The Last Homework

Christian Rinderknecht

11 June 2009

Red-Black Trees (RBT) are binary search trees such that

- each node is either red or black;
- the root is black;
- no red node has a red child;
- any path from the root to a leaf has the same number of black nodes.

These invariants ensure that the height of a RBT is at most $2\lfloor\lg(n+1)\rfloor$, where $\lfloor x\rfloor$ is the greatest integer which is lower than $\lg(n+1)$. As a consequence, the longest path from root to leaf is, at worst, twice as long as the shortest: this is the balancing property which makes RBT a data structure efficiently accessed. The purpose of the homework is to write the function `add/2` that takes an item and a RBT and returns the tree with the item (the same tree if the item was already present). The following remarkably simple presentation is due to Chris Okasaki.

Let us assume that the inserted item is not in the tree. The insertion of an item starts as in a binary search tree: going down until an empty node is reached. Then add a **red** leaf. On the way up, the tree may need to be rebalanced. The following four cases of unbalance are possible:

The double circle means "black" and a single circle means "red". These four cases, if any (otherwise no change), must be transformed into

Also, make sure that the root of the rebalanced tree is made **black**.