Homework 2  Power function and binary trees

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

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

Prove the following property of $\text{pow}$. You may need to prove auxiliary lemmas.

**Theorem** "$\text{pow } x (m \cdot n) = \text{pow } (\text{pow } x m) n$"

Define a datatype $\text{tree}$ 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 $\text{count}$ which returns the total number all nodes (i.e., of leafs and inner nodes) of such binary trees.

Consider the following recursive function:

**fun** $\text{explode} :: \text{n} \Rightarrow \text{tree} \Rightarrow \text{tree}$ where

"$\text{explode } 0 \ t = t$" |
"$\text{explode } (\text{Suc } n) \ t = \text{explode } n \ (\text{Node } t \ t)$"

Experiment how $\text{explode}$ 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 $\text{pow}$.

Prove that your equation is correct.