COMP 282

Lecture 14

2-3-4 Tree

Deletion

Philosophy

- Insertion attempts to make nodes empty enough so that the leaf node can always accept a new item.
- Deletion does the opposite.
- Deletion attempts to fill nodes so that the leaf node can spare at least one item.
- Our goal therefore is to gurantee that our focus can only be on 3-nodes or 4-nodes.

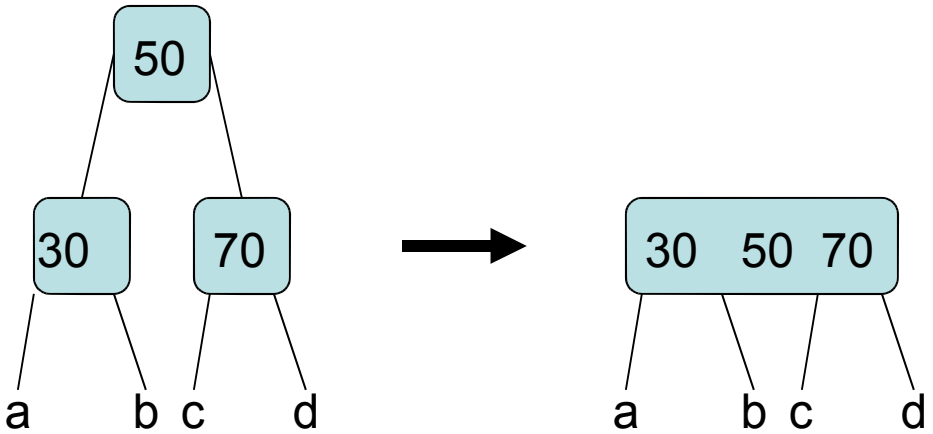
Special Case

- The tree has only one node (root) with one item.

- If that item is the one to be deleted then we simply make the tree empty (root=null). Otherwise we descend the tree and swap with in-order successor.

Other root cases… [case R1]

- Reduce the tree if the root node and both of its children are 2-nodes. Then begin descent/search.



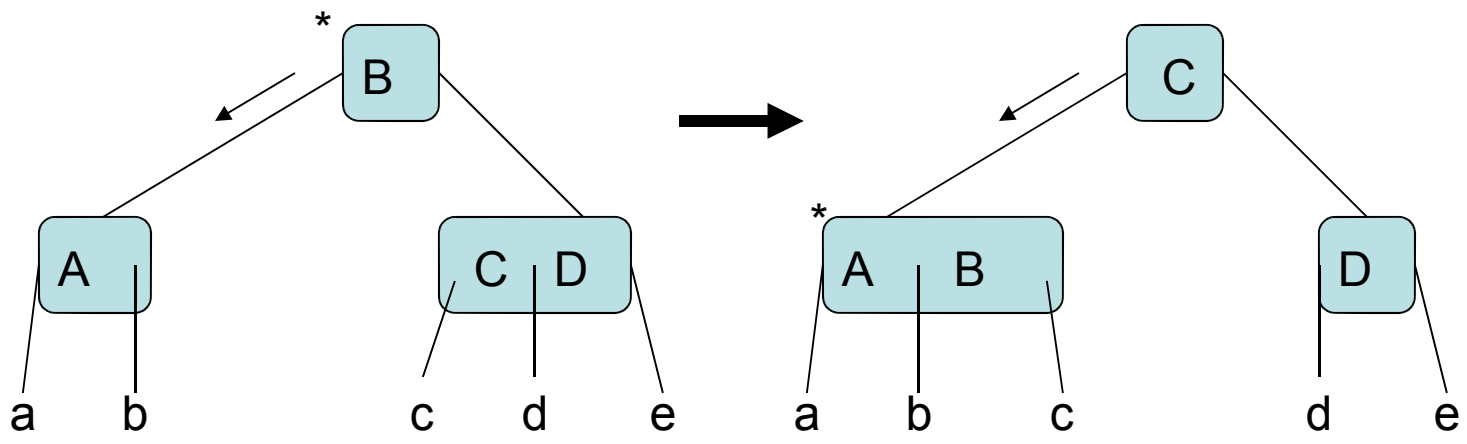
- This is how the height of the tree will decrease.

[Case R2l and R2r]

- You only need to perform topology modifcations if you are about to focus on a 2-node.
- If the root is a 2-node you need to decide which branch is important to you.
  - If the root is theItem then choose the right branch (in-order successor),
  - Otherwise choose the branch based on search key value.

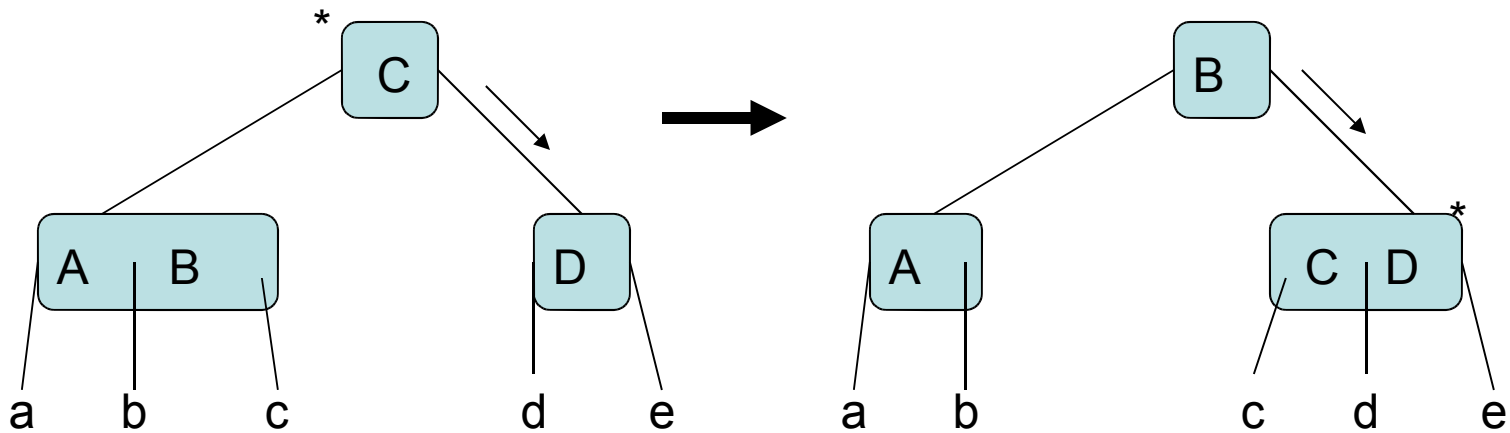
[Case R2l]

