

Traza de reescritura — BSTree (HIDE)

Aplicación paso a paso de los axiomas a un término concreto, según la especificación *BSTree-hide.nereus*

Término a evaluar:

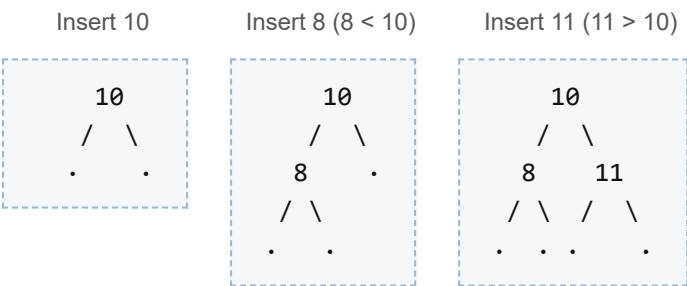
```
BSTree_Delete(  
  BSTree_Insert(  
    BSTree_Insert(  
      BSTree_Insert(BTree_Empty() as BSTree, 10),  
        8),  
      11),  
    10)
```

Resultado esperado: borrar 10 (raíz con dos hijos) sustituye la raíz por el mínimo del subárbol derecho (11) y borra 11 de ese subárbol.

0. Convenciones

- . abrevia `BTree_Empty()` as `BSTree` en las vistas de árbol.
- El redex (subterm sobre el que se aplica la regla en cada paso) se resalta en amarillo.
- Las ligaduras (*bindings*) de cada axioma se listan antes del término reescrito.

1. Estado del árbol después de cada Insert (visión semántica)



2. Traza formal

Paso 1 — caso base de BSTree_Insert sobre el más interno

Axioma aplicado:

```
BSTree_Insert(BTree_Empty() as BSTree, i) =  
  BTree_Create(BTree_Empty() as BSTree, BTree_Empty() as BSTree, i) as BSTree;
```

Binding: $i = 10$.

```
BSTree_Delete(  
  BSTree_Insert(  
    BSTree_Insert(  
      BTree_Create(BTree_Empty() as BSTree, BTree_Empty() as BSTree, 10) as BSTree,  
      8),  
    11),  
  10)
```

Paso 2 — BSTree_Insert con $i < j$ ($8 < 10$)

Axioma aplicado:

```
(i < j) implies  
BSTree_Insert(BTree_Create(b1, b2, j) as BSTree, i) =  
  BTree_Create(BSTree_Insert(b1 as BSTree, i), b2, j) as BSTree;
```

Bindings: $b1 = \text{BTree_Empty}() \text{ as BSTree}$, $b2 = \text{BTree_Empty}() \text{ as BSTree}$, $j = 10$, $i = 8$. Condición $8 < 10$ ✓

```
BSTree_Delete(  
  BSTree_Insert(  
    BTree_Create(  
      BSTree_Insert(BTree_Empty() as BSTree, 8),  
      BTree_Empty() as BSTree,  
      10) as BSTree,  
    11),  
  10)
```

Paso 3 — caso base de BSTree_Insert (otra vez) sobre el subárbol izquierdo

Mismo axioma del Paso 1, con $i = 8$.

```
BSTree_Delete(  
  BSTree_Insert(  
    BTree_Create(  
      BTree_Create(BTree_Empty() as BSTree, BTree_Empty() as BSTree, 8) as BSTree,  
      BTree_Empty() as BSTree,  
      10) as BSTree,  
    11),  
  10)
```

Paso 4 — BSTree_Insert con $i > j$ ($11 > 10$)

Axioma aplicado:

```
(i > j) implies
BSTree_Insert(BTree_Create(b1, b2, j) as BSTree, i) =
  BTree_Create(b1, BSTree_Insert(b2 as BSTree, i), j) as BSTree;
```

Bindings: $b1 = \text{BTree_Create}(\cdot, \cdot, 8)$ **as** **BSTree**, $b2 = \text{BTree_Empty}()$ **as** **BSTree**, $j = 10$, $i = 11$.
Condición $11 > 10 \checkmark$

```
BSTree_Delete(
  BTree_Create(
    BTree_Create(BTree_Empty() as BSTree, BTree_Empty() as BSTree, 8) as BSTree,
    BSTree_Insert(BTree_Empty() as BSTree, 11),
    10) as BSTree,
  10)
```

Paso 5 — caso base de BSTree_Insert sobre el subárbol derecho

Mismo axioma del Paso 1, con $i = 11$.

```
BSTree_Delete(
  BTree_Create(
    BTree_Create(BTree_Empty() as BSTree, BTree_Empty() as BSTree, 8) as BSTree,
    BTree_Create(BTree_Empty() as BSTree, BTree_Empty() as BSTree, 11) as BSTree,
    10) as BSTree,
  10)
```

Hasta acá hemos construido el árbol con 10 a la raíz, 8 a la izquierda y 11 a la derecha. Ahora arranca el `BSTree_Delete`.

