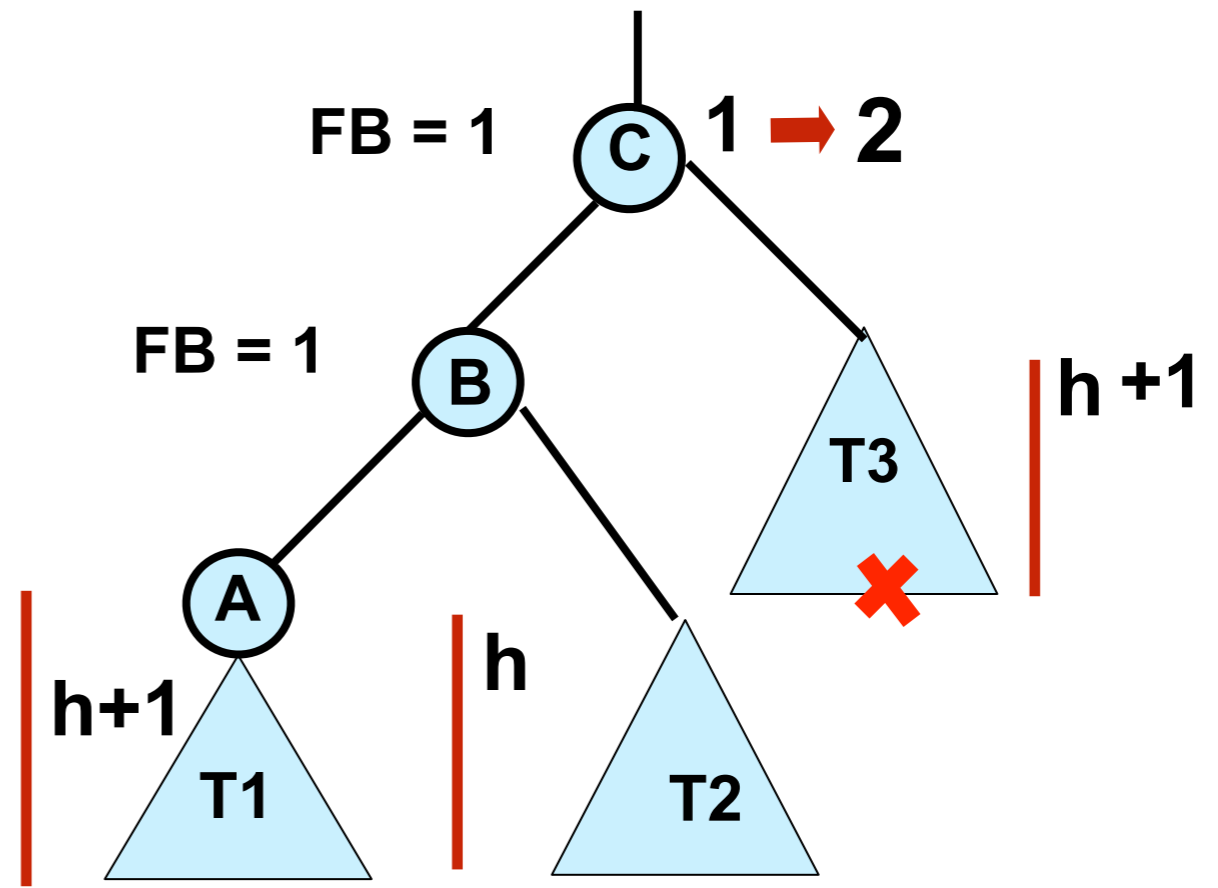
La funzione di cancellazione viene modificata in modo tale che risalendo verso la radice si aggiornano i fattori di bilanciamento nei nodi.

Dopo ogni aggiornamento si controlla se il valore del FB (la differenza tra le altezze) è legale , se non lo è si chiama la funzione di ribilanciamento.