- Wish to descend down left branch and left branch is a 2-node



[Case R2r]

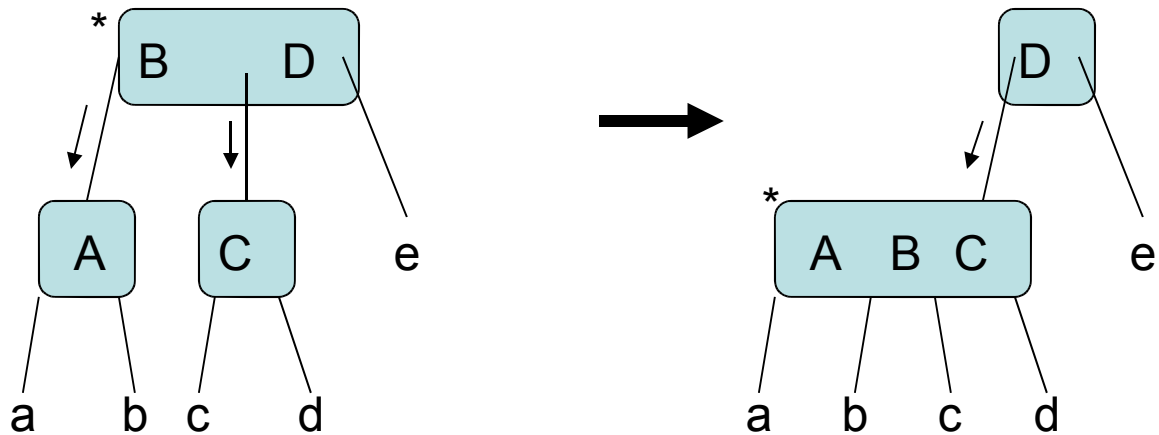
- Desire to traverse right branch.



- From this point on *all* nodes that you focus on will be guaranteed to not be 2-nodes and can therefore spare at least one item.

Cases [F3L-M2] and [F3M-L2]

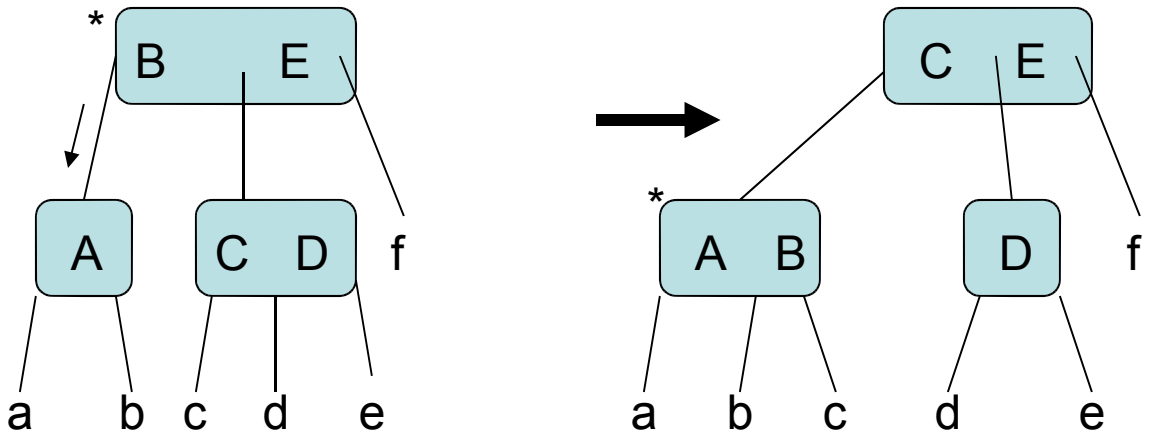
- Current focus is a three node and destination is either the left or middle branch and both of these branches are 2-nodes:



- After restructure blindly descend left.

[Case F3L-M3]

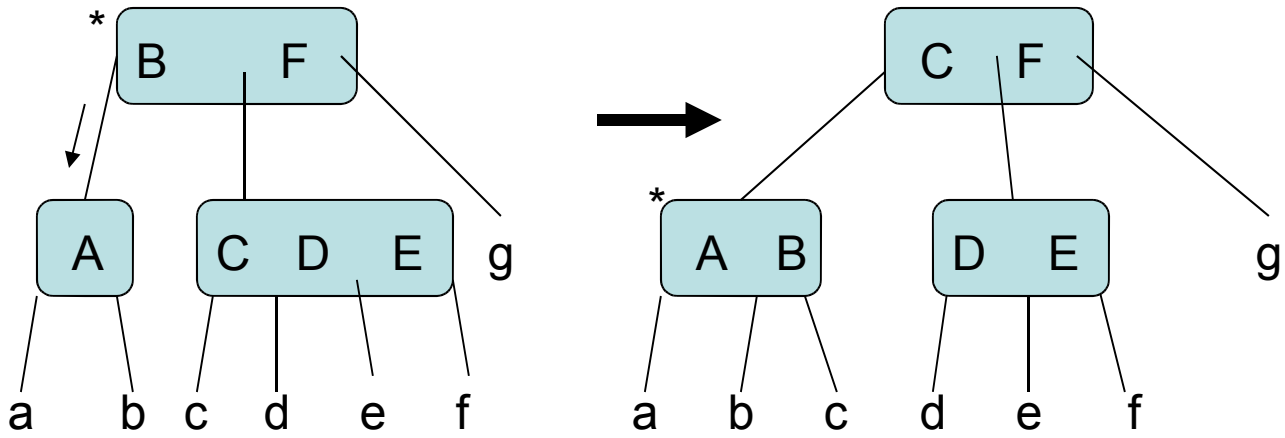
- Focus is a 3-node, destination is left (2-node) and middle branch is 3-node.



