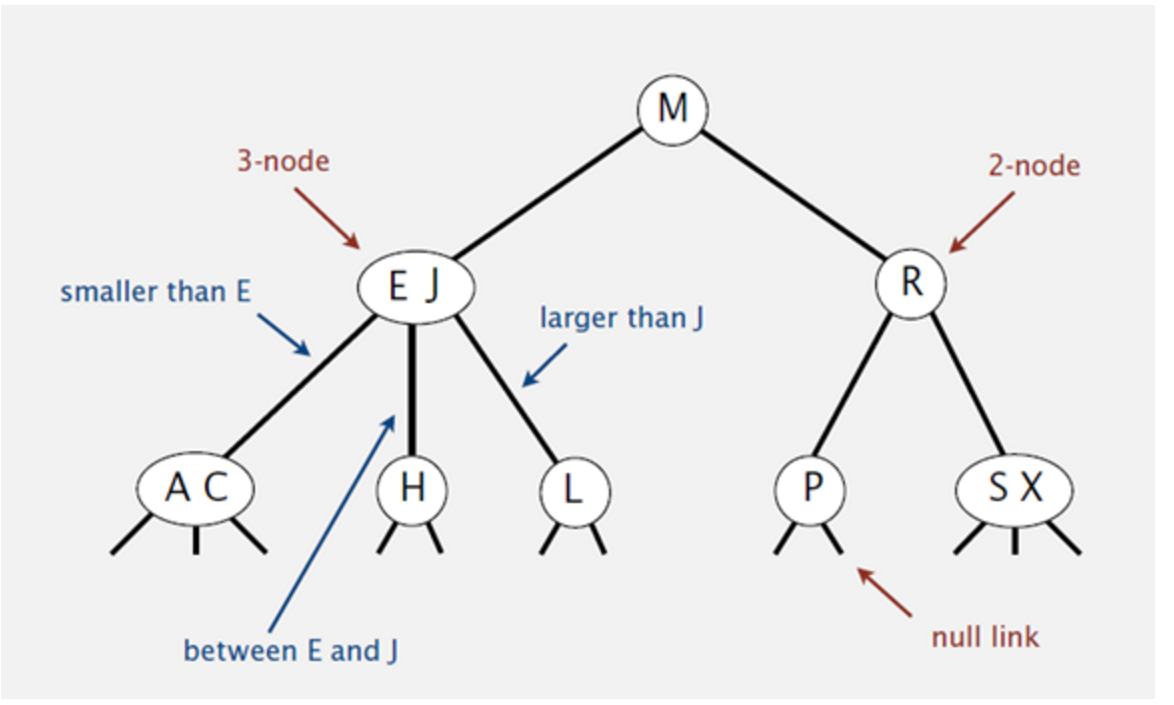
2-3 Search Tree

和二叉树不一样，2-3树允许每个节点保存1个或者2个值。对于普通的2节点(2-node)，他保存1个key和左右2个子节点。对应3节点(3-node)，保存两个Key和3个子节点。2-3查找树的定义如下：