Casi di sbilanciamento cancellazione



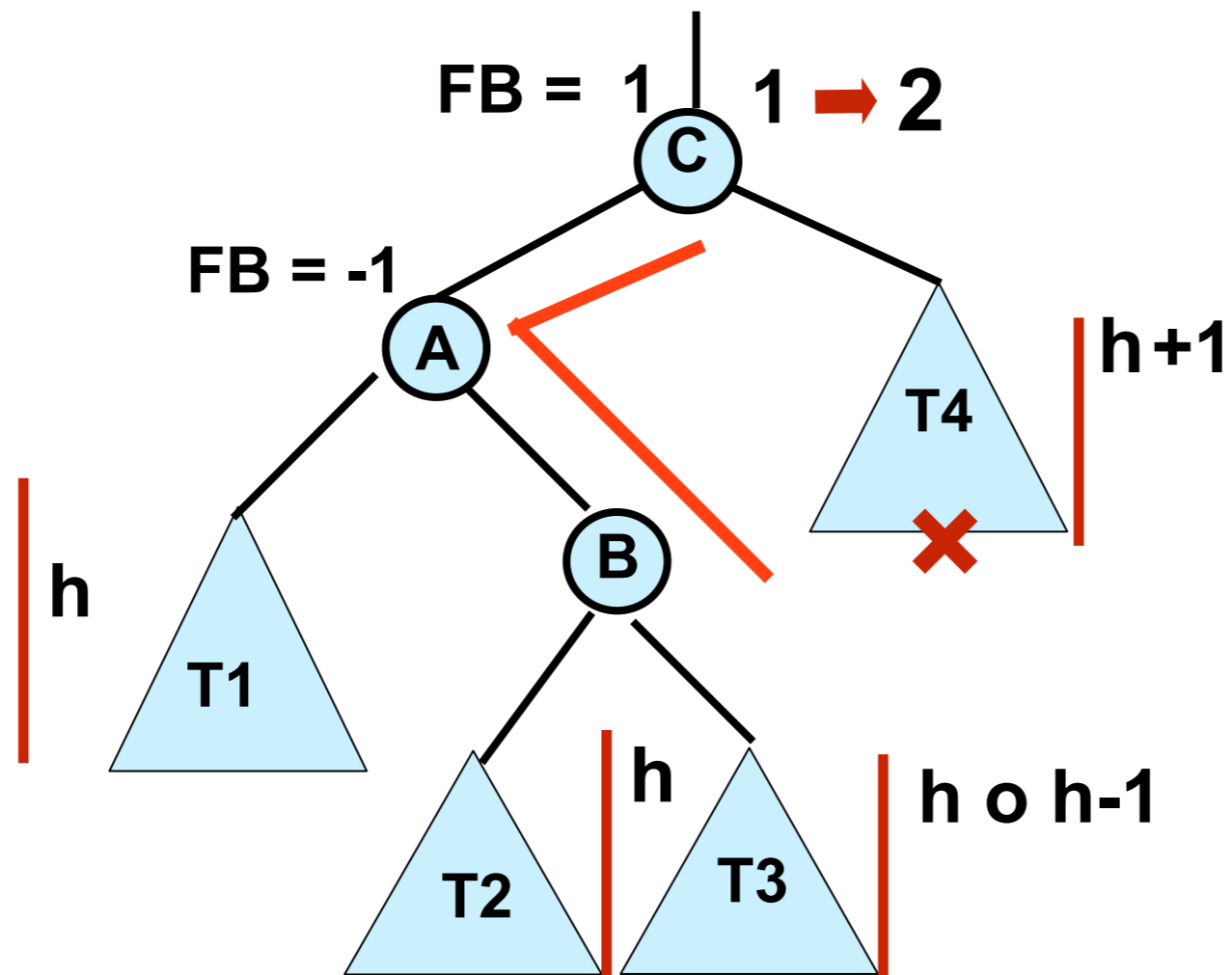
left- left 1



left- left 2

A questi si devono aggiungere i simmetrici: right-right 1, right-right 2

Casi di sbilanciamento cancellazione



left- right

Pseudocodice ribilanciamento

Ribilanciamento(T)

input: T è il riferimento al primo antenato del nuovo nodo inserito il cui fattore di bilanciamento è illegale, cioè -2 o 2

prec: T è un AVL, salvo la violazione in T

post: l'albero radicato in T è ribilanciato

```
if (T.bf == 2) %L'altezza è aumentata a sinistra
```

```
  P=T.left
```

```
  if (P.bf == -1) %L'altezza è aumentata a destra: siamo nel caso Left Right
```

```
    leftRotate(P) % ci si riconduce al caso Left Left
```

```
  rightRotate(T); % il bf è 0 o 1, l'altezza è aumentata a sinistra, caso Left Left
```

```
    (è 1 nel caso inserimenti, può essere 0 nel caso di cancellazioni)
```

```
if (T.bf == -2) %L'altezza è aumentata a destra
```

```
  P=T.right
```

