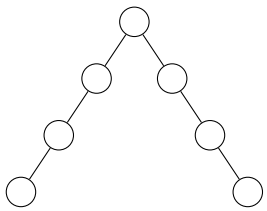
S. Schwoon / D. Suwimonteerabuth Winter Term 2008/09

Fundamental Algorithms

Solution Keys 7

1. (a) The condition is not strict enough. Consider the following (unbalanced) tree:



   (b) The tree can still be too unbalanced. For instance, the tree in (a) is still possible.

   (c) The condition is too strict. With height h, only trees with $2^h - 1$ nodes are possible.

2. (a) The following procedure performs a double rotation corresponding to situation (ii). Let `LL` and `RR` be "left" and "right" single rotations for situations (i) and (iv), respectively.

   Procedure `LR`
   $p \rightarrow$ `.left` := `RR`$(p \rightarrow$ `.left`$)$;
   return `LL`(p);

   (b) Let `LL`, `LR`, `RL`, `RR` denote procedures that fix the situations (i)–(iv), respectively. Inserting new nodes into the tree results in the following sequence of rotations:

   `LL`, `RR`, `RR`, `RR`, `RL`, `RL`, `RR`, `LL`, `LL`, `LL`, `LR` .

3. Given a height $h \geq 1$, call the following procedure `create`$(h, 1)$ to create a tree.

   Procedure `create`
   Input: a height h, a value k
   Output: (i) a minimal AVL tree of height h, containing n nodes labeled with values starting from k; (ii) the value $k + n$

   p := `new BinHNode`;
   if $h \leq 2$ **then**
   $p \rightarrow$ `.height` := 1; $p \rightarrow$ `.left` := $p \rightarrow$ `.right` := `NIL`;
   if $h = 1$ **then** $p \rightarrow$ `.value` $= k$; **return** $(p, k + 1)$; **fi**;
   q := `new BinHNode`; $q \rightarrow$ `.value` := k; $q \rightarrow$ `.height` := 2;
   $q \rightarrow$ `.left` := `NIL`; $q \rightarrow$ `.right` := p;
   $p \rightarrow$ `.value` := $k + 1$;
   return $(q, k + 2)$;
   fi
   $(p \rightarrow$ `.left`$, k)$:= `create`$(h - 2, k)$;
   $p \rightarrow$ `.value` := k; $p \rightarrow$ `.height` := h;
   $(p \rightarrow$ `.right`$, k)$:= `create`$(h - 1, k + 1)$;
   return (p, k);