- Blindly descend to left subtree.

[Case F3-M4]

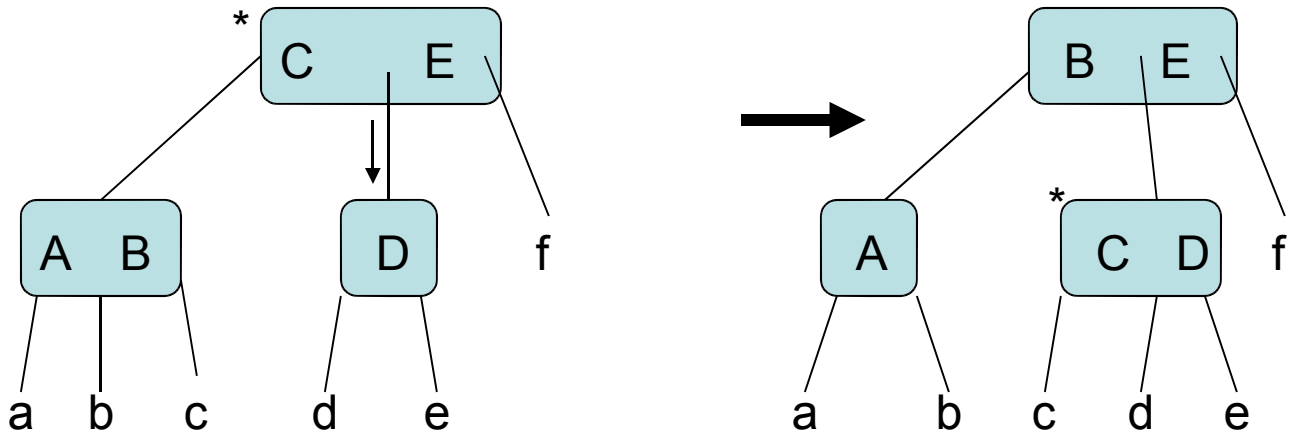
- Focus is a 3-node, destination is left and middle branch is 4-node.



- Blindly descend to left subtree.

[Case F3M-L3]

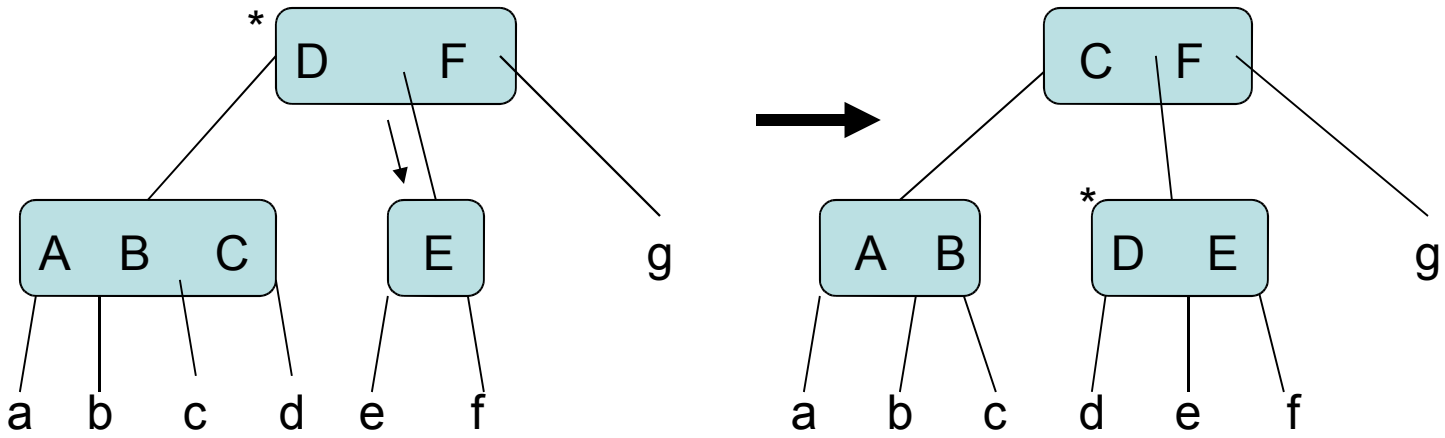
- Focus is a 3-node, destination is middle and left branch is 3-node.



- Blindly descend to middle subtree.

[Case F3M-L4]

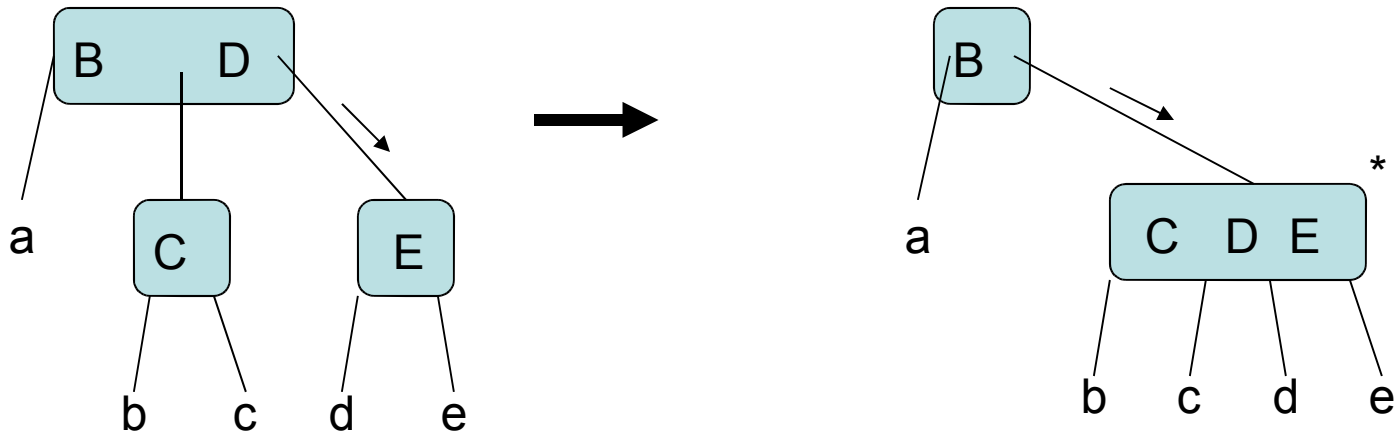
- Focus is a 3-node, destination is middle and left sibling is a 4-node.