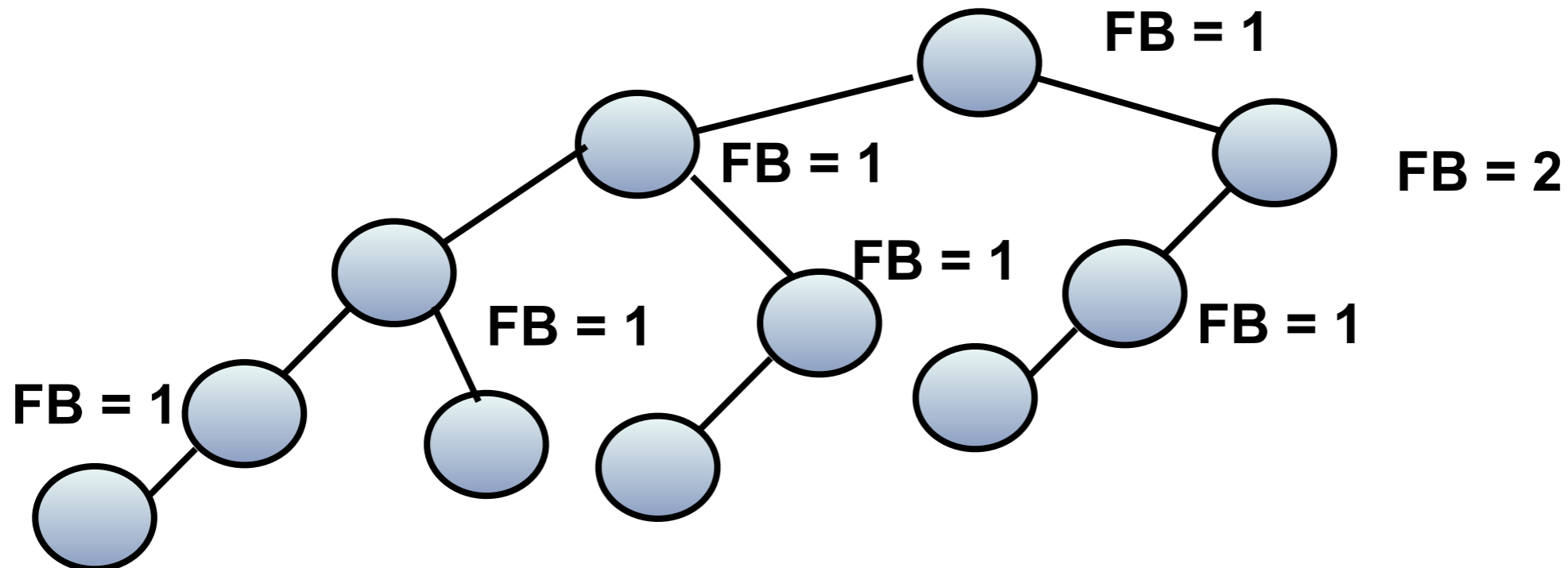
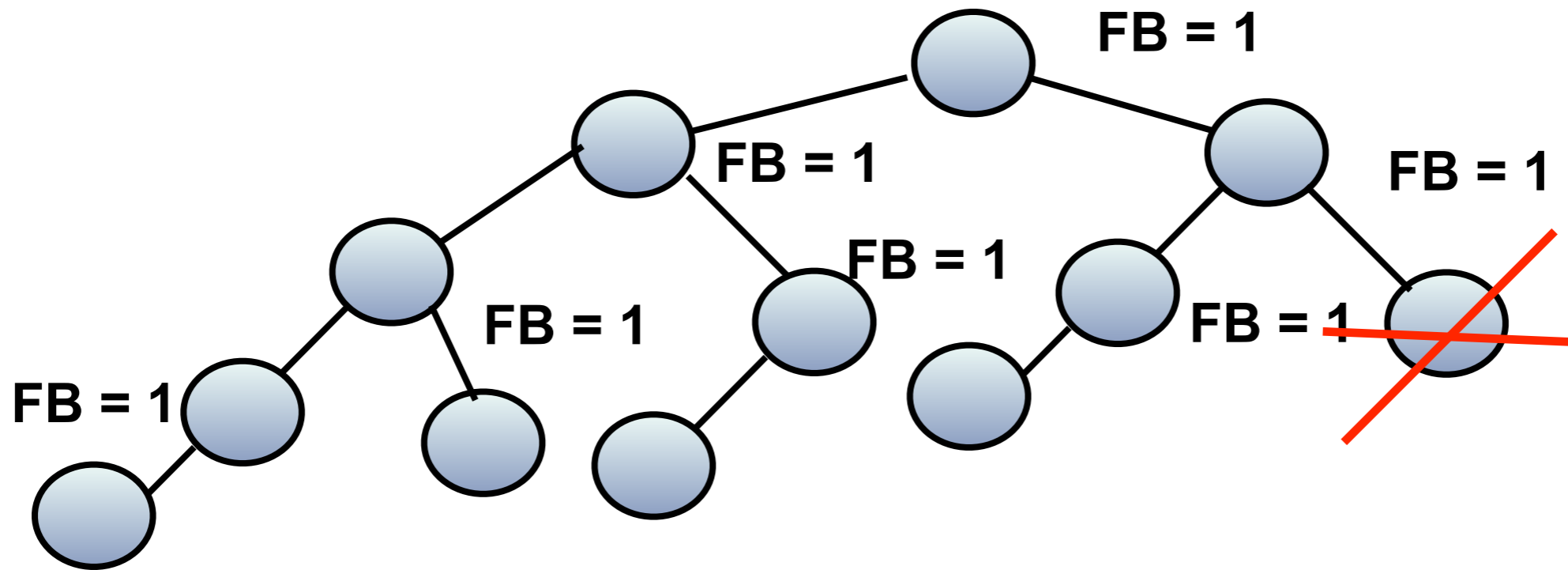
```
  if (P.fb) == 1 %L'altezza è aumentata a sinistra: siamo nel caso Right Left
```

```
    rightRotate(P) % ci si riconduce al caso Right Right
```

