Technische Universität München
Institut für Informatik
Prof. Tobias Nipkow, Ph.D.
Fabian Huch

SS 23
13. 6. 2023

Functional Data Structures

Exercise Sheet 8

Exercise 8.1 Joining 2-3-Trees

Implement and prove correct a function to combine two 2-3-trees of equal height, such that the inorder traversal of the resulting tree is the concatenation of the inorder traversal of the arguments, and the height of the result is either the height of the arguments, or has increased by one. Use *'a upI* to return the result, similar to *Tree23_Set.ins*:

fun *join* :: *"'a tree23 ⇒ 'a tree23 ⇒ 'a upI"*

lemma *join_inorder*:
fixes *t1 t2* :: *"'a tree23"*
assumes *"height t1 = height t2"*
assumes *"complete t1" "complete t2"*
shows *"inorder (treeI (join t1 t2)) = (inorder t1 @ inorder t2)"*

lemma *join_complete*:
fixes *t1 t2* :: *"'a tree23"*
assumes *"height t1 = height t2"*
assumes *"complete t1" "complete t2"*
shows *"complete (treeI (join t1 t2)) ∧ hI (join t1 t2) = height t2"*

Hints:

- Try to use automatic case splitting (*auto split*: ...) instead of explicit case splitting via Isar (There will be dozens of cases).
- To find bugs in your join function, or isolate the case where your automatic proof does not (yet) work, use Isar to perform the induction proof case by case.

Exercise 8.2 Bounding the Size of 2-3-Trees

Show that for complete 2-3-trees, we have:

$log_3\ (\ s(t)\ +\ 1\ )\ \leq\ h(t)\ \leq\ log_2\ (s(t)\ +\ 1)$

Hint: It helps to first raise the two sides of the inequation to the 2nd/3rd power. Use sledgehammer and find-theorems to search for the appropriate lemmas.

lemma *height_bound_upper*: *"complete t ⟹ height t ≤ log 2 (size t + 1)"*

lemma *height_bound_lower*:
  assumes *"complete t"*
 shows *"log 3 (size t + 1) ≤ height t"*

Homework 8.1 Bit-Vectors

Submission until Monday, June 19, 23:59pm.

A bit-vector is a list of Booleans that encodes a finite set of natural numbers as follows: A number i is in the set, if i is less than the length of the list and the ith element of the list is true. That means that the abstraction function is:

$bv\_set\ l = \{i.\ i < length\ l \wedge l\ !\ i\}$

Define the other operations of the Set interface (including delete) and interpret the locale!

Hints:

- Compose existing functions rather than defining your own. Elegant definitions won't even need a single case distinction!
- The syntax to update the n-th element of a list is: $xs[n := x]$.
- To get a clickable template for your interpretation proof, start it with: *proof*(*standard*,*goal_cases*).

Homework 8.2 Bounding the size of 2-3-Trees

Submission until Monday, June 19, 23:59pm.

Show that a 2-3-tree has only 3 nodes, if and only if its number of leafs is 3 to the power of its height. One direction is quite easy, the other one requires more proving.

Hint: What is the general relation between *size1* and *height*?

theorem *complete_3_tree_height*: *"complete t ⟹ is_3_tree t ⟷ size1 t = 3^height t"*