- Blindly descend to middle subtree.

Case [F3R-M2]

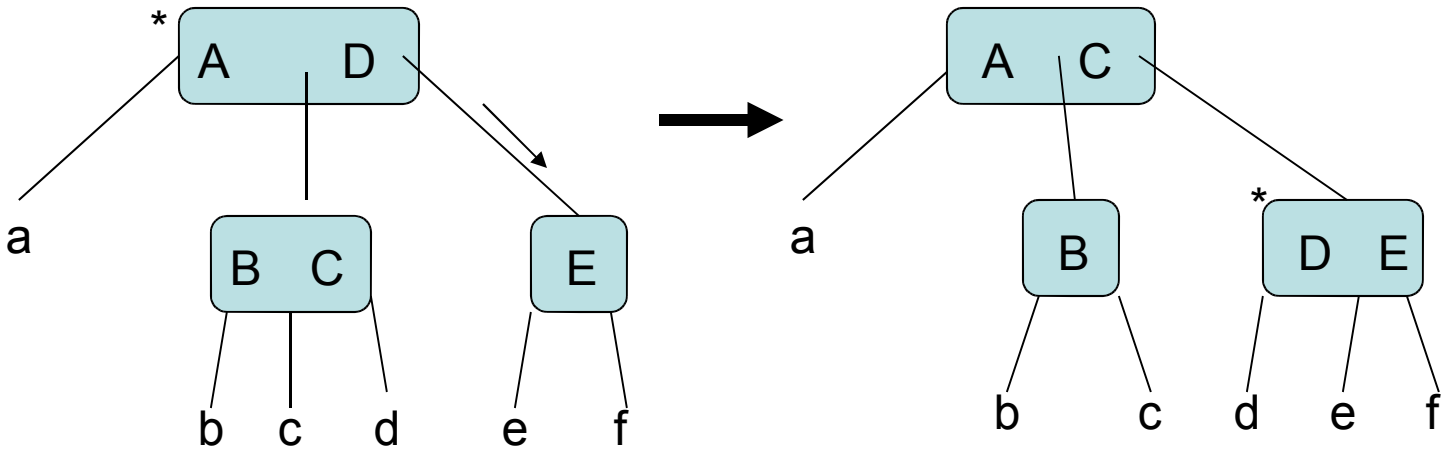
- Focus is a 3-node and destination is right with a 2-node middle



- Blindly descend right afterwards.

[Case F3R-M3]

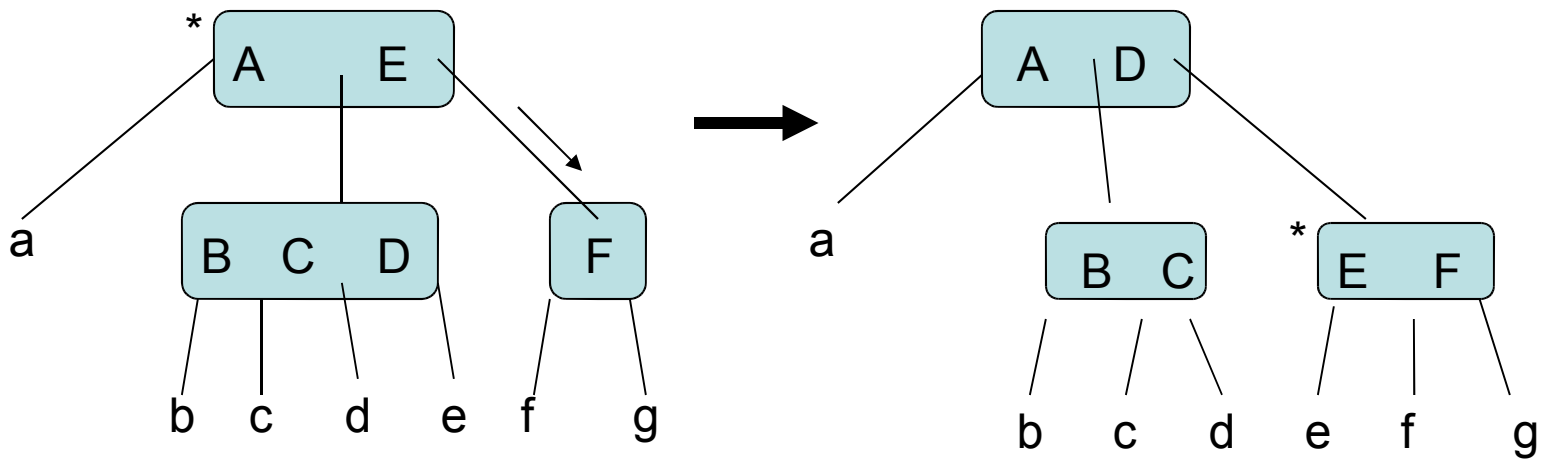
- Focus is a 3-node, destination is right and middle branch is 3-node.



- Blindly descend to right subtree.