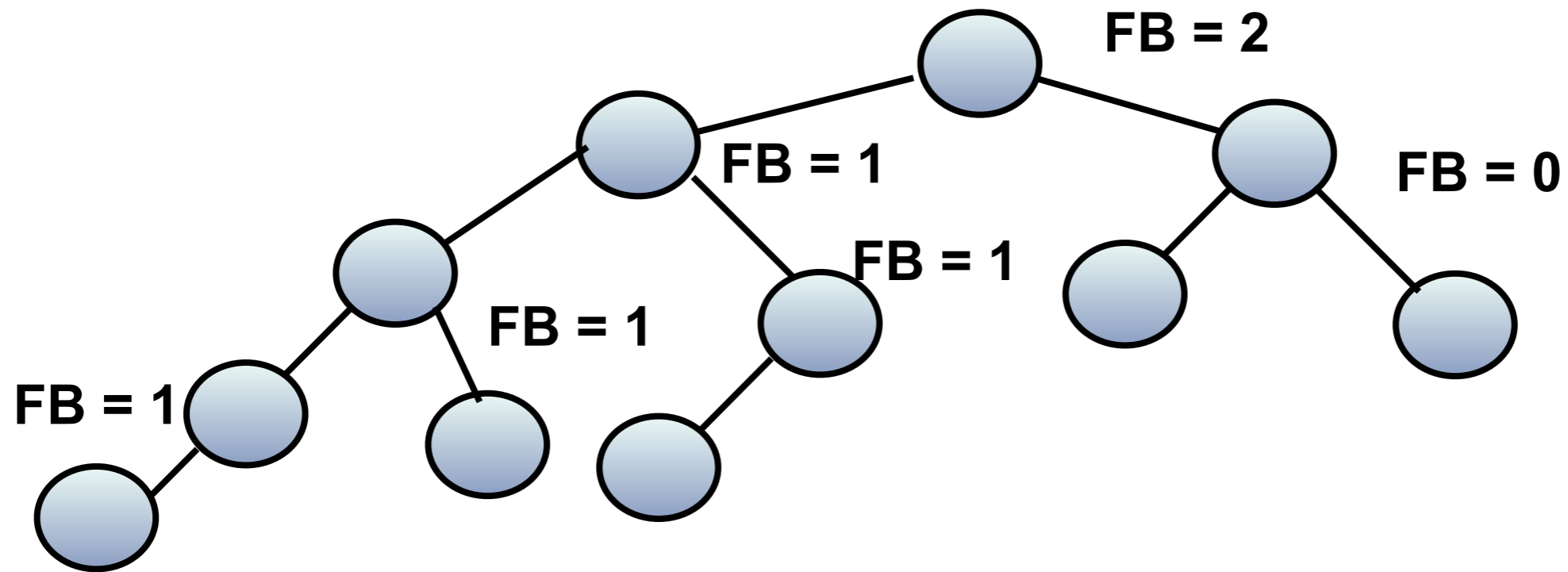
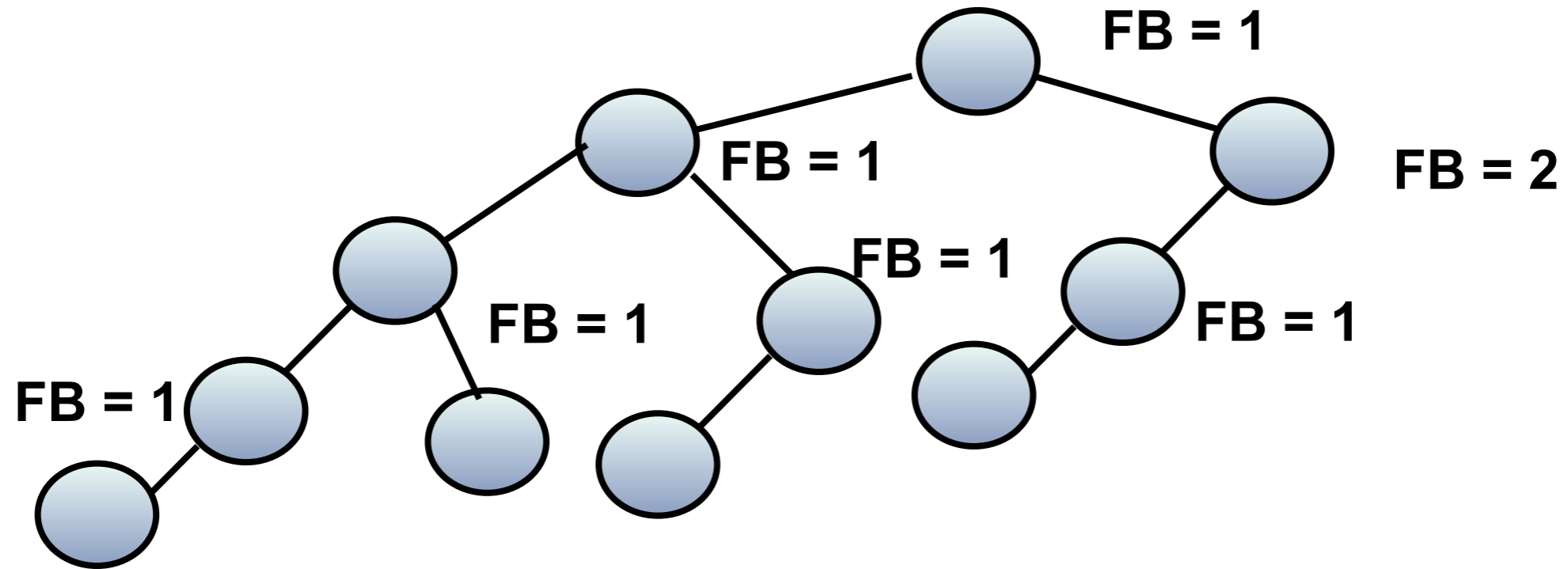
```
  leftRotate(T); % il bf è 0 o -1 l'altezza è aumentata a destra, caso Right Right
```

```
    (è 1 nel caso di inserimenti, può essere 0 nel caso di cancellazioni)
```

