

Pictorial Summary of B-Tree Deletion

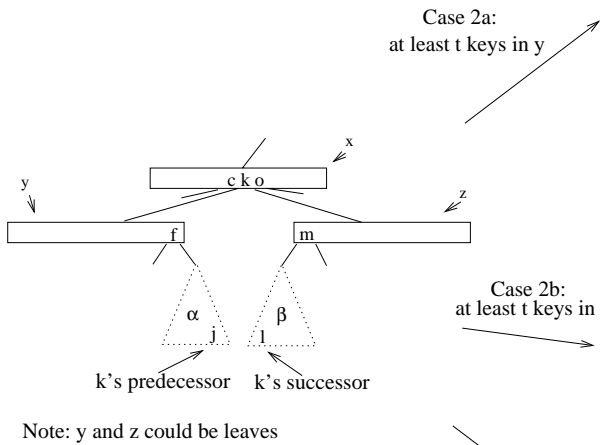
Key idea is to ensure as you move down the tree that for the nodes visited (i.e. those on the path from the root to the node with the key to delete), the number of keys is always at least one more than the minimum allowed (i.e. at least $1 + (t-1) = t$) with the exception of the root.

Repeat until Case 1 is reached:

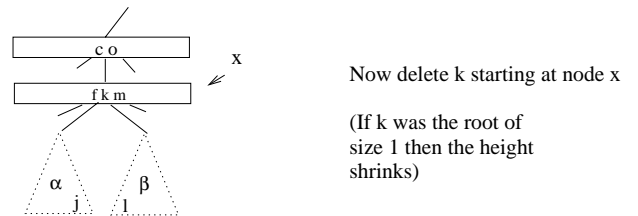
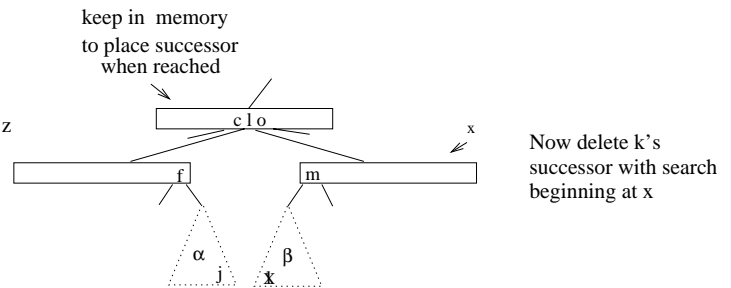
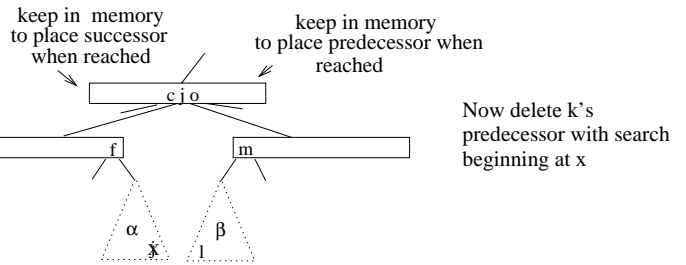
Let x be the current node when searching for the node with key k to delete.

Case 1: x is a leaf (which must then contain k). Then just delete k

Case 2: k is in x where x is not a leaf
 y and z are children surrounding k

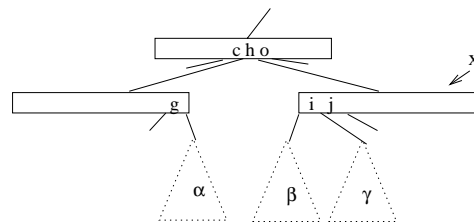
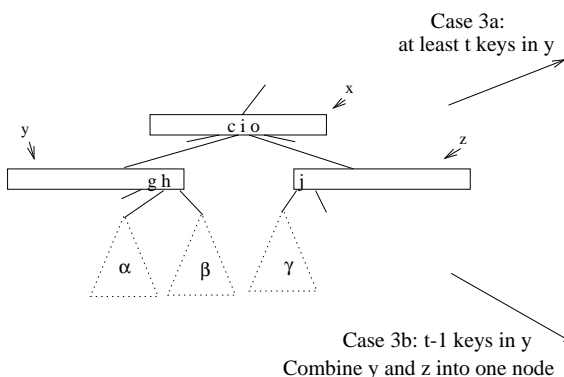


Case 2c: $t-1$ keys in y and z
 Combine y and z into one node



Case 3: Node we want to go to next has only $t-1$ keys

z is the node you want to go to next in the search, and y is a neighboring sibling
 (y could be right of z , and y and z could both be leaves)



In both cases you now continue the search for the desired key continuing from x