[Case F3R-M4]

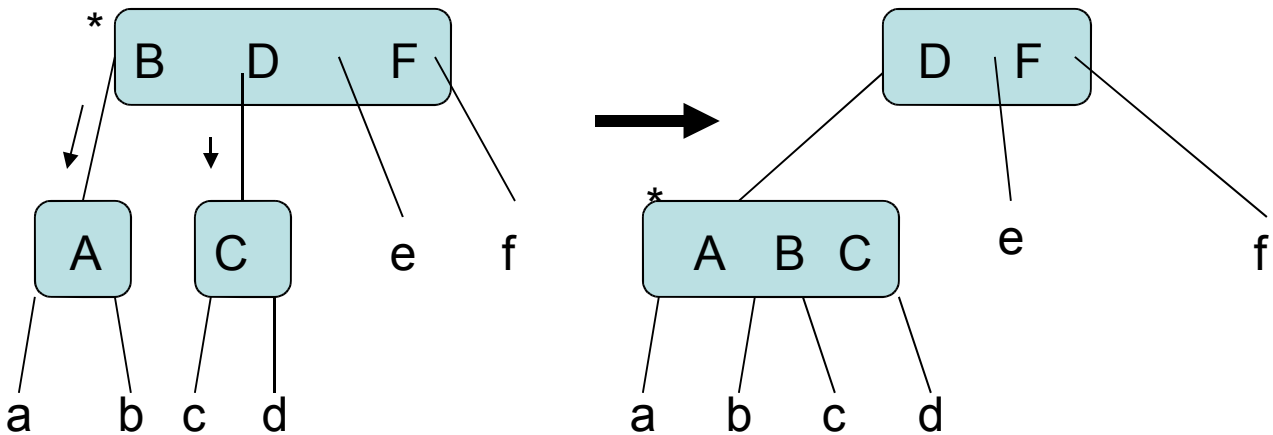
- Focus is a 3-node, destination is right and middle branch is 4-node.



- Blindly descend to right subtree.

Cases [F4L-M_R2] and [F4M_L-L2]

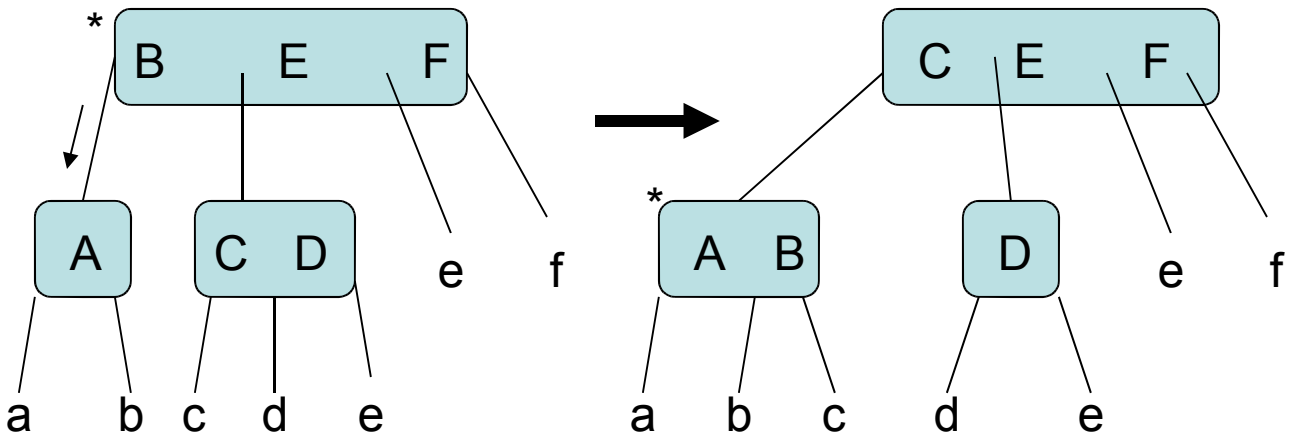
- Focus is a 4-node, destination is left or left-middle and both branches are 2-nodes.



- Descend to left (or left-middle) branch

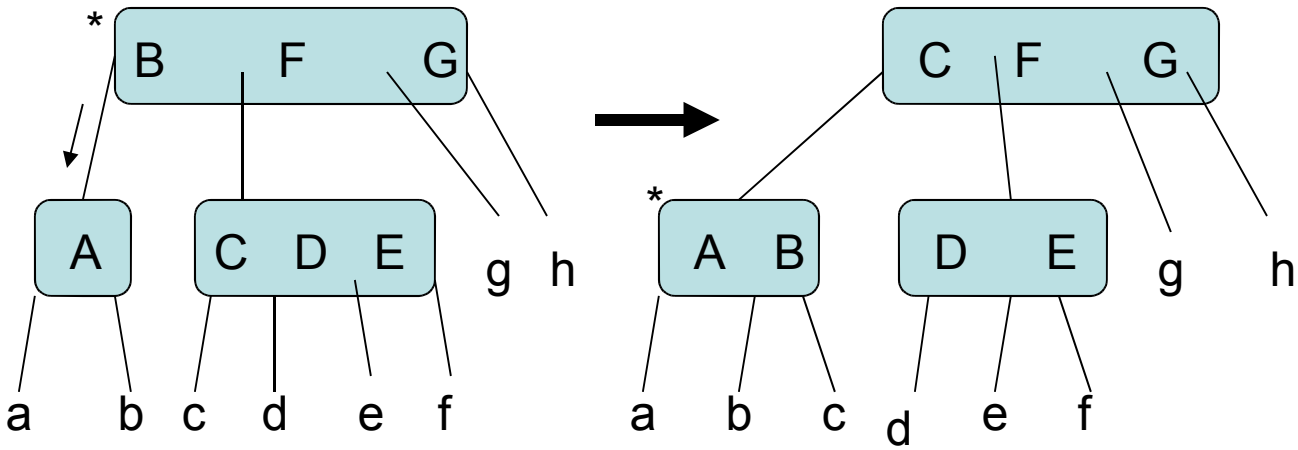
[Case F4L-M_L3]

- Focus is a 4-node, destination is left and left-middle branch is 3-node.