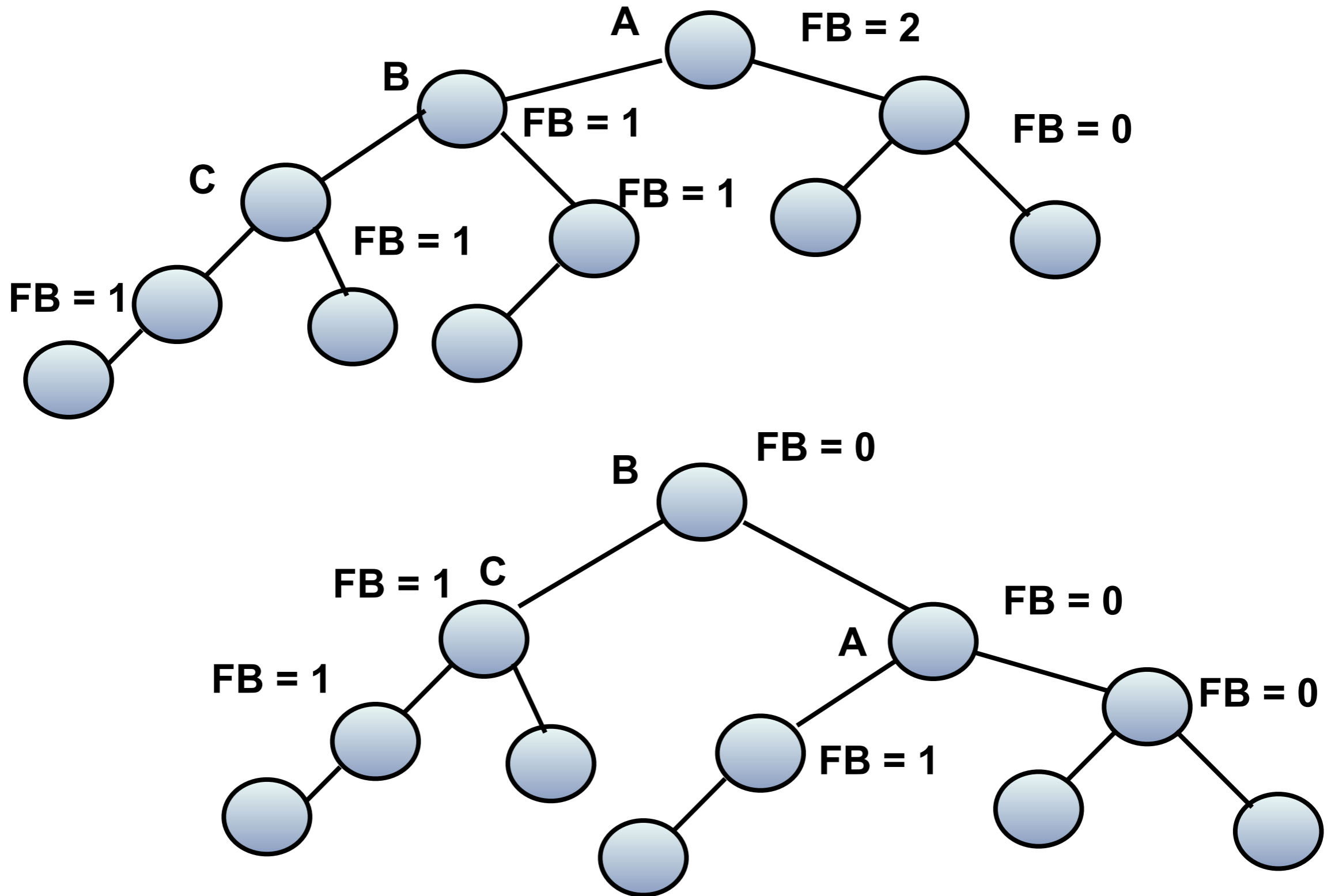
Esempio cancellazione



Esempio cancellazione



Esempio cancellazione



Una soluzione diversa

Qualche volta si fa uso della cosiddetta cancellazione “lazy”: si aggiunge ad ogni nodo un flag che segnala se il nodo è presente o no. Quindi la cancellazione è solo una ricerca con modifica del campo flag da “presente” a “assente”.

Quando il numero dei nodi “assenti” supera una certa percentuale si ricostruisce l’albero inserendo solo i nodi presenti. Anche l’inserimento può avvantaggiarsi di questo approccio, se si vuole inserire un nodo “assente” basta cambiare il flag a “presente”