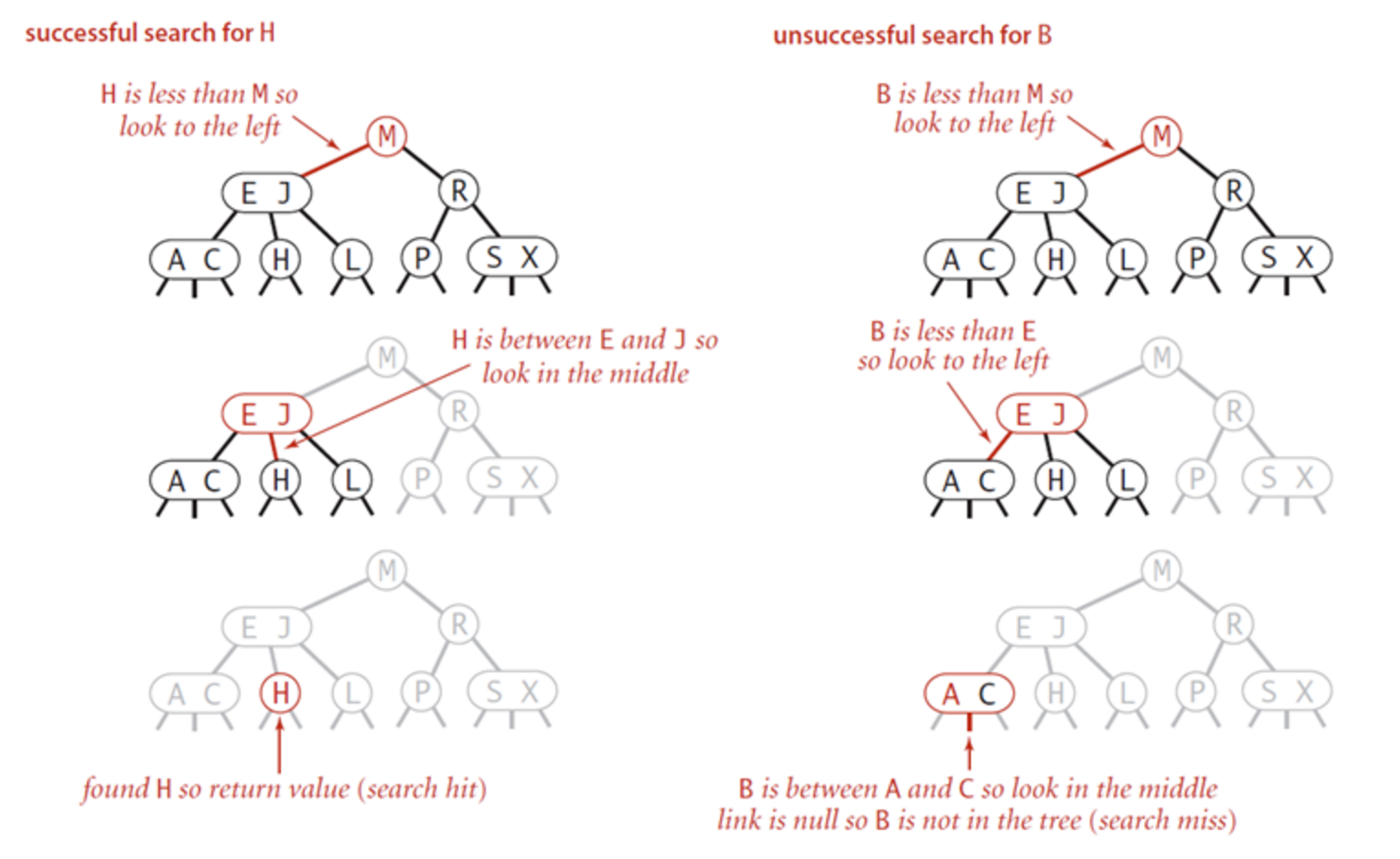
1. 要么为空，要么：

2. 对于2节点，该节点保存一个key及对应value，以及两个指向左右子节点的指针，左节点也是一个2-3节点，所有的值都比key要小，右节点也是一个2-3节点，所有的值比key要大。

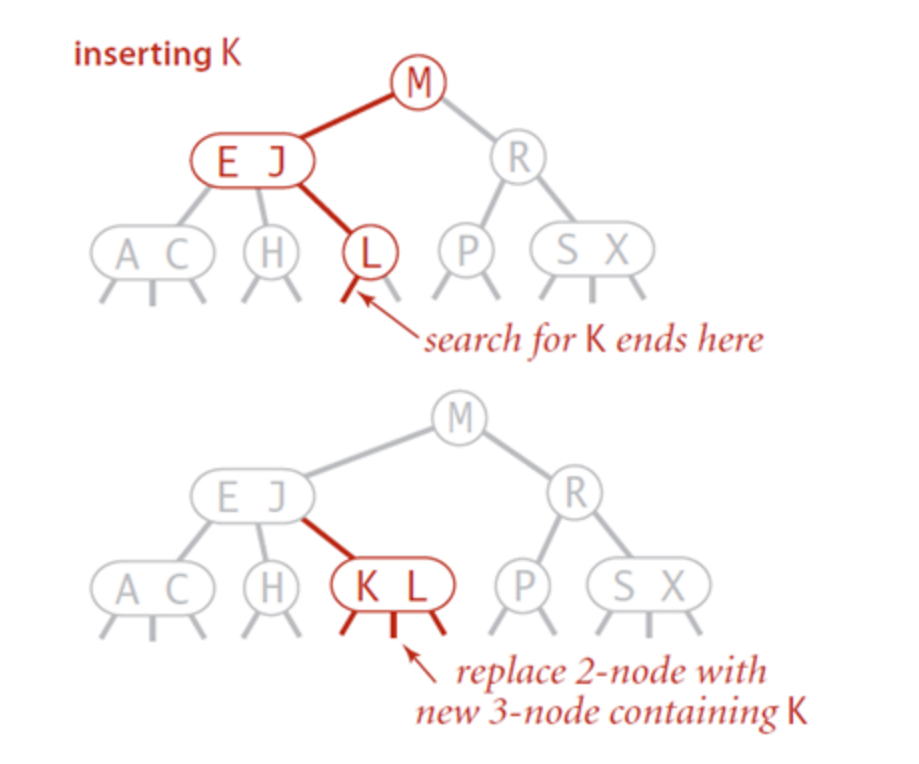
3. 对于3节点，该节点保存两个key及对应value，以及三个指向左中右子节点的指针。左节点也是一个2-3节点，所有的值均比两个key中的最小的key还要小；中间节点也是一个2-3节点，中间节点的key值在两个跟节点key值之间；右节点也是一个2-3节点，节点的所有key值比两个key中的最大的key还要大。