[Case F4L-M_L4]

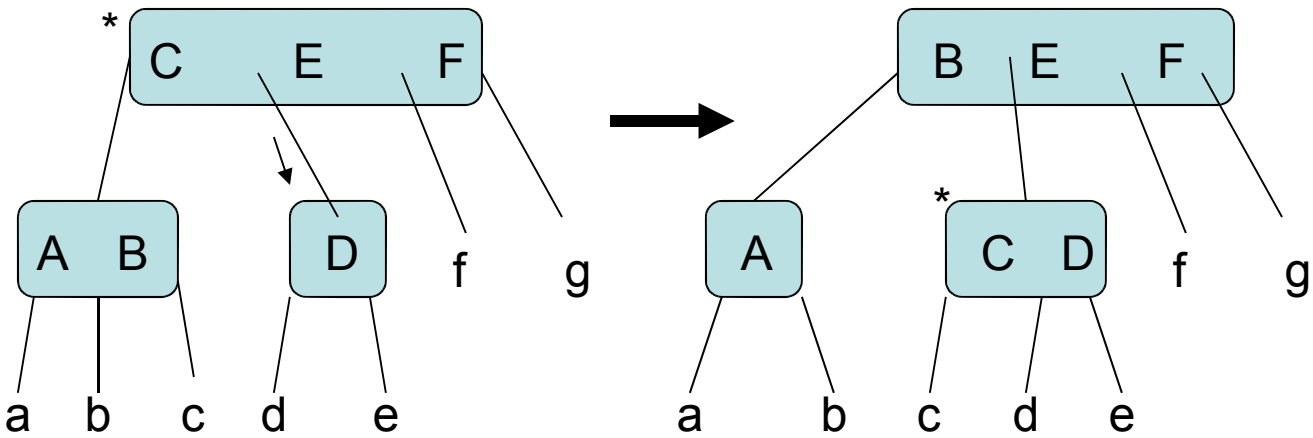
- Focus is a 4-node, destination is left and left-middle branch is 4-node.



- Descend to left branch

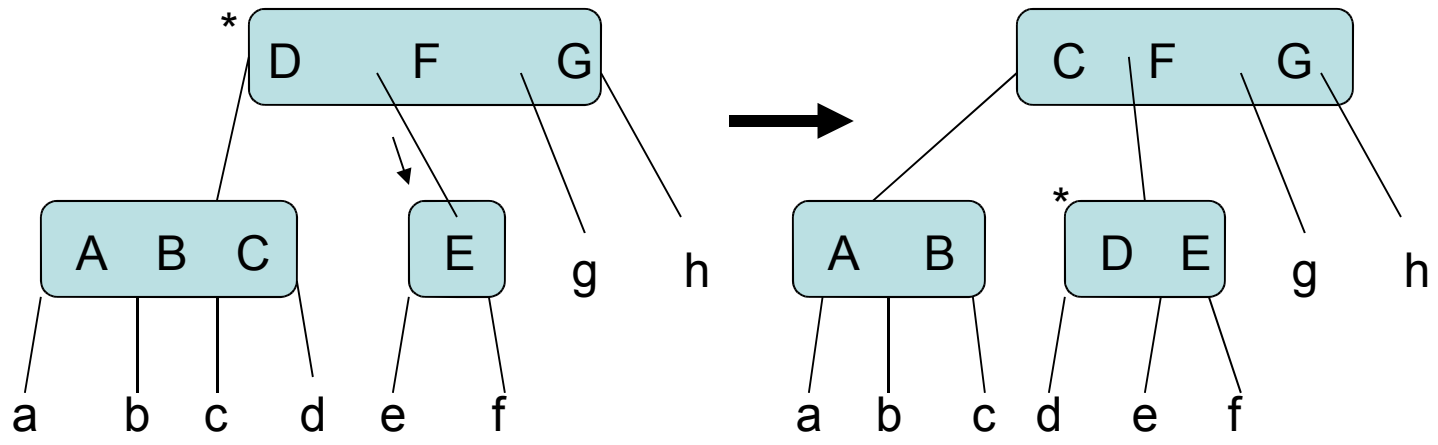
[Case F4M$_L$-L3]

- Focus is a 4-node, destination is left-middle and left branch is 3-node.



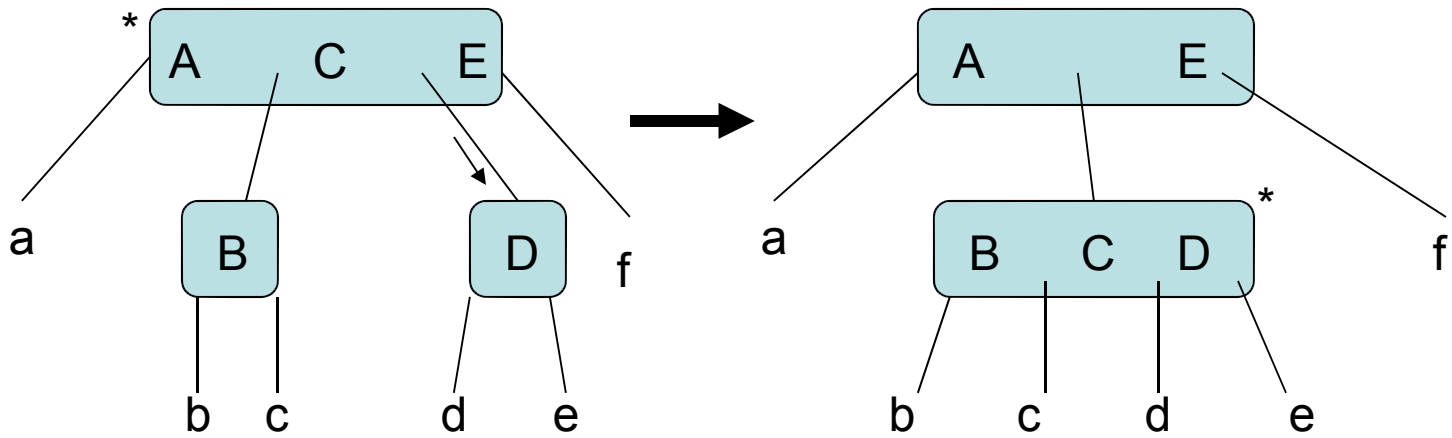
[Case F4M$_L$-L4]

- Focus is a 4-node, destination is left-middle and left branch is 4-node.



