CS476 Homework 8 Due at 10:45 am on Tuesday 10/17

Your solutions, as well as screenshots of your tool interactions for Problem 2, should be emailed to `clarage2@illinois.edu`. In addition, your screenshots for all tool interactions *should be present in the same pdf containing your answers to the homework's problems.*

1. Prove **Ex**.14.3 in pg. 14 of Lecture 14.

2. Consider the definition of binary trees with natural numbers on the leaves given in Lecture 16. Complete the module `NAT-TREE+AC` given below by defining with confluent and terminating equations the two functions called `leaves` and `inner`, that count, respectively, the number of leaf nodes of a tree, and the number of nodes in a tree that are *not* leaf nodes. For example, for the tree `((0 ^ s(0)) ^ 0) ^ s(0)` there are 4 leaf nodes (namely `0`, `s(0)`, `0`, and `s(0)`, and 3 inner nodes (corresponding to the 3 different occurrences of the `^` operator).

```
set include BOOL off .

fmod PEANO+AC is sort Nat .
    op 0 : -> Nat [ctor metadata "0"] .
    op s : Nat -> Nat [ctor metadata "4"] .
    op _+_ : Nat Nat -> Nat [assoc comm metadata "8"] .
    vars N M : Nat .
    eq N + 0 = N .
    eq N + s(M) = s(N + M) .
endfm


fmod NAT-TREE+AC is protecting PEANO+AC .
    sort  Tree .  subsort Nat < Tree .
    op _^_ : Tree Tree -> Tree [ctor metadata "6"] .
    op inner : Tree -> Nat [metadata "10"] .  *** counts inner nodes
    op leaves : Tree -> Nat [metadata "12"] . *** counts tree leaves
    vars N M : Nat .  vars T1 T2 : Tree .
    *** add equations for leaves and inner here
endfm
```

Once you have defined and tested your definitions for `leaves` and `inner` do the following, *including screenshots for each tool used* in your solutions for this homework:

- state a theorem, in the form of an equation, that gives a general law stating, for any tree `T`, the exact relation between the numbers `leaves(T)` and `inner(T)`
- give a mechanical proof of that theorem using Maude's NuITP.

50% Extra Credit. The correctness of the NuITP proof of the above property depends on checking some executability conditions. You can get 50% extra credit if you check the following conditions:

- check that `NAT-TREE+AC` is confluent using the Church-Rosser Checker

- check that NAT-TREE+AC is terminating using the MTA with the RPO order specified by the metadata annotations
- check that NAT-TREE+AC is sufficiently complete using the SCC tool