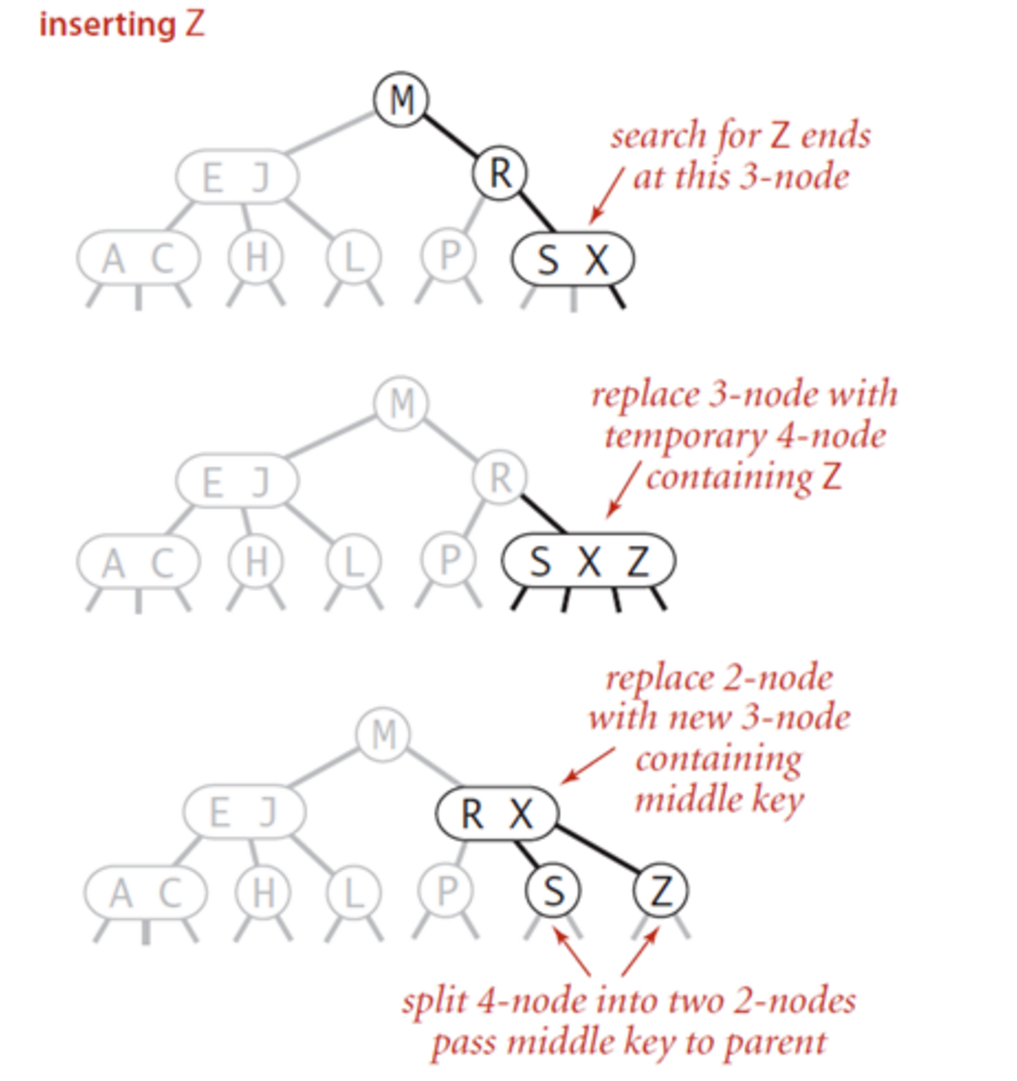
查找



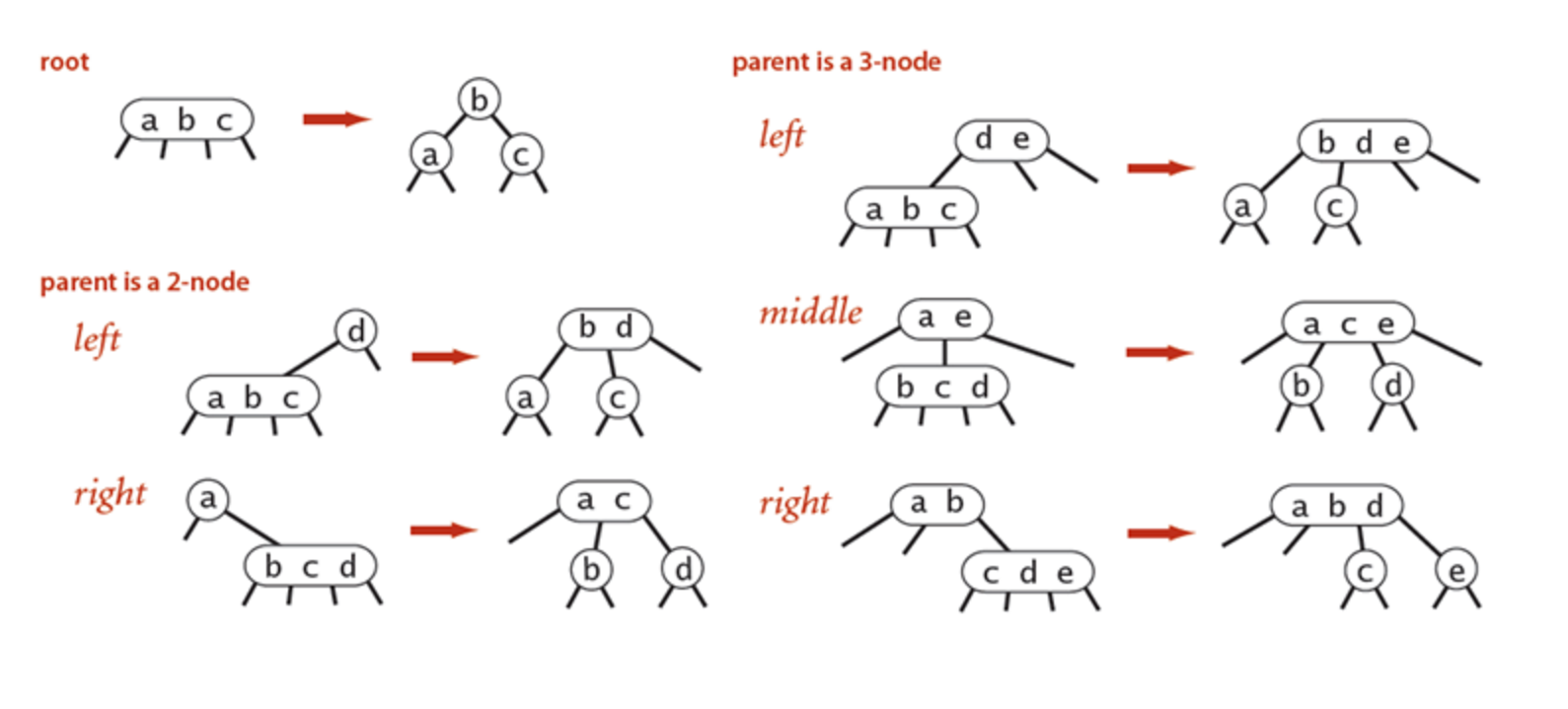
往2-node里插入



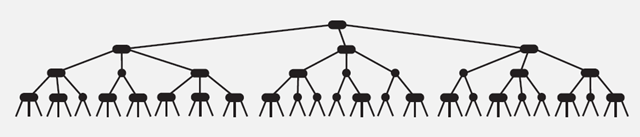
往3-node里插入



重新平衡



完全平衡的2-3查找树如下图，每个根节点到叶子节点的距离是相同的：



2-3树的查找效率与树的高度是息息相关的。

* 在最坏的情况下，也就是所有的节点都是2-node节点，查找效率为lgN
* 在最好的情况下，所有的节点都是3-node节点，查找效率为log3N约等于0.631lgN

Inserting in RB Tree

