Idea

...

Definition

We define T, the set of 2-3-Trees, inductively:

1. $Nil \in T$
2. $Two(l, k, r) \in T$ if and only if:
   (a) $l, r \in T$
   (b) $k \in Keys$
   (c) $l < k < r$
   (d) $l.height() = m.height()$
3. $Three(l, k_1, m, k_2, r) \in T$ if and only if:
   (a) $l, m, r \in T$
   (b) $k_1, k_2 \in Keys$
   (c) $l < k_1 < m < k_2 < r$
   (d) $l.height() = m.height() = r.height()$

There are also 4-Nodes:

$$Four(l, k_l, m_l, k_m, m_r, k_r, r)$$

is a 4-Node iff:

1. $l, m_l, m_r, r \in T$
2. $k_l, k_m, k_r \in Keys$
3. $l < k_l < m_l < k_m < m_r < k_r < r$
4. $l.height() = m_l.height() = m_r.height() = r.height()$

We define $insert : T \times Keys \to T$ using 3 auxilliary methods:

1. $ins : T \times Keys \to T^*$ $t.ins(k)$ may not contain more than one 4-Node, which has to be directly below the root-node - or if t consists of only one node, then the 4-node may be at the root too.
2. $restore : T^* \to T^*$ $restore$ moves the 4-node up to the root.
3. $grow : T^* \to T$

Implementation

TBD