Implementazione DICTIONARY con ABR

```
// RIMOZIONE DI UN NODO
TREE removeNode(TREE T, ITEM k)
    // individuo il nodo da rimuovere
    TREE u ← lookupNode(T, k)

    // se il nodo da rimuovere è presente nell'albero...
    se u ≠ nil allora
(1)     // ...e non ha figli
        se u.left == nil and u.right == nil allora
            se u.parent ≠ nil allora // se esiste il padre
                link(u.parent, nil, k)  // rimuovo il puntatore al figlio

            // rimuovo direttamente il nodo
            delete u
(3)     // ...ed ha due figli
        se u.left ≠ nil and u.right ≠ nil allora
            TREE s = successorNode  // individuo il successore
            link(s.parent, s.right, s.key)  // collego il sottoalbero destro

            // copio il successore
            // nella posizione del nodo rimosso
            u.key ← s.key
            u.value ← s.value

            // rimuovo il successore
            delete s
(2)     // ...ed ha un solo figlio (sinistro)
        se u.left ≠ nil and u.right == nil allora
            link(u.parent, u.left, k)  // collega il figlio al padre

            se u.parent == nil allora // se il padre non esiste
                T == u.right  // il figlio diventa la radice
        // ...ed ha un solo figlio (sinistro)
        altrimenti
            link(u.parent, u.right, k)  // collega il figlio al padre

            se u.parent == nil allora // se il padre non esiste
                T == u.right  // il figlio diventa la radice

    // restituisco la radice
    ritorna T
```