Paso 6 — BSTree_Delete caso F (j = i, ambos hijos no vacíos)

Axioma aplicado:

```
(j = i and not BTree_IsEmpty(b1) and not BTree_IsEmpty(b2)) implies
BSTree_Delete(BTree_Create(b1, b2, i) as BSTree, j) =
  BTree_Create(
    b1,
    BSTree_Delete(b2 as BSTree, BSTree_Min(b2 as BSTree)),
    BSTree_Min(b2 as BSTree)
  ) as BSTree;
```

Bindings:

b1 = BTree_Create(·, ·, 8) as BSTree → no vacío ✓

b2 = BTree_Create(·, ·, 11) as BSTree → no vacío ✓

i = 10, j = 10 → j = i ✓

```
BTree_Create(
  BTree_Create(BTree_Empty() as BSTree, BTree_Empty() as BSTree, 8) as BSTree,
  BSTree_Delete(b2 as BSTree, BSTree_Min(b2 as BSTree)),
  BSTree_Min(b2 as BSTree)
) as BSTree
```

Con b2 = BTree_Create(·, ·, 11) as BSTree. Pendiente: evaluar las dos sub-expresiones BSTree_Min y BSTree_Delete sobre b2.

Paso 7 — calcular BSTree_Min(b2)

Axioma aplicado (rama izquierda vacía):

```
BTree_IsEmpty(b1) implies
BSTree_Min(BTree_Create(b1, b2, i) as BSTree) = i;
```

Sobre BTree_Create(BTree_Empty() as BSTree, BTree_Empty() as BSTree, 11) as BSTree:

b1 = BTree_Empty() as BSTree → BTree_IsEmpty(b1) = True ✓

Resultado: BSTree_Min(b2) = 11

Paso 8 — calcular BSTree_Delete(b2, 11)

Axioma aplicado (caso C: j = i, ambos hijos vacíos):

```
(j = i and BTree_IsEmpty(b1) and BTree_IsEmpty(b2)) implies
BSTree_Delete(BTree_Create(b1, b2, i) as BSTree, j) =
  BTree_Empty() as BSTree;
```

Sobre BSTree_Delete(BTree_Create(·, ·, 11) as BSTree, 11):

b1 = · → IsEmpty ✓, b2 = · → IsEmpty ✓, i = 11, j = 11 ✓

Resultado: BSTree_Delete(b2, 11) = BTree_Empty() as BSTree

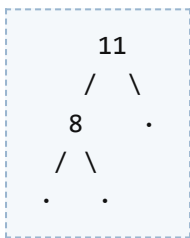
Paso 9 — sustituir los resultados parciales en el término del Paso 6

```
BTree_Create(  
  BTree_Create(BTree_Empty() as BTree, BTree_Empty() as BTree, 8) as BTree,  
  BTree_Empty() as BTree,  
  11  
) as BTree
```

3. Forma normal

```
BTree_Create(  
  BTree_Create(BTree_Empty() as BTree, BTree_Empty() as BTree, 8) as BTree,  
  BTree_Empty() as BTree,  
  11  
) as BTree
```

Vista de árbol del resultado:



Lectura semántica: al borrar 10 (raíz con dos hijos no vacíos), el axioma F reemplaza la raíz por $\min(\text{subárbol derecho}) = 11$ y borra ese mínimo del subárbol derecho. Como el subárbol derecho era el nodo hoja 11, queda vacío. El BST resultante tiene raíz 11 y único hijo izquierdo 8 (con sus dos hojas vacías). Se preserva la invariante BST: todo elemento del subárbol izquierdo (8) es menor que la raíz (11).

Por qué importa el casteo del Paso 1: sin `as BTree` en los hijos del Paso 1, los términos producidos no encajarían como `b1, b2: BTree` en los patrones de los pasos 2 y 4. La reescritura se trabaría justo después del primer `Insert`. El cambio realizado en el axioma base habilita la cadena completa de 9 pasos que mostramos.

Traza generada para la spec `BSTree-hide.nereus`. La forma normal del término es un BST sintácticamente válido construido únicamente desde los constructores básicos públicos `BTree_Empty` y `BTree_Insert` (modulo la equivalencia algebraica con `BTree_Create`, oculto vía `HIDE`).