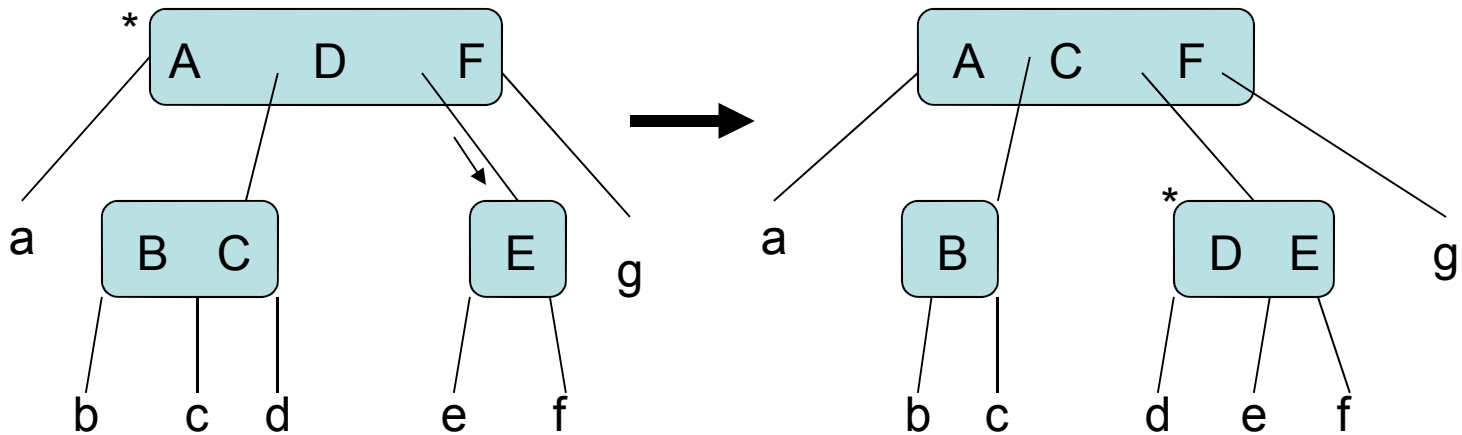
[Case F4M_R-M_L2]

- Focus is a 4-node, destination is right-middle and left-middle branch is 2-node.



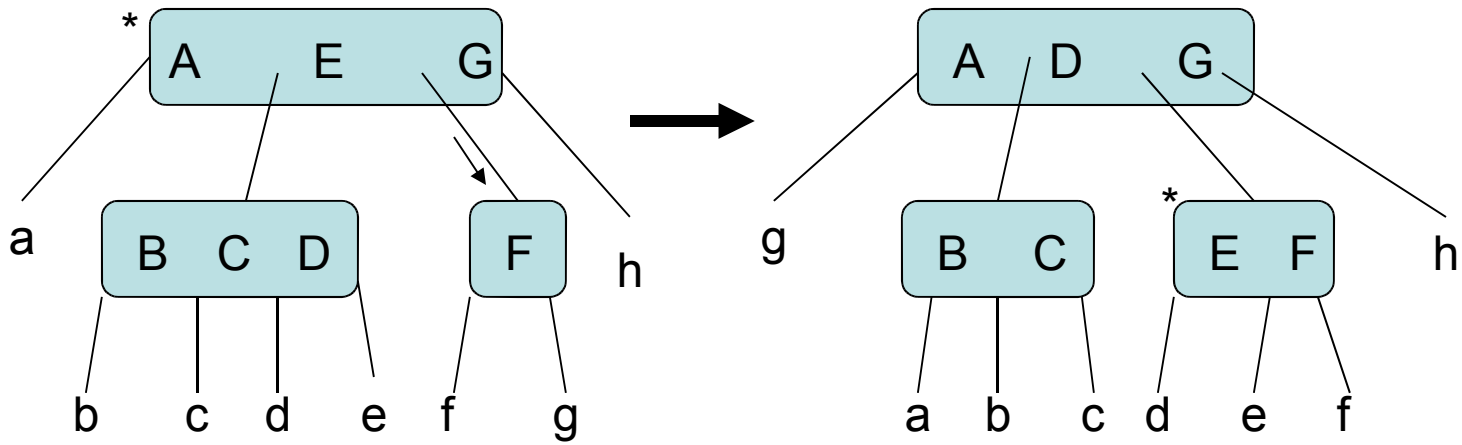
[Case F4M_R-M_L3]

- Focus is a 4-node, destination is right-middle and left-middle branch is 3-node.



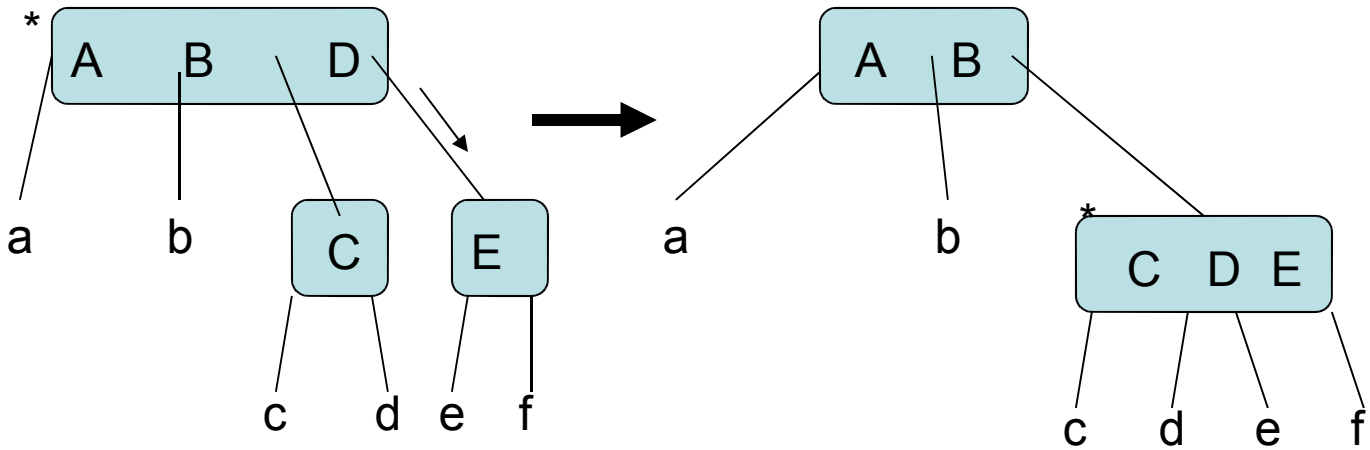
[Case F4M_R-M_L4]

- Focus is a 4-node, destination is right-middle and left-middle branch is 4-node.



