Technische Universität München
Institut für Informatik
Prof. Tobias Nipkow, Ph.D.
Sascha Böhme, Alexander Krauss

WS 2010/11
1. 11. 2010

Semantics of Programming Languages

Exercise Sheet 2

Homework 2 Power function and binary trees

Submission until Wednesday, November 10, 2010, 12:00 (noon).

Define the recursive function *pow* which computes, for two natural numbers n and m, the value n^m. You may use the predefined natural number operators $+$ and $*$.

Prove the following property of *pow*. You may need to prove auxiliary lemmas.

theorem *"pow x (m * n) = pow (pow x m) n"*

Define a datatype *tree* of plain binary trees, that is, binary trees which do not store any information, neither in leafs nor in inner nodes. Moreover, write a function *count* which returns the total number all nodes (i.e., of leafs and inner nodes) of such binary trees.

Consider the following recursive function:

fun *explode* :: *"nat ⇒ tree ⇒ tree"* **where**
"explode 0 t = t" |
"explode (Suc n) t = explode n (Node t t)"

Experiment how *explode* influences the size of binary trees and find an equation expressing the relation between the count of a tree t and the count of the tree after exploding it by an arbitrary number n. Hint: you may re-use the previously defined function *pow*. Prove that your equation is correct.