Case [F4R-M_R2]

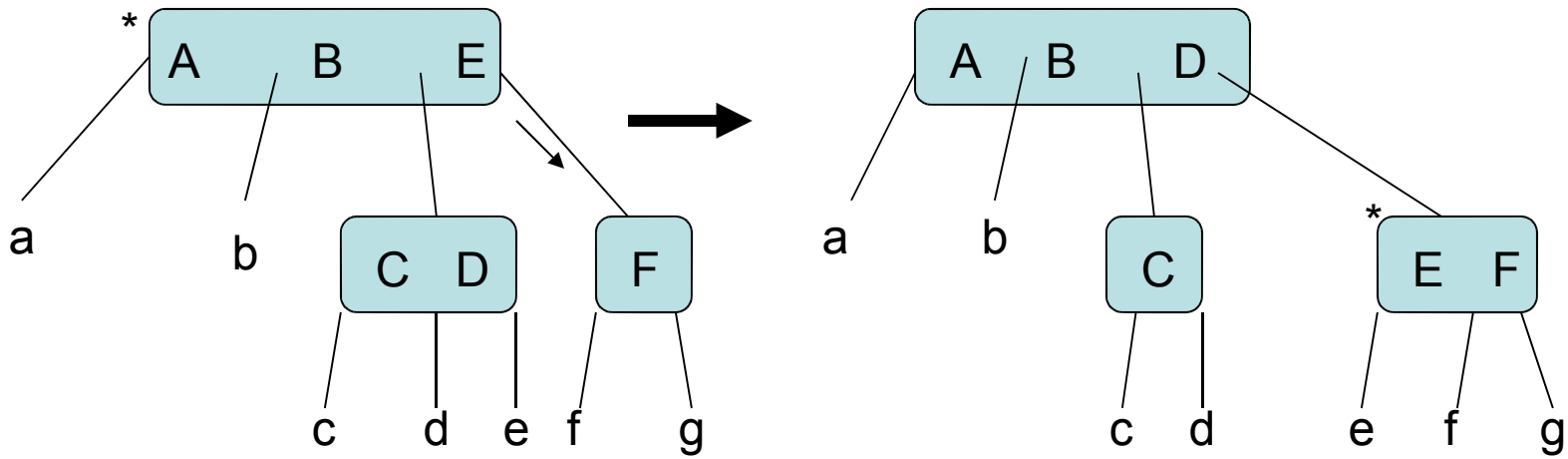
- Focus is a 4-node, destination is right and right-middle branch is 2-node.



- Descend to right-middle (or right) branch

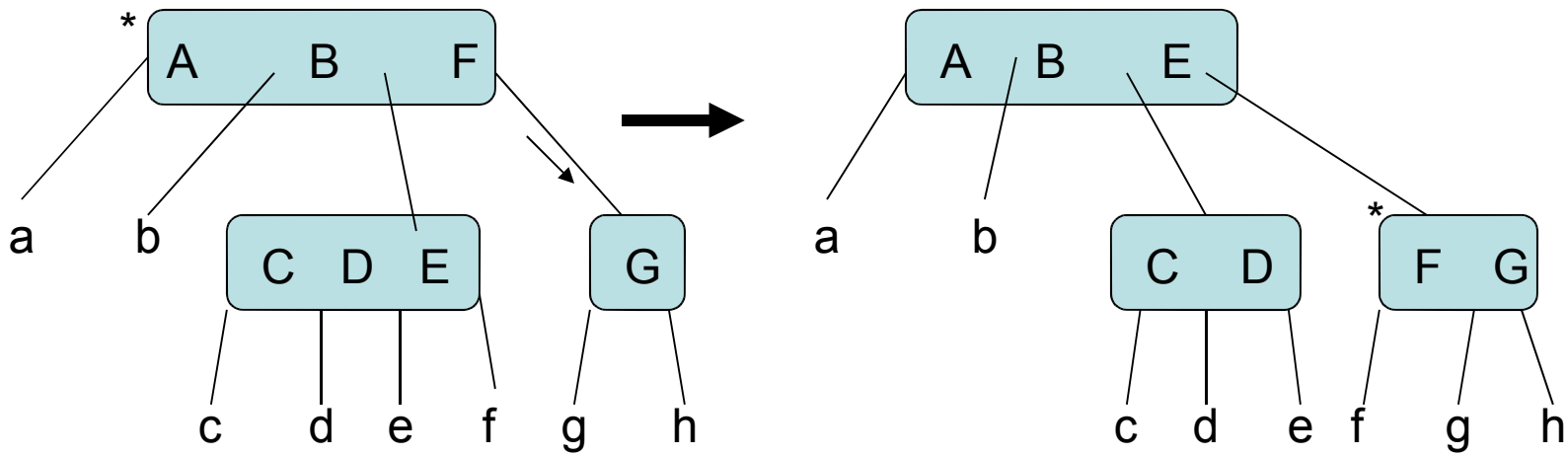
[Case F4R-M_R3]

- Focus is a 4-node, destination is right and right-middle branch is 3-node.



[Case F4R-M_R4]

- Focus is a 4-node, destination is right and right-middle branch is 